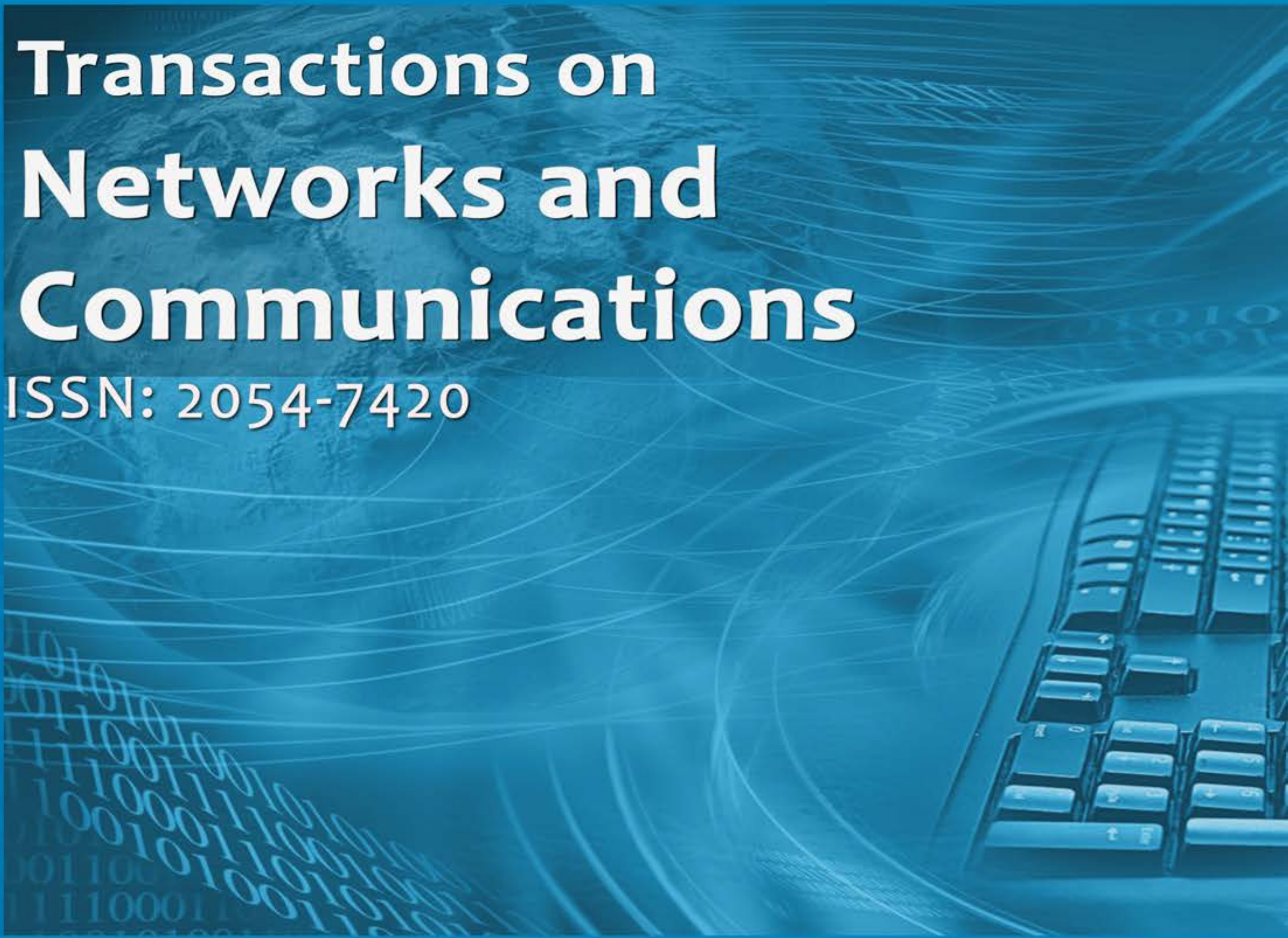
Volume 7 No. 1, February 2019



Transactions on Networks and Communications

ISSN: 2054-7420

Society for Sciences and Education - United Kingdom

TABLE OF CONTENTS

Editor In Chief

EDITORIAL ADVISORY BOARD

DISCLAIMER

All the contributions are published in good faith and intentions to promote and encourage research activities around the globe. The contributions are property of their respective authors/owners and the journal is not responsible for any content that hurts someone’s views or feelings etc.

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TNC TRANSACTIONS ON NETWORKS AND COMMUNICATIONS

VOLUME 7, NO. 1
ISSN: 2054 -7420

Merging Securely M2M Protocols, Internet of Things and Cloud Computing

Dina Darwish
The International Academy for Engineering and Media Science, Egypt
dina.g.darwish@gmail.com

ABSTRACT

The Internet of Things provides new ways for communication through the Web world using object-enabled networks. At the same time, M2M devices intercommunication and their communication through the web if they were connected to the Internet, presents new challenges, especially in security that traditional communication models have not yet fully solved. Because of their inborn un-watched, minimal effort and mass-sent nature, M2M devices, and remote communication architectures and solutions for these devices, would encapsulate new dangers in security. These threats are not fully faced by use of security technologies and methods implemented in existing wireless devices, cellular networks or WLANs. The use of cloud computing gives a convenient, on demand and scalable network access to a shared pool of configurable computing resources and devices. This paper concentrates on a secure method to integrate the M2M protocols with the Internet of Things (IoT) and Cloud Computing under the name of Secure Machine-to-Internet Clouding (SM2IC) architecture. The secure design for integrating M2M protocols, along with IoT and cloud computing is proposed. To apply this design, an IoT enabled smart home scenario was examined to analyze secure communication between M2M devices and IoT applications. Also, the cloud computing is used to include different cloud applications, such as, IaaS, PaaS, and SaaS for monitoring the quality of service of M2M devices through IoT applications. Then, simulations were performed to test the proposed security technique, followed by conclusions and future work.

Keywords: Cloud computing; Internet of Things, M2M protocols, Secure Integration, Connected M2M Devices.

1 Introduction

Internet of Things (IoT) is considered as a technology aimed at providing customers with smarter services by linking different devices to the Internet and enabling these devices to exchange information with each other. IoT has been distinguished as a developing technology in numerous IT trend reports [1], and the number of IoT devices is proposed to increase [2,3]. It is suggested by some IT trend reports that the worldwide IoT market will be worth billions of dollars by 2022 [4]. The interconnection between the different kinds of IoT devices is a key issue for the achievement of IoT, in light of the fact that the numbers of IoT devices is developing ceaselessly. IoT standardization bodies have completed several endeavors to understand the interconnection issue. Numerous Web of Things (IoT) platforms were outlined and

DOI: 10.14738/tnc.71.5939
Publication Date: 02nd February 2019
URL: http://dx.doi.org/10.14738/tnc.71.5939

executed over the previous decade. In any case, most platforms were applied in light of particular solutions or created to address certain domain issues.

To interconnect different proprietary platforms and deliver common IoT services to customers, there is a strong need to create a standardized IoT platform. To face interoperability issues, seven standard development organizations (SDOs) started a global standards project named oneM2M [5], by providing scalable and interoperable IoT standards for communication of devices and services. The goal of oneM2M is to present a single horizontal service platform, that can be implemented in different industries to deliver smarter IoT services to users and to exchange and share data among IoT applications. Machine-to-machine (M2M) communication is one of the next frontiers in wireless communication. There exist a large number of possibilities, in terms of new use cases, services and applications, that is suggested to result from communication between M2M devices. M2M can present benefits for the production and market opportunities for various manufacturers of M2M devices and components, service providers, and network operators.

Due to the huge number of M2M devices expected to be used, in a highly distributed network, enforcement of traditional security methods will not be practical because of the high cost of implementation of these devices. Also, deploying the conventional centralized IT security network model, protected by a firewall, is challenged by the need for a dispersed model, so, de-centralized methods for realizing security must be accessed. The growing direction towards using de-centralized systems creates a lot of situations in which enforcement of security, is accompanied by a controlled risk. The principles of how to enforce security embraced by traditional concepts of access control are being changed by a shift to implement "trust." An entity is considered "trusted" if it behaves correctly as expected to achieve its intended goal. By including pieces of the enforcement tasks to trusted elements distributed in a system, trust relationships can be created. This is the most important part in the organizational method of separate tasks within IT security. This security model, which is balanced between trust and enforcement, produces a useful, practical and scalable approach and, can be used for M2M communication.

Security is one of the main problems in any information system, including M2M systems. With a big market for M2M devices and networks, M2M systems require to be properly designed and implemented. Many applications envisioned for M2M can be done if security is properly considered from the beginning. There is a need for good security mechanisms and procedures, also, various characteristics of M2M systems and applications may constitute challenges to the design of useful security mechanisms.

A few troubles, for example, the support of heterogeneous communication advancements and protocols for communication between M2M devices, the restrictions on equipment of numerous M2M detecting and actuating platforms, and security desires from clients require to be recognized. Numerous lessons and specialized solutions have been achieved from research in many fields, for example, mobile ad hoc networks [6] [7] or remote sensor networks [8], but, M2M frameworks still need new strategies in security. The employment of different wired and wireless communication innovations guided by the utilization of a typical service platform decides the careful assessment of the applied cryptographic algorithms. The support of communication needs proper distinguishing strategies. The properties and asset restrictions of M2M systems form difficulties to the plan of suitable security innovations, that can deal with distinctive detecting and actuating M2M devices. Contingent upon security desires for the M2M systems and applications, clients will require systems that permit the control of how much individual data is anchored, while certain applications will require a specific level of individual data to be ensured [9].

Cloud computing is depicted by the National Institute of Standard and Technologies (NIST) [10] as follows: "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." The fundamental thought behind Cloud computing began to pick up fame after that Google's Chief Eric Shmidt utilized it in 2006 [11], and along the most recent years, Cloud computing has influenced IT industry. The presence of virtually unlimited storage and processing capabilities at low cost has created a new computing model, inside which virtual resources can be used on-demand.

Cloud computing [12] provides also a new method for design, development, test, deployment, run and maintenance of applications on the Internet. It is required to take care of running operating systems, networks, load balancing, routers, firewalls, and storage by the application developer, and at the same time, integrating these things and, enabling them to interact with the system. Also, it is important to take care by the developer of scalability, because it determines how the application can fit many geographically distributed users. The Cloud user; developer or consumer, can reach the Cloud services over the Internet, and the Cloud users must pay for time and services they are in need. The Cloud can also be expanded to implement large numbers of service requests. Cloud computing considers the micro-lifecycle management of applications, and enables application managers to concentrate on application design and surveillance. The Cloud computing platform is composed of different services for developing, testing, running, deploying, and maintaining applications on the Cloud. This direction for delivering services over the Internet, has been widely used by large companies, such as, Amazon, Google and Facebook and so on to gain both economic and technical benefits. Cloud Computing is considered as a disruptive technology with huge impact on the delivery of Internet services as well as for the IT sector.

The Internet of Things and Cloud computing are both considered as emerging technologies and they possess their own features. Things are connected to their virtual representations on the Internet and are reached through the Internet (i.e. Things as services) [13]. Cloud computing implements the utility model, which allows end-users to use and consume services in an efficient and pay-per-use way.

However, various technical and business-related issues are still unsolved. Certain issues have been determined for each service model, these issues are related to security, privacy, and service-level agreements, which need to be addressed for users [14]. Moreover, the lack of standard APIs makes extracting code and data from a site difficult for customers. Also, public Cloud customers are exposed to price increases, reliability problems or even to providers going out of business.

In this paper, a secure approach for integrating M2M protocols with the internet of things through cloud computing, namely secure machine-to-internet clouding (SM2IC) is proposed, and tested using the appropriate software tools. In section 2, a background on work done in cloud computing, Internet of Things and M2M communication was presented as well as the need for their integration. In section 3, description of current M2M protocols adopted by ETSI. In section 4, a detailed description of the proposed secure machine-to-internet clouding architecture is given. In section 5, description of simulation environment was made. In section 6, description of parameters used during simulation was done. In section 7, experimental results were provided. In section 8, conclusions and future work were given. Finally, references were provided.

2 Merge of Cloud computing, Internet of Things, and M2M Communication

In the following sections, few important characteristics of Cloud are going to be described. The design of Cloud can be partitioned into four layers: datacenter (equipment), infrastructure, platform, and application [11]. Every one of them is considered as a service for the layer above and as a consumer for the layer beneath. By and by, Cloud services can be gathered in three fundamental categories: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). SaaS is related to the monitoring of applications running on Cloud environments. Applications can typically be reached by a thin client or a web browser. PaaS is related to platform-layer resources (such as, operating system support, software development frameworks, etc.). IaaS is related to delivering processing, storage, and network resources, enabling the consumer to manage the operating system, storage and applications. It has attracted the biggest interest so far. There are different types of Clouds' deployment that have been determined [10, 11], as mentioned in the following: (i) Private Cloud designed for use by a single organization, typically possessed, directed, and operated by the organization itself; (ii) Community Cloud – designed for use by a specific community of consumers that have common concerns; (iii) Public Cloud – designed for open use by the public; (iv) Hybrid Cloud – is composed of two or more distinct Cloud infrastructures (private, community, or public); (v) Virtual Private Cloud – designed to address issues related to public and private Clouds, benefiting from advantage of virtual private network (VPN) technologies for enabling business owners to setup desired network settings (such as, security, topology, and so on).

Cloud computing model frees the business owner from the need to invest in the infrastructure, enabling him to rent resources according to his needs and only pay for the usage, then, it becomes attractive. Also, it enables decreasing operating costs, as service providers do not have to rent capacities according to peak load, and resources are left when service demand becomes low. In addition to these economic advantages, Cloud computing provides a number of technical benefits, such as, energy efficiency, optimization of hardware and software resource utilization, elasticity, performance isolation, and flexibility. The two terms of Cloud and IoT have evolved rapidly and independently. These terms are different from each other and, their characteristics are complementary, as Table [15] shows.

Table 1. Complementary aspects of Cloud and IoT

	IoT	**Cloud**
Displacement	pervasive	centralized
Reachability	limited	ubiquitous
Components	Real world things	Virtual resources
Computational capabilities	Limited	Virtually unlimited
Storage	Limited or none	Virtually unlimited
Role of the Internet	Point of convergence	Means for delivering services
Big data	source	Means to manage

For this reason, many researchers have suggested complementary characteristics of cloud and IoT, and have proposed integration, generally to obtain benefits in specific application scenarios [16, 17]. Generally, IoT can benefit from the virtually unlimited capabilities and resources of Cloud to compensate its technological limitations (such as, storage, processing, communication). Cloud can provide an efficient solution for IoT service management and composition as well as for applying applications and services that benefit from the things or the data generated by them. Also, cloud can exploit IoT by expanding its scope to manage real world things in a more distributed and dynamic manner, and for presenting new services in a large number of real life cases. Cloud can deliver the intermediate layer between the things and the applications, preventing all the complexity and functionalities necessary to use the latter. This has impact on future application design, because information collecting, executing, and transmission will create new challenges, especially in a multi-cloud environment. It is believed that Cloud fills some gaps of IoT (such as, the limited storage). And, some see IoT filling gaps of Cloud (such as, the limited scope). Most of these drivers pushing cloud and IoT integration fall in three categories that are communication, storage, and computation, while there exist other basic traversal drivers. IoT is characterized by a very high heterogeneity of devices, technologies, and protocols, but, it lacks different important characteristics such as scalability, interoperability, flexibility, reliability, efficiency, availability, and security. On the other hand, Cloud has proved to deliver them [18, 19], then, they can be identified as some of the main transversal drivers for cloud and IoT integration. There are two other transversal drivers, which are the ease of use and the reduced cost delivered by both users and providers of applications and services [19].

3 M2M communication and high level architecture

3.1 Background on M2M communication

M2M communication tries to reach the vision of connected things, or what is meant by Internet of Things (IoT) [20] [4], through a variety of possible uses in a world where intelligent applications provide a better and safer world. Also, the number of connected devices is rapidly increasing. International Data Corporation expects there will exist around 15 billion devices communicating over the network by the year 2015 [21], while Cisco Internet Business Solutions Group (IBSG) expects 25 billion devices connected to the Internet by 2015 and 50 billion by 2020 [22]. Machina Research white paper mentioned that by 2022, there will exist around 18 billion M2M connections in the world, up from approximately 2 billion today [23]. Ericsson claims that their vision of more than 50 billion connected devices by 2020 may appear realizable and within reach using the right approach [24]. Due to this rapid expansion, the concept of M2M communication is having more and more significance. Interoperability, between devices based on various access network technologies (e.g. mobile (2G/3G/4G), Wi-Fi, Bluetooth), with different platforms and data models is still very limited.

M2M (Machine-to-Machine) communication is initiated between two or more entities without any direct human intervention [25]. Actors in this environment are broad range of communication capable devices, such as, computers, mobile phones, tablets, a variety of sensors, actuators, pieces of industrial and medical equipment, and other everyday devices [26].

3.2 M2M high level architecture defined by ETSI

ETSI's work determined a high-level architecture view describing all constituents of M2M systems, their roles, and relationships. The high-level architecture of M2M system is composed of two main parts, which are Device and Gateway Domain, and a Network Domain, as shown in Figure 1 [27]. The device and gateway domain is consisting of the following elements:

- M2M Device: executes M2M Device Applications (DA) using M2M Device Service Capabilities Layer (DSCL).

- M2M Gateway: executes M2M Gateway Applications (GA) using M2M Gateway Service Capabilities Layer (GSCL).
- M2M Area Network: conveys connectivity based on Personal or Local Area Network technologies (e.g. ZigBee, Bluetooth) between M2M devices and M2M gateways.

The network domain is comprising of the following components:

- M2M Access Network: empowers M2M devices and M2M gateways to communicate with the Core Network. It can rely upon any of the following existing access network solutions: Digital Subscriber Line (DSL), satellite, GSM EDGE Radio Access Network (GERAN), Universal Terrestrial Radio Access Network (UTRAN), evolved UTRAN (eUTRAN), Wi-Fi (IEEE 802.11), and Worldwide Interoperability for Microwave Access (WiMAX), that can be used for M2M communication when needed.
- M2M Core Network: allows interconnection with other networks, delivers IP connectivity or other connectivity choices, service and control functions, and roaming. Similar to access network, it can depend on different existing core networking (CN) solutions (3GPP CN, ETSI Telecoms & Internet converged Services & Protocols for Advanced Networks (TISPAN) CN, and 3GPP2 CN) that can be changed to meet certain M2M communication requirements when needed.
- M2M Network Service Capabilities Layer (NSCL): delivers shared M2M functions by different M2M applications.
- M2M Applications: execute the service logic and implement M2M service capabilities available through open interfaces.
- M2M Network Management Functions: is composed of all the functions, such as, provisioning, supervision, and fault management needed to deal with access and core networks.
- M2M Management Functions: is composed of all the functions, such as, M2M Service Bootstrap Function (MSBF) implemented to simplify the bootstrapping of permanent M2M service layer security credentials needed to deal with M2M service capabilities in the network domain.

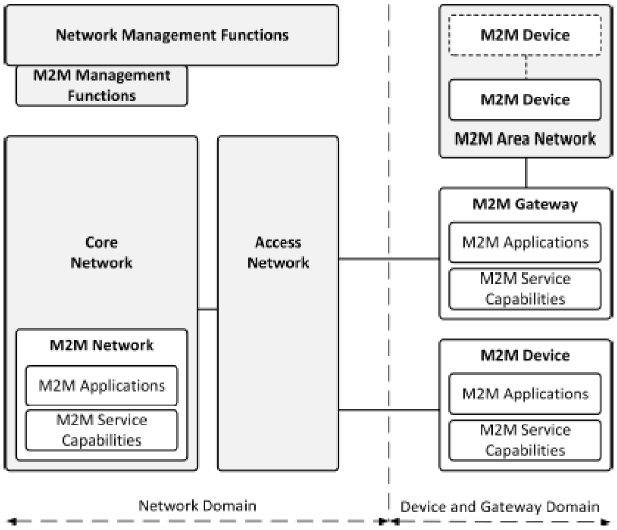


Fig. 1. High-level architecture of M2M system defined by ETSI

4 The proposed Secure Machine-to-Internet Clouding (SM2IC) security technique

4.1 Communication process in the proposed security technique

The idea proposed in this paper is based on presenting secure smart home connections between the user and his home devices. So, any user can open, shut down and monitor his home devices from outside his home using his smartphone or any similar device, besides, monitoring and controlling the quality of service delivered from his home devices from abroad.

The proposed technique provides a detailed description of how to initiate a secure connection from a user's device, such as, PC or tablet and so on to a central controller home device. This central controller device then sends signals to other desired home devices (e.g. microwave, TV, Lighting system and so on.) to control the status of these devices. The user's request communicates with the Internet using the hypertext transfer protocol, known as HTTP, then, the internet passes this request to the central device using the Https (or secure hypertext transfer protocol), also, the central controller device's hardware is Raspberry pi enabled, and this device can exchange data with the internet using Get and Post commands through Https. In its turn, the central controller device communicates in both directions with home devices previously mentioned using M2M network through one of the following technologies (Bluetooth, ZigBee, radio waves, Wi-fi ……. and so on).

The user's holding his device with cloud applications (such as, PaaS, IaaS and SaaS) receives a feedback from the home devices to the central device then to the Internet to his device about the completion of the secure connection and about the status of his home devices. The cloud applications residing in the user's device allow the user to monitor the quality of service (QoS) delivered through the whole process at the end home devices through giving precise measurements about the different parameters relating to the connection, internet, central device and controlled home devices, and this enables the user to control and modify the desired parameters according to what he expects.

The whole process starting from the connection initiation to the connection completion, must be secure and uncompromised. How security is achieved through this proposed security protocol, will be discussed in details through the following subsections. Fig.2 shows the whole connection process of the proposed security technique from the beginning to the end.

In Fig.3, the connection process of the proposed security technique is demonstrated focusing on the type of networks supporting M2M communication used and at the same time showing the different layers of the Internet of things. First, the user sends a request from his device containing cloud applications to change the status of his home devices. In this case, the user carrying his device represents the IoT application layer. The request traverses the Internet using Http protocol, then, the Internet forwards the user's request to M2M core network, which is responsible of connectivity with other networks, and includes both M2M applications and M2M service capabilities. Then, the request passes to the access network. The Internet, the M2M core network and the access network compose the IoT network layer. The request goes then to the M2M gateway, which contains both M2M applications and M2M service capabilities.

The M2M gateway represents the IoT service and application support layer. The user's request then reaches the M2M controller device, which contains M2M applications and M2M service capabilities. This

M2M controller device sends the user's request to the M2M Area Network, which works according to one of the following technologies (such as, Bluetooth, ZigBee, radio waves, Wi-fi …). In turn, the M2M Area Network forwards the user's request to the desired M2M devices, which include M2M applications and M2M service capabilities. The M2M controller device, M2M Area Network and M2M devices constitute the IoT device layer. Then, the feedback carrying the devices status is sent to the direction of the user, traversing all the preceding layers. The user can monitor and modify the home devices state using cloud applications according to his will. Then, the communication process in this technique occurs in both directions (indicated by lines with arrows in both directions, in Fig.3).

In the following subsections, the proposed security technique is going to be described in details. Fig.4 illustrates the proposed security technique for M2M communication through steps. In the first step, the user initiate a secure connection from his device, which contains cloud applications to monitor or modify the parameters of the whole process. The encryption and decryption processes are performed according to a proposed security technique in this paper, named Double Key Secure Internet (DKSI), and this new security technique is going to be explained in section 4.2.

The user's device must generate a connetion request number, composed of the user's device ID encrypted using the proposed Double Key Secure Internet (DKSI) tehnique and the key used to encrypt it, as well as, an authentication setting number, containing the user's password encrypted using the DKSI technique by the same key used to encrypt the user's device ID. The user's connection request traverses the Internet, the M2M core network, access network and M2M gateway to reach the M2M controller.

In the second step, the M2M controller checks the connection request number and the authentication setting number by decrypting the device's ID and the user's password using DSA or RSA technique with the first encryption key sent by the user. If the user's device ID and the user's password are verified, then, the controlller device can transport the requested tasks to the desired devices.

In the third step, the M2M controller device generates a temporary conection key, containing the controller device ID encrypted using the DKSI encryption technique by a new generated key from the M2M controller, then, the temporary connection key is encrypted for the second time using the user's first key generated at the beginning of the connection with the DKSI encryption technique. Then, the user's request containing connection keys passes from the M2M controller device to the M2M required devices through M2M area network.

In the fourth step, the required M2M devices have to check the temporary connection key containing the controller device ID and the second encryption key generated by the controller device, as well as the first encryption key sent from the user's device. First, the connection key is decrypted using the user's device generated key, then, it is decrypted for the second time using the second key generated by the controller device. Once, the controller device's ID is checked and verified, then, the desired devices status can be changed.

Also, each connected device has to create a new connection key, containing its ID encrypted with the DKSI technique using the second key, which was generated by the controller device, then, this connection key is encrypted for the second time using the user's device generated key. Then, the new connection key has to reach the controller device.

In the fifth step, the controller device checks the connected M2M devices IDs by decrypting the connected devices IDs using firstly the user's device generated key, then, secondly using the controller's device generated key with the DKSI technique. Once, their IDs are verified, then, the controller device encrypt its own ID and the connected M2M devices IDs separately using the user's device generated key with the DKSI technique forming two new connection keys, and then relay them to the user's device.

In the sixth step, the user decrypts the controller device's ID and the connected M2M devices IDs, using the key generated from his device with the DKSI technique to be verified. Then, the user can monitor the connected devices from his device and send a feedback through his device to change connected devices status. Apart from the different steps of the security technique, the desired devices status are sent from the user at the connection initiation, to the M2M controller device by passing through the Internet, the M2M core network, the access network, and the M2M gateway. Then, the M2M controller device retransmits the desired devices required status to the requested devices by passing through the M2M area network. Then, the requested devices resend back their encrypted IDs and their modified status to the M2M controller device again through the M2M area netwok. Finally, the M2M controller device resends the desired devices encrypted IDs and their modified status back to the user's device by traversing the M2M gateway, access network, M2M core network and the Internet. And, the user can modify the home devices status again as desired.

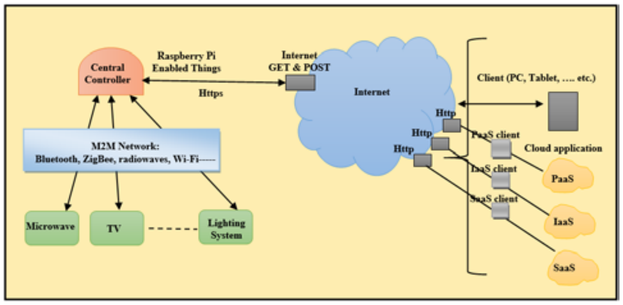


Fig.2 Description of the whole connection process of the proposed security technique

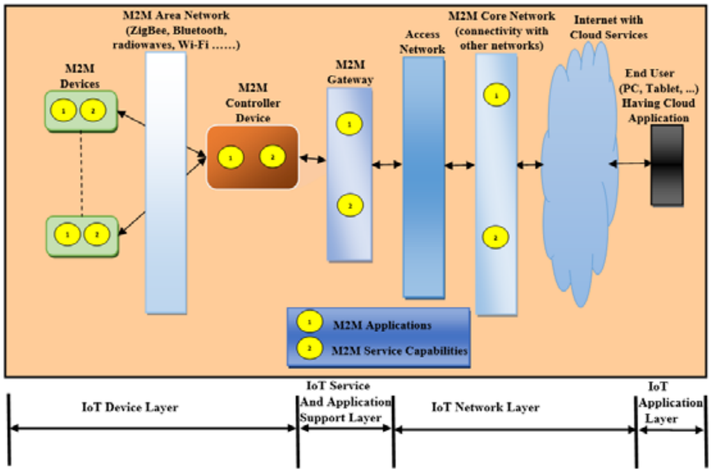
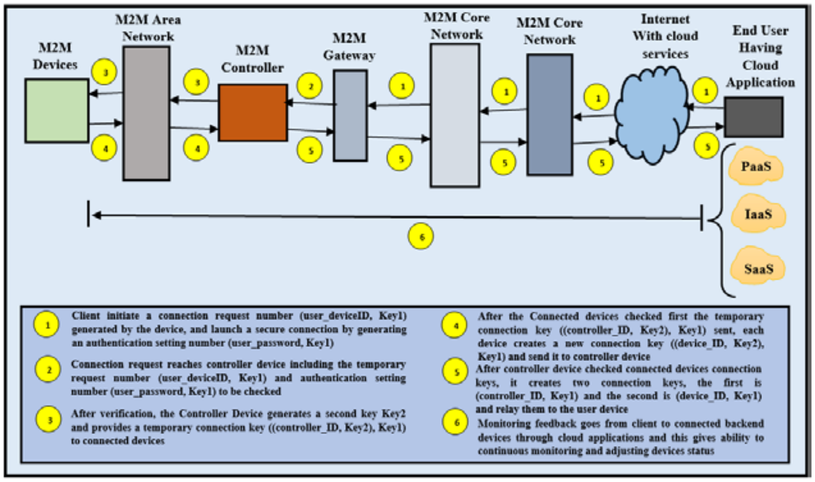

M2M Area Network (ZigBee, Bluetooth, radiowaves, Wi-Fi ......)
M2M Devices
1
2
M2M Controller Device
M2M Gateway
Access Network
M2M Core Network (connectivity with other networks)
Internet with Cloud Services
End User (PC, Tablet, ...) Having Cloud Application
1 M2M Applications
2 M2M Service Capabilities
IoT Device Layer
IoT Service And Application Support Layer
IoT Network Layer
IoT Application Layer

Fig.3 The whole connection process of the proposed security technique focusing on the different networks types and the IoT layers



1 Client initiate a connection request number (user_deviceID, Key1) generated by the device, and launch a secure connection by generating an authentication setting number (user_password, Key1)

2 Connection request reaches controller device including the temporary request number (user_deviceID, Key1) and authentication setting number (user_password, Key1) to be checked

3 After verification, the Controller Device generates a second key Key2 and provides a temporary connection key ((controller_ID, Key2), Key1) to connected devices

4 After the Connected devices checked first the temporary connection key ((controller_ID, Key2), Key1) sent, each device creates a new connection key ((device_ID, Key2), Key1) and send it to controller device

5 After controller device checked connected devices connection keys, it creates two connection keys, the first is (controller_ID, Key1) and the second is (device_ID, Key1) and relay them to the user device

6 Monitoring feedback goes from client to connected backend devices through cloud applications and this gives ability to continuous monitoring and adjusting devices status

Fig.4 the proposed security technique steps for M2M communication

4.2 Encryption and decryption techniques in the proposed security technique

4.2.1 Hash Function Used in the Proposed Security Technique

Most known cryptography techniques, such as RSA [28], DSA [29] or elliptic curve [30] encryption techniques utilize hash functions to ensure security of the user's data. Hash functions are created in one-way and can not be reversed or decrypted. In the proposed security technique (DKSI), the hash function SHA-2 was used. The SHA-2 is going to be described in the following section.

SHA-2 (Secure Hash Algorithm 2) [31] comprises of a set of cryptographic hash functions made by the United States National Security Agency (NSA). Cryptographic hash functions are considered as mathematical tasks that process digital data; by looking at the processed "hash" to a known and evaluated hash value, a person can estimate the data's integrity.

Also, evaluating the hash of a downloaded document and contrasting the outcome with a formerly created hash result can identify whether the download has been changed or altered. A key property of cryptographic hash functions is their resistance against collision; no one is capable of discovering two distinctive input values that deliver the same hash output. SHA-2 has huge changes from its antecedent, SHA-1.

The SHA-2 family is comprising of six hash functions with digests (hash values) that are 224, 256, 384 or 512 bits: SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256. SHA-2 was announced in 2001 by the National Institute of Standards and Technology (NIST) a U.S. federal standard (FIPS). Today, the best public attacks break preimage resistance for 52 rounds of SHA-256, and collision resistance for 46 rounds of SHA-256. The table 2 below show the most common properties for sha-256 hash function.

Table 2. the most common properties for sha-256 hash function

Algorithm and variant		Algorithm size (bits)	Internal state size (bits)	Block size (bits)	Max message size (bits)	Rounds	Operations	Security bits (Info)	Example Performance (MiB/s)	First published
SHA-2	SHA-224 SHA-256	224 256	256 (8x 32)	512	2^{64} - 1	64	And, Xor, Rot, Add (mod 2^{32}), Or, Shr	112 128	139	2001

Based on its characteristics, Sha- 256 was chosen to be used as a hash function in the proposed technique, due to its relative proven strength against attacks and collisions.

4.2.2 Proposed DKSI (named double key secure internet) encryption technique

In this paper, a new encryption technique to ensure security of data transactions is proposed. This technique implements SHA-2 (as a hash function) instead of SHA-1, because SHA-2 provides better security as explained in previous section 4.2.2. The proposed DKSI technique is performed on the user's ID, password and generated key. It depends on transforming the user's device ID into binary representation, then performing some transformations and logical operations on it, and retransforming it using a hash function along with some mathematical operations. The steps of the proposed DKSI encryption technique are described below as follows:

1. Take the user's device ID in combined representation using letters and numbers, for example, EAT56TYUI
2. Transform each letter or number to its ASCII code, then to its binary representation, composed of 8 bits; 0's and 1's
3. Generate the first user's key of a random 0's and 1's, but it must be the same length of the Device ID binary representation
   using Key = rand(1, len), where len is the length of the Device ID binary representation
4. ANDing the binary representation of the Device ID and the generated key
5. Shift to the left twice the output result and the generated key and put 2 0's at the left of both of them
6. The first two bits shifted from the left, can be: 00, 01, 10 and 11, and put them at the end of the right of both of the result and the key

Note: the output result represents the Device ID primary encryption. Steps 5 and 6 is performed for both the result from step 4 and the user generated key. The output length from step 6 is equal to the length of the binary representation of the Device ID plus two bits, and the length of the key in binary is increased by two bits.

7. Transform the primary encrypted device_id (Device ID') to numerical representation
8. Make hash function of the primary encrypted device_ID using SHA-2/256(Device ID')
9. Apply mathematical operations to the to the hashed primary encrypted device_ID after transforming it to hexadecimal format and using the logarithmic function, as follows, Final encrypted Device_id = [(numerical (hashed Device_ID'))*log (len)] ^{K}
10. Transform the user generated shifted key from its binary representation to numerical representation to become (key')
11. Apply some modifications on the user generated key before sending it, by New_Key = (K*key')/(M*P)

Assumptions for this encryption technique are described below:

- ☒ Where K, P and M are known numbers saved at the user's device, central controller, home devices,
- ☒ Also, the user_mobile_id and password are known to the central device
- ☒ The central device_id is known to the home devices
- ☒ The home devices ids are known to the the central device_id
- ☒ The central device_id and home devices ids are known to the user's mobile

Note: the user password is encrypted the same way as user's device ID following steps to 9 by using the same user generated key, and finally, the encrypted Device_ID, the encrypted password along with the transformed key are all sent to the central controller device for verification.

The proposed DKSI encryption technique would be applied later on central device ID, as well as home devices IDs by using the same 9 steps performed on the user's device_ID. At the central controller device, the received encrypted Device_ID and user's password are decrypted using the transformed user's key sent with them, once they are verified, the central controller device generates a second key using the same steps 3, 5, 6, 10 and 11 as performed for the user generated key. But, in this case, there exist double encryptions for the central device ID, by encrypting it using key 2 (central device generated key) then using key 1 (user's device generated key) implementing the same steps to 9 as done for the encryption of the user's Device_ID and password at the user's device first, and the central Device ID, along with key 1 and key 2 are sent to home devices for verification.

Then, at the home devices, once the central device _ID is verified after decrypting it, the home devices status are changed as required by the user, and the home devices IDs are encrypted using the same steps to 9 mentioned above using key2. Then, the encrypted home devices IDs are sent with key 2 back to the central controller device. Then, the central controller device checks the home devices IDs by decrypting them and comparing them to the IDs stored inside it, once, the home devices IDs are verified, the central controller device Id and the home devices IDs are encrypted by key 1 using the proposed DKSI technique steps to 9, and then, sent along with key 1 to the user's device to be verified. Finally, at the user's device, the received central controller device ID and home devices IDs are decrypted using key 1 to be checked and compared to the ones stored inside the user's device, once they are verified, a monitoring feedback can propagate easily from the user to the central controller device and the connected backend home devices to enable the user easily to monitor the status of these devices. The decryption process is going to be described in the following section.

4.2.3 Proposed DKSI (named double key secure internet) decryption technique

The inverse of the steps explained in the encryption is done in the decryption to recover the original user Device_ID and password, to check them against saved ones in the central device. The following decryption steps are reapplied every time there is a need for decryption in the proposed security DKSI technique. The decryption steps of the proposed DKSI security technique are described below:

Recover first the key by using these steps;

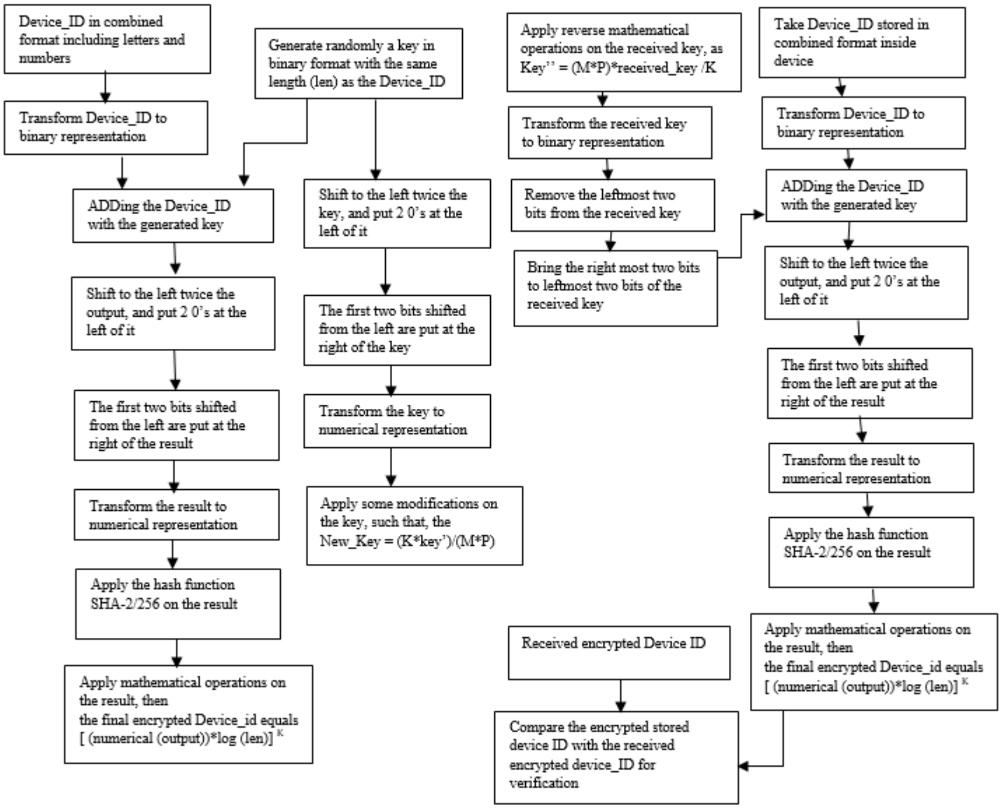
1. Apply reverse mathematical operations to that performed in encryption on the received key such as, Key'' = (M*P)*received_key /K;
2. Transform the received modified key (key'') from numerical representation to 8 bit binary representation.
3. Remove the leftmost two bits in the binary representation that were added during the encryption process.

4. Bring the rightmost two bits that were shifted during the encryption to the leftmost two bit locations, then, the original key was restored and the original Device_ID need to be found.

Recover second the original Device_ID by implementing these steps;

1. Apply the steps to 9 mentioned in the encryption above using the recovered key on the user's Device_ID, which is stored inside the central controller device to encrypt it, the result of the encryption is a vector.
2. Compare the newly encrypted user's Device_ID by the central controller device with the received encrypted user's Device_ID from the user's device, because the hash function SHA-2 cannot be reversed, so we have to repeat the DKSI encryption steps on the stored user's Device_ID. If the newly encrypted user's Device_ID matched the received encrypted user's Device_ID, then, the user's Device_ID is verified, and same steps of encryption to 9 are applied on the user's password stored inside the central device using the recovered key, then, the newly encrypted password is compared to the received encrypted user's password to be verified. Once both the user's Device_ID and password are verified, then, encrypt the central controller device ID using key2 generated by it, then, by using the recovered key 1, and send them to the home devices for verification, and so on as explained earlier in the DKSI technique.

Note: the decryption steps are repeated every time there is a need during the DKSI security technique. Fig. 5 illustrates encryption and decryption processes of the DKSI security technique.


Device_ID in combined format including letters and numbers
Generate randomly a key in binary format with the same length (len) as the Device_ID
Transform Device_ID to binary representation
ADDing the Device_ID with the generated key
Shift to the left twice the key, and put 2 0's at the left of it
Shift to the left twice the output, and put 2 0's at the left of it
The first two bits shifted from the left are put at the right of the key
The first two bits shifted from the left are put at the right of the result
Transform the key to numerical representation
Transform the result to numerical representation
Apply some modifications on the key, such that, the New_Key = (K*key')/(M*P)
Apply the hash function SHA-2/256 on the result
Apply mathematical operations on the result, then the final encrypted Device_id equals [(numerical (output))*log (len)] K
Apply reverse mathematical operations on the received key, as Key'' = (M*P)*received_key /K
Take Device_ID stored in combined format inside device
Transform the received key to binary representation
Transform Device_ID to binary representation
Remove the leftmost two bits from the received key
ADDing the Device_ID with the generated key
Bring the right most two bits to leftmost two bits of the received key
Shift to the left twice the output, and put 2 0's at the left of it
The first two bits shifted from the left are put at the right of the result
Transform the result to numerical representation
Apply the hash function SHA-2/256 on the result
Received encrypted Device ID
Apply mathematical operations on the result, then the final encrypted Device_id equals [(numerical (output))*log (len)] K
Compare the encrypted stored device ID with the received encrypted device_ID for verification

DKSI Technique Encryption Steps

DKSI Technique Decryption Steps

Fig.5 Encryption and decryption of the DKSI security technique

5 Simulation Environment

5.1 MATLAB Simulink

Simulink [32] is made of a block diagram environment for multidomain simulation and Model-Based Design. It allows simulation, automatic code generation, and continuous test and verification of embedded systems. Simulink possesses a graphical editor, adaptable block libraries, and solvers for modeling and simulating dynamic systems. It is converged with MATLAB, enabling the user to integrate MATLAB algorithms into models and generate simulation results to MATLAB for investigation and assessment. Engineers everywhere use Simulink to realize their ideas off the ground, including reducing fuel emissions, developing safety-critical autopilot software, and designing wireless LTE systems.

Simulink delivers built-in support for prototyping, testing, and executing models on low-cost target hardware, such as Arduino, LEGO MINDSTORMS NXT, and Raspberry Pi. A client can create algorithms in Simulink for control systems, robotics, sound processing, and computer vision applications and see them working progressively.

Using Simulink Desktop Real-Time, a user can run Simulink models in real time on Microsoft Windows PCs and MacOS and link to a range of I/O boards to create and manage a real-time system. To process a model in real time on a target computer, Simulink Real-Time for Hardware-In-the-Loop (HIL) simulation, rapid control prototyping, and other real-time testing applications can be used.

With Simulink, the user can develop algorithms and models, and process them on low-cost embedded hardware including Arduino, LEGO MINDSTORMS NXT and EV3, and Raspberry Pi. Development for a range of embedded hardware applications such as control systems, robotics, audio processing, and computer vision can be performed. Simulink support for low-cost embedded hardware is existing in student and home-use versions.

5.2 Raspberry pi 3 general specifications

The Raspberry Pi [33] is composed of a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to be used in teaching of basic computer science in schools and in developing countries. The original model became far more popular than expected, selling outside of its target market for uses such as robotics. Peripherals (including keyboards, mice and cases) are not included with the Raspberry Pi. A few accessories anyway have been incorporated into several official and informal bundles. A few generations of Raspberry Pis have been discharged. Raspberry Pi 3 Model B was discharged in February 2016 and was packaged with on-board WiFi, Bluetooth and USB boot capacities. As of January 2017, Raspberry Pi 3 Show B was the freshest mainline Raspberry Pi.

All Raspberry Pi models possess a Broadcom system on a chip (SoC), which includes an ARM compatible central processing unit (CPU) and an on-chip graphics processing unit (GPU, a VideoCore IV). CPU speed varies from 700 MHz to 1.2 GHz for the Pi 3, and on board memory varies from 256 MB to 1 GB RAM. Secure Digital(SD) cards are utilized to store the operating system and program memory in either the SDHC or MicroSDHC sizes. Most boards possess between one and four USB slots, HDMI and composite video output, and a 3.5 mm phono jack for audio. Lower level output is provided by a number of GPIO pins which support common protocols like I²C. The B-models have an 8P8C Ethernet port, and the Pi 3 has on board Wi-Fi 802.11n and Bluetooth. The Raspberry Pi 3, has a quad-core Cortex-A53 processor. This model was expected to be highly dependent upon task threading and instruction set use. The Raspberry Pi 3 is equipped with 2.4 GHz WiFi 802.11n (150 Mbit/s) and Bluetooth 4.1 (24 Mbit/s) based on Broadcom BCM43438 FullMAC chip with no official support for Monitor mode but used through unofficial firmware patching and also has a 10/100 Ethernet port.

The Raspberry Pi Foundation recommends the use of Raspbian, a Debian-based Linux operating system. Other third party operating systems available via the official website are Ubuntu MATE, Snappy Ubuntu Core, Windows 10 IoT Core, RISC OS and specialised distributions for the Kodi media center and classroom management. It presents Python and Scratch as the main programming language, with support for many other languages. The default firmware is closed source, while an unofficial open source is available. Many other operating systems can also execute on the Raspberry Pi.

5.3 Simulink Support Package for Raspberry pi capabilities and features

Simulink Support Package for Raspberry Pi empowers you to create algorithms in Simulink, a block diagram environment for designing dynamic systems and creating algorithms, and execute them independently on your Raspberry Pi. The support package broadens Simulink with blocks for adjusting your Raspberry Pi, sending and accepting UDP packets, and reading and writing information from sensors. This includes writing information to the free ThingSpeak information aggregation service for Internet of Things applications.

In the wake of making your Simulink demonstrate, you can simulate it, tune algorithm parameters until you get it just right, and download the finished algorithm for independent execution on the device. With the MATLAB Function block, you can incorporate MATLAB code into your Simulink model. Using Simulink support package for Raspberry Pi, you compose the algorithm in Simulink and implement it to the Raspberry Pi utilizing automatic code generation. Execution is then performed on the Raspberry Pi. Utilizing Simulink for Raspberry Pi programming empowers you:

- Create and mimic your algorithms in Simulink and utilize automatic code generation to execute them on the device
- Incorporate signal processing, control configuration, state logic, and other advanced math and engineering schedules in your Raspberry Pi programming projects
- Intelligently tune and advance parameters as your algorithm runs on your Raspberry Pi

Notwithstanding utilizing Simulink Support Package for Raspberry Pi, you can deliver clear and convenient C code from MATLAB algorithms and actualize it on a Raspberry Pi utilizing Raspberry Pi support from MATLAB Coder. Simulink Support Package for Raspberry Pi influences you to create algorithms that execute independent on your Raspberry Pi. The support package broadens Simulink with blocks to guide Raspberry Pi digital I/O and read and write information from them. In the wake of building up your Simulink model, you can mimic it and download the finished algorithm for independent execution on the device. One particularly useful (and unique) capability provided by Simulink is the ability to tune parameters live from your Simulink model while the algorithm executes on the hardware.

5.4 Laptop specifications

Simulation of the smart home scenario was accomplished on a Dell laptop model Inspiron 15500 series, having the following specifications illustrated in table m below. A 64-bit operating system was used in simulation, because Matlab R2017a must be installed on 64-bit operating system machine. The Laptop used in simulation has 8 GB RAM and 2.40 GHz speed, since Matlab R2017a requires a computer with good speed and acceptable RAM. The rest of the Dell Laptop specifications is mentioned in the following table 3.

Table 3. Operating System Specifications

Laptop Model	Dell model Inspiron 15 5000 series includes Nvidia Geoforce and Ubuntu
Laptop processor	Intel(R) Core(TM) i7-5500 CPU @ 2.40GHz 2.40 GHz
System Type	64 bit operating system
Maximum possible array MATLAB can create	12445 MB
Memory available for all arrays and data	12445 MB
Memory used by MATLAB	2402 MB
Computer Physical Memory (RAM)	8102 MB
Physical memory and paging system	14263 MB

5.5 Smart home scenario to be implemented using MATLAB

The proposed security technique to be implemented in a smart home scenario can be built using MATLAB R207a Simulink Raspberry pi toolbox. The scenario begins when the user clicks from his android mobile phone a button to request the change of home devices status or switching them ON using android toolbox blocks; which provides blocks enabling user interaction inside his android enabled smartphone. Raspberry pi toolbox provides blocks to simulate a Wi-Fi UDP send and receive blocks for wireless communication. So, the proposed security technique is composed of three main parts; the first part represents the user clicking on his smartphone button to switch on/off home devices, then the request is sent wirelessly to the Raspberry pi enabled controller device. The second part represents the raspberry pi enabled controller device sending wirelessly the user request to the raspberry pi enabled home devices, after verifying the coming data. The third part represents the raspberry pi enabled home devices after verifying received data, change the status of home devices; here switching Raspberry pi LED ON/OFF. Figure 6 represents the main blocks of the smart home scenario to be implemented using MATLAB Simulink illustrating how they exchange data wirelessly. The figures of three parts constituting the proposed security technique are provided in the experimental results section.

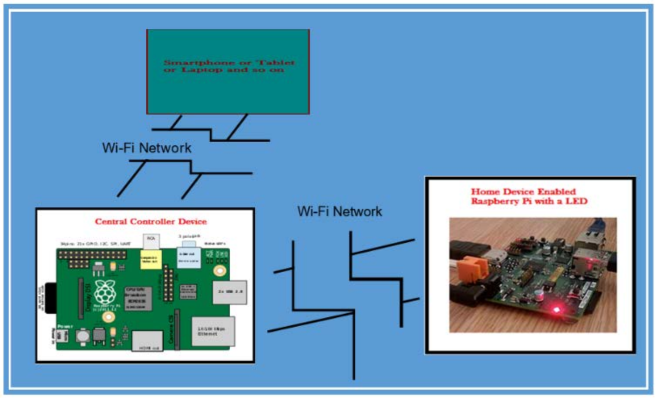

Smartphone or Tablet or Laptop and so on
Wi-Fi Network
Central Controller Device
Wi-Fi Network
Home Device Enabled Raspberry Pi with a LED

Fig.6 Real life scenario to be implemented using MATLAB Simulink blocks and Raspberry pi toolbox

6 Description of parameters used in simulation

The parameters evaluated during simulations are going to be described in the following subsections.

6.1 Response time

Real-time systems exist in the world around us. A modern car is considered as an example of a real-time system. Any person using the car will most likely want to have guarantees about the car's behavior. If the brakes need to be replaced in a nearby future, a lamp should indicate this, and not by the user who declares that there no longer exists any braking effect. A task is a program that performs some service or functionality in the system, like checking the brakes. A task's reaction time can be portrayed as the required time for checking the brakes. To be fit for giving ensures in continuous real-time systems, one must know the response times of tasks. If a message is sent from a source to a destination, the response time can be calculated as the time the message takes from the source to the destination. The factors are considered for evaluating the response time are the bandwidth of the network and the message size, and both are determined by symbols as follows:

M: the message size
$B_{wireless}$: the bandwidth of the wireless network

$$\textit{Response time} = M / B_{wireless}$$

6.2 Memory consumption

To calculate the total memory consumed [34], it is necessary to calculate the number of concurrent users, the domain of the system, the amount of memory required per user, the buffer cache compensation, the number of virtual machines allocated and the system excess rate to estimate the memory size based on the data obtained through the investigation. Also, the system domain contains spaces for the OS, DMBS, engine, middleware engine, and other utilities. The result of the estimation of the memory amount can be expressed as follows.

$$\textit{Total Consumed Memory} =(T1+M * q) * p * o$$

T1= The total memory for the system domain
M = The number of the virtual machines allocated
q = The amount of the required memory per user
p = Buffer cache compensation
o = system excess rate
By using the formula above, the result can be calculated as (384 + 959 * 2) * 1. 2 * 1. 3 = 3,591 MB. In consideration of the unit of memory expansion, the amount will be estimated as to b be 8,192MB.

6.3 Power consumption of transmitted signals

There exist cell phone base station tower networks across many nations globally, but there are still many areas within those nations that do not have good reception. Some provincial regions are probably not going to be successfully covered, in light of the fact that the cost of raising a cell tower is too high for just a couple of clients. In high reception zones, it is discovered that basements and the insides of vast buildings have poor reception. Weak signal strength can likewise be caused by damaging interference of the signals from nearby towers in urban territories, or by the construction materials used in few buildings, bringing about fast weakening of signal strength. Vast buildings for example warehouses, hospitals and manufacturing plants regularly have no usable signal more distant than a couple of meters from the outside walls. This is especially valid for the networks, which work at higher frequency, in light of the fact that these signals are lessened quickly by mediating obstacles, despite the fact that they can utilize reflection and diffraction to go around obstacles. The estimated received signal strength [35] in a mobile device can be calculated as follows:

$$dBm_e = -113.0 - 40.0\ \log_{10}\left(\frac{r}{R}\right)$$

More general, you can take the path loss exponent into consideration:

$$dBm_e = -113.0 - 10.0\,\gamma\ \log_{10}\left(\frac{r}{R}\right)$$

If the mobile device exists at *cell radius* distance from the cell tower, the *received power* is calculated as −113 dBm. The effective path loss is based on the frequency, the topography, and the environmental conditions. Actually, one could use any known *signal power* dBm_0 at any distance r_0 as a reference:

$$dBm_e = dBm_0 - 10.0\,\gamma\ \log_{10}\left(\frac{r}{r_0}\right)$$

Table 4 ilustrates the parameters of the received power signal.

Table 4. The parameters of the received power signal

Parameter	Description
dBm_e	Estimated received power in mobile device
−113	Minimum received power
40	Average path loss per decade for mobile networks
r	Distance mobile device - cell tower
R	Mean radius of the cell tower
γ	Path loss exponent (average value of 4 for mobile networks)

6.4 Bit error rate

In digital transmission, the number of **bit errors** [36] is considered as the number of received bits of an information flow over a communication channel that have been altered because of noise, interference, distortion or bit synchronization errors. The **bit error rate** (**BER**) is the number of bit errors per unit time. The **bit error ratio** (also **BER**) is the number of bit errors partitioned by the total number of exchanged bits amid an examined time interval. Bit error ratio is a unitless performance measure, frequently introduced as a percentage. The **bit error probability** p_p is the normal estimation of the bit error ratio. The bit error ratio can be resolved as a rough estimate of the bit error probability. This estimate is precise for quite a long time interval and a high number of bit errors.

Estimating the bit error ratio empowers individuals to choose the convenient forward error rectification codes. Since most such codes rectify just bit-flips, but not bit-inclusions or bit-erasures, the Hamming distance metric is considered as the proper technique to gauge the number of bit errors. Numerous FEC coders additionally constantly measure the current BER. A more broad technique for estimating the number of bit errors is the Levenshtein distance. The Levenshtein distance measurement is more appropriate for estimating raw channel performance before frame synchronization, and when utilizing error correction codes created to amend bit-inclusions and bit-erasures, such as Marker Codes and Watermark Codes. The BER is depicted as the likelihood of a bit distortion due to electrical noise $w(t)$. On the account of a bipolar NRZ transmission, we have $x_1(t) = A + w(t)$ for a "1" and $x_0(t) = -A + w(t)$ for a "0". Every one of $x_1(t)$ and $x_0(t)$ has a period of T. Realizing that the noise has a bilateral spectral density $\frac{N_0}{2}$,

$x_1(t)$ is $\mathcal{N}\left(A, \frac{N_0}{2T}\right)$

and $x_0(t)$ is $\mathcal{N}\left(-A, \frac{N_0}{2T}\right)$.

Coming back to BER, we have the likelihood of a bit distortion $p_e = p(0|1)p_1 + p(1|0)p_0$.

$p(1|0) = 0.5\,\operatorname{erfc}\left(\frac{A+\lambda}{\sqrt{N_o/T}}\right)$ and $p(0|1) = 0.5\,\operatorname{erfc}\left(\frac{A-\lambda}{\sqrt{N_o/T}}\right)$

where λ is considered as the threshold of choice, set to 0 when $p_1 = p_0 = 0.5$.

We can use the average energy of the signal $E = A^2T$ to suggest the last expression:

$p_e = 0.5\,\operatorname{erfc}\left(\sqrt{\frac{E}{N_o}}\right)$.

6.4 Strength of the password

6.4.1 Password Cracking

In cryptanalysis and computer security, password cracking [37] is considered as the way toward recuperating passwords from information that have been stored or transferred by a computer system. A common technique (brute-force attack) is to attempt surmises over and again for the password and check them against an accessible cryptographic hash of the password.–The objective of password cracking can be to enable a client to recoup a forgotten password (introducing a totally new password is to a lesser degree a security hazard, but it needs System Administration privileges), to get unauthorized access to a system, or as a preventive measure by system executives to check for easily crackable passwords. On a file-by-file premise, password cracking is made to gain access to digital evidence, for which a judge has empowered access however the specific file's access is confined. The best cracking password techniques are; dictionary attack, brute force attack, rainbow table attack, blogs, phishing, social engineering, malware, offline cracking, shoulder surfing, spidering, guess, and port scan attack.

6.6.2 Password Strength

Password strength [38] is depicted as the measure of a password's ability to resist password cracking attacks. In its typical shape, it estimates how many attempts an assailant who does not have direct access to the password would need, overall, to get it accurately. The strength of a password is considered a function of length, complexity, and unpredictability. Utilizing strong passwords diminishes large danger of a security break, yet strong passwords do not eliminate the requirement for other viable security controls.The strength of a password is defined by;

- **Length**: the number of characters the password incorporates.
- **Complexity**: does it utilize a blend of letters, numbers, and symbols?
- **Unpredictability**: is it something that can be speculated effectively by an assailant?

Let’s now look at a practical example. We will use three passwords namely

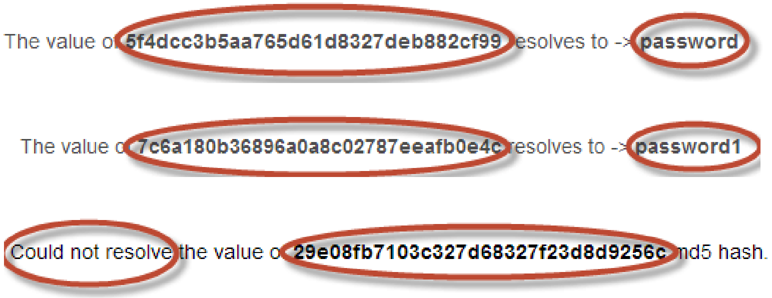
1. *password*
2. *password1*
3. *#password1$*

The higher the strength number, better the password. Let’s suggest that we have to save our above passwords using md5 encryption. We will use an online md5convertor to convert our passwords into md5 hashes. The table 5 below shows the password hashes.

Table 5. the passwords’ hashes

Password	MD5 Hash
password	5f4dcc3b5aa765d61d8327deb882cf99
password1	7c6a180b36896a0a8c02787eeafb0e4c
#password1$	29e08fb7103c327d68327f23d8d9256c

We will now use http://www.md5this.com/ to crack the above hashes. The images below illustrate the password cracking results for the above passwords. Fig.7 illustrates passwords’ hashes.



The value of **5f4dcc3b5aa765d61d8327deb882cf99** resolves to -> **password**

The value of **7c6a180b36896a0a8c02787eeafb0e4c** resolves to -> **password1**

Could not resolve the value of **29e08fb7103c327d68327f23d8d9256c** md5 hash.

Fig.7 passwords’ hashes

From the above outcomes, we figured out how to break the first and second passwords. We didn't figure out how to break the third password which is longer, perplexing and unexpected.

6.6.3 Types of Attacks against hash functions used in Passwords Encryption and their resistance

1. Collision attack

In cryptography, a **collision attack** [39] on a cryptographic hash tries to find two inputs generating the same hash value, i.e. a hash collision. This is different than a preimage attack where a specific target hash value is determined. There are briefly two types of collision attacks:

Collision attack

Find two different messages *m1* and *m2* such that *hash(m1) = hash(m2)*.

Chosen-prefix collision attack

Given two different prefixes *p1, p2* find two appendages *m1* and *m2* such that *hash(p1 ∥ m1) = hash(p2 ∥ m2)* (where ∥ is the concatenation operation).

2. Preimage Attack

In cryptography, a **preimage attack** [40] on cryptographic hash functions attempts to discover a message, that has a specific hash esteem (value). A cryptographic hash function should resist attacks on its preimage. With regards to attack, there exist two kinds of preimage resistance:

- *preimage resistance*: for basically all pre-determined outputs, it is considered as computationally infeasible to discover any input that hashes to that output, i.e., given y, it is hard to discover an x to such an extent that $h(x) = y$.
- *second-preimage resistance*: it is considered computationally infeasible to discover any second input which has an indistinguishable output as that of a predetermined input, i.e., given x, it is hard to discover a second preimage $x' \neq x$ with the end goal that $h(x) = h(x')$.

3. Collision resistance

Collision resistance [41] is considered as a property of cryptographic hash functions: a hash function H is collision resistant, if it is hard to find two inputs that hash to the same output; that is, two inputs a and b such that $H(a) = H(b)$, and $a \neq b$.

Collision resistance is an even harder property, which we still need for most usages of hash functions:
It is hard to find a pair of messages x1≠x2x1≠x2 with H(x1)=H(x2)H(x1)=H(x2).

Each hash function with bigger number of inputs than outputs will essentially cause collisions. Considering a hash function for example SHA-256, that produces 256 bits of output from an (discretionarily) arbitrarily extensive input. It must produce one of 2^{256} outputs for every member of a much bigger set of inputs. Collision resistance does not imply that no collisions exist; essentially that they are elusive.

4. Preimage resistance

Preimage resistance [42] is considered as the most fundamental characteristic of a hash function, which can be thought. It implies:

For a given h in the output space of the hash function, it is difficult to discover any message x with H(x)=h. Hard means takes additional time/costs than any (speculative aggressor) hypothetical attacker can contribute. In practice, uniqueness is not characterized by the (dynamic) abstract theoretical non-presence of collisions, but by the (pragmatic) practical non-presence of collisions. So as to discover a collision in SHA-256, you would (presumably) probably need to run the algorithm somewhere in the range of 2^{128} times. It is far-fetched that this will happen at any point in the near future, regardless of whether you check the total number of times SHA-256 will ever be processed by anybody in the whole universe combined. SHA-256 is thought to be practically difficult to "crack", that is, to recover the original plaintext from the hash.

7 Experimental Results

In the following subsections, the main parts of the security proposed technique are described. Then, each parameter of the five parameters described in the previous section 6 resulting from simulation is explained below.

7.1 Proposed security technique main parts

Part 1: android enabled device sending user’s device id and password to controller device

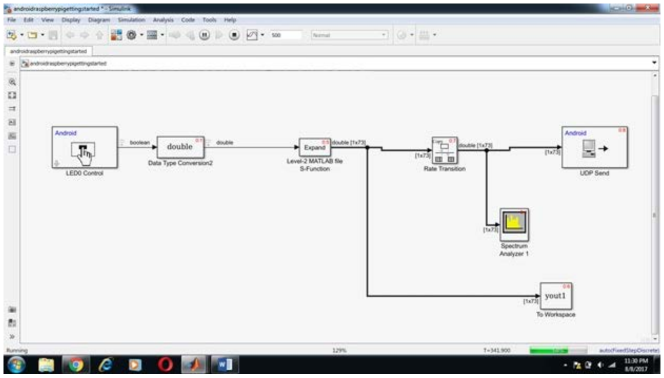


Fig.8 Process of sending user data from android device to controller device during simulation

In part 1, the user's sends the device id and password and key used for encryption to the controller device during simulation. Part 1 is represented in Fig.8. Main blocks of part 1 are: Android control button (from android toolbox and used for turning on/off remote devices through the controller device), double block (used to convert data to double), Level 2 MATLAB s-function (used for performing encryption), the To workspace block (carries yout1 3D array; which results after encryption data as mentioned in the proposed technique and represents the encrypted signal,and its role is to output the encrypted signal in the workspace), the rate transition block (to transform data in way that it can appear in the spectrum analyzer), the spectrum analyzer 1 block (to show signal), the android UDP send block (to send encrypted signal wirelessly to the controller device).

Part 2: controller device sending the controller device id and encryption keys to home devices

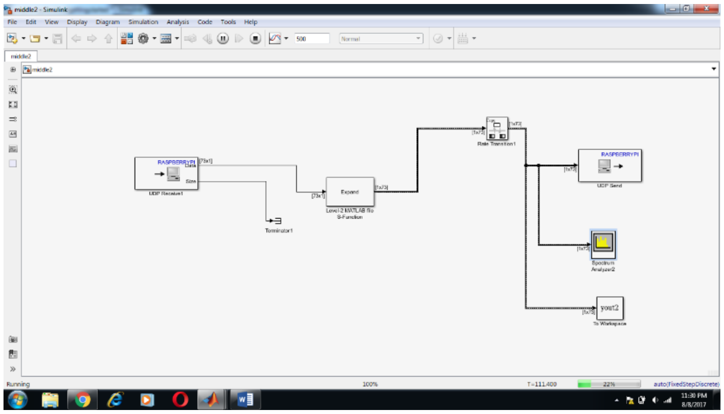


Fig.9. Process of controller device sending data to home devices during simulation

Part 2 represents the process of controller device sending data to home devices wirelessly during simulation. Part 2 is illustrated in Fig.9. Main blocks of part 2 are: the raspberry pi UDP receive block inside the controller device (from the raspberry pi toolbox (raspberry pi 3 model B); used to receive wirelessly the signal coming from the android device), the level-2 MATLAB function block (used for verifying the encrypted received signal and then sending encrypted device id and encryption keys to raspberry pi enabled home devices after verification), the rate transition block (used to enable encrypted signal that results from the level-2 MATLAB function to be displayed in the spectrum analyzer 2), the to workspace block (carries yout2 3D array; which results after encryption data and represents the second encrypted signal,and its role is to output the second encrypted signal in the workspace to be displayed), the raspberry pi UDP send block (from the raspberry pi toolbox (raspberry pi 3 model B), used to send data wirelessly to the raspberry pi enabled home device/s).

Part 3: the home devices verify the coming data and turn on/off connected device/s

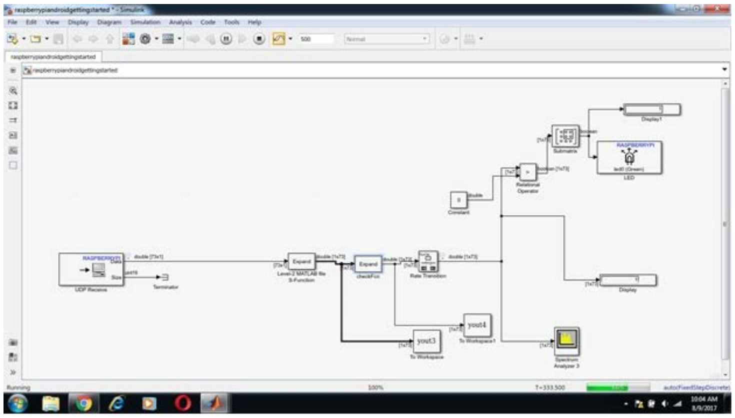


Fig.10 Process of homes devices verifying data and turning device LED ON/OFF during simulation

Part 3 represents the process of home devices verifying data and turning device LED ON/OFF during simulation. Fig.10. illustrates part 3. The main blocks of part 3 are: raspberry pi UDP receive block (from the raspberry pi toolbox (raspberry pi 3 model B); and shows the raspberry pi home enabled device receiving data wirelessly from the raspberry pi controller device), the Level-2 MATLAB s-function block (verify the received encrypted controller device id, after verification, send signals to next checkFCn block), the checkFCn MATLAB block (used to change the home device status if received data are true), the to workspace block yout3 (outputs 3D array resulting from the Level-2 s-function to the workspace), the to workspace block yout4 (outputs 3D array resulting from thr checkFcn block to the workspace), the rate transition block (allows received signal to be displayed in the spectrum analyzer 3), the spectrum analyzer 3 (displays received signals), the display block (display the received signal as a numerical array), the relational operator block (checks if the signal is greater than 0, its outputs 1 and then turns on the LED conncted to the device; else if the the signal is smaller than 0; it outputs 0 then turn off the LED device), the submatrix block (changes the size of the array to fit the LED input), the raspberry pi LED block (from the raspberry pi toolbox (raspberry pi 3 model B), represents the LED of the raspberry pi connected enabled home device).

Let's look at signals as they appeared in the spectrum analyzer. Signals appear as a line means devices receiving 0's or no data sent or received. The below two figures show the signal carrying the encrypted data as well as the keys when sent and received. Fig.11 shows spectrum analyzer 1 & 2 when receiving signal carrying encrypted data in parts 1 & 2. Fig.12 illustrates spectrum analyzer 3 upon receiption of signal carrying encrypted data in part 3.

The received signal in part 3 named yout 3 as appearing in the MATLAB interface in numerical representation is shown in Fig.13 below. The sent signals gives 1's after sending the encrypted data and keys during simulation.

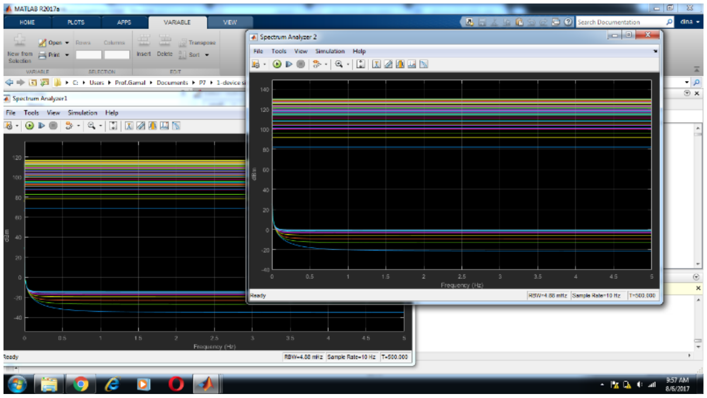


Fig.11. Spectrum analyzer 1 & 2 showing encrypted signal receiption

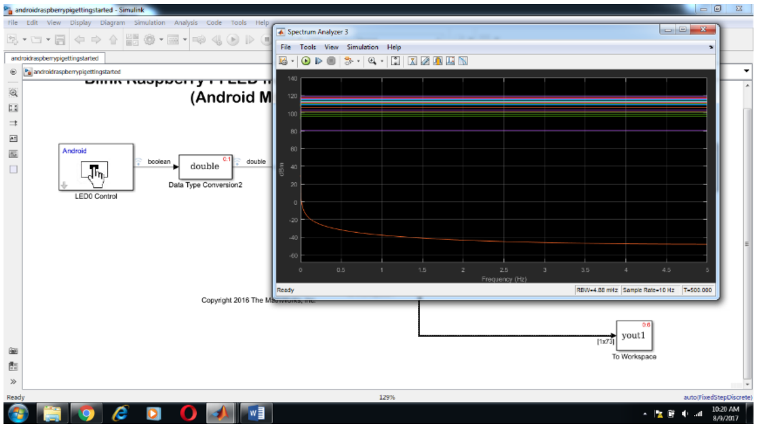
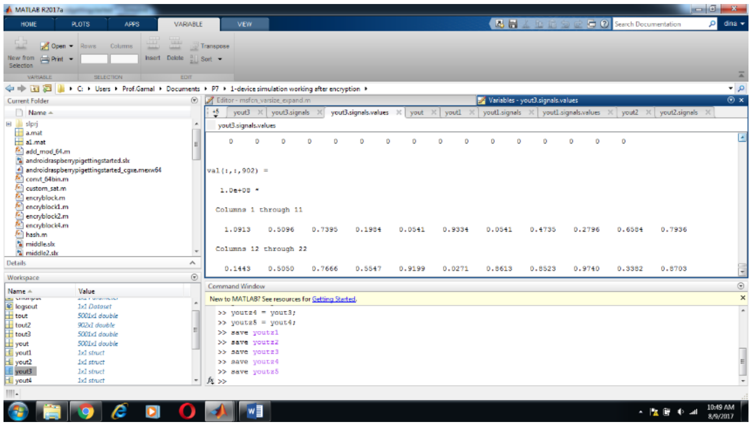


Fig.12. Spectrum anlyzer 3 showing encrypted signal receiption



```
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0


val(:,:,902) =

  1.0e+08 *

 Columns 1 through 11

   1.0913    0.5096    0.7395    0.1984    0.0541    0.9334    0.0541    0.4735    0.2796    0.6584    0.7936

 Columns 12 through 22

   0.1443    0.5050    0.7666    0.5547    0.9199    0.0271    0.8613    0.8523    0.9740    0.3382    0.8703
```

```
>> youtz4 = yout3;
>> youtz5 = yout4;
>> save youtz1
>> save youtz2
>> save youtz3
>> save youtz4
>> save youtz5
>>
```

Fig.13. Encrypted received signal in part 3 named yout3 in the MATLAB interface

7.2 Response time

To know simulation time from the beginning of part 1, when the user presses the button to change connected home devices passing through the controller device to the end of part 3 when the home device LED is turned ON in case the received data carrying user device's id and user's password, and controller device id are all correct, a stop block is added in part 3; which represents the connected enabled raspberry pi home device with a LED. The role of the stop block is to stop simulation when a signal carrying data greater than 0's is entered, it means when the encrypted verified signal arrives to turn ON (in this case) the home device's LED. There are two display blocks; one display to show the signal after transforming it to fit the LED size to 1's, the other display to show the final encrypted signal received in numerical representation; which is connected to the stop block. After stopping the simulation, the response time is given, with analysis on the time partitioning across different tasks; in other words, how each task takes time during simulation obtained from report analyzer tool inside MATLAB. A stop block is added in part 3 to stop simulation when the received encrypted data are verified and the LED is turned ON to measure the response time. The response time here from the beginning to the end of the simulation is 24.24 s, but it is organized across many tasks, it means that each task takes an amount of time to be executed during simulation. So that, the compile and link task takes approximately more than 80% of the total response time, but some other tasks; such as display.outputs.major task takes a very small percentage of the response time not increasing than 1% of it. Also, Fig.14 shows the response time; as explained in section 6, organized across different tasks as obtained from the report analyzer tool in MATLAB. Table 6 illlustrates the tasks composing the total simulation time, and the percentage of each task from simulation time.

Table 6. The percentage of each task from simulation time

Task	Time(%)
total simulation time from start to end	100
simulation Phase	77.3
compile and link Phase	10.2
initialization phase	8.6
simulation.outputs.major	7.9
simulation.setupruntimeresources	6.3
spectrumanalyzer.setupruntimeresources	6.1
M-S-function.outputs.major	4
termination phase	3.9
simulation.cleanupruntimeresources	3
M-S-function2.outputs.major	2.8
simulation.update	0.6
spectrumanalyzer.update	0.6
toworkspace.outputs.major	0.3
spectrumanalyzer.cleanupruntimeresources	0.2
M-S-function3.outputs.major	0.2
display.setupruntimeresources	0.1
display.cleanupruntimeresources	0.1
stop.outputs.major	0.1
S-function4.outputs.major	0.1
display.outputs.major	0.1
s-function5.outputs.major	0.1
toasyncqueueblock.setupruntimeresources	0.1

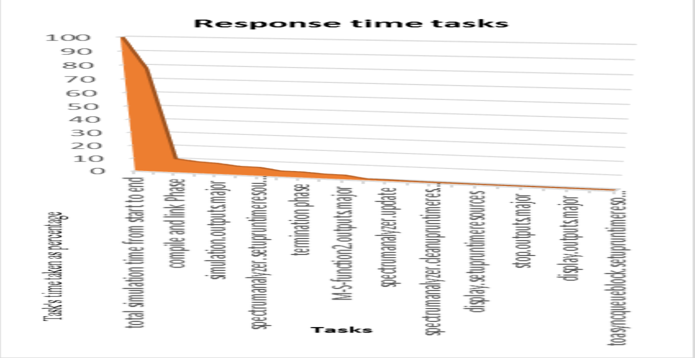


Fig.14 shows the response time organized across different tasks during simulation

7.3 Memory consumption

During the simulation, when the simulation time increases from 100 s to 1000 s, memory consumed increases gradually from below 5 MB to near 35 MB as shown in Fig.15. The experimental results proved that increasing the simulation time, increases the amount of memory consumed in Megabytes. Fig.15 shows memory consumption in Megabytes when the simulation time increases from 100, 200, 300, 400, 500, 600, 700, 800, 900 to 1000 seconds. But the amount of memory consumed expressed in Megabytes in general is good, and not very large. Table 7 shows memory consumption in (MB) versus simulation time.

Table 7. Memory consumption in (MB) versus simulation time

time	100	200	300	400	500	600	700	800	900	1000
Memory consumption during simulation (MB)	3.4438	6.2819	10.3102	13.7435	17.1767	20.7017	24.1501	27.5986	31.0471	34.4956

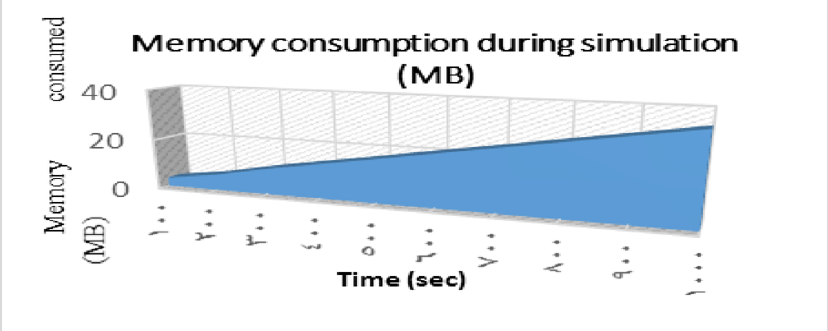


Fig.15 Memory consumption (in Megabytes) versus simulation time (in seconds)

7.4 Power of signals consumed

The experimental results showed that increasing the simulation time gradually from 100, 200, 300, 400, 500, 600, 700, 800, 900 to 1000 s, increases slightly by small portions the amount of power consumed in decibels of signals sent; either at the android device, the raspberry pi enabled controller device or the the raspberry pi enabled home device, and in some cases, the consumed power can decrease a little and increase again; for example at a simulation time of 600 seconds in the android device. In general, the power of signals consumed during simulation proved to be good and reasonable at the android device, the controller device and the home device, and ranges from 152 dB to 156 dB. Fig.16, Fig.17, Fig.18 and Fig.19 illustrates the power of signals consumed expressed in Megabytes in the android device, controller device, the home device and all the mentioned three graphs are added in the last graph respectively. Table 8 shows power consumed in (dB) versus simulation time.

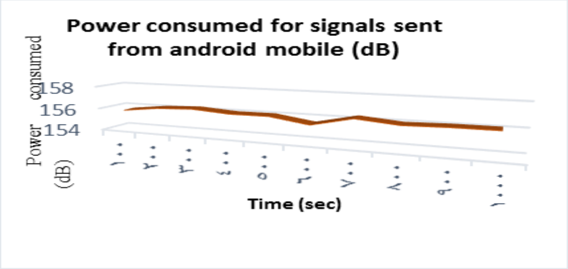


Fig.16 Power consumed (in Megabytes) versus the simulation time (in seconds) at the android device

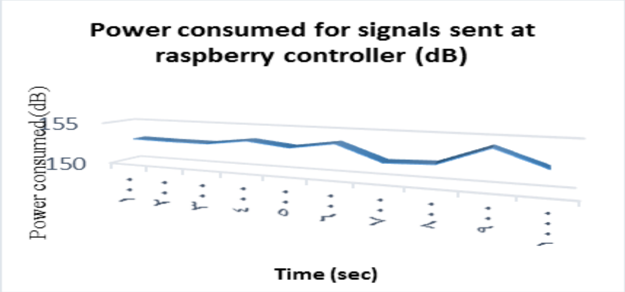


Fig.17 Power consumed (in Megabytes) versus the simulation time (in seconds) at the raspberry pi controller device

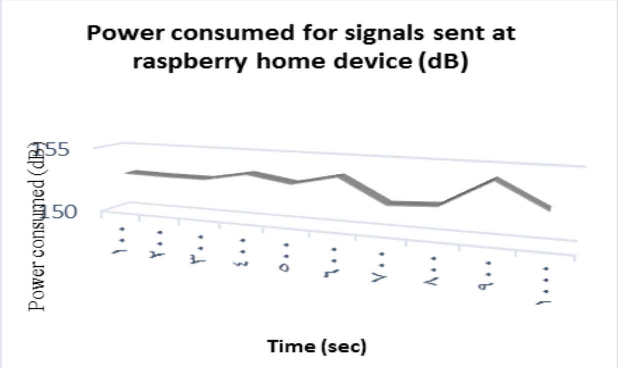


Fig.18 Power consumed (in Megabytes) versus the simulation time (in seconds) at raspberry pi home device

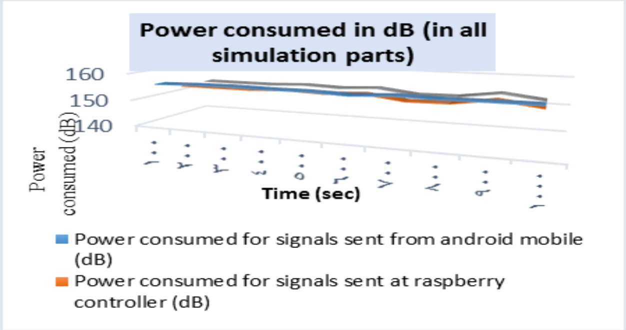


Fig.19 Power consumed (in Megabytes) versus the simulation time (in seconds) at the android device, the controller device and the home device

Table 8. Power consumed in (dB) versus simulation time

Time	100	200	300	400	500	600	700	800	900	1000
Power consumed for signals sent from android mobile (dB)	155.768	156.1378	156.2714	156.0421	156.0652	155.5702	156.3233	156.0253	156.1118	156.1221
Power consumed for signals sent at raspberry controller (dB)	152.9724	152.9094	152.9071	153.5063	153.0104	153.7477	152.0055	152.1923	154.1718	152.4356
Power consumed for signals sent at raspberry home device (dB)	152.9724	152.9094	152.9071	153.5063	153.0104	153.7477	152.0055	152.1923	154.1718	152.4356

7.5 Bit error rate during simulation

To measure bit error rate, it is required to make some adjustments on the three main parts of the proposed security technique blocks. First, a packet Output block is added at part 1 before the android UDP send block (from the android toolbox), the block has three outputs; number of ticks, data_ready and data_error; which represents the third output of the packet output block and is used to measure errors that occurred during sending data wirelessly from the android device to the raspberry pi enabled controller device. Also in part 1, a packet input block is added; which has four outputs, captured data at the controller device, the data_ready, the data_error and the number of ticks. Also, the data_error in packet input block is used to measure errors in data received wirelessly at the IP address of the controller device. The spectrum analyzer shows signal at the android device, and the scope shows signal data at the controller device. In part 2, a packet input block is added, with four outputs; which are captured data at the controller device, data_ready, data_error and number of ticks. The data error output of the packet input block represents errors in data sent wirelessly from the controller device. Also in part 2, a packet output block is added, which has three outputs, number of ticks, data_ready and data_error. The data error represents errors in data received wirelessly at the IP address of the home device. The error of data sent from the android device and errors of data received at the controller device are measured. The error of data sent from the controller and errors of data received at the home device are measured. Fig.20, Fig.21, Fig.22, Fig.23, Fig.24 illustrate the percentage of data errors that occurred versus a simulation time of 100, 200, 300, 400, 500, 600, 700, 800, 900 and 1000 s, at the android device (sender), the controller device (receiver), the controller device (sender), the home device (receiver), and all the preceding graphs grouped in one graph respectively. The experimental results showed that the errors that occurred during transmission wirelessly is in general good, since it ranges from 0.5% to 20%. Table 9 shows data error percentage versus simulation time.

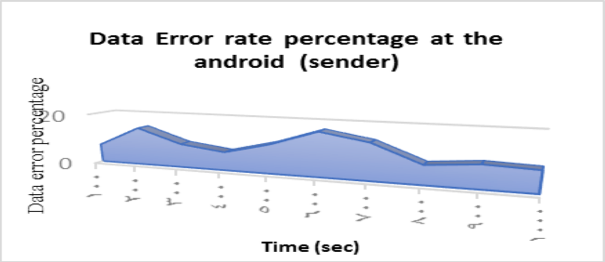


Fig.20 Percentage of data errors that occurred versus the simulation time at the android device as a sender

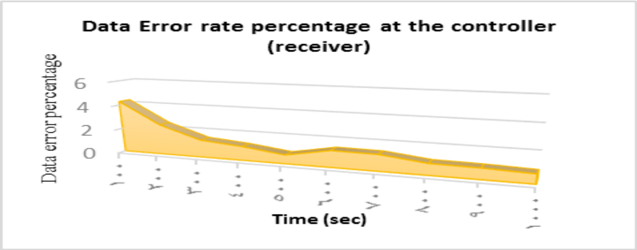


Fig.21 Percentage of data errors that occurred versus the simulation time at the controller device as a receiver

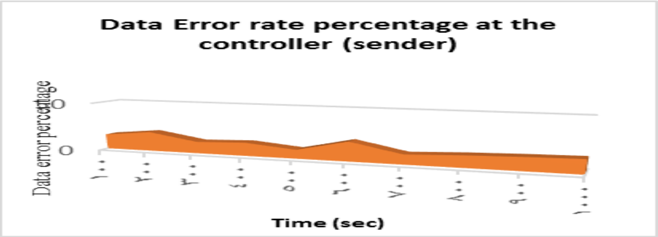


Fig.22 Percentage of data errors that occurred versus the simulation time at the controller device as a sender

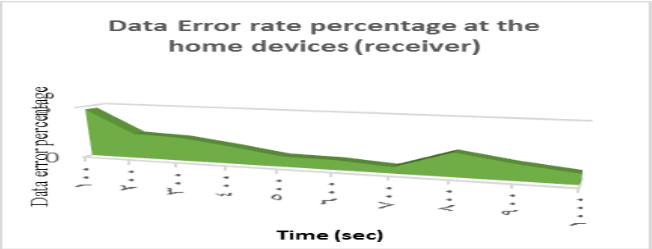


Fig.23 Percentage of data errors that occurred versus the simulation time at the home device as a receiver

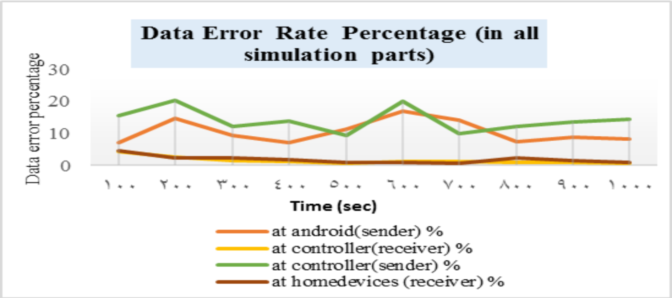


Fig.24 Percentage of data errors that occurred versus the simulation time at the android device (sender), the controller device (receiver), the controller device (sender), the home device (receiver)

Table 9. Data error percentage versus simulation time

data error percentage										
Time	**100**	**200**	**300**	**400**	**500**	**600**	**700**	**800**	**900**	**1000**
at android (sender) %	7.0929	14.7426	9.2636	7.1482	11.2777	16.9805	14.0409	7.2491	8.7657	8.0992
at controller (receiver) %	4.2957	2.4988	1.4329	1.1497	0.7199	1.3498	1.3141	0.9124	0.8666	0.7499
at controller (sender) %	15.3846	20.09	12.1293	13.7716	9.1782	20.0133	9.9843	12.0985	13.4207	14.2986
at home devices (receiver) %	4.6953	2.4488	2.2659	1.6746	0.9998	0.9332	0.5999	2.2122	1.4998	0.9799

7.6 Password strength checking

In the final parameter, it is required to test the strength of the device id for example, using mathematics once, and using the simulation inside the MATLAB R2017a. To find the possible combinations of finding a device id (for example). The basic formula used for finding a given combination is given by:

C(n,k) = n!/(k!(n–k)!)

Here, n is the total number of items and k is the number of members or items chosen from total number of given n. This can also be written as the binomial coefficient (n k) as below:

(n(n–1)(n–2)…(n–k+2)(n–k+1))(k(k–1)(k–2)…2.1)(n(n–1)(n–2)…(n–k+2)(n–k+1))(k(k–1)(k–2)…2.1)

So, in our case the total number of possible combinations is calculated as follows, we have 127 (n) characters at the computer keyboard, and the device id is composed of 10 (k) items (numbers and letters; capital or small), we have:

C(127,10) = 127!/10!(127-10)! =
127.126.125.124.123.122.121.120.119.118.117!/1.2.3.4.5.6.7.8.9.10.117!
= 7.588684395810302e+20/3628800 = 209123798385425 possible combinations.

After 209 trillion trials 209123798385425/5100290 = 20501167.42238432 hours of trials/24= 854215.3092660131 days of trials/365 = 2340.315915797296 years of trials. On an ordinary computer it means it is very difficult to regenerate the device_id in an ordinary computer using trials due to the huge number of possible combinations. Generating a code inside MATLAB R2017a to try to find the device id, and giving the number of trials or iterations during the simulation, gives us the below figures. Fig.25 shows the total number of trials to find the device id (in this case) versus the total number of hours of simulation. After all these trials, the device id was not found. Table 10 shows number of trials versus number of simulation hours.

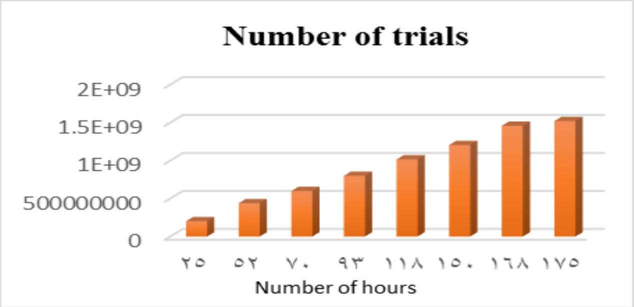


Fig.25 Number of trials to find the device id versus the number of hours of simulation taken

Table 10. Number of trials versus number of simulation hours

Number of hours	25	52	70	93	118	150	168	175
Number of trials	2.03E+08	4.39E+08	5.99E+08	7.99E+08	1.02E+09	1.20E+09	1.46E+09	1.52E+09

8 Conclusion and Future Work

From the results obtained from experimental simulations, it is concluded that the proposed security technique (SMI2C) provides good response time, reasonable amount of memory consumed, and power consumed during simulation, good bit error rate and strong technique for protecting passwords without any additional overheads on the proposed system. So, the proposed technique is useful in encryption as it protects user data during transmission between different devices and has many benefits.

In the Future work, a secure technique for internet could be developed taking into consideration reducing energy consumed during transmission; also, reducing memory consumed during signals transmissions could be studied. There are a lot of other parameters that can be considered as areas of research while designing a secure technique for online data transmissions in the future; such as speed of transmission, bit error rate and so on.

REFERENCES

[1] Gartner's hype cycle special report for 2015, Gartner Inc., 2015. [Online]. Available: http://www.gartner.com/technology/research/hype-cycles/.

[2] Gubbi J., et al., *Internet of Things (IoT): A vision, architectural elements, and future directions*, Future Gener. Comput. Syst., 2013. 29 (7): p. 1645–1660.

[3] Miorandi D., et al., *Internet of things: Vision, applications and research challenges*, Ad Hoc Network, 2012. 10 (7):p. 1497–1516.

[4] Yasumoto K., Yamaguchi H., and Shigeno H., *Survey of real-time processing technologies of IoT data streams*, J. Inf. Process, 2016. 24 (2):p. 195–202.

[5] Husain S., et al., *Recent trends in standards related to the internet of things and machine-to-machine commun*., 2014, 4 (6).

[6] Djenouri D., Khelladi L., and Badache N., *A Survey of Security Issues in Mobile Ad-hoc Networks and Sensor Networks*, IEEE Communications Surveys, 2005. 7 (4):p. 2-28.

[7] Cho J.-H., Swami A., and Chen R., *A Survey on Trust Management for Mobile Ad-hoc Networks*, IEEE Communications Surveys & Tutorials, 2011. 13 (4):p. 562-583.

[8] Wang Y., Attebury G., and Ramamurthy B., *A Survey of Security Issues in Wire-less Sensor Networks*, IEEE Communications Surveys Tutorials, 2006. 8 (2):p: 2-23.

[9] Cha I., et al., *Trust in M2M Communication*, IEEE Vehicular Technology Magazine, 2009. 4 (3): p. 69-75.

[10] Mell P. and Grance T., *The nist definition of cloud computing*, National Institute of Standards and Technology, 2009. 53 (6) article 50.

[11] Zhang, Q., Cheng, L., and Boutaba, R., *Cloud computing: state-of-the-art and research challenges*. Journal of internet services and applications, 2010. 1 (1):p. 7-18.

[12] Zhou J., et al., *Cloud Architecture for Dynamic Service Composition*, International Journal of Grid and High Performance Computing, 2012. 4 (2):p. 17-31.

[13] Christophe, B., et al., *The web of things vision: Things as a service and interaction patterns*. Bell Labs Technical Journal, 2011. 16 (1):p. 55-61.

[14] Subashini, S., and Kavitha, V., *A survey on security issues in service delivery models of cloud computing*. Journal of Network and Computer Applications, 2011. 34 (1):p. 1-11.

[15] Botta A., et al., *Integration of Cloud Computing and Internet of Things: a Survey*, Journal of Future Generation Computer Systems, September 18, 2015.

[16] Gomes, M. M., Righi, R. d. R., and da Costa, C. A., *Future directions for providing better iot infrastructure*. In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication (UbiComp '14 Adjunct.)*, 2014, p. 51-54.

[17] Alhakbani, N., ei al., *A framework of adaptive interaction support in cloud-based internet of things (iot) environment*. In: Internet and Distributed Computing Systems. Springer, 2014, p. 136-146.

[18] Fox, G. C., Kamburugamuve, S., and Hartman, R. D., *Architecture and measured characteristics of a cloud based internet of things*. In: Collaboration Technologies and Systems (CTS), 2012 International Conference on. IEEE, 2012, p. 6-12.

[19] Dash, S. K., Mohapatra, S., and Pattnaik, P. K*., A Survey on Application of Wireless Sensor Network Using Cloud Computing*. International Journal of Computer science & Engineering Technologies, 2010. 1 (4):p. 50-55.

[20] Atzoria L. and Giacomo Morabito A.I., *The Internet of Things: A Survey*, Computer Networks, 2010. 54 (15):p. 2787-2805.

[21] Gantz J., *The Embedded Internet: Methodology and Findings*, 2009. [Online]. Available: https://www.bryankorourke.com/blog/2010/3/11/the-embedded-internet-15-billion-devices-by-2015.html

[22] Evans D., *The Internet of Things: How the Next Evolution of the Internet Is Changing Everything*, 2011. [Online]. Available: http://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

[23] Hatton M., *The Global M2M Market in 2013*, Machina research whitepaper, 2013.

[24] Emmerson B., *M2M: The Internet of 50 Billion Devices*, Win-Win, 2010, pp. 19-22.

[25] M2M. [Online]. Available: SingTel M2M, http://info.singtel.com/large-enterprise/about-m2m.

[26] Watson D.S., et al., *Machine-to-Machine (M2M) Technology in Demand Responsive Commercial Buildings*, in Proceedings of the ACEEE Summer Study on Energy Efficiency in Buildings, 2004, pp.1-14.

[27] ETSI, TS 102 690 M2M Functional Architecture, 2011.

[28] RSA. [Online]. Available: https://en.wikipedia.org/wiki/RSA

[29] Digital signature algorithm. [Online]. Available: https://en.wikipedia.org/wiki/Digital_Signature_Algorithm

[30] Elliptic curve cryptography. [Online]. Available: https://en.wikipedia.org/wiki/Elliptic-curve_cryptography

[31] SHA-2. [Online]. Available: https://en.wikipedia.org/wiki/SHA-2

[32] MATLAB Simulink. [Online]. Available: https://www.mathworks.com/products/simulink

[33] Raspberry Pi. [Online]. Available: https://en.wikipedia.org/wiki/Raspberry_Pi

[34] Choil K.-H., et al., *Method of Calculating the Server Capacity for Cloud Computing for SaaS*, International Journal of Software Engineering and Its Applications, 2015. 9 (11):p. 117-126 .

[35] Signal strength. [Online]. Available:

https://en.wikipedia.org/wiki/Signal_strength_in_telecommunications

[36] Bit error rate. [Online]. Available: https://en.wikipedia.org/wiki/Bit_error_rate

[37] Password cracking. [Online]. Available: https://en.wikipedia.org/wiki/Password_cracking

[38] Password cracking of an application. [Online]. Available:]https://www.guru99.com/how-to-crack-password-of-an-application.html

[39] Collision attack. [Online]. Available: https://en.wikipedia.org/wiki/Collision_attack

[40] Preimage attack. [Online]. Available:

[41] Collision resistance. [Online]. Available: https://en.wikipedia.org/wiki/Collision_resistance

[42] Preimage resistance and collision resistance. [Online].Available: https://crypto.stackexchange.com/questions/1173/what-are-preimage-resistance-and-collision-resistance-and-how-can-the-lack-ther

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TNC TRANSACTIONS ON NETWORKS AND COMMUNICATIONS

VOLUME 7, NO. 1
ISSN: 2054 -7420

Deployment of an Energy Efficient Routing Protocol for Wireless Sensor Networks Operating in a Resource Constrained Environment

Agbotiname Lucky Imoize[1,2], Taiwo Oyedare[1], Chibuike Gerald Ezekafor[2], and Sachin Shetty[3]
[1]Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, USA
[2]Department of Electrical and Electronics Engineering, University of Lagos, Akoka Lagos, Nigeria
[3]Virginia, Modeling, Analysis and Simulation Center, Old Dominion University, Norfolk, VA, USA
aimoize@vt.edu, toyedare@vt.edu, jralde@yahoo.com, sshetty@odu.edu

ABSTRACT

Wireless Sensor Networks often exist in a resource constrained environment and the most critical limitation is energy or battery life. In most cases, the batteries cannot be recharged during operation and cannot be replaced during transmission without interfering with the quality of service. An efficient routing protocol that can prolong the life span of the energy source for mobile wireless sensor network (WSN) is proposed in this paper. Hierarchical routing protocol improved the life span by more than six times compared to direct techniques. Sets of varied parameters were used to obtain simulation results. A sample typical of a hospital layout was used in this paper. Wards were assigned varied sensor range which depicts number of patients that fall into these banded regions of clusters. Service centers were created to ensure that random mobilization of sensors or patient per round is achieved for one of the simulation option. Simulation rounds were carried out for non-hierarchical (direct) and low energy adaptive clustering hierarchy (LEACH) based routing protocols. Model validation was done by setting up a real life test using Wi-Fi sensor node module ESP8266 properly distributed, and this was done without mobility. Effect of base station location was also investigated and results tabulated. With improved battery and network life span from both simulation and live validation for hierarchical routing, quality of service is greatly improved in health-care deliveries; cost of replacing energy sources is reduced. Results were also compared with other similar works to support our conclusion. Though results were not exact when compared, parameters used could cause tolerable difference. Furthermore, same result could not be achieved for different locations, varied number of wards, varied number of sensor nodes per wards as well as variations in other modeling parameters.

Keywords: Energy efficient routing protocol, Wireless sensor network, Adaptive clustering, Healthcare, Base station.

1 Introduction

The revolution caused by the emergence of wireless sensor technology (WST) cannot be over emphasized. The trend is quite significant and the application is almost limitless. Researchers have consistently explored this field of great interest and new grounds and methodologies have been identified [1]-[2]-[3]-[4] thereby making the challenges of today success stories of tomorrow. Wireless sensor network

DOI: 10.14738/tnc.71.5975
Publication Date: 02nd February 2019
URL: http://dx.doi.org/10.14738/tnc.71.5975

application used to be a stationary based technology. With the demands for real time control and monitoring in various areas of interests, mobile wireless sensor network became a necessity. Our world today is enjoying the great dividends of wireless sensor networks, and more useful applications of WSNs are envisaged in the near future.

Telemedicine, military surveillance, industrial and environmental monitoring and other areas of application have become popular today. The value added to both life expectancy and other nature's course is quite notable. Wireless body sensor network mostly applied for medical information monitoring and control process is designed as a wearable system or implanted into the human body parts. The fact that sensor nodes need longer battery life to sustain the wireless sensor activities and the maintenance routine required when these sensors approach the end of their lifetime is of great concern. It is also more challenging to manage this in a mobile wireless sensor network [5]. Cost is always a key factor in approaches and models, hence, it is very imperative to explore this area and come up with a model that attempts to proffer an acceptable solution on how energy source can be optimized to extend wireless sensor network (WSN) lifetime. In this medical application, the role of sensors is to obtain vital signs and intelligently relay them to designated receiver or receiving node. In essence, a standard sensor node for this function consists of four major sections; the sensing unit, communication unit, processing section and the power unit. All these sections work seamlessly to achieve a quality throughput.

However, there are lots of limitations and performance/system integrity threat currently affecting the deployment and maximal usage of WSN. These WSNs have stern resource constraints and energy conservation cannot be overemphasized [6]. Since all these building blocks/units of the sensor node rely greatly on power to function, it implies that constant energy availability and sustainability is a major threat to wireless body sensor node network [7]. There has been various works on improving on the energy efficiency of each sensor node. All these are aimed at either reducing the power consumption of the network or increasing the battery life span. Local processing by nodes has also been considered though this implies complex circuitry and more power consumption. Adding mobility to the WSN system increases the complexity as it becomes more challenging to achieve maximum energy conservation or power efficiency. Amongst many other energy optimization methods, Hierarchical routing model has proven to present a better result. Hierarchical routing technique involves creating clusters for sensor nodes, assigning sensor nodes to cluster zone depending on the parameter used, and then selecting a cluster head which helps to obtain data from members of the same cluster for relay to other cluster heads or base station (BS). In this scenario, energy depletion on the network is reduced as well as service cost.

This study focuses on this technology and the methodology applied in this model. It also highlights few other challenges peculiar to mobile sensor nodes. Furthermore, this study is useful in deciding the best cluster formation for an health care environment before the deployment of mobile sensor nodes technology in such an environment. One should note however, that the best cluster formation for one environment may not necessarily be the best in another environment. This study therefore works with a typical predetermined environmental layout. In practice however, the layout parameters within this study will have to be re-defined for the environment under study.

The organization of this paper is as follows. Section II presents the literature survey of related works on the challenges with WSN and applications, routing protocols based on flat, location and hierarchy model and general challenges. Section III provides a theoretical background for routing protocol challenges as well as common classification of routing protocols for WSN. Section IV covers proposed hierarchical routing technique via a detailed description of methodology of the model, parameters used and their descriptions, assumptions, mathematical expression of model and employed algorithm. Focus was more on the Hierarchical-based routing technique using the LEACH approach for healthcare application. Section V presents the simulations, graphical and tabulated results and indications. Section VI concludes the paper, states useful contributions and recommendations for future studies.

2 Related work

Wireless sensor networks have various applications and characteristics. However, the main tasks of the nodes that make up these networks involves sensing and collecting, processing and transmitting data to a site where it is required for either tracking or monitoring functions [7].

Effective power management is a major focus while designing protocols for WSN. There have also been efforts in the recent technology to reduce power consumption of sensor node by local processing [8], which means that sensor nodes now have the capacity to process information more faster before transmission. This has rather increased the circuitry complexity and does not relatively reduce power consumption. Though the average power consumption of wireless sensor devices has exhibited a downward trend over the decades, battery power innovation have not followed this trend. Hence, the smart techniques to consume and optimize power in wireless sensor network is crucial.

Gupta et al. [9] provided the H-LEACH, a typical hybrid system that uses fixed clusters and dynamic cluster head (CH). In their simulation, the authors did not consider key aspects such as energy consumption by the cluster head based on the fact that they were dynamically chosen and would account for a negative impact on the information sent to the base station.

Heinzelman et al. [10], reported a proposal based on the asumption that the entire sensor nodes are potentially taken as base stations. Hence, all nodes disseminate information to other nodes across the network. Rumor routing as established by Braginsky et al. [12] is suited for applications where geographical routing can be very difficult. Zabin et al. [13] reported a proposal on energy aware routing and the reliable, energy efficient protocol (REEP). The later is a new energy awareness routing. Generally, REEP enables sensor nodes to quickly establish more reliable and efficient paths for data transmission and this increases network lifetime.

Heienzelman et al. [11] noted that Low Energy Adaptive Clustering Hierarchy (LEACH) protocol is one of the popular algorithms for WSN. Here, clusters of sensor nodes formed are focused on the received signal quality and the use of a local CH as a relay or backbone to the base station was presented. Rather than every individual sensor node sending packet to the BS, only a designated CH node will perform this action at any given round and this greatly reduces energy consumption in a way similar to [14].

Stationary and mobile wireless sensor applications both have diverse challenges and concerns but they share these concerns in common; transmission power, data storage and aggregation capacity, computational capacities and energy consumption [8]. Effective Power management is a major focus while designing protocols for WSN. In this study, the proposed Hierarchical (LEACH) protocol is employed to optimize the energy efficiency of a WSN in a healthcare environment.

3 Theoretical background

A. Routing Protocol Challenges and Design Issues

Though the emergence of WSN has transformed various aspect of today's technology, there are various challenges and limitations [15] which are consistently reassessed and investigated with the hope of improving functionality and quality of service. Energy supply bandwidth limitation for the links and computational storage capacity will always be areas of research. Improving longevity and lifetime of a WSN network is a priority. To design routing techniques that will cater for these limitations, we need to put to bare, these few challenging factors discussed as follows:

1) ***Network Dynamics***: In earlier applications of sensor nodes, it was always considered that nodes are stationary. For this architectural condition, challenge is minimal when compared with more recent applications where mobility is as important as sensing, for example, health care and wildlife monitoring. It is more challenging to route and consume less energy in mobile WSN. Whether we are talking about tracking or monitoring, the network is dynamic and such requires constant updates and message alerts to keep to the node and environmental requirements. This is a task that requires smart routing protocol.

2) ***Transmission Media***: In mobile sensor network, Multi-hopping is expected. Multi-hopping is done via a wireless link. There is a need to address the issues inherent in wireless channels that could negatively impair performance of the sensor node. In the design of such networks, there is a need to consider the use of Time Division Multiple Access (TDMA) for the Media access protocol. This helps greatly in conserving the energy of the network when compared with the Code Division Multiple Access (CDMA).

3) ***Node Deployment***: Deployment of sensor nodes can be a deterministic or randomized process and this as well affect the performance of protocols of choice. In either case, sensors are manually positioned and the transmission route is predetermined or positions are randomized making it possible to determine routes. The experience creates an ad-hoc infrastructure, which greatly poses challenges in coverage and optimization of clustering.

4) ***Fault Tolerance***: When a sensor node fails, the entire activities of the sensor network, especially when such failure is caused by power depletion, physical damage or environmental constraints, should not be affected. In scenarios when this failure occurs, there must be a MAC and routing protocols that should cater for raising new links and connections as an alternative to keep the network activities alive especially collating data to the base stations. What this means is that multiple redundancy of paths and task channel should always be considered to improve on fault handling capacity of a mobile sensor network.

5) ***Quality of Service***: In most mobile WSN applications like healthcare, there is time limit for data aggregation and transmission to the base station. In this scenario, the moment sensed data is not relayed to the base station for onward processing; data becomes invalid after a particular amount of time. In some applications, the network lifetime and power conservation is of much importance than the integrity of data. This implies that we need to consider the protocol that will suit each kind of application.

6) ***Coverage***: Area coverage of the sensor nodes is of great importance in WSN design. Coverage in mobile wireless sensor network can either mean the communication coverage or the sensing coverage. For which ever context we consider, the coverage affects the accuracy and range of sensor node performance. This has to be considered as a concern when designing a WSN.

7) ***Energy Consumption:*** Since sensor nodes exhaust their energy supply during transmission and computation in WSN setting, it is always important to apply energy conservation techniques and model in network design. Life span of a sensor node is dependent on the battery life as the source of energy [11]. Once depleted and there are no alternative route provision via redundancy, WSN is affected. There is a need to keep an eye on this challenge in design.

B. Classifications of Routing Protocols for WSNs

Various routing protocols have evolved basically to avert the challenges and poor services experienced in this resource constrained wireless sensor network architecture. Generally, routing protocols can be in flat-based routing [16], Location based routing or Hierarchical based routing [17]. The routing model decided in each design is dependent also on; the type of network structure, the initiator of communication, network operation carried out using this protocol and the type of communication routes processed from source to sink. Routing protocol is flat-based if all sensor nodes within the network are assigned the same role and function. In location based protocols [18], sensor node functionality ties greatly to the position of these nodes in the network, while in hierarchical based routing [18], nodes are seen to carry out various roles in the network based on assigned role and hierarchy.

A routing protocol can be considered to be adaptive if the nodes have the ability to adjust intelligently to existing parameters for proper network and node functioning. In this form of protocol, it can be multi-path, query or negotiation based. It can also be QoS based or dependent on the protocol operations. Routing protocols also can be reactive, proactive or hybrid depending on how data routing is established from source to destination. We also have cooperative routing where data aggregates to a central point for further processing before sending to a base station, this is to ensure cost is minimized [19].

4 Proposed Hierarchical Routing Technique

There are various methods and techniques used in hierarchical routing. However, the basic concept is that sensor nodes are clustered, and within the formed cluster, a node is picked as the cluster head depending on the amount of energy left within that node. All other nodes then forward their data directly to the selected head. Note however that the cluster head will have its energy depleted faster than any other node in the cluster as it has to send more data to the next cluster or the base station than the other sensor nodes. The position of the cluster head is therefore rotated on every iteration of the simulation loop to keep the residual energy within the sensor nodes balanced.

In a typical health care unit, there will be movement of the patients from one location to the other. Since sensors are attached to a patient, his/her movement causes the position of a sensor node to change relative to its cluster head. If a sensor node is moved out of proximity from its cluster head, it may have to join a new cluster to reduce the energy consumed in sending its packet. If and when the sensor node comes back to its previous location, it rejoins its original cluster. Modeling the change in location of a sensor node however is not a straight forward model as this is statistical in nature. A patient may or may not decide to move from his/her location. This behavior is modeled by creating some locations on the model called "service points" where a patient may need to go every once in a while. The service points are deployed as well as the sensors at the start of the simulation. Randomly, a sensor node is selected and mobilized to a service point, kept in that region for some simulation rounds and then moved back to its original location.

A. Methodology

The method involved in modeling the proposed hierarchical routing technique in MATLAB used the object oriented programming (OOP) approach. In this approach, each entity in the model is programmatically abstracted or represented. In this model, the following entities are identified: the BS, the WSN, the cluster, servicing point and the ward.

1) ***The Base Station (BS)***: Every data acquired by the WSN is assumed to be destined for a data center meant to aggregate such data, process them and could be stored in memory for further re-transmission. The data center is described in the model as the base station. However, note that the base station is modeled differently from the wireless sensor node even though they share some similarities such as the ability to receive data from a neighboring WSN. The different model used is to account for the fact that the base station energy cannot be depleted given that the BS is grid powered. In the model, the base station has these properties; Location and packet received. Location considers the X and Y coordinate positions for the BS while the packet received depicts the set of data that has been relayed or logged at the base station.

2) ***Wireless Sensor Node (WSN):*** A sensor unit consists of the sensor and the analogue to digital converter circuitry. Generally, sensors collect physical data and convert to an electrical signal. This signal is in a format that can be interpreted by other processing units.

Here, it should not be assumed that the location is fixed during the cause of this simulation; this means a patient may relocate to a new vector coordinate on the simulation space. The sensor node model has the following properties; energy, location, whether dead or not, and packet to transmit.

During simulation, the residual energy of a sensor node is always re-estimated every time a sensor receives or sends a data packet (DP). If residual energy (RE) within such sensor is not enough to receive the incoming packet or send outgoing packet, the sensor is marked as dead and will no longer participate in the network. The incoming or outgoing packet is therefore lost. Wireless Sensor Nodes are also mobile in this case. This has an implication on the geographical location of the WSN during this simulation. When a sensor moves, chances are that it will no longer be in proximity to the current cluster to which it belongs. The sensor will therefore be detached from its current cluster and attached to another. In reality, the selection of which sensor to move and the location of a sensor node when it moves is statistical in nature as there is no way to programmatically define which sensor moves and the new location of this sensor node when it moves. To circumvent this problem, sensors are selected randomly using a random number generator. The location of the Service Center (SC) to mobilize the sensor node to is also randomized. The selected sensor is therefore relocated to the SC, kept for a number of simulation rounds and finally moved back to its original location.

3) ***The Cluster***: The intended optimization to be performed in this study is meant to be accomplished using cluster formation. A cluster is simply a group of wireless sensor node in proximity to each other. Clusters are formed geographically by dividing the simulation area into some banding rectangles. Sensor nodes falling within the same banding rectangle are assigned to a common cluster. After cluster formation, a cluster head (CH) election is performed using a cost function as described in later part of this section. The sensor node with the minimum cost is delegated as CH. Due to the dynamic position of sensor nodes, CH election is repeated on every simulation round as the CH itself may have migrated away from the cluster

leaving the cluster without a head. Therefore, the election is performed over and over again to establish or ascertain that there will be a head within the cluster.

4) ***The Service Center (SC)***: This makes provision for the mobility of the sensor nodes. Service centers are strategically positioned at the beginning of the simulation and they define certain locations that a patient with a WSN attached to may have to migrate to once in a while. The number of patients that are mobilized to service center is a pre-determined value in this paper.

5) ***The Ward***: Ward as used here is a hospital location where patients are cared for specially. This is always made for patients on admission. Patients with various levels of attention are allocated to various wards. It can also be based on gender, age or treatment type. In this paper, our wards are modeled to have patients with wearable or implanted wireless sensor nodes which are grouped based on zones or clusters.

4.1 System Energy Model of a Sensor Node

As stated in [20], the energy suitable for a sensor node to transmit its packets to a base station is a function of the number of data in the packet, the distance separating the sensor node from the destination of the packet is as shown in figure 1

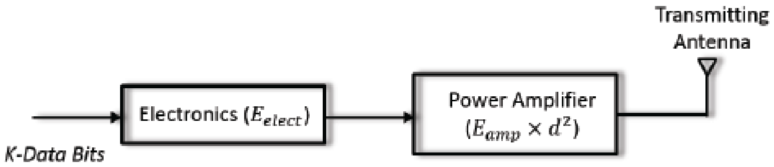

Transmitting Antenna
Electronics (E_{elect})
Power Amplifier ($E_{amp} \times d^2$)
K-Data Bits

Figure. 1. Energy model diagram for a sensor node (sending)

where *k* stands for the number of data bits to be transmitted from a sensor node.

E_{elect} is energy required by the electronics to send one data bit.

E_{amp} is energy required by the power amplifier to send one data bit through a distance of 1*m*.

d represents the separation distance between the transmitter and the receiver.

The total energy required to send the k-data bits is mathematically expressed in equation 1

$$E_{tx} = (E_{elect} \times k) + (E_{amp} \times d^2 \times k) \quad (1)$$

While receiving a packet by a cluster head, an amount of energy is expended in receiving the packet. This is denoted as in figure 2. This is mathematically expressed as shown in (2)

$$\mathrm{E_{rx}} = (\mathrm{E_{elect}} \times \mathrm{k}) \quad (2)$$

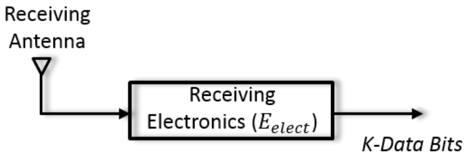


Fig. 2: Energy model diagram for a sensor node (receiving)

4.2 The Proposed Routing Protocol Algorithm

The proposed routing algorithm focuses on extending the life-cycle of the network. First, sensor nodes are geographically formed into clusters depending on the proximity of the nodes to each other. This is done by breaking down the work area in to a specific number of rectangles. Sensor nodes falling into the same rectangle are grouped together to form a cluster, and the cluster head is then chosen depending on the amount of energy left within the sensor node and the distance between the sensor node and the base station. This is to limit the possibility of the data being sent from a sensor node traveling back and forth. The selection of the cluster head therefore is a function of two variables, henceforth denoted as cost. The cost function is expressed in equation 3.

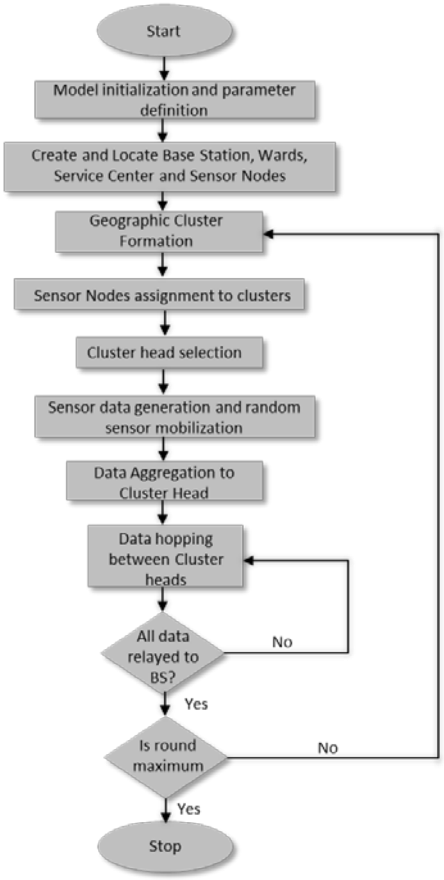

Start
Model initialization and parameter definition
Create and Locate Base Station, Wards, Service Center and Sensor Nodes
Geographic Cluster Formation
Sensor Nodes assignment to clusters
Cluster head selection
Sensor data generation and random sensor mobilization
Data Aggregation to Cluster Head
Data hopping between Cluster heads
All data relayed to BS?
No
Yes
Is round maximum
No
Yes
Stop

Fig. 3: A typical algorithm for the Hierarchical based routing model

$$E_{cost} = (E_i - E_s) \times W_s + d_{bs} \times W_b \tag{3}$$

where E_{cost} is the cost function

E_i represents initial energy assigned to all sensor nodes

E_s represents energy residue within a sensor node

W_s represents a weight factor for the energy residue

W_b represents a weight factor for the energy consumed d_{bs} represents separation distance between a sensor and the BS.

The cost is then computed for all the sensor nodes within a cluster and the one with the minimal cost is chosen to be the cluster head. When the head has been selected, all sensor nodes aggregate their data to the head.

For the data to reach its destination at the BS, a multihop approach was used where the data get routed to where the data is routed through the closest cluster head until it arrives at the base station.

While routing the data however to the next cluster head, a predictive function should be established as there may be a number of clusters that the data can be channeled through. The forward prediction establishes the amount of power that will be consumed if the data is channeled through a particular branch in the cluster tree. Starting from the root of the tree, a walk is successively made from the root to the base station through different paths and the energy consumed is recorded. At the end of the walk, the path that consumes the minimum energy is taken and the data is forwarded to the next cluster head in that path.

The algorithm followed by the simulation is summarized as; ward positioning, service center location, sensor node deployment, cluster formation, cluster head selection, data aggregation and data multi-hopping.

1) ***Ward Positioning***: The positioning of the ward used in this simulation is towards the sides of the simulation area/landmark chosen.

2) ***Service Center Location***: A patient may need to migrate from his/her ward for one reason or the other and this is based on the assumption that the sensor node is attached to the body of the patient. This migration causes the location of the sensor node to change. The sensor detaches itself from a cluster and joins another one. When the patient resume back to his ward, the sensor node rejoin its initial cluster.

3) ***Sensor Node Deployment***: At the start of the simulation, sensors are created and positioned on the simulation field. Each sensor node is also assigned an equal amount of energy. The location of the sensor is done in a manner that reflects a real world scenario where patients are expected to be assigned into a ward. Sensor nodes are deployed on a per-ward-basis with a particular amount of sensor nodes in a ward.

4) ***Cluster Formation***: Sensor nodes are always grouped in to clusters to conserve the energy consumed by the network. The cluster formation algorithm is derived on the basis of the geographical location of a sensor node. Due to the mobility of the patient however, cluster grouping is repeated on every simulation cycle as a sensor may have moved away from its last known location. To find an optimum point for the network, the clusters' geographical area is varied between simulations. Increasing this area causes more sensor nodes to fall within a particular area and vice versa.

5) ***Cluster Head Selection***: As stated earlier, the cluster head is elected based on a cost function. The sensor node with a minimum cost gains the position of a cluster head.

6) ***Data Aggregation***: At the onset of every simulation round, a specific amount of data packet is assigned for each sensor node to transmit. After cluster head election, the entire sensor nodes within the cluster send their packets down to the cluster head. At this stage, the cluster head does not send any data, but rather merges the incoming packet with its own data packet.

7) ***Multi-Hopping***: After the aggregation stage, every cluster head has to send its packet and its children packet to the next cluster head nearest to the base station. The packets then hop from one cluster head to the next until the base station is reached. The cluster head to hop data to is also selected based on a tree walk from the cluster head to the base station via many routes. The route with the minimum power requirement is taken. Figure 3 shows the algorithm for the model.

5 Results and Discussion

In the previous section, an improved LEACH protocol for mobile wireless sensor node was proposed. Here we provide a simulation of our improved protocol and estimate the performance of the protocol under varying simulation conditions. There are varieties of tools that could have been used for this modeling, but the environment used for the simulation is MATLAB. This program was chosen as the preferred language because of its availability and ease of use.

A. Simulation Setup and Scenarios

In this simulation, a health care environment layout was created with a dimension of 400m by 400m. Within this boundary, the base station is positioned at the center with a vector coordinate of 200, 200. Followed by this is positioning the service centers, wards and finally sensor nodes within the wards. Table 1 shows the simulation parameters used in this study.

Simulation was performed by varying the cluster grouping of the network. Clusters were formed geographically by dividing the working space vertically and horizontally into different sizes starting from 1 to 15. For example, when forming clusters with two groups per length and two groups per breadth, the environment looks like figure 4.

The description of the symbols in figure 4 is explained briefly as follows:

i. Cluster group formations are marked with Purple
ii. Base Station marked in Cyan Dots and Circles
iii. Service centers marked with Cyan Lines
iv. Wards marked with Blue Lines
v. Sensor Nodes marked with Red Dots
vi. Cluster Heads marked with Red Circular Dots

Figure 4 indicates the geographic formation of the cluster. Sensor Nodes falling within the same cluster box are automatically placed in the same cluster. Note however that even if sensors fall within the same ward, and are divided into different cluster by a cluster line, they will be placed in different clusters as depicted in figure 5.

Table I: List of Defined Parameters and Values Used in Simulation

Parameter	**Value**
Landmark Length	400m
Landmark Breadth	400m
Number of Base Station	1
Base Station Location	[200, 200]
Number of Service centers	2
Service Center Locations[X, Y, L, B]	[[252005050][3752005050]]
Number of Wards	16
Wards Location[X, Y, L, B]	[25 350 48 48] [75 350 48 48] [125 350 48 48] [175 350 48 48] [225 350 48 48] [275 350 48 48] [325 350 48 48] [375 350 48 48] [25 50 48 48] [75 50 48 48] [125 50 48 48] [175 50 48 48] [225 50 48 48] [275 50 48 48] [325 50 48 48] [375 50 48 48]
Initial Sensor Energy	10J
Transmitter Electronics Energy Per bit Etx	5nJ
Transmitter Electronics Energy Per bit Erx	5nJ
Transmitter Power Amplifier Per bit	5nJ
Maximum Simulation Rounds	1500
Sensors per Ward	5
Maximum Sensors on Transit	5
Maximum Sensor Speed	1m/s
Sensor Data packet size	2000 bits
Cluster Formation Range	1 : 15

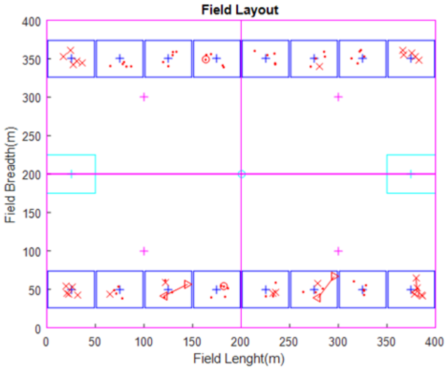


Fig. 4: A Sample 2 × 2 cluster field layout for defined 400m ×400m Area

Sensors drawn in triangular symbols above as shown in figure 4 and 5 indicate mobile sensors currently on transit to or from a service center. For each simulation rounds, the energy left within each sensor is estimated and when a sensor does not have enough energy to send its data, it is marked dead as depicted in figure 6.

Figure 6 shows the conclusive layout of a simulation where all sensors are dead as marked with an ’x’. At the end of each simulation, the network lifetime of the system is estimated from a graph depicting the number of active nodes plotted against the simulation rounds in figure 7.

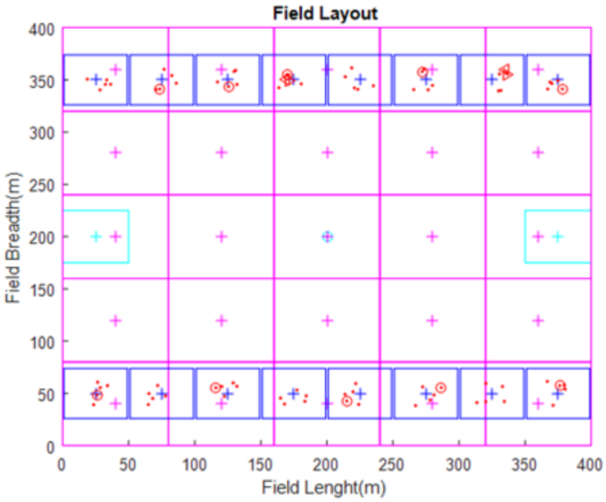


Fig. 5: Sample 5x5 cluster formation with a ward mapped to different cluster/zone

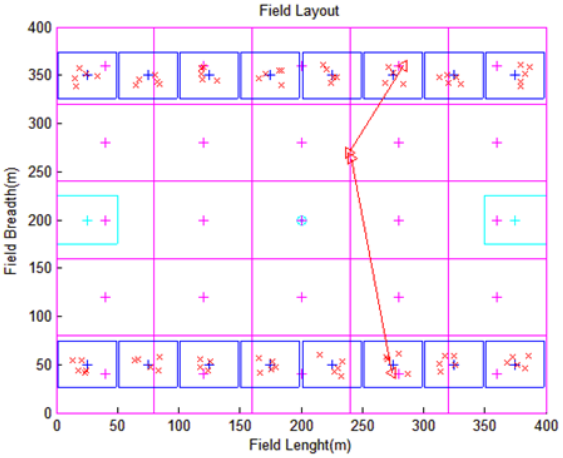


Fig. 6: Sample 5 × 5 cluster depicting dead sensors

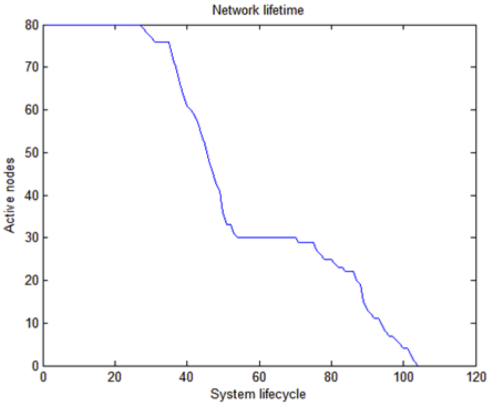


Fig. 7: Sample of a network lifetime graph indicating active nodes per round

B. Simulation Results

1) ***Combined Network Lifetim***e: Having repeated the stated procedure, some of the graphs for the cluster groups between non-hierarchical and hierarchical are generated and presented in figures 4 - 6. It should

be noted that in order to present brief results, we show cluster results for only 2 × 2 and 5 × 5 clusters. It is important to note that we see similar trends for other cluster sizes.

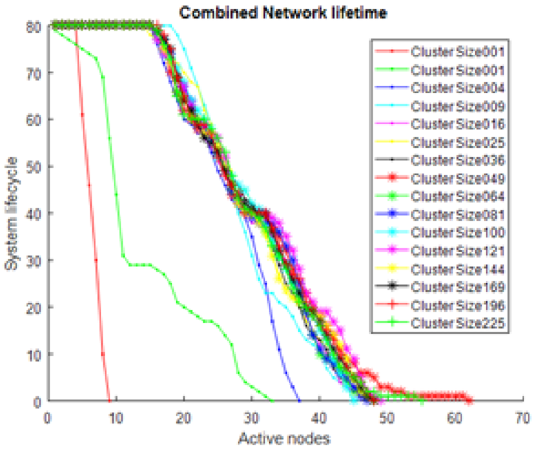


Fig. 10: Combined graphical network lifetime for considered routing protocols and cluster formation

From figure 10, it can be deduced that the "First-to-die" sensor time of the system is relatively proportional to the cluster groups. In the non-hierarchical network, the first sensor is observed dead at round xx. While in the hierarchical network, this time increases considerably. All these can be observed in table II:

The cluster formation that gave the best network lifetime was observed at 6 × 6 hierarchical cluster division. It should be noted, however, that the results obtained in this study are only applicable for the layout used. A change in the environment layout may definitely give a new result entirely. The results obtained here should therefore not be taken as a generic solution applicable to other instances.

Table II: Statistical average round for non-hierarchical and hierarchical orders

Routing Protocol/Order	Clustering	First-To-Die Round	Active node life time at 50%	Last-to-die sensor round	Statistical Average weight of rounds
N.H.O.	0	4	6	9	6.3
H.O. 1	1×1	2	10	32	14.7
H.O. 4	2×2	15	29	40	28
H.O. 9	3×3	17	27	46.5	30.2
H.O. 16	4×4	15	26	46.5	29.2
H.O. 25	5×5	16	30	47	31
H.O. 36	6×6	15	32	54	33.7
H.O. 49	7×7	16.5	34	47.5	32.7
H.O. 64	8×8	15	32	48	31.7
H.O. 81	9×9	15	29	46	30
H.O. 100	10×10	15	32	46.5	31.2
H.O. 121	11×11	15	28	50	31
H.O. 144	12×12	15	29	46	30
H.O. 169	13×13	14.5	27	47	29.5
H.O. 196	14×14	14.5	27	48	29.8
Hierarchic order 225	al 15×15	14.5	30	50	31.5

N.H.O. = Non Hierarchical Order; H.O. = Hierarchical Order

Table III shows a comparison between previous approaches and the approach followed in this paper. First, a comparison is made between the proposed model and the models reported in [11] and [14]. The statistical average weight of rounds in [14]; for non-hierachical is 300, 1115 for the hierarchical LEACH comprising of 5 cluster heads, and 3.7 for the round increase ratio. For [11], the round first node dies is 217, and the round last node dies is 468, and the statistical average weight of rounds is 451, for the non-hierarchical. Here, hierarchical LEACH with 5 cluster heads show a round first node dies of 1848 and round last node dies of 2609, with a statistical average weight of rounds increase ratio of 4.9. Finally, the proposed LEACH with 6 cluster heads reveal 1583 round first node dies, 2793 for the round last node dies, and 2188 statistical average weight of rounds. Here, results show that the proposed model agree closely with the existing models presented in [11] and [14].

Table III: Combining and comparing results

Simulation/ Model Type	Routing Protocol	Round First node dies	Round Last node dies	Statistical Average weight of rounds
Yadav model [14]	Non-Hierarchical	120	240	300
	Hierarchical (LEACH) (5 Cluster Heads)	930	1300	1115
	Round increase ra- tio	7.8	5.4	3.7
Own simulation using model [14]+ mobile sensors	Non-Hierarchical	35	74	54.5
	Hierarchical (LEACH) (3 Cluster Heads)	712	1793	1252.5
	Round increase ra- tio	20.3	24.2	22.98
Heinzelman model [11]	Non-Hierarchical	217	468	451
	Hierarchical (LEACH) (5 Cluster Heads)	1848	2609	2228.54
	Round increase ra- tio	8.5	5.6	4.9
Own simulation using model [11]+ mobile sensors	Non-Hierarchical	57	131	94
	Hierarchical (LEACH) (6 Cluster Heads)	1583	2793	2188
	Round increase ra- tio	27.8	21.32	23.7
Own Visual Studio Life/ validation model results	Non-Hierarchical	6	108	56
	Hierarchical (LEACH) (1 Cluster Heads)	89	128	172.5
	Round increase ra- tio	14.8	1.2	3.08

6 Conclusion and Future Work

A. Conclusion

This study has shown that quality of service, cost of maintenance and replacements of energy sources, healthcare giver efficiency can be greatly improved by implementing Hierarchical (LEACH) routing protocol. Cluster formation will be dependent on the size of locations in question and application. Choice of simulation parameters, locations of sensor nodes and base stations as well as other factors will always impact on results as seen in the comparison tables shared in section V of this paper.

In this study, results show that hierarchical routing technique based on low energy adaptive clustering protocol increased the energy efficiency of WSN for the predetermined simulation condition. It was shown that clustering and hierarchical routing techniques greatly improve the network lifetime and depending on the number of clusters formed, the network lifetime varies. It was observed however that the rise in the network lifetime with more clusters stopped at certain cluster level after which further increase in the cluster numbers resulted in either no increase in network lifetime or a further decrease.

B. Future Work

Future studies could use a practical healthcare facility layout as the basis for simulation. In this paper, we did not consider the case where patient wards are located in structures with more than one floor. Therefore, it is recommended that future works put this into consideration. Furthermore, in order to limit the amount of energy expended in sending data packets, these packets may be compressed prior to transmission. However, if the compression algorithm used consumes a noticeable amount of power, it should be carefully investigated.

C. Miscellaneous - Conflict of Interest Declaration

The authors declare that there is no conflict of interest regarding the publication of this paper.

REFERENCES

[1] D. Shinghal, N. Srivastava, et al., “Wireless sensor networks in agriculture: for potato farming,” 2017.

[2] B. Wang, X. Gu, L. Ma, and S. Yan, “Temperature error correction based on bp neural network in meteorological wireless sensor network,” International Journal of Sensor Networks, vol. 23, no. 4, pp. 265–278, 2017.

[3] Gauravpaliwal and Pankajkasar, “Article: Wireless body area net- work for ubiquitous mhealth mobile patient monitoring systems: Architecture, opportunities and challenges,” IJCA Proceedings on National Conference on Emerging Trends in Computer Technology, vol. NCETCT, pp. 1–6, December 2014.

[4] M. H. Anisi, G. Abdul-Salaam, M. Y. I. Idris, A. W. A. Wahab, and Ahmedy, “Energy harvesting and battery power based routing in wireless sensor networks,” Wireless Networks, vol. 23, no. 1, pp. 249–266, 2017.

[5] K. Fang, C. Liu, and J. Teng, "Cluster-based optimal wireless sensor deployment for structural health monitoring," Structural Health Monitoring, vol. 17, no. 2, pp. 266–278, 2018.

[6] B. Lo and G.-Z. Yang, "Body sensor networks-research challenges and opportunities," 2007.

[7] C. Umamaheswari, J. Gnanambigai, et al., "Energy optimization in wireless sensor network using sleep mode transceiver," Global Journal of Research In Engineering, vol. 11, no. 3, 2011.

[8] J. Stankovic, Q. Cao, T. Doan, L. Fang, Z. He, R. Kiran, S. Lin, S. Son, R. Stoleru, and A. Wood, "Wireless sensor networks for in- home healthcare: Potential and challenges," in High confidence med- ical device software and systems (HCMDSS) workshop, vol. 2005, 2005.

[9] V. Gupta and M. Doja, "H-leach: Modified and efficient leach protocol for hybrid clustering scenario in wireless sensor networks," in Next-Generation Networks, pp. 399–408, Springer, 2018.

[10] J. Kulik, W. Heinzelman, and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor net- works," Wireless networks, vol. 8, no. 2/3, pp. 169–185, 2002.

[11] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in System sciences, 2000. Proceedings of the 33rd annual Hawaii international conference on, pp. 10–pp, IEEE, 2000.

[12] D. Braginsky and D. Estrin, "Rumor routing algorthim for sensor networks," in Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, pp. 22–31, ACM, 2002.

[13] F. Zabin, S. Misra, I. Woungang, H. F. Rashvand, N.-W. Ma, and M. A. Ali, "Reep: data-centric, energy-efficient and reliable routing protocol for wireless sensor networks," IET communications, vol. 2, no. 8, pp. 995–1008, 2008.

[14] L. Yadav and C. Sunitha, "Low energy adaptive clustering hierarchy in wireless sensor network (leach)," International journal of com- puter science and information technologies, vol. 5, no. 3, pp. 4661– 4664, 2014.

[15] R. S. Istepanian, E. Jovanov, and Y. Zhang, "Guest editorial introduction to the special section on m-health: Beyond seamless mobility and global wireless health-care connectivity," IEEE Transactions on information technology in biomedicine, vol. 8, no. 4, pp. 405–414, 2004.

[16] R. C. Shah and J. M. Rabaey, "Energy aware routing for low energy ad hoc sensor networks," in Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE, vol. 1,pp. 350–355, IEEE, 2002.

[17] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," IEEE wireless communications, vol. 11, no. 6, pp. 6–28, 2004.

[18] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," Ad hoc networks, vol. 3, no. 3, pp. 325– 349, 2005.

[19] S. Dai, X. Jing, and L. Li, "Research and analysis on routing pro- tocols for wireless sensor networks," in Communications, Circuits and Systems, 2005. Proceedings. 2005 International Conference on, vol. 1, pp. 407–411, IEEE, 2005.

[20] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive proto- cols for information dissemination in wireless sensor networks," in Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, pp. 174–185, ACM, 1999.

Society for Science and Education
United Kingdom

TNC Transactions on Networks and Communications

Volume 7, No. 1
ISSN: 2054 -7420

Moment Generating Function Approach in Diversity Combiner with M-PSK over Hoyt Fading Channel

Adeyemo Z.K, Ajadi, A.S., Semire, F.A., Ojo, S.I.
Department of Electronic and Electrical Engineering,
Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria.
zkadeyemo@lautech.edu.ng

ABSTRACT

Wireless communication system is the processing, transmitting and receiving signals over an open space. This system suffers from time varying environment disturbance due to propagation of signals through different paths which prompt the signals to have different statistical distributions. The existing modified diversity combiners diversity such as Maximal Ratio Combiner (MRC), Equal Gain Combiner (EGC) with one Match Filter (MF) and one Radio Frequency (RF) chain, lack closed form expression. Therefore, in this paper, Moment Generating Function (MGF) approach is used to analyse the performance of the modified MRC over Hoyt fading distribution through the closed form expression. A closed formed expression is developed using the existing modified MRC with one Radio Frequency (RF) chain and one Matched Filter (MF) at the RF stage. Binary data of ten thousand bits are generated randomly as source data and modulated with M-ary Phase Shift Keying (M-PSK). The modulated signal is passed through the Hoyt fading channel which is then modeled using Moment Generating Function (MGF) approach in order to generate the resultant signals. The resultant signals at varying paths 'L' (2, 3 and 4) and Hoyt fading factors'q' (0.2, 0.4 and 0.6) are combined using the modified MRC. The output signal is passed through a comparator which compares the output signal with 9.5 dB set as threshold value. The outage probability of the modified MRC using M-PSK (2-PSK and 4-PSK) at varying 'L' and 'q' is evaluated, while processing time is used to compare the performance of the conventional MRC with the modified MRC. The results obtained with the modified MRC using closed form expression give lower Outage Probability (OP) and processing time. The research can be used by satellite communication system designers to improve the quality of service.

Keywords: Radio Frequency (RF), Moment Generating Function (MGF), Maximum Ratio Combiner (MRC), M-PSK, Match Filter (MF)

1 Introduction

Wireless communication is the transfer of information between two or more locations that are not connected by an electrical conductor. Wireless communication systems that have been deployed are first, second and third generations. The fourth and fifth generations systems are currently under deployment in some developing countries. The first generation wireless systems are analog in nature and make use of Frequency Division Multiple Access (FDMA) as multiple access technology. The second generating systems are digital in nature and use Time Division Multiple Access (TDMA) or Code Division Multiple

DOI: 10.14738/tnc.71.5972
Publication Date: 09nd February 2019
URL: http://dx.doi.org/10.14738/tnc.71.5972

Access (CDMA) as multiple access scheme. This has some features like data encryption, authentication, automatic location services and so on. The third generation systems are also digital in nature and are equipped with infrastructure to support Personal Communications Systems (PCS) such as Low Earth Orbit (LEO), satellite networks, wireless Asynchronous Transfer Mode (ATM) networks, mobile Internet Protocol (IP) and so on. Fourth and fifth generation systems have enhanced features over the third generation such as higher spectral efficiency, higher data rates, downlink speed of 100Mbps and uplink of 50Mbps. They use multicarrier modulation especially Orthogonal Frequency Division Multiple Access (OFDMA) and Multiple antenna that is Multiple Input Multiple Output (MIMO) to achieve the requirement of the system, but these are currently being deployed. Wireless technologies according to [30] use radio waves to make the distance between the transmitter and receiver appears to be short.

Signal transmitted through the wireless channels suffers from time varying channel which makes the received signal unpredictable due to fluctuations. This phenomenon is known as fading, which may vary with time, geographical location, or radio frequency, and is often modeled as a random process [15, 25, 21, 29]. Fading in wireless communication may either be due to multipath propagation referred to as multipath fading or to shadowing from obstacles affecting the signal propagation [2, 3, 10, 12]. Multipath propagation is a phenomenon that results in radio signals reaching the receiving antenna through two or more paths, which are due to atmospheric ducting, reflection, refraction, mountains and buildings. The multipath propagation effects are interference, signal delay and phase shifting of the signal. In wireless communication, multipath causes error and affects the quality of the received signal. It can be modeled statistically using different fading distributions such as Rayleigh, Rician, Nakagami, Weibull and Lognormal fading distributions [10, 11, 13, 17, 22, 24].

Nakagami distribution is of different forms namely: Nakagami q, Nakagami m and so on. Nakagami-q is also known as Hoyt distribution, this is being used in describing the short term signal variation of wireless communication systems [29]. Hoyt channel model has been statistically used in modeling of satellite communication. The faded signal through Hoyt fading channel can be processed by different mitigating techniques such as diversity technique, equalization technique and so on. Diversity technique has gained popularity in combating the effects of fading at the receiver. The basic principle of diversity is to create multiple copies of uncorrelated signals and combine them in an optimum way. Diversity can be classified as: space, frequency, time and polarization [9, 33, 31].

However, multiple signal copies must be uncorrelated or weakly correlated for diversity combining to be most effective. Hence, different types of diversity are normally efficient in different scenarios. The most widely used techniques are namely: Selection Combining (SC), Maximal Ratio Combining (MRC), Equal Gain Combining (EGC), and Hybrid Combining [26, 28, 20, 23, 18]. It is a known fact that Maximal Ratio Combiner (MRC) provides better performance than all other diversity combining techniques in multipath environment. However, this conventional MRC has the highest complexity of all combining techniques due to many hardwares involved. Modified Maximal Ratio Combiner which will have a reduced complexity relative to the conventional MRC scheme is being researched into, but lacks a closed form expression. Maximal Ratio Combiner outperforms others but at the expense of hardware complexity [28, 34].

This paper addresses the challenges of hardware complexity and closed form expression posed by the modified MRC with single MF and RF chain in Hoyt fading channel. Although, there are commonly used performance metrics in wireless communication namely: Signal-to-Noise Ratio (SNR), Average Outage

Duration (AOD), Bit Error Rate (BER), Amount of Fading (AF), and Outage Probability (P_{out}) [4]. However, Outage Probability which is the instantaneous error of probability that exceeds a specified value according to Goldsmith (2005) is employed in this paper for evaluating the performance.

2 Wireless Communication Channel

Signal transmission in wireless system is achieved through the inherent broadcast nature of electromagnetic waves. However, the electromagnetic waves are not guided to the receivers through some media in wireless communications. Therefore, in wireless communication systems, sending of radio signals is through the space, the signal radiated from the antenna reaches the receiver through different paths, this phenomenon which is known as multi-path fading in wireless communications, and dictates the quality of the received signal [8, 10, 16]. The limiting factors affecting the performance of radio propagation models are: path loss, shadowing, time dispersion, time variance, fading, interference and noise.

2.1 Path Loss

Path loss is the reduction in power density of an electromagnetic wave as it propagates through space. This phenomenon leads to signal attenuation [28]. According to [9], the path loss 'P_L ' is given as:

$$P_L = 20log_{10}\left(\frac{4\pi d}{\lambda}\right) \tag{1}$$

where: P_L is the path loss in decibels,
λ is the wavelength,
d is the transmitter- receiver distance.

2.2 Shadowing

Shadowing is a random variable which varies according to types of environment where radio wave propagates and does not depend on the distance between the receiver and the transmitter. This is experienced due to absorption of radio waves in the propagation by scattering structures. In term of the quality, shadowing loss varies faster than the path loss, but it is slower than the fading [24, 27].

2.3 Interference

This is a distortion which disrupts a signal as it travels along a channel, it can be expressed as addition of unwanted signal to a useful signal. The source of interference in a radio system can be originated at the system itself or it can be located at external source. The two major types of interference are Inter-Symbol Interference (ISI) and Co-Channel Interference (CCI), [28, 32].

2.4 Inter- Symbol Interference (ISI)

This is a form of distortion in which one symbol interferes with subsequent symbols and thus, making the communication less reliable, [28, 32].

2.5 Co-Channel Interference (CCI)

This is the overlapping of signal unintentionally from different radio transmitters with the same frequency. There are several causes of co-channel radio interference, such as: adverse weather conditions, poor frequency planning and overcrowded of radio spectrum, [28, 32].

3 Statistical Representation of Fading Channel

Multipath fading can be modeled with different statistical distributions in different environments. The commonly used statistical distributions in wireless communication are: Rayleigh, Rician and Nakagami distributions because of their close approximation to reality.

3.1 Rayleigh Distribution

In Rayleigh fading, channel model assumes that all the components that make up the resultant received signal are reflected or scattered and there is no direct path from the transmitter to the receiver. The Rayleigh distribution P_{rayl} (u) envelope of a received signal is given by Rappaport [28] as:

$$P_{rayl}\ (\mathrm{u})\ = \frac{u}{\sigma^2} exp - \left(\frac{r^2}{2\sigma^2}\right) \qquad 0 \le u \le \infty \tag{2}$$

where: u is the amplitude of the received signal

σ is the root mean square (rms) value of the received signal

σ^2 is the time-averaged power of the received signal

$2\sigma^2$ is the pre-detection mean power of the received signal

3.2 Rician Distribution

Normally, there exists a dominant Line of Sight (LOS) path in addition to numerous diffused multipath components between the transmitter and receiver in micro-cellular environments. In such a case, the other faded signal components are superimposed on the dominant component and the resultant signal amplitude follows a Rician distribution, with the ratio between the LOS and diffused components denoted by the Rice factor k which is equal to the ratio of power of line of sight component to the average power of the scattered components. The complex envelope of the received signal $P_{Rician}(u)$ is given by [28] as

$$P_{Rician}(u) = \frac{u}{\sigma^2} exp - \left(\frac{u^2+A^2}{2\sigma^2}\right) I_o\left(\frac{Au}{\sigma^2}\right) \qquad u \ge 0 \tag{3}$$

where: A is the peak amplitude of the dominant component,

$I_o(\frac{Au}{\sigma^2})$ denotes the modified Bessel function of the first land,

`u is the amplitude of the received signal.

$\frac{r^2}{2}$ is the instantaneous power

σ is the standard deviation of the local power

The distribution is expressed in term of the Rice factor k=A/($2\sigma^2$) which is the ratio of the dominant components power to the variance of the combined power of the multipath components. As the power in the dominant component decreases to zero, the Rician distribution can be shown to approach the Rayleigh distribution.

3.3 Nakagami-m distribution

It is possible to describe both Rayleigh and Rician distributions with the help of a fading channel model using the Nakagami distribution. Nakagami-m distribution often gives the best fit to land mobile and indoor mobile propagation, as well as scintillation of ionospheric radio links. It fits experimental data much better than a Rayleigh or Rician distribution [29, 31] The Nakagami-m distribution envelope of the received signal $P_{Nak}(u)$ is given by [24, 31] as

$$P_{Nak}(u) = \frac{2}{\Gamma(\mathrm{m})}\left(\frac{m}{2\sigma^2}\right)^m u^{2m-1} \exp\left(\frac{-mu^2}{2\sigma^2}\right) \qquad u \geq 0 \tag{4}$$

where: ‘σ’ is the received rms envelope level,
m is the fading severity index,
Γ (·) denotes the gamma function,
u is the amplitude of the received signal

When m =1, Nikagami distribution degenerates to a Rayleigh distribution and closely approximates the Rician distribution via relationship m = (k=1)^2/ (2k=1), [24].

3.4 Nakagami –q distribution

The Nakagami –q distribution also refers to as Hoyt distribution is commonly used to describe a short-term signal variation of certain wireless communication system subject to fading. Specifically, the Hoyt channel has been applied in satellite based cellular communications to characterize more severe fading condition than those modeled by Rayleigh. The Probability Density Function (PDF) of Nakagami-q distribution ‘$P_{Nakq'}$ (u) is given by [4] as:

$$P_{\mathrm{Nakq}}\ (\mathrm{u}) = \frac{u}{\sigma_1\sigma_2} \exp\left(\frac{u^2}{4}\left(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}\right)\right) I_o\left(\frac{u^2}{4}(\sigma_2^2 - \sigma_1^2)\right) \tag{5}$$

where: I_o (.) denotes the zeroth order modified based function of the first kind,
u is the amplitude of the received signal.
σ_1 and σ_2 are zero mean and variance, respectively.

4 MRC with Single MF and Single RF Chain

Conventional MRC uses many RF chain and Match Filters (MFs) due to multiple copies of the transmitted signal resulting in high hardware complexity. But MRC with single MF and single RF chain makes use of only one RF chain and one MF [9]. This is performed by co-phasing and weighing the received signals through different branches before adding. This type of MRC is shown in Figure 1, where $H_{1,}$ H_2 …H_L represent the Hoyt fading through different paths over which the transmitted signals S_1 (t), S_2 (t)…S_L (t) propagate to the receiver. The faded signals are combined by MRC and passed through only one RF chain and one MF for reception.

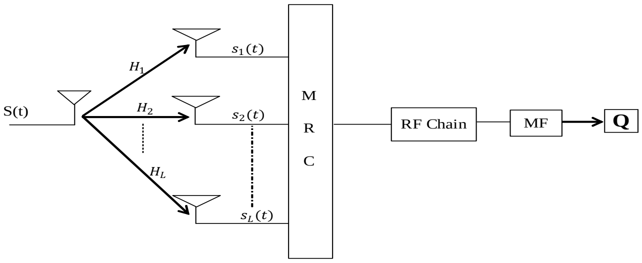


Fig.1: MRC with single MF and single RF chain

5 Development of Closed Form Expression for System Model

In signal transmitting over Hoyt fading channel, the baseband representation of the received signal ‘y’ is given by [29] as:

$$\mathrm{y = Xs + n} \tag{6}$$

where; ‘X’ is the channel fading envelop which is Hoyt distributed
‘s’ is the transmitted baseband signal and
‘n’ is the Additive White Gaussian Noise (AWGN)

The receiver weights all the inputs signals separately and sums them to make a decision. The transmitted symbols of the instantaneous output SNR per symbol 'γ'is expressed by [29] as

$$\gamma = \frac{E_b}{N_o}\alpha^2 \tag{7}$$

Mathematically, instantaneous output from MRC γ_{MRC} can be expressed as:

$$\gamma_{MRC} = \frac{E_b}{N_o} ({\alpha^2}_1 + {\alpha^2}_2 + \cdots + {\alpha^2}_L) = \frac{E_b}{N_o}\alpha^2 \tag{8}$$

where; α^2= (${\alpha^2}_1 + {\alpha^2}_2 + \cdots + {\alpha^2}_L$), which means that the PDF of γ_{MRC} can be obtained from the PDF of the Random Variable RV α^2.

The Nakagami-q (q being the fading severity parameter) distribution is generally used to characterize the environment that is more severe than Rayleigh fading. The corresponding PDF of the fading envelope f $\alpha(\alpha)$ is given by [29] as:

$$\mathrm{f}\,\alpha(\alpha) = \frac{(1+q^2)\alpha}{q\Omega}\, \mathrm{exp}\, [- \frac{(1+q^2)^2\alpha^2}{4q^2\Omega}]\, \mathrm{I_O}\, [\frac{(1-q^4)\alpha^2}{4q^2\Omega}] \, ; \alpha \geq 0 \tag{9}$$

where $\Omega = \mathrm{E}[\alpha^2]$ and $\mathrm{I_O}$(.) denotes zeroth order modified Bessel function of first kind.

According to [29], the SNR per symbol of the channel f_γ (γ) is distributed as

$$f_\gamma\,(\gamma) = \frac{(1+q^2)\alpha}{2q\gamma_a}\, \mathrm{exp}\, [- \frac{(1+q^2)^2\gamma^2}{4q^2\gamma_a}]\, \mathrm{I_O}\, [\frac{(1-q^4)\gamma}{4q^2\gamma_a}] \, ; \gamma \geq 0 \tag{10}$$

Finding the kth moment using (10), the indefinite integrals involving exponential and Bessel functions are solved using Marcum's Q function and the closed form expression of kth moment of γ output SNR obtained as:

$$E(\gamma^k) = \Gamma(1+k){}_2\mathrm{F}_1\,(-\frac{k-1}{2}, \frac{-k}{2}; 1, (\frac{1-q^2}{1+q^2})^2)\gamma_a^k \tag{11}$$

In order to quantify the performance in term of OP, MGF based approach is used with Pade Approximation (PA) technique to find simple way of evaluating rational expression for the MGF.

5.1 Moment Generating Function (MGF) of the SNR Output

The MGF $M_\gamma(s)$ is the Laplace transform of the PDF of a distribution and is given by [1, 29] as

$$M_\gamma(s) = E(\mathrm{e}^{s\gamma}) = \int_0^\infty \mathrm{e}^{s\gamma} P_\gamma(\gamma)\, d\gamma \qquad \gamma > 0 \tag{12}$$

By expanding ($\mathrm{e}^{s\gamma}$) in equation (12), results in

$$\mathrm{M}_\gamma(\mathrm{s}) = \sum_{k=0}^\infty \frac{s^k\, \Sigma(\gamma^k)}{k!} = \sum_{k=0}^\infty \frac{s^k\, H_k}{k!} \tag{13}$$

where $\sum(\gamma^k)$ = $\mathrm{H_k}$ and is equal to the kth moment represented by equation (12)

$$M_\gamma(s) = \frac{(-1)^k\Gamma(1+k)}{k!}{}_2\mathrm{F}_1\,(-\frac{k-1}{2}, \frac{-k}{2}; 1, (\frac{1-q^2}{1+q^2})^2)X^{\mathrm{k}} \text{ where } X = s\gamma_a \tag{14}$$

when $k = 0$ in 3.10, the coefficient of the expression becomes one. Using MATLAB 7.0 software when $q = 0{:}\,0.2{:}\,1,$ k ranges from 0 to 19.

$$M_\gamma(s) = 1\text{-}0.8893x + 0.6870x^2 - 0.4276x^3 + 0.1545x^4 + 0.0885x^5 - 0.2653x^6 + 0.3539x^7 - 0.3500x^8 + 0.2667x^9 - 0.1313x^{10} - 0.0213x^{11} - 0.1559x^{12}\,{}^{-}0.2441x^{13} + 0.2699x^{14} - 0.2320x^{15} + 0.1432x^{16} - 0.0267x^{17} + 0.0894x^{18} + 0.1786x^{19} \quad (15)$$

Padé Approximants (PA) of the MGF of SNR Output

Employing the work of [6, 8] a PA which is a rational function approximation of a power series is used to rationalize equation 3.12 in order to obtain its inverse Laplace transform. In this case, $M_\gamma(s)$ has a specified order B for the denominator and A for the numerator, where A and B are positive integers [1].

$$R(x) = \frac{\sum_{i=0}^{A} a_i x^i}{1+\sum_{j=0}^{B} b_j x^j} = M_\gamma(s) = \sum_{n=0}^{N} C_n x^k \quad (16)$$

where C_n are the coefficients of equation (16), B= A+ 1. This implies that order of denominator is greater than that of the numerator by 1. a_i and b_j are real coefficients which are determined by solving the set of A + B + 1 equations, with an assumption that b_0=1

$$\sum_{j=0}^{B} b_j\, c_{B-1-j+l} = 00 \le l \le B \quad (17)$$

After solving for the values of $b_{j,}$ the set a_i is obtained through back substitution from,

$$a_i = c_i + \sum_{p=1}^{\min(B,i)} b_i\, c_{i-p} = 00 \le l \le A \quad (18)$$

MATLAB was used to generate the coefficients a_i and b_j. The Hankel matrix rank deficient above N= 15. and the PA of equation (16) is given in equation.

$$R(x) = \frac{1+7.0528\text{x}+ 22.1531\text{x}^2+ 40.4388\text{x}^3 +46.8885\text{x}^4+35.3433\text{x}^5+ 16.9064\text{x}^6+4.6910\text{x}^7+ 0.5779\text{x}^8}{1+7.921x+28.5288x^2+60.7805x^3+845825x^4+79.6900x^5+50.8123x^6+21.1380x^7+5.2049x^8+0.5779x^9} \quad (19)$$

5.2 Outage probability of MRC Output

The signal outage probability is defined as the probability that the instantaneous SNR will fall below a certain threshold.

$$\text{P}_{\text{out}}\,(\gamma_{th}) = \text{Pr}\,(\,SNR) \le \gamma_{th}) \quad (20)$$

For MRC receiver with L identical and independently distributed channels, the signal outage probability is given by

$$\text{P}_{\text{MRC\_out}}\,(\gamma_{th}) = \frac{1}{2\pi j}\int_{\varepsilon-j\alpha}^{\varepsilon+j\alpha} \frac{[M_\gamma(S)]^L}{S} e^{S\gamma_{th}}\text{ds} \quad (21)$$

where ε is a properly chosen constant in the region of convergence of complex s-plane. Since $M_\gamma(s)$ is in term of rational function, it is, therefore, possible to use partial fraction expression of $\frac{[\,M_\gamma(s)]^L}{s}$ in (21) to evaluate outage probability, meaning

$$\begin{aligned}\text{P}_{\text{out}}\,(\gamma_{th}) &= \frac{1}{2\pi j}\int_{\varepsilon-j\alpha}^{\varepsilon+j\alpha} \sum_{i=1}^{N_p} \frac{\lambda_i}{S+P_i} e^{S\gamma_{th}}\text{ds} \\ &= \frac{1}{2\pi j}\sum_{i=1}^{N_p}\int_{\varepsilon-j\alpha}^{\varepsilon+j\alpha} \frac{\lambda_i}{S+P_i} e^{S\gamma_{th}}\text{ds} \quad (22) \\ &= \sum_{i=1}^{N_p} \lambda_i\, e^{P_i\gamma_{th}}\end{aligned}$$

where p_i are N_p poles of rational function in 's' with $'\lambda_i'$ its residues. Each term inside the summation in (8) represents a simple rational function form.

6 Development of the System Simulation Model

The system model consists of a transmitter to process the randomly generated data which is available within the MATLAB simulation software for transmission by converting the source data into bits, reshaping and modulating with M-PSK signaling scheme. The Square-Root Raised Cosine (SRRC) filter is used at both transmitter and receiver to reduce the effect of the Inter Symbol Interference (ISI) and the match filtering criterion to maximize SNR of the system. The acquired data is processed using modified Maximal Ratio Combining technique. The complete system model for this work is shown in Figure 2. The system model is simulated using MATLAB simulation software with some simulation parameters contained in Table 1. in accordance with the wireless standards. The Outage probability is calculated by equating the threshold value to be 9.5 dB.

Table 1: System simulation parameters for modified MRC

Parameter	Specification
Modulation schemes	M-PSK
Fading	Hoyt
Number of MRC paths	3,4,5
Carrier Frequency	1800MHz
Bandwidth of symbol	250kHz
Delay spread	250ns
Noise	AWGN
Transmit Filter	Square Root Raised Cosine
Receiver Filter	Square Root Raised Cosine
Roll of factor	0.25
Number of samples/symbol	10
SNR	0:2:12
Number of symbol (data length)	10,000

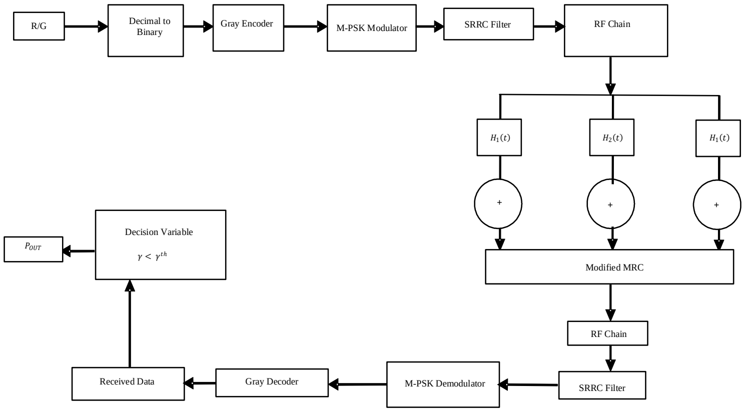


Figure 2: The Developed System Simulation Model.

7 Results and Discussion

The results obtained for the modified MRC at varying paths ‘L’(2,3,4) and different Hoyt fading factors-q 0.2, 0.4 and 0.6 using M-PSK signaling schemes are presented in Figures. 3-11. Figure 3 shows the OP values versus Signal to Noise Ratio for the modified MRC for varying ‘L’ at $q = 0.2$ using Binary Phase Shift Keying (BPSK) scheme. The OP values obtained at the threshold value of 9.5 dB for L=2, 3 and 4 are 0.1499, 0.1354 and 0.1217, respectively. Also, Figure 4 depicts the simulated OP versus signal to Noise Ratio of the modified MRC at q= 0.4 using 2-PSK in which OP values obtained at 9.8 dB are 0.99001, 0.97111, 0.90017 for L=2, 3 and 4, respectively. Similarly, Figure 5 shows the OP versus Signal to Noise Ratio of the modified MRC at q=0.6 using 2-PSK signalling scheme. It can be confirmed from Figure 5 that the OP values obtained are 0.97111, 0.95002 and 0.90001 for L=2, 3 and 4, respectively, while Figure 6 reveals the OP versus SNR of the modified MRC at q=0.2 using 4-PSK signaling scheme known as Quadrature Shift Keying (QPSK) scheme. The OP values obtained for Figure 6 at SNR of 9.8 dB are 0.9900, 0.9982 and 0.98223 for L=2,3,4, respectively. Table 2 contains the processing time of the conventional MRC and the modified MRC at q=0.2 with 2-PSK (BPSK) scheme. It can be confirmed that the processing time (s) at q=0.2 for 2-PSK (BPSK) scheme for conventional MRC are 3.88527, 4.72021 and 5.470033s for 'L'=2, 3 and 4, respectively, while the processing time at q=0.2 with 2-PSK for the modified MRC are 0.744267, 1.503282 and 2.18500 for L=2,3 and 4, respectively. The results obtained with the modified MRC using closed form expression give lower OP and processing time. The results are justifiable in that only one RF chain and one MF are used at RF stage, thereby reducing the hardware complexity as a result of reduction of processing time. It is also observed that as the number of paths 'L' increases, the OP decreases while the processing time increases, thereby improving the performance. It can also be deduced as Hoyt fading factor 'q' increases indicating decreases in severity of fading resulting in decrease in probability of outage. The result could be used by satellite communication system.

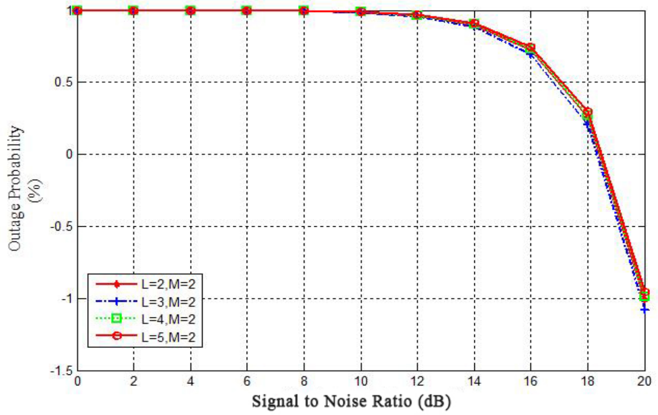


Figure 3: Simulated P_{out} of 2-PSK (BPSK) versus Signal-to-Noise Ratio (dB) with the Modified MRC for Varying L at q=0.2.

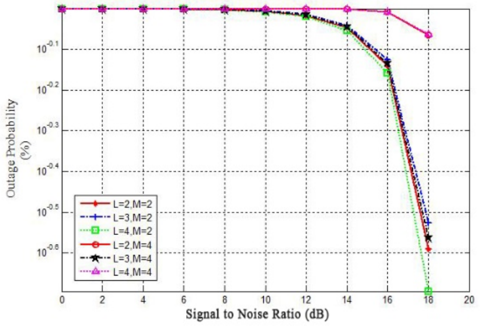


Fig. 4: Simulated P_{out} versus Signal-to-Noise Ratio (dB) with the Modified MRC for Varying 'L' and M at q=0.4.

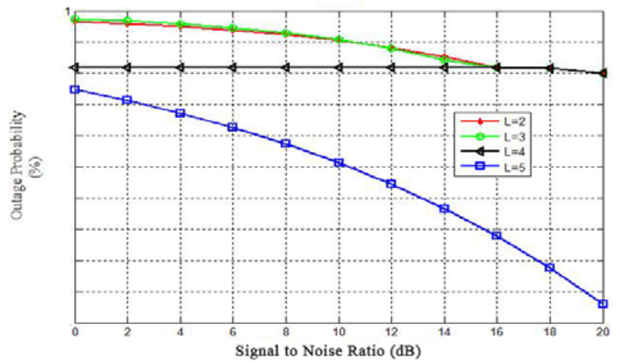


Figure 5: Outage Probability versus Signal-to-Noise Ratio (dB) with Modified MRC for Varying 'L' at q=0.6

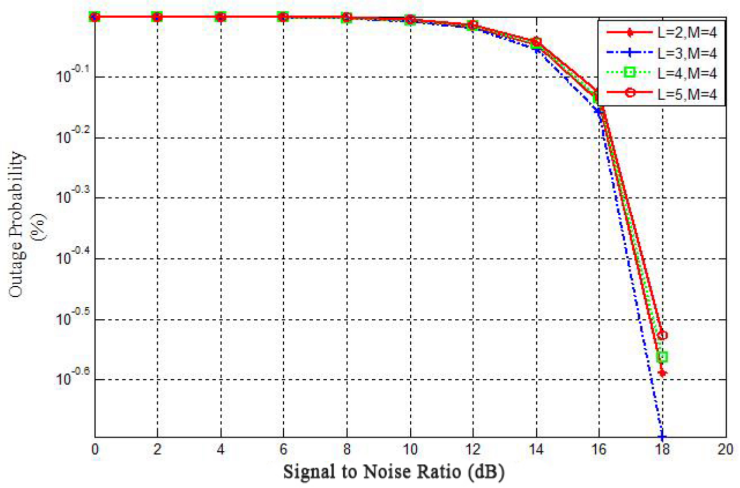


Figure 6: Simulated P_{out} of 4-PSK (QPSK) versus Signal-to-Noise Ratio (dB) with the Modified MRC for Varying L at q=0.2

8 Conclusion

The Moment Generating Function (MGF) approach in MRC with single MF and single RF chain using M-PSK over Hoyt fading channel has been developed. The modified MRC with single RF chain and single MF has been investigated over Hoyt fading channel. The closed form expression for the modified MRC over different Hoyt fading factors using MGF approach has been developed for various paths 'L' (2,3,4). The developed channel model has been incorporated in the system model using M-PSK signaling scheme at different constellations. The system model has been simulated using MATLAB simulation software and evaluated using outage probability. It has been shown that the modified MRC using closed form expression gives lower outage probability and processing time.

REFERENCES

[1] Abramovitz, M., and Stegun, I.A, "Hand book of Mathematical functions with formulas, graphs, and mathematical tables" Dover, New York, USA. (1992),

[2] Adeyemo, Z. K. and Raji T. I, "Effects of Diversity combining in mobile Terrestrial Environment" Continental Journal of Engineering Sciences 5:27-37. (2010)

[3] Adeyemo, Z. K. and Ojedokun Isaac A., "EGC Receiver using single Radio Frequency chain and single matched filter over combined Rayleigh and Rician Fading channels" ARPN Journal of Engineering and Applied Science, (2014). 9(7): 1819-6608.

[4] Anamllyas, Ejaz Ansari and Saleem Akhtar "Accurate BER\SER Analysis and performance of different modulation schemes over wireless fading channel," Science Inernation, (2013). 25(2):367-374.

[5] Adeyemo Z.K, Rabiu E.O and Abolade R.O. Offset Phase Shift Keying modulation in Multiple-Input Multiple-Output Spatial Multiplexing; Transaction on Networks and Communications, (2015): 3(20): 117-127.

[6] Amindavar H., and Ritcey J. A., "Pade approximation of probability density functions", *IEEE Transaction Aerospace.* Electron System, (1994). 30(1):416-424

[7] Annamalai A., and Tellambura C., "Performance evaluation of generalized selection Diversity systems over Nakagami–m fading channels", Wireless Communications and Mobile Computing, (2003). 3(1):99 – 116.

[8] Blaunstein, N., and Christodoulou, C. G., "Radio Propagation and Adaptive Antennas for Wireless Communication Links", John Wiley and Sons, Inc., Hoboken, New Jersey. (2007),

[9] Brennan, D.G, "Linear Diversity Combining Techniques," Proceeding of the IEEE*,* (2003), *91* (2):1075-1102

[10] Dighe, P. A., Mallik, R. K., and Jamuar, S. S., "Analysis of transmit-receive diversity in Rayleigh Fading*",IEEE Transactions on communication,* (2003). 51(1):694 – 703.

[11] Dimitris A. Zogas "Equal Gain Combining over Nakagami-n (Rice) and Nakagamiq(Hoyt) Generalized fading channels*"* IEEE Transactions on wireless communications, (2005). 4(2): 374-379

[12] Goldsmith, A., "Wireless Communications*,*" Standford University Press, California. (2005),

[13] Helstrom C.W., "Probability and Stochastic processes for engineers", Second edition Macmillan, New York, USA. (1991),

[14] Hoyt, R., "Probability functions for the modulus and angle of the normal complex variate," Belly Syst. Tech. J., (1947). 26(1):318-359.

[15] Jahn, A., "Propagation Considerations and fading counter measures for Mobile Multimedia Services", Int. Journal of Satellite Communication*,* (2001). 19(3), pp 223-250

[16] Jochen S., Mobile Communications*,* 2nd Ed. Pearson Education Ltd., India. (2006).

[17] Jyoteesh Malhotra, Ajay Sharma, K., and Kaler R.S., "On the performance analysis of wireless receiver using generalized- gamma fading model", Annals of Telecommunications. (2009). 46(1) :147-157.

[18] Karagiannidis G. K., Sagias N. C., and Zogas D. A. "Error analysis of M-QAM with equal Gain Combining diversity over generalized fading channels", IEEE Transactions on wireless communications, (2005). 152(1):69-74.

[19] Kim S.W. and Wang Z. "Maximum Ratio Diversity Combining Receiver using single Radio Frequency chain and single Matched Filter", IEEE Globecom proceedings, (2007). 2(1):269-274.

[20] Lee T., and Lee Z "A Beam Space Diversity Combiner for sector Division Multiple Access Communications", IEEE Antennas and Propagation Society Symposium Digest, Atlanta, (1998) 1(1):372 – 375.

[21] Malhotra J. “Investigation of M-QAM and MPSK with EGC in Generalized Flat Fading channels”, Journal of Advances in information Technology, (2011). 2(4):250-256.

[22] Malhotra J., Sharma A. K., and Kaler R. S. “On the performance Analysis of wireless receiver using generalized-gamma fading Model”, Annals of Telecom, (2009), Vol. 64, No. 1, pp147-153.

[23] Mohammed J., Leszek S., and Mustapha B., “Outage probability of Diversity combining Receivers in Arbitrarily Fading channels, (2011)

[24] Nakagami M., “The m-distribution- A General Formula of intensity Distribution of Rapid Fading” Statistical Methods of Radio Wave Propagation, pergamon (1960). 1(1):3-36.

[25] Patzoid, M., “Mobile Fading Channels” John Wiley and Sons, Ltd, Baffins Lane, Chichester, West Sussex, po191 IUD, England. (2002).

[26] Pornchai, S., Wanaree, W. and Sawasd, T. “Performance of M-PSK in Mobile Satellite Communication Over Combined Ionosphere Scintillation and Flat Fading Channels with MRC Diversity”*,* IEEE Transactions on Wireless Communications, (2009), Vol. 8, No. 7.

[27] Proakis, J. G, “Digital Communications*”,* McGraw-Hill companies, incorporation. International Edition, 2001.

[28] Rappaport, T.S., “Wireless Communication Principles and Practice” 2nd edition Prentice Hall of India Private limited view Delhi, (2002), pp1-526.

[29] Simeon M.K.and Alouini, M. S, “Digital Communications Over Fading Channels*”,*2nd edition, John Wiley and Sons, Incorporation, Hoboken, New Jersey, (2005), Pp1 – 523.

[30] Stewart, K.A., Labedz, and sohrabi K., “Wideband channel measurements at 900MHZ” in Proc. IEEE Vehicular Technology Conf. (VTC’95), Chicago IL, (1995) pp 236-240.

[31] Tang L. and Hongbo Z., “Analysis of Nakagami Fading channel with MATLAB” Asifat Pacific Conference on Environmental Electromagnetic CEEM Hang zhou, China. (2003), Pp490 – 494.

[32] Tse, D., and Viswanath, P., “Fundamentals of Wireless Communication” 1st Edition; Cambridge University Press, New York. (2005).

[33] Ye, Z. and Satorius “Channel modeling and simulation for mobile user objective system (MUOS) – part 1: flat scintillation and fading”, in proceeding IEEE ICC, (2003), 5(1):3508-3510.

TNC TRANSACTIONS ON NETWORKS AND COMMUNICATIONS

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

VOLUME 7, NO. 1
ISSN: 2054 -7420

Fault-Tolerant Adaptive Routing Algorithm for 2D Torus Network

Tsukasa-Pierre Nakao, Yasuyuki Miura, Naohisa Fukase
Shonan Institute of Technology, Kanagawa, JP;
nakaot@center.shonan-it.ac.jp; miu@info.shonan-it.ac.jp; fukasen@center.shonan-it.ac.jp
Tsujido-Nishikaigan, Fujisawa, Kanagawa, Japan

ABSTRACT

A 2D torus network is one of the most popular networks for parallel processing. We have researched the North-South First (NSF) routing which is applicable to a 2D torus and combines the north-first (NF) and south-first (SF) methods. We focused on the proposal of a routing algorithm aimed at avoiding congestion of the crowded network. It was superior in congestion tolerance but not in fault tolerance. We have therefore been researching algorithms considering fault tolerance of the NSF method. In this paper we propose an NSF-FT method which is a new routing algorithm with improved fault tolerance. We evaluated the congestion resistance and fault tolerance of the proposed method by dynamic communication performance evaluation by simulation. The software simulation showed that the proposed algorithm has higher performance.

Keywords: Network on Chip; Interconnection Network; Adaptive Routing; Turn Model; Fault Tolerance.

1 Introduction

The interconnection network is an important topic in the field of parallel processing. Parallel computers have processing elements (PEs) that are directly connected to a network such as a *k*-ary *n*-cube. Parallel processing is also performed in a *Network on Chip* (NoC) between PEs located on one chip. Many different interconnection networks for parallel processing have been proposed, and the 2D torus network is one of the most popular networks for parallel processing.

The routing algorithms of interconnection networks are classified into deterministic routing, in which paths are fixed, and adaptive routing [1-11], in which paths are changed to avoid failures or congestion. Because of its tolerance to failures and congestion, adaptive routing has been the topic of a lot of research. Various adaptive routing algorithms have been proposed for *k*-ary *n*-cubes [6]-[10, 12, 13].

[However, these methods require additional hardware for virtual channels comparison with deterministic routing (its name is *Dimension Order Routing*, DOR). For example, Dally et al. proposed two types (dynamic and static) of *Dimension Reversal routing* [7]. Those methods require at least one (dynamic) or two (static) additional virtual channels for achieving adaptive routing. The fully adaptive routing for *k*-ary *n*-cubes can also be based on Duato's method [11], but this method requires one additional virtual channel for the bypass path needed for achieving adaptive routing.

DOI: 10.14738/tnc.71.6032
Publication Date: 09th February 2019
URL: http://dx.doi.org/10.14738/tnc.71.6032

A number of adaptive routing algorithms based on the turn model [14-17] do not need additional virtual channels. However, most of these algorithms cannot be applied to torus networks without change. If an adaptive routing algorithm for a torus network could be realized by modifying the turn model, it would be possible to realize adaptive routing without having to install additional virtual channels [18].

We have previously proposed the North-South First (NSF) algorithm, which is the combination of North First and South First algorithms [19-23]. Since up to now we focused on the proposal of a routing algorithm aimed at avoiding congestion of the interconnection network, the fault tolerance of the NSF algorithm was not sufficient.

We improved the NSF algorithms and propose an improved North-South First method (NSF-IP, NSF-ImProved), which is a fault-tolerant routing algorithm. And we evaluated both its congestion resistance and its fault tolerance in dynamic communication performance by simulation[23]. However, the fault tolerance of NSF-IP is not enough [23]. It is thought that the fault tolerance can be by changing the routing policy when a packet arrives at a faulty PE.

In this paper we propose the *NSF-FT* (NSF- Fault-Tolerant) algorithm to improve the fault tolerance by improving the NSF-IP algorithm. We evaluate its congestion resistance and fault tolerance by software simulation.

2 2D Torus Network

The structure of a 2D torus network is shown in Figure 1. The network has an *N*×*N* two-dimensional structure, and its four edges are connected by wraparound links. It is used in many parallel computers and some interconnection networks.

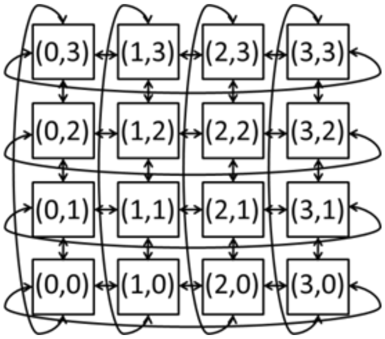


Figure 1: Structure of a 2D-torus network.

Dimension-order routing (DOR) is generally used for deterministic routing on a 2D torus. In DOR, the packet moves on channels in the y direction before moving in the x direction. To avoid deadlocks on a 2D torus, DOR needs two virtual channels (channel-L and channel-H).

- Choose channel-L when starting routing in the y direction.
- When the head of the packet passes through a wraparound link, move the packet to channel-H.
- When the routing in the y direction is completed, move the packet in the x direction; use channel-L regardless of the current channel.
- When the head of a packet passes through a wraparound link, moves the packet to channel-H. Use channel-H until the routing finishes.

The link selection function and channel selection function of dimension-order routing on an N × N torus network are shown in Figures 2 and 3.

```
/ Link Selection Function for Dimension-Order Routing
Link_Select_DOR (cx, cy, dx, dy)
  cx, cy;  // current node  0 ≦ cx, cy ≦ N－1
  dx, dy;  // destination   0 ≦ dx, dy ≦ N－1
{
  if(cy≠dy){                     // dimension Y
     dist_y = (N+dy-cy)%N;
     if(1≦dist_y ≦N/2)           return Y+;
     else                        return Y-;
  }
  else if(cx≠dx){                // dimension X
     dist_x = (N+dx-cx)%N;
     if(1≦dist_x ≦N/2)           return X+;
     else                        return X-;
  }
  else return OUT;

}
```

Figure 2: Link selection function of the dimension-order routing.

```
// Channel Selection Function for Dimension-Order Routing
Channel_Select_DOR (cd, cc, nd)
   cd;               // current direction    ∈{Y+, Y-, X+, X-}
   cc;               // current channel      ∈{L, H, W}
   nd;               // next direction       ∈{Y+, Y-, X+, X-}
{
   if(cc∈L)  return L;                // before wrap around
   else if(cc∈W)  return H;           // in wrap around
   else
      if(cd∈{X+,X-} & nd ∈{Y+,Y-})
             return L;    // Y-routing → X-routing
      else   return H;                // after wrap around
}
```

Figure 3: Channel selection function of the dimension-order routing.

Figures 2 and 3 show the link selection function and channel selection function of DOR on an N×N torus. Here the address of each PE of the torus is shown in terms of its coordinates (*x*,*y*). Moreover, the y-direction channels are written as Y+ and Y−, and the x-direction channels are written as X+ and X−. The four inputs of the link selection function indicate the *x* and *y* coordinates of the present PE, and the *x* and *y* coordinates of the destination PE. The function outputs the link of either X+, X−, Y+, Y− or ``OUT'', which is an output link to a node.

The three inputs of the channel selection function correspond to the current direction, current channel, and direction of the next hop. The current direction and the direction of the next hop have four states, i.e., X+, X−, Y+, and Y−. The current channel has three states, i.e., channel-L (L), channel-H (H), and wraparound channel (W). Although the output has two states (L and H), it unconditionally serves as W when the selected link is a wraparound link.

3 Adaptive Routing of *k*-ary *n*-cube

3.1 Turn Model

The turn model [15] is used by some adaptive routing algorithms [16, 17]. Packet cycles can be prevented by adding a restriction to a path change (turn) of a packet. In the case of a 2D mesh, there are eight kinds of turn, and the various turn model methods put restrictions on two of the eight turns. There is essentially no difference between these methods other than the choice of turn to be restricted. In this paper, we shall incorporate the North First (NF) algorithm and South First (SF) algorithm into one (NSF) and apply it to a 2D torus.

3.2 North First (NF) Routing

The turn model of DOR for a 2D mesh is shown in Figure 4, and the turn model of the NF algorithm is shown in Figure 5. DOR restricts four out of eight turns, whereas the NF algorithm restricts only two, i.e., X− (left, west) → Y+ (upper, north) and the X+ (right, east) →Y+ (upper, north). The South First algorithm, by which the Y− (South) direction is chosen at the beginning of a routing path, is similar.

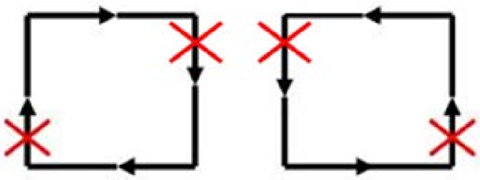


Figure 4: Turn model of dimension-order routing for a 2D-mesh network.

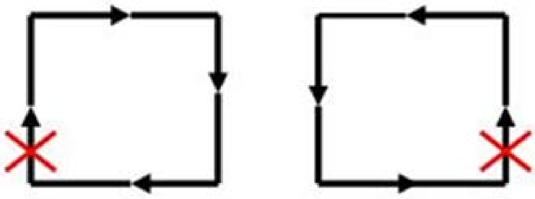


Figure 5: Turn model of north-first (NF) algorithm for 2D-mesh network.

3.3 Application of the Turn Model to a Torus Network

When applying a turn model such as the NF algorithm to a torus network without change, the following differences from the case of a mesh network have to be considered.

(1) In a torus network, when the packet passes through a wraparound channel, a deadlock by cyclic dependency can occur. Therefore, it is necessary to impose an additional restriction.

(2) At least two virtual channels are needed for routing in a torus network. As a result, adaptive routing with higher pliability is attained by applying different turn models to each channel.

An example of a cyclic dependency that occurs in the NF algorithm is shown in Figure 6. Here, packets A–D mutually block a path, causing a deadlock. By contrast, the deadlock does not happen in DOR because packets A and C do not turn in Figure 6. This problem illustrates that in adaptive routing on a torus network it is necessary to take into consideration complicated turn restrictions. Our method deals with this issue by applying the NF and SF algorithms to channel-H and channel-L.

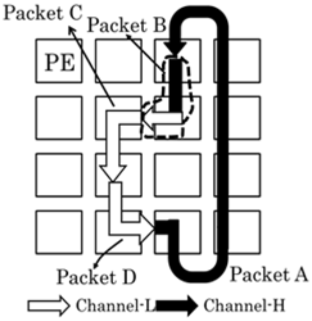


Figure 6: Cyclic in the NF algorithm running on a torus network.

4 NSF Algorithm

When the turn model of NF or SF is applied to a 2D torus directly, cyclic dependency like that shown in Figure 6 occurs because of channel wrap-around. To avoid the deadlock described above, additional restrictions have to be put on the NF and SF algorithms:

1) The SF algorithm does its routing on channel-H. However, a cycle may occur when a path is chosen in which a packet returns to channel-L through channel-H, and for this reason, DOR is carried out instead of the adaptive routing. In DOR, the x-direction channel chosen after a vertical (y-direction) wraparound channel has to be channel-L.

2) The NF algorithm does its routing on channel-L. Because the path of channel-H → channel-L exists after a wraparound channel, the cycle shown in Figure 6 occurs. As shown in Figure 6, though, the cycle can be avoided by adding one more restriction to the other two. Here, three restrictions are put on eight turns, specifically, right → upper, left → upper, and right → lower. This algorithm was named {\it restricted North First (rNF) [19]. The turn model of rNF is shown in Figure 7.

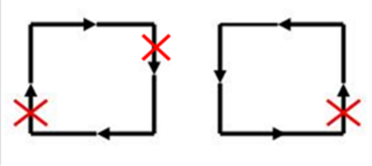


Figure 7: Restricted north-first routing.

From here on, all channels will be described in terms of their dimension $d \in \{X, Y\}$, direction $\delta \in \{+, -\}$, channel type $c \in \{L, W, H\}$, i.e., $(d\delta, c)$. X means X dimension, Y means Y dimension, and L L, W, and H means channel-L, wraparound channel, and channel-H. $(d+, c)$ and $(d-, c)$ will be shown as a set, written as $(d\pm, c)$.

4.1 Routing Algorithm of NSF Routing

In our method, the restricted NF algorithm is carried out in channel-L and the SF algorithm is carried out in channel-H. Since $(Y+, L)$ and $(Y+, H)$ are respectively used in the restricted NF algorithm and SF algorithm, we will study cases in which $(Y+, c)$ is used and not used and cases in which the horizontal and vertical wraparound channels are used and not used.

Figures 8 and 9 show the link selection function and channel selection function of the proposed method on an $N \times N$ torus. As in the case of DOR in Figure 2, the link selection function outputs X+, X−, Y+, Y−, or ``OUT'' (an output link to a node). The proposed method needs the ``current channel'' as an input in addition to the inputs of DOR.

The channel selection policy varies depending on whether $(Y+, c)$ is used or not. If it is used, adaptive routing is carried out only when the wraparound channels are not to be used from that point on. If $(Y+, c)$ is not used, the restricted NF algorithm is carried out from the source PE until the first wraparound channel (or destination PE) is reached.

The algorithm of Figure 8 first determines, in procedure ①, whether the wraparound links of X and Y are used. In this case, the determination is based on the X and Y coordinates of the source and destination PEs as follows:

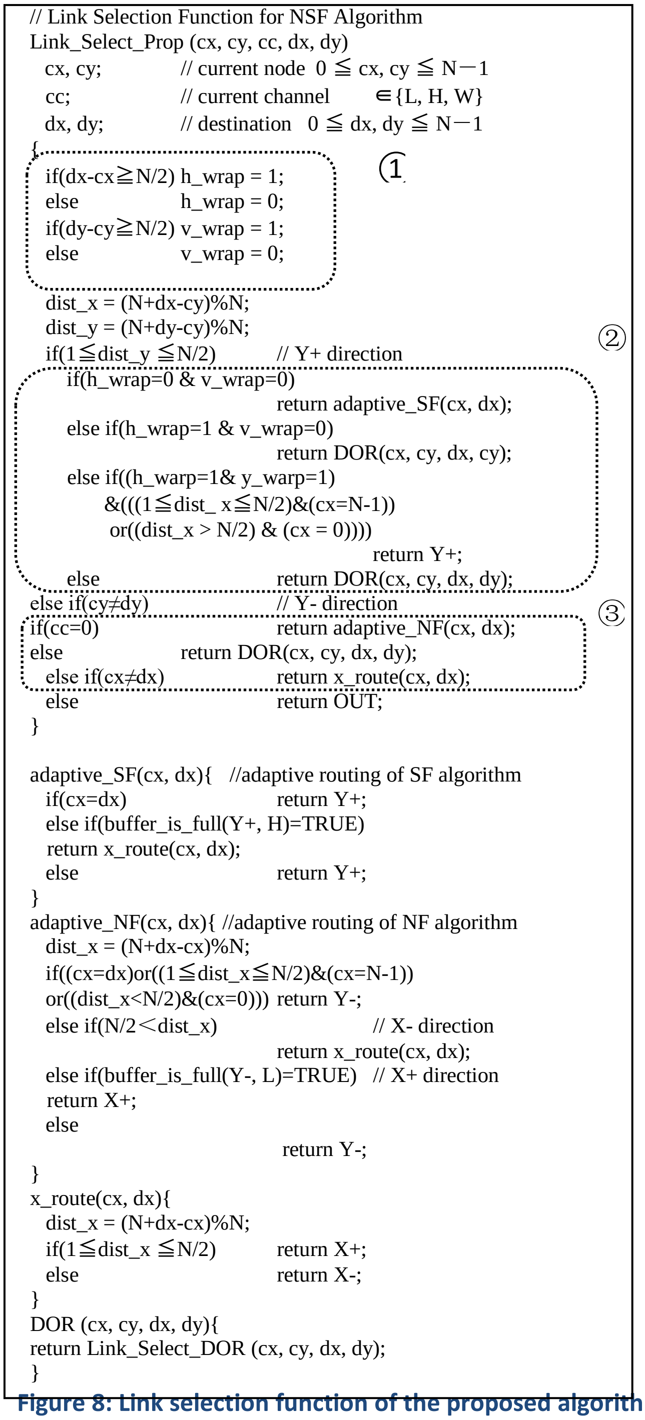
- When the difference between the X coordinates of the current PE and destination PE is less than *N*/2, h_wrap is set to 0 because the wraparound channel of the x direction is not straddled. If not, h_wrap is set to 1.
- When the difference between the Y coordinates of the current PE and destination PE is less than *N*/2, v_wrap is set to 0. If not, v_wrap is set as 1.

Next, the link is chosen on the basis of whether the Y+ channel (channel (Y+,*c*)) is used or not, as follows:

- When $(Y+, c)$ is used, the procedure ② is carried out. In this case, since the restricted NF in channel-L is equivalent to DOR, only the adaptive routing of the SF method in channel-H is carried out. If neither wraparound channel is used in going from the current PE to the destination, the packet can be sent over channel-H and routing can be continued. Thus, adaptive routing can be carried out with the SF method. The only other case in which channel-H may be used is after the packet has passed through a vertical wraparound channel $(Y+, W)$ and is due to pass through a horizontal wraparound channel $(X\pm, W)$. Even in this case, it is thought that adaptative routing using the SF method is possible. However, since it is difficult to prove that is deadlock-free, only the X-directional routing is carried out from $(Y+, W)$ to $(X\pm, W)$ and SF is applied after the packet has passed through $(X\pm, W)$. So the load concentrates in part of the network (near PE(0, 0)).

- When $(Y+, c)$ is not used, the procedure ③ is carried out. Since the SF method in channel-H is equivalent to DOR, only the adaptive routing of the restricted NF method in channel-L is carried out. In this case, the following restriction is added in order to make the order of passage in a wraparound channel into $(Y-, W) \rightarrow (X\pm, W)$.
  - ➢ Restricted NF is carried out only when$(Y-, W)$ is not be passed from the current PE to the destination or the next channel is not $(X\pm, W)$. DOR is carried out otherwise.

Besides the three inputs of the channel selection function of DOR in Figure 3, the channel selection function needs four inputs that indicate the *x* and *y* coordinates of the source and destination PEs. These new inputs can be used to judge the possibility of the packet passing through a wraparound channel. Based on the judgment, channel-H is chosen only when the wraparound channel is not to be used and $(Y+, c)$ is to be used. DOR is carried out otherwise. As in the case of DOR, the output has two states, L and H. However, an output unconditionally serves as W when the selected link is a wraparound link.



```
// Link Selection Function for NSF Algorithm
Link_Select_Prop (cx, cy, cc, dx, dy)
  cx, cy;          // current node  0 ≦ cx, cy ≦ N－1
  cc;              // current channel     ∈{L, H, W}
  dx, dy;          // destination   0 ≦ dx, dy ≦ N－1
{
  if(dx-cx≧N/2) h_wrap = 1;                    ①
  else          h_wrap = 0;
  if(dy-cy≧N/2) v_wrap = 1;
  else          v_wrap = 0;

  dist_x = (N+dx-cy)%N;
  dist_y = (N+dy-cy)%N;                                          ②
  if(1≦dist_y ≦N/2)           // Y+ direction
     if(h_wrap=0 & v_wrap=0)
                              return adaptive_SF(cx, dx);
     else if(h_wrap=1 & v_wrap=0)
                              return DOR(cx, cy, dx, cy);
     else if((h_warp=1& y_warp=1)
          &(((1≦dist_ x≦N/2)&(cx=N-1))
           or((dist_x > N/2) & (cx = 0))))
                                          return Y+;
     else                     return DOR(cx, cy, dx, dy);
else if(cy≠dy)                // Y- direction                    ③
if(cc=0)                      return adaptive_NF(cx, dx);
else              return DOR(cx, cy, dx, dy);
  else if(cx≠dx)              return x_route(cx, dx);
  else                        return OUT;
}

adaptive_SF(cx, dx){   //adaptive routing of SF algorithm
  if(cx=dx)                   return Y+;
  else if(buffer_is_full(Y+, H)=TRUE)
  return x_route(cx, dx);
  else                        return Y+;
}
adaptive_NF(cx, dx){ //adaptive routing of NF algorithm
  dist_x = (N+dx-cx)%N;
  if((cx=dx)or((1≦dist_x≦N/2)&(cx=N-1))
  or((dist_x<N/2)&(cx=0)))  return Y-;
  else if(N/2＜dist_x)                    // X- direction
                              return x_route(cx, dx);
  else if(buffer_is_full(Y-, L)=TRUE)   // X+ direction
  return X+;
  else
                               return Y-;
}
x_route(cx, dx){
  dist_x = (N+dx-cx)%N;
  if(1≦dist_x ≦N/2)          return X+;
  else                        return X-;
}
DOR (cx, cy, dx, dy){
return Link_Select_DOR (cx, cy, dx, dy);
}
```

Figure 8: Link selection function of the proposed algorithm.

```
// Channel Selection Function for NSF Algorithm
Channel_Select (cx, cy, dx, dy , cd, cc, nd)
   cx, cy;  // current node 0 ≦ cx, cy ≦ N－1
    dx, dy;              // destination  0 ≦ dx, dy ≦ N－1
    cd;                  // current direction ∈{Y+, Y-, X+, X-}
    cc;                  // current channel  ∈{L, H, W}
    nd;                  // next direction  ∈{Y+, Y-, X+, X-}
{
  if(dx-cx≧N/2)                    h_wrap = 1;
  else                             h_wrap = 0;
  if(dy-cy≧N/2)                    v_wrap = 1;
  else                             v_wrap = 0;

    dist_y = (N+dy-cy)%N;
  if((1≦dist_y ≦N/2)                          // Y+ direction
           & (h_wrap=0 & v_wrap=0))           return H;
  else                                        // Others
     return DOR_Channel (cd, cc, nd);
}

DOR_Channel (cd, cc, nd){
  return Channel_Select_DOR (cd, cc, nd);
}
```

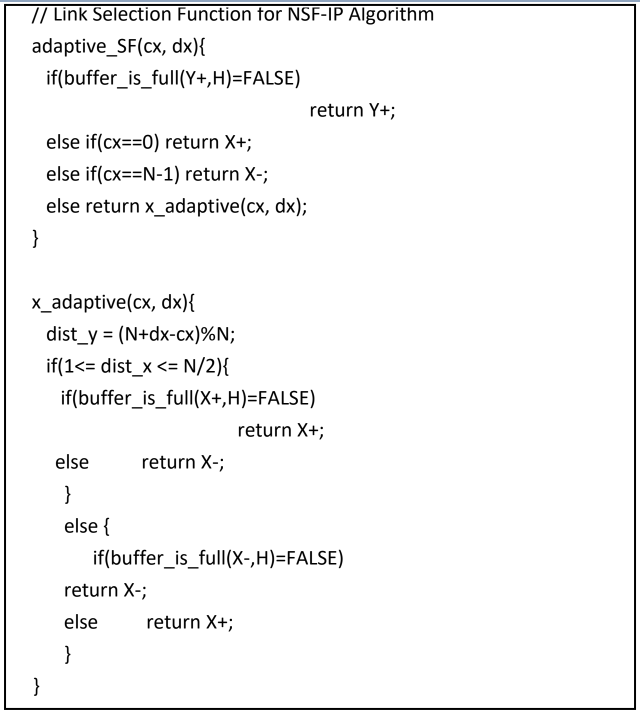
Figure 9: Channel selection function of the NSF algorithm.

4.2 NSF-IP Routing

The NSF-IP method proposed in this paper has improved fault tolerance. In this method, the SF method executed with channel-H is changed to a method that does not necessarily pass the shortest path. The routing algorithms of the SF method part before and after the change are shown in Figures 10 and 11, respectively.

```
// Link Selection Function for NSF Algorithm

adaptive_SF(cx, dx){   //adaptive routing of SF algorithm

     if(cx=dx)                          return Y+;

     else if(buffer_is_full(Y+, H)=TRUE)

     return x_route(cx, dx);

     else                               return Y+;

}
```

Figure 10: SF routing of the link selection function of the NSF algorithm.



```
// Link Selection Function for NSF-IP Algorithm
adaptive_SF(cx, dx){
  if(buffer_is_full(Y+,H)=FALSE)
                                    return Y+;
  else if(cx==0) return X+;
  else if(cx==N-1) return X-;
  else return x_adaptive(cx, dx);
}

x_adaptive(cx, dx){
  dist_y = (N+dx-cx)%N;
  if(1<= dist_x <= N/2){
    if(buffer_is_full(X+,H)=FALSE)
                              return X+;
   else        return X-;
    }
    else {
        if(buffer_is_full(X-,H)=FALSE)
    return X-;
    else        return X+;
    }
}
```

Figure 11: SF routing of the link selection function of the NSF-IP algorithm.

The main difference between before and after improvement is that adaptive routing that selects both the - direction and the + direction is performed in the horizontal direction (X direction). This enables routing avoiding congestion and failure regardless of whether it is the shortest route or not.

4.3 Deadlock Avoidance

A channel dependency graph is drawn in order to prove that the routing algorithm described in the previous section does not cause a deadlock [11, 24, 25]. The channel dependency graph is a directed graph in which nodes (channels) with dependencies are connected by an arrow. Specifically, nodes (channels) with dependencies are pairs of nodes (channels) in which a packet may be directly transmitted and received while routing.

First, the channel dependency graph is drawn. Then, each channel is numbered. If it is proved that the channel numbers are in ascending order (or descending order) in the direction of the arrows of the channel dependency graph, deadlock does not happen. In such a case, the channels are said to have an ordered relation and the corresponding channel will not cause a cyclic dependency.

A routing algorithm based on the turn model generally assigns numbers to the output channels from the PE on the basis of the PE address. As mentioned above, a 2D torus network has two virtual channels. Accordingly, the following 4-dimensional channel numbers CN are given to the 4 links × 2 channels (=8 channels) in each PE of an $N \times N$ torus.

Here, x $(0 \leq x \leq N-1)$ and y $(0 \leq y \leq N-1)$ are the x and y coordinates of the PE address, $d \in \{Y+, Y-, X+, X-\}$ is the direction of the channel, and $ch \in \{\mathrm{L}, \mathrm{H}, \mathrm{W}\}$ is the type of channel. g_m, c_1, g_s and c_2 denote the main group, first coordinate, sub group, and second coordinate, respectively. These values are numbered as follows:

$$CN(x, y, d, ch) = (g_m, c_1, g_s, c_2) \quad (1)$$

- Main Group g_m

The channel direction order is based on d and ch. Table 1 lists the values of g_m.

Table 1: Values of g_m determined by g_m and ch.

d	ch	g_m
Y+	L,W	0
Y-, X-	L,W	1
Y-	H	
X+	L,W	2
X+, X-, Y+	H	3

- First Coordinate c_1

The value c_1 is based on g_m (see Table 2).

Table 2: Value of c_1 determined by g_m.

g_m	c_1
0	y
1	$N - x$
2	0
3	y

- Sub Group g_s

The sub group g_s value determines the order of channels in the same g_m. g_s is set to 0 at g_m=0 and g_m=2. Table 3 lists the values of g_s at g_m =1 and g_m=3.

Table3: Value of g_s

g_m	d	ch	g_s
1	Y-	L,W	0
	Y-	H	1
	X-	L,W	2
3	X+, X-	H	0
	Y+	H	1

- Second Coordinate c_2

The second coordinate c_2 determines the order of the same d and ch. The c_2 values of , $N - y$, x , and $N - x$ correspond to d values of $Y+, Y-, X+$, and $X-$.

Figure 12 illustrates the channel numbers for each channel. Here, deadlocks can be avoided because channel numbers will be in ascending order through a routing path [20][21].

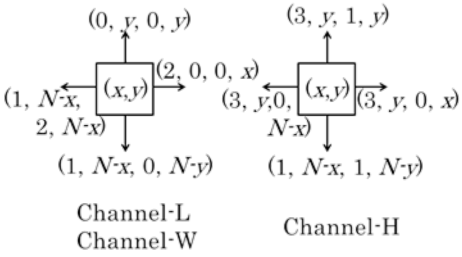


Figure 12: The channel number.

5 NSF- Fault-Tolerant (NSF-FT) Algorithm

The NSF-FT method improves the fault tolerance by moving the head of packet to channel-H when the following condition is satisfied. Examples of avoidance of faulty PE on NSF-FT are shown in Figure 13. Figure 13 shows a situation in which a packet is transferred from the source PE to the destination PE. In the conventional method, channel-L is firstly used and the transfer by path ① is carried out. When there is a faulty point as shown in the Figure 13, since the route of the packet is blocked by the failure point at point α, it is impossible to reach the destination PE. Therefore, when the next hop PE is in a faulty state, the head of the packet is compulsorily moved to channel-H. This enables adaptive routing based on the SF method. As a result, the path of the packet changes as shown in the figure, and the path ② is chosen by using channel-H and packet can avoid the faulty PEs.

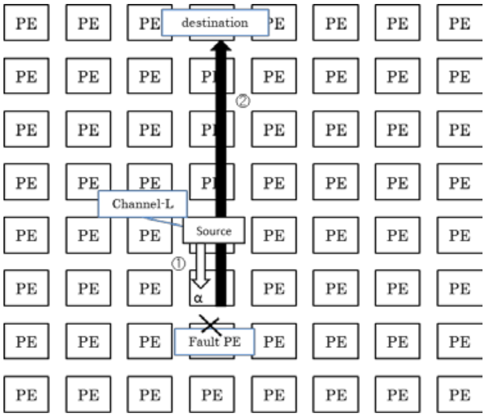


Figure 13: Examples of assumed failure patterns.

Figure 14 shows the link selection function of the NSF-FT algorithm. "Out_NSF_IP" is the selection result of the link selection function according NSF_IP. "Out_NSF_IP_H" is the selection result of the link selection function by SF algorithm in channel-H. In addition, the status (Out_NSF_IP) returns the state of the link selection destination PE by NSF_IP. The value of “Fault” is returned in the case where the PE is faulty. In the algorithm of the figure, firstly the PE of the link selection destination by NSF_IP is confirmed. In the case where the PE is not faulty, the selection result of the link selection function by NSF_IP is selected. If the PE is faulty, routing by the SF algorithm is carried out.

```
// The link Selection Function of the NSF-FT Algorithm.
Link_Select_NSF_FT(cx, cy, cc, dx, dy)
{
  Out_NSF_IP = Link_Select_NSF_IP (cx, cy, cc, dx, dy);
  if(dy>cy)  Out_NSF_IP_H = adaptive_SF(cx, dx)
     else       Out_NSF_IP_H = DOR(cx, cy, cc, dx, dy);

  if(status(Out_NSF_IP) == fault)
    return Out_NSF_IP_H;
  else
    return Out_NSF_IP;
}
```

Figure 14: Link selection function of the NSF-FT algorithm.

Figure 15 shows a channel selection function of NSF-FT. "Ch_NSF_IP" in the figure shows the selection result of the channel selection function by NSF_IP. In the channel selection function shown in the figure, firstly the PE of the link selection destination by NSF_IP is confirmed. In the case where the PE is not faulty, the selection result of the channel selection function by NSF_IP is selected.

If the PE is faulty, channel-H is selected.

```
// The Channel Selection Function of the NSF-FT Algorithm.
Channel_Select_NSF_FT(cx, cy, cc, dx, dy)
{
  Out_NSF_IP = Link_Select_NSF_IP(cx, cy, cc, dx, dy);
  Ch_NSF_IP = Channel_Select(cx, cy, dx, dy, cd, cc, nd);

  if(status(Out_NSF_IP)==fault)
    return H;
  else
    return Ch_NSF_IP;
}
```

Figure 15: Channel selection function of the NSF-FT algorithm.

5.1 Deadlock Avoidance of NSF-FT

NSF-FT is the algorithm that the packet moves from channel-L or channel-W to channel-H. From Figure 12, the case that the channel number has descending order by moving from channel-L to channel-H is only the case that the transfer is from channel $(X+, L)$ or $(X+, W)$ to $(Y-, H)$. In such cases, the channel numbers become from $(2, 0, 0,\ x)$ to $(1,\ N-x,\ 1,\ N-y)$. Therefore, it has to be proven that such case does not occur.

The cases that the transfer from channel $(X+, L)$ or $(X+, W)$ to $(Y-, H)$ are as following three cases:

① The path of $(X+, L) \rightarrow (Y-, L) \rightarrow (Y-, H)$.
② The path from $(X+, L)$ to $(Y-, H)$ via $(X+, W)$ and $(X+, H)$.
③ At $(X+, L)$ or $(X+, W)$, the packet moves to channel-H because of the blocking of faulty PE. In this case, $(Y-, H)$ is used in channel-H.

From the turn model of Figure 7, the turn from right to lower is not permitted in the rNF method in channel-L. Therefore, the transmission from $(X+, L)$ to $(Y-, L)$ such as ① does not occur. Also, the turn from right to lower is not permitted in the SF method in channel-H. Therefore, the transmission from $(X+, H)$ to $(Y-, H)$ such as ② does not occur.

From the above reason, the turn from right to lower is not permitted in every channel of channel-L, channel-W, and channel-H. Therefore, in the case that the destination PE is at lower-right from the current PE, the Y coordinate of the network is aligned first and then $(X+, c)$ are used. In such cases, $(Y-, c)$ cannot be used after $(X+, L)$ or $(X+, W)$, so the transmission such as ③ does not occur. Thus, the transmission from $(X+, L)$ or $(X+, W)$ to $(Y\pm, c)$ such as ③ does not occur.

Moreover, channel numbers seem not to be in ascending order because the value of Sub Group g_s becomes small in the case from $(X-, L)$ or $(X-, W)$ to $(Y-, H)$ (channel number becomes from $(1,\ N - x, 2, N - x\ )$ to $(1, N - x, 1, N - y)$). However, the value of first Coordinate c_1 increases when $(X-, L)$ is transmitted. Therefore, the order relation is maintained.

6 Evaluation of Congestion Resistance

A communication performance was evaluated by a software simulator that transmits 50000 cycles of packets for a 16 × 16 torus network with 256 PEs, and the evaluation results were shown. In the software simulator used in this evaluation, functional modules of routers and processor cores such as crossbar switch, FIFO, multiplexer, demultiplexer, and control unit are faithfully implemented, and is written by C language [26, 27].

Evaluation by Uniform communication pattern was carried out using this simulator. The Uniform communication pattern is a simulation performed randomly for both the start point and the end point when sending a packet.

The dynamic communication performance of an interconnection network is characterized by message latency and network throughput. Message latency refers to the time elapsed from the instant when the first flit is injected into the network from the source to the instant when the last flit of the message is received at the destination. Average transfer time is the average value of the message latency for all packets. Network throughput refers to the maximum amount of information delivered per unit of time through the network. It is the average value of the number of flits that a PE receives in each clock cycle. In the evaluation of dynamic communication performance, flocks of messages are sent in the network to compete for the output channels. Packets are transmitted by the request-probability *r* during *T* clock cycles, and the number of flits that reached at destination PE and their transfer time are recorded. Then the average transfer time and throughput are calculated and plotted with average transfer time on the horizontal axis and throughput on the vertical axis. The process of performance evaluation is carried out with changing the request-probability *r*.

Figure 16 shows the average transfer time for network throughput obtained by simulation. The horizontal axis of the graph is throughput and the vertical axis is average transfer time. As shown in Figure 16, it was revealed that the NSF, NSF-IP, and NSF-FT methods have higher throughput than the DOR. In addition, when the NSF-IP, NSF-FT, and NSF methods were compared, the throughput was almost the same, and no significant performance degradation due to proposed method was observed. In a communication pattern in which the interconnection network is crowded as a whole like random communication, the effect of avoiding congestion is limited by adaptive routing in a method that does not use additional virtual channels.

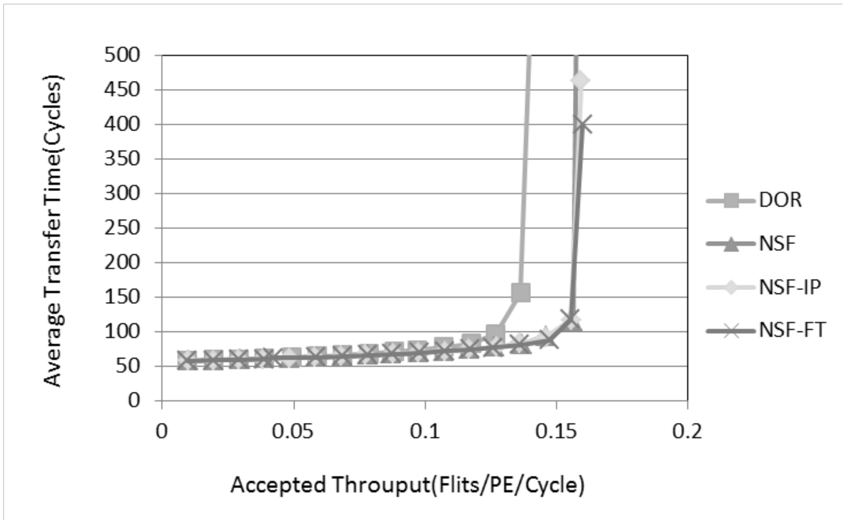


Figure 16: Result for uniform traffic.

7 Evaluation of Fault Tolerance

In the same way as in the previous section, communication performance was evaluated by a software simulator that transmits packets for a 16 × 16 torus network with 256 PEs. On the communication pattern of the evaluation of this section, one “session” is defined as “All PEs in the network send one packet to another PE. In this case, packets blocked by faulty PEs remain in the network. In this evaluation, the communication of 1, 3, and 5 sessions were carried out in the network with some faulty patterns, and the number of non-arrival packets (packets that do not arrive at the destination because they were blocked by a faulty PE) with DOR，NSF，NSF-IP，and NSF-FT were tracked. We tracked the results of 10 times simulation and plotted the average values of the number of non-arrival packets in Tables 4, 5, and 6. How much the proposed algorithm reduces the number of non-arrival packets can be determined by tracking the non-arrival packets. “Loop 1”, “Loop 3”, and “Loop 5” in the following subsections are the communication patterns of 1, 3, and 5 sessions.

7.1 Four PEs Near the Center Are Faulty

Table 4 shows the results of evaluation of “four PEs near the center are faulty PEs”. Since four PEs are set as faulty PEs, $256 - 4 = 252$ packets are sent between PEs per loop. In the “loop 1”, the numbers of non-arrival packets for NSF, NSF-IP, and NSF-FT were lower than that for DOR and the numbers of non-arrival packets for NSF-IP and NSF-FT were a little lower than that for NSF. In the “loop 3” and “loop 5”, the number of non-arrival packets for NSF-FT was slightly lower than that for NSF-IP.

The “loop 3” results are more than ten times larger than the “loop 1” results. This is the result of another packet being blocked in the path that was blocked by the faulty node. As a result, it is thought that such a result was obtained because a large amount of unarrived packets remains in the Interconnection network. The improvement of NSF-IP compared with DOR and NSF is due to removal of constraints of the shortest path, which increased the number of selectable paths. And the reason NSF-FT is slightly better than the other methods is that it avoids faulty points by the method described in Figure 13.

Table 4: Number of non-arrival packets on 4 central points of failure.

Algorithm	Loop 1	Loop 3	Loop 5
NSF-FT	14.8	179.0	639.8
NSF-IP	14.7	188.3	648.8
NSF	16.8	204.9	681.2
DOR	21.1	251.5	742.4

7.2 Four Corner PEs Are Faulty

Table 5 shows the results of evaluation of average non-arrival packets where four PEs at four corners including wraparound link are faulty PEs. In all the loops 1, 3, and 5, the number of non-arrival packets of the proposed method is decreasing, and the effect by implementing the proposed method is shown.

Table 5: Number of non-arrival packets on 4 corner failure.

Algorithm	Loop 1	Loop 3	Loop5
NSF-FT	13.1	182.8	652.5
NSF-IP	15.9	213.4	698.2
NSF	15.7	210.4	693.4
DOR	19.2	248.6	740.9

7.3 PEs Are Randomly Broken

It was assumed that 1, 2, 4, 8, and 16 PEs are randomly broken in a 16 × 16 two-dimensional torus network simulator having 256 PEs, and the number of non-arrival packets were measured using the DOR, NSF, NSF-IP, and NSF-FT algorithms. The average numbers of non-arrival packets in loops 1, 3, and 5 are shown in Figures 17, 18, and 19. The vertical axis of each graph is the average number of non-arrival packets. Table 6 shows the ratio of average number of non-arrival packets in NSF-FT compared to other methods. The value of "DOR → FT" in Table 6 is (number of non-arrival packets for NSF-FT) / (number of non-arrival packets for DOR). Likewise, "IP → FT" and "NSF → FT" are (non-arrival packet number for NSF-FT) / (number of non-arrival packets for NSF-IP) and (number of non-arrival packets for NSF-FT) / Packet number). Shaded areas in the tables are portions where the improvement by NSF-FT are not seen.

Although the dispersion of non-arrival packets in NSF-FT and NSF-IP of 8 faulty PEs in Figure 17 and NSF-FT and NSF-IP of 16 faulty PEs in Figure 18 were large, the number of arrival packets of NSF-FT was improved in all methods in others. The reason the dispersions were large at two points in Figures 17 and 18 is that NSF-FT caused more packets to flow into the channel-H than other methods did. It causes the congestion of channel-H, and then it causes the increase of the number of cycles and the increase of the number of non-arrival packets. From the above, one can infer that NSF-FT improves fault tolerance in the case that the number of faulty PE is not large or in a situation of congestion or moderate congestion.

The average transfer times required for loops 1, 3, and 5 are shown in Figures. 20, 21, and 22. The number of cycles required for transfer is plotted as the vertical axis. Figures 20, 21, and 22 show that the average transfer times were slightly larger in DOR and NSF-FT.

The large average transfer time of DOR is thought to be due to the long waiting time of packets in the network because there are few choices regarding the course of the packet. The reason the average transfer time of NSF-FT becomes longer is the longer transfer path itself. As described in Figure 13 in section 5, NSF-FT tends to lengthen the transfer path because it is a method of changing the rule of link selection by the faulty PE. This can be considered a cause of the above-mentioned dispersion.

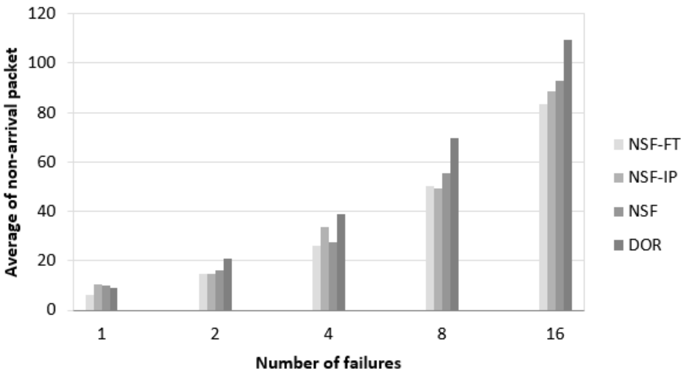


Figure 6: Average of non-arrival packet in loop 1.

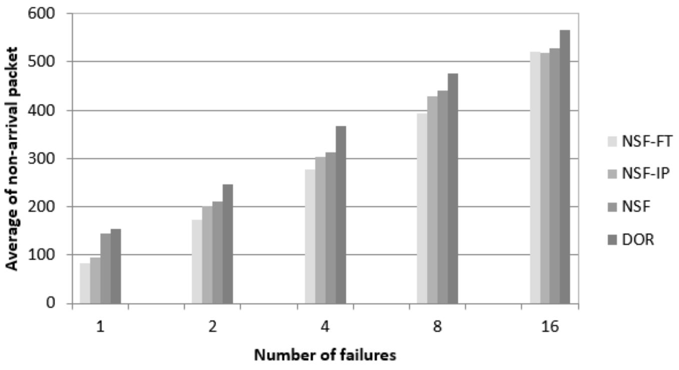


Figure 7: Average of non-arrival packet in loop 3.

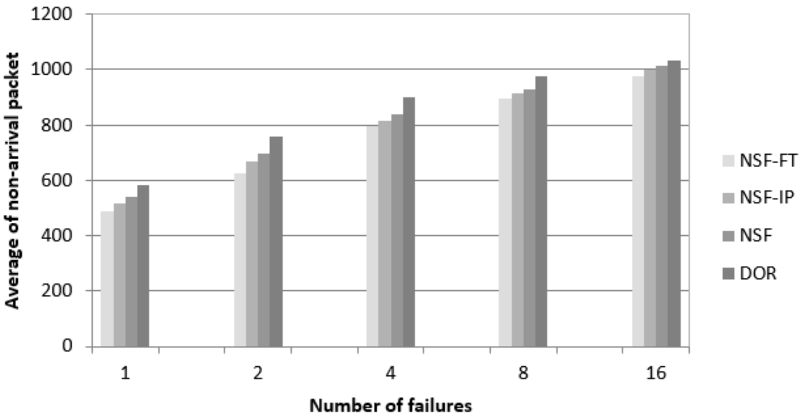
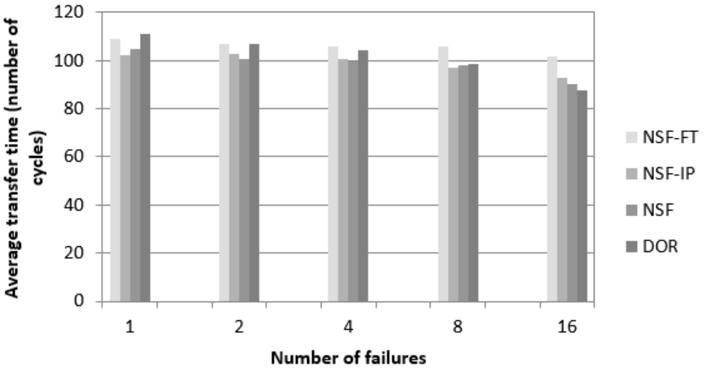


Figure 8: Average of non-arrival packet in loop 5.



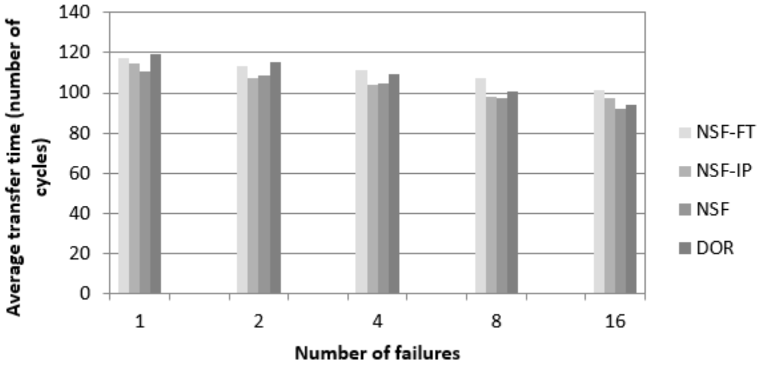


Figure 9: Average transfer time on loop 3.

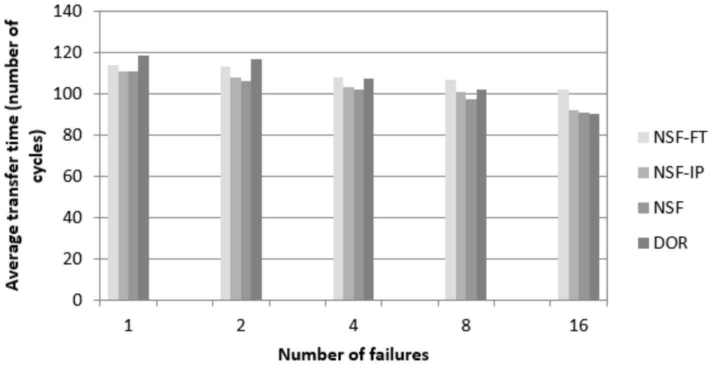


Figure 10: Average transfer time on loop 5.

Table 6: The ratio of NSF-FT average non-arrival packets compared to other methods.

	Number of faulty PEs	Loop 1	Loop 3	Loop 5
DOR→FT	1	69.2%	54.0%	83.2%
	2	70.5%	70.1%	82.2%
	4	67.0%	75.1%	88.6%
	8	72.1%	82.3%	92.2%
	16	76.1%	92.0%	94.3%
IP→FT	1	60.0%	87.2%	94.0%
	2	99.3%	85.9%	93.2%
	4	77.6%	90.8%	97.7%
	8	101.8%	91.7%	98.0%
	16	93.8%	100.1%	97.3%
NSF→FT	1	64.3%	57.4%	89.9%
	2	91.4%	82.5%	89.8%
	4	94.5%	88.4%	94.9%
	8	91.1%	88.9%	96.5%
	16	89.9%	98.8%	96.3%

8 Conclusion

In this paper we proposed the NSF-FT method, which is a routing algorithm that improves the fault-tolerance of NSF method based on the turn model and is applicable to a 2D torus. And its congestion resistance and fault tolerance were evaluated by software simulation.

As for the congestion resistance, the throughput obtained with NSF-FT is better than that obtained with DOR. And even compared with NSF and NSF-IP, there was no throughput reduction due to NSF-FT implementation.

Regarding the evaluation of fault tolerance assuming a random faulty PEs, it is clear that the packet arrival rate in congested situations has been increased by the NSF-FT method. As the result, the fault tolerance was improved.

We are planning to extend the proposed method to the *n* -dimensional torus because it can be applied to any two dimensions in an *n*-dimensional torus network. Specifically, the following algorithm can be considered for the *n*-dimensional torus, whereas the routing order of 0 dimension → 1 dimension→ 2 dimension → ... (N−1)-dimension is carried out in DOR.

- At first, the NSF method and so on are applied for dimensions 0 and 1.
- Next, the NSF method or the like is similarly applied for dimensions 2 and 3.
- And then, the NSF method and so on are applied until dimension N−1.

Furthermore, future work will further improve fault tolerance by combining the proposed method with one in which the rectangular areas of faulty PEs are handled as faulty area [28].

REFERENCES

[1]. J. Y. Ngai and C. L. Seitz, A framework for adaptive routing in multicomputer networks, ACM SIGARCH Computer Architecture News, vol. 19, no. 1, pp. 6–14, 1991.

[2]. T. Schonwald, J. Zimmermann, O. Bringmann, and W. Rosenstiel, Fully Adaptive Fault-Tolerant Routing Algorithm for Network-on-Chip Architectures, Digital System Design Architectures, Methods and Tools, pp. 527–534, 2007.

[3]. M. M. Hafizur Rahman, Yukinori Sato, Yasuyuki Miura, and Yasushi Inoguchi, Dynamic Communication Performance of a Hierarchical 3D-Torus Network, IASTED, In 10th International Conference Parallel and Distributed Computing and Networks (PDCN 2011), 2011.

[4]. Y. Miura and S. Horiguchi, An Adaptive Routing for Hierarchical Interconnection Network TESH, Proc. of the Third International Conference on Parallel And Distributed Computing, Applications and Technologies, pp. 335–342, 2002.

[5]. Y. Miura, Masahiro Kaneko, M. M. Hafizur Rahman, and Shigeyoshi Watanabe, Adaptive Routing Algorithms and Implementation for TESH Network, Communications and Network (CN), vol. 5, no. 1, pp. 34–49, 2013.

[6]. W. J. Dally, Performance Analysis of k-ary n-cube Interconnection Networks, IEEE Trans. on Computers, vol. 39, no. 6, pp. 775–785, 1990.

[7]. W.J. Dally and Hiromichi Aoki, Deadlock-Free Adaptive Routing in Multicomputer Networks Using Virtual Channels, IEEE Trans. on Parallel and Distributed Systems, vol. 4, pp. 466–475, 1993.

[8]. M. P. Merlin and J. P. Schweitzer, Deadlock Avoidance in Store-and-Forward Networks-1: Store and Forward Deadlock, IEEE Trans. On Communications, vol. COM-28, no. 3, pp. 345–354, 1980.

[9]. W. J. Dally and C. L. Seitz. Deadlock-Free Message Routing in Multiprocessor Interconnection Networks. IEEE Trans. on Computers, vol. C-36, no. 5, pp. 547–553, 1987.

[10]. C. S. Yang, Y. M. Tsai, and Y. L. Tsai, Adaptive Routing in k-ary n-cube Multicomputers, Proc. of 1996 International Conference on Parallel and Distributed Systems(ICPADS'96), pp. 404–411, 1996.

[11]. J. Duato, A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks, IEEE Trans. on Parallel and Distributed Systems, vol. 4, no. 12, pp. 1320–1331, 1993.

[12]. D. H. Linder and J. C. Harden, An adaptive and fault tolerant wormhole routing strategy for k-ary n-cubes, IEEE Trans. on Computers, vol. C-40, no.1, pp. 2–12, 1991.

[13]. R. S. Ramanujam and Bill Lin, Destination-based adaptive routing on 2D mesh networks, 2010 ACM/IEEE Symposium onArchitectures for Networking and Communications Systems (ANCS), pp. 1–12, 2010.

[14]. C. J. Glass and L. M. Ni, Maximally Fully Adaptive Routing in 2D Meshes, Proc. of The 19th International Symposium on Computer Architecture, pp. 278–287, 1992.

[15]. C. J. Glass and L. M. Ni, The Turn Model for Adaptive Routing, Proc. of The 25th Annual International Symposium on Computer Architecture, pp. 441–450, 1998.

[16]. Jie Wu, A Fault-tolerant and Deadlock-free Routing Protocol in 2D Meshes Based on Odd-even Turn Model, IEEE Trans. on Computers, vol. 52, no. 9, pp. 1154–1169, 2003.

[17]. A. Jouraku, M. Koibuchi, and H. Amano, An Effective Design of Deadlock-Free Routing Algorithms Based on 2D Turn Model for Irregular Networks, IEEE Trans. on Parallel and Distributed Systems, vol. 18, no. 3, pp. 320–333, 2007.

[18]. W. J. Dally, Virtual-Channel Flow Control, IEEE Trans. on Parallel and Destributed Systems, vol. 3, no. 2, pp. 194–205, 1992.

[19]. K. Matoyama, Y. Miura, and S. Watanabe, Adaptive routing of the 2-D torus network, Proc. of FIT2009, RC-005, 2009 (in Japanese).

[20]. Y. Miura, K. Shimozono, K. Matoyama, and S. Watanabe, An Adaptive Routing of the 2-D Torus Network Based on Turn Model, Proc. of 4th International Workshop on Advances in Networking and Computing, pp. 587-591,December 2013.

[21]. Yasuyuki Miura, Kentaro Shimozono, Kazuya Matoyama, and Shigeyoshi Watanabe, The Static and Dynamic Performance of an Adaptive Routing Algorithm of 2-D Torus Network Based on Turn Model, Proc. of the 2014 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA’ 14), pp. 114–120, July 2014.

[22]. Yasuyuki Miura, Kentaro Shimozono, Naohisa Fukase, Shigeyoshi Watanabe, and Kazuya Matoyama, An Adaptive Routing Algorithm of 2-D Torus Network Based on Turn Model: The Communication Performance, International Journal of Networking and Computing (IJNC), pp. 223–238,January 2015.

[23]. Tsukasa-Pierre Nakao and Yasuyuki Miura, The Study on Adaptive Routing Algorithm of 2-D Torus Network with Fault Tolerance, IEICE Technical Report (FIIS64),October 2017.

[24]. J. Duato A Necessary and Sufficient Condition for Deadlock-Free Adaptive Routing Wormhole Networks, Proc. of the International Conference on Parallel Processing, vol. 1, pp. 142-149, 1994.

[25]. E. Fleury and P. Fraigniaud, A General Theory for Deadlock Avoidance in Wormhole-Routing Networks, IEEE Trans. on Parallel and Distributed Systems, vol. 9, no. 7, pp. 626–638, 1998.

[26]. Naohisa Fukase, Yasuyuki Miura, Shigeyoshi Watanabe, and M. M. Hafizur Rahman, The Performance Evaluation of a 3D Torus Network Using Partial Link-Sharing Method in NoC Router Buffer, IEICE Trans. on Information and Systems, vol. E100–D, no. 10, October 2017.

[27]. Naohisa Fukase, Yasuyuki Miura, and Shigeyoshi Watanabe, Link-Sharing Method of Buffer in Router Circuit of Direct-Connection Network, IEEJ Trans EIS, vol. 132, no. 10, pp. 1675–1688, 2012.

[28]. N. Aosaka, Y. Fukushima, M. Fukushi, and M. Kameyama, Fault-Tolerant Cogestion-Avoidance Routing for 2D-Mesh Network-on-Chip,Technical report of IEICE. FIIS10, no. 271, March 2010.

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TNC TRANSACTIONS ON NETWORKS AND COMMUNICATIONS

VOLUME 7, NO. 1
ISSN: 2054 -7420

Slope-based Empirical Path Loss Prediction Models for Rural Networks at 2.4 GHz

[1]Jean Louis Ebongue Kedieng Fendji, [2]Nelson Maguelva Mafai, [3]Jean Michel Nlong
[1]Department of Computer Engineering, UIT, University of Ngaoundéré, Cameroon;
[2]Ministry of Scientific Research and Innovation, Yaoundé, Cameroon;
[3]Inter States University, Sangmélima, Cameroon;
jlfendji@univ-ndere.cm ; nelsonmafai@gmail.com; jmlong@yahoo.fr

ABSTRACT

Despite the plethora of works on empirical path loss prediction in wireless networks, just a little is addressing rural environments. In this work, we consider slope-based empirical path loss models in wireless networks at 2.4 GHz using off-the-shelf 802.11n (one transmitter and two receivers at 150Mbp and 300Mbps). We define three scenarios usually observed in rural environment. Subsequently, we do a measurement campaign and compare results to selected prediction models. We later propose a new model based on Liechty model. The new model is compared to Liechty model in Non-Line of Sight (NLOS) and combined (LOS and NLOS) scenarios. The Liechty model provided a better prediction in NLOS scenario while the new model outperforms in combined scenario. In addition, we observe that the data rate also influences the prediction. Especially in free space scenarios, the receiver with the greater data rate provides a smaller mean error and standard deviation.

Keywords: Attenuation, measurement, 802.11n, network planning, rural area.

1 Introduction

Wireless networks are incontestably an appealing solution to bridge the digital divide between rural and urban regions [1]. This easy-to-deploy technology, especially in hard-to-wire regions or emergency situation [2], can provide bad results and be useless if the network is not well planned [3]. The difficulty when planning a wireless network is to predict the quality of links by estimating the path loss of the signal.

A frequently used tool to predict the quality of signal is the empirical path loss model. However, such a model is usually tied to a specific environment because of the particular configuration. This configuration depends on the devices used as transmitter and the receiver, the distance between them, the frequency of the signal, and the height of the antenna. Despite the great number of path loss models [4], [5], [6], [7], [8], [9], just few are focusing on propagation at 2.4 GHz [10], [11].

However, with the vulgarization of off-the-shelf technologies like IEEE 802.11n, wireless networks at 2.4 GHz and 5 GHz represent currently an attractive solution to connect rural regions. There is therefore a need to predict the signal path loss at these particular frequencies in rural environment. This is a prior task before estimating the cost of deploying such an infrastructure in rural areas [12].

DOI: 10.14738/tnc.71.6162
Publication Date: 01st March 2019
URL: http://dx.doi.org/10.14738/tnc.71.6162

This paper is an extended version of [13]. In this paper, we are interested to provide a more precise empirical model for predicting signal path loss at 2.4Ghz using off-the-shelf 802.11n in outdoor environment. For this purpose, we do a measurement campaign in three different scenarios (free space, raised space and built space) using two different receivers (150Mbps and 300Mbps). Afterwards, we compare the gathered data to the attenuation predicted by selected empirical models. Liechty model [11] shows itself as the more precise empirical model by providing the lowest mean error in raised space and built space using both receivers. Finally, we improve this model to obtain a better prediction model by considering the distance to the first breakpoint.

The rest of the paper is organized as follows: section 2 introduces to empirical models and presents the selected models for this study; the methodology, material, scenario and data collection approaches are presented in section 3. Section 4 is dedicated to the numerical analysis of data and section 5 presents and evaluates a new model.

2 Prediction Models for Wireless Networks

The prediction of path loss in wireless networks have been tackled by several works. They can be grouped into two classes: theoretical (deterministic) models and empirical ones.

Theoretical models, also called deterministic approaches, are based on the physical phenomena of radio wave propagation. There are different types of theoretical methods: multi-ray models taking into account transmitted and reflected rays [14], [15], and digital or discrete models depending on Maxwell's equations [16], [17]. But in practice, the implementation of deterministic models usually requires a huge knowledge of the field, which is sometimes nearly impossible to obtain in some cases. In addition, deterministic models make use of complex algorithms which usually require a lot of computation depending on the expected accuracy of the model. For this reason, deterministic models are used generally for indoor environments or to well-defined and small size outdoor environments.

A totally different approach for predicting path loss without an exact knowledge of the environment has emerged with the development of statistics and probability. In this approach, the calculation of the signal path loss is made along a single radius shown by the line connecting the transmitter and the receiver. This second class of models is called empirical models.

2.1 Characteristics of path loss models

The path loss of a signal is the ratio between the transmitted power and the received power using the following expression (1).

$$PL = 10\log_{10}\frac{P_t}{P_r} \tag{1}$$

Where:

PL : Signal path loss (dB)

P_t : Transmitted Power (dBm)

P_r : Received Power (dBm)

By using the Friis equation [18], and considering antennas as isotropic, we obtain (2)

$$PL_{fs} = 20\log_{10}\frac{(4\pi d)}{\lambda} \qquad (2)$$

With $c = \lambda f$, f frequency in Hz and $c = 3.10^{8} m.s^{-1}$, we finally obtain (3) by changing the unit of frequency and the distance

$$PL_{fs} = \left\{ \begin{matrix} -92.4 + 20\log_{10} d + 20\log 10 f \\ -32.4 + 20\log_{10} d + 20\log 10 f \end{matrix} \right\} \begin{matrix} (3.1) \\ (3.2) \end{matrix} \qquad (3)$$

With d in km and f in GHz in (3.1) or MHz in (3.2).

Path loss models are built from the basic model (3). The calculation of signal path loss between the transmitter and the receiver considers in practice other factors such as obstacles and height of antennas.

The list of some common path loss models is given in Table 1.

Table 1. Common empirical path loss models

<table>
<tr><th>Models</th><th>Condition</th><th>Cite</th></tr>
<tr><td>|-Free space</td><td></td><td>[18]</td></tr>
<tr><td>|-Egli</td><td>$f \in]30; 3000[$</td><td>[4]</td></tr>
<tr><td>|-One Slope</td><td></td><td>[19]</td></tr>
<tr><td>|-Dual Slope</td><td></td><td>[19]</td></tr>
<tr><td>|-Log Normal Shadowing</td><td></td><td>[20]</td></tr>
<tr><td>|-Partitioned</td><td></td><td>[20]</td></tr>
<tr><td>|-Liechty</td><td>$f \approx 2400$</td><td>[11]</td></tr>
<tr><td>|-Okumura</td><td>$f \in]150; 1920[$; $d \in]1; 100[$; $h_{tx} \in]30; 200[$; $h_{rx} \in]3; 10[$</td><td>[5]</td></tr>
<tr><td>|-Okumura-Hata</td><td>$f \in]150; 1500[$; $d \in]1; 10[$; $h_{tx} \in]30; 200[$; $h_{rx} \in]1; 10[$</td><td>[6]</td></tr>
<tr><td>|-COST 231 Hata</td><td>$f \in]150; 2000[$; $d \in]1; 20[$; $h_{tx} \in]30; 200[$; $h_{rx} \in]1; 10[$</td><td>[21]</td></tr>
<tr><td>|-Hata-Davidson</td><td>$f \in]150; 1500[$; $d \in]1; 300[$; $h_{tx} \in]30; 200[$; $h_{rx} \in]1; 10[$</td><td>[24]</td></tr>
<tr><td>|-Rural</td><td>$f \in \{160; 400; 900\}$</td><td>[8]</td></tr>
<tr><td>|-ITU-R</td><td>$1.5 < f < 2$; $1 < d < 10$; $30 < h_{tx} < 200$; $1 < h_{rx} < 10$</td><td>[26]</td></tr>
<tr><td>|-ECC-33</td><td>$700 \leq f \leq 3500$; $1 < d < 10$; $20 < htx < 200$; $1 < h_{rx} < 10$</td><td>[25]</td></tr>
<tr><td>|-Green Obaidat</td><td></td><td>[22]</td></tr>
<tr><td>|-Erceg</td><td>$f \approx 2000$; $1 < h_{rx} < 2$</td><td>[7]</td></tr>
<tr><td>|-SUI</td><td>$2500 < f < 2700$; $0.1 < d < 8$; $20 < h_{tx} < 80$; $2 < h_{rx} < 10$</td><td>[23]</td></tr>
<tr><td colspan="3">f: frequency (MHz); d: distance between the transmitter and the receiver (Km)
h_{tx}: height of transmitter antenna (m); h_{rx}: height of the receiver antenna (m)</td></tr>
</table>

2.2 Selected models

The present study focuses on five prediction models generally used in micro-cellular areas: One Slope, Dual Slope, Log Normal Shadowing, Partitioned, and Liechty.

2.2.1 One Slope Model

The One Slope model is a Log Distance based prediction model. In [18], authors show how the attenuation of the signal weakens with the distance. The attenuation on an exponent which indicates how the path loss increases rapidly with distance. The expression of the path loss is given by (4).

$$PL_{1Slope} = PL_{fs}(d_0) + 10n\log_{10}\left(\frac{d}{d_0}\right) \tag{4}$$

Where:

$PL_{fs}(d_0)$: Signal path loss in free space at distance d$_0$ (m)

d_0 : Distance of reference (m)

d : Distance between the transmitter and the receiver (m)

n : Path loss exponent of the environment

The expression of $PL_{fs}(d_0)$ is given by (3). Prior measurements are required in order to calibrate the parameter n by using the least squares method. An obvious limit of this model is that it does not consider obstacles between the transmitter and the receiver.

2.2.2 Dual-slope Model

The Dual-slope model extends the One Slope model [18]. The first slope considers Line Of Sight (LOS), and the second slope considers Non-Line Of Sight (NLOS). The expression of the path loss is given by (5).

$$PL_{2Slope} = PL_{fs}(d_0) + \begin{cases} 10n_1\log_{10} d & (5.1) \\ 10n_1\log_{10} d_{bp} + 10n_2\log_{10}\dfrac{d}{d_{bp}} & (5.2) \end{cases} \tag{5}$$

With $1m < d \le d_{bp}$ in (5.1) and $d > d_{bp}$ in (5.2).

Where:

$PL_{fs}(d_0)$: Signal path loss in free space at distance d$_0$ (dB)

d_0 : Distance of reference (m)

d : Distance between the transmitter and the receiver (m)

d_{bp} : Distance between the transmitter and the first obstacle (m)

n_1 : Path loss exponent of the environment $d \le d_{bp}$

n_2 : Path loss exponent of the environment $d > d_{bp}$

The expression of $PL_{fs}(d_0)$ is still given by (3) and values of the exponents n_1 and n_2 are determined using the least squares method and field measurements.

2.2.3 Log Normal Shadowing Model

The Log Normal Shadowing Model is another improvement of One Slope model [18]. It considers that the signal path loss at a distance is in fact a random variable, due to the effect of multipath propagation and obstacles encountered by the wave. The expression of the path loss is given by (6).

$$PL_{LN-Sha} = PL_{fs}(d_0) + 10n\log_{10}\left(\frac{d}{d_0}\right) + \chi_\sigma \tag{6}$$

Where:

$PL_{fs}(d_0)$: Signal path loss in free space at distance d_0 (dB)

d_0 : Distance of reference (m)

d : Distance between the transmitter and the receiver (m)

χ_σ : Shadowing effect (dB)

n : Path loss exponent of the environment

The expression of $PL_{fs}(d_0)$ is still given by (3). χ_σ is a Gaussian random variable (in dB) with zero mean and standard deviation σ . Values of n and σ are also determined from field measurements [18].

2.2.4 Partitioned Model

This model is also an extension of the One Slope model. It has four different expressions of signal path loss depending on the distance between the transmitter and the receiver. The expression of the path loss is given by (7) [24].

$$PL_{part} = PL_{fs}(d_0) + \begin{cases} 20\log_{10} d & (7.1) \\ 20 + 30\log_{10}\dfrac{d}{10} & (7.2) \\ 29 + 60\log_{10}\dfrac{d}{20} & (7.3) \\ 47 + 120\log_{10}\dfrac{d}{40} & (7.4) \end{cases} \tag{7}$$

With $1m < d \le 10m$ in (7.1), $10m < d \le 20m$ in (7.2), $20m < d \le 40m$ in (7.3) and $d > 40m$ in (7.4).

Like in previous models, the expression of $PL_{fs}(d_0)$ is also given by (3).

2.2.5 Liechty Model

The Liechty Model has been proposed by Christopher Lorne Liechty [11]. The model extends the One slope model in order to consider the attenuation due to obstacles such as trees and buildings. The expression of the path loss is given by (8).

$$PL_{Liechty} = PL_{fs}(d_0) + 10n\log_{10}\left(\frac{d}{d_0}\right) + \sum_i num_i * a_i \quad (8)$$

Where:

$PL_{fs}(d_0)$: Signal path loss in free space at distance d₀ (m)

d_0 : Distance of reference (m)

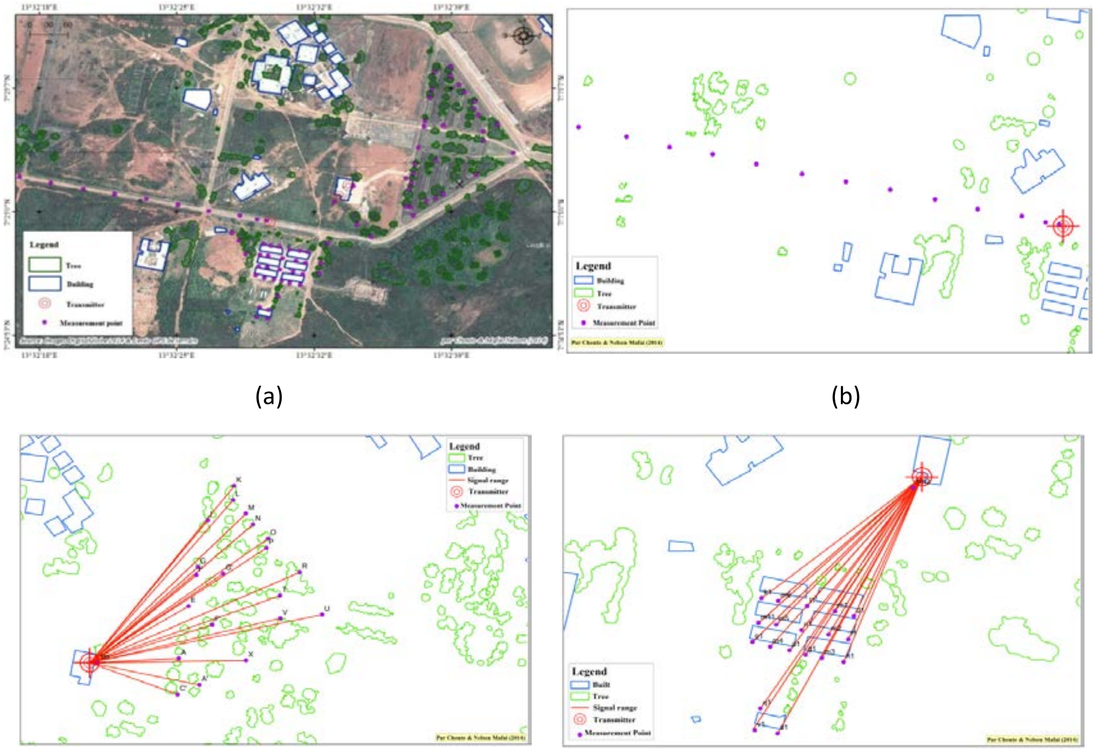
d : Distance between the transmitter and the receiver (m)

num_i : Number of obstructions of type i

a_i : Attenuation of obstructions of type i (dB)

n : Path loss exponent of the environment

The expression of $PL_{fs}(d_0)$ is given by (3), as it has been the case in previous models.



(a)

(b)

(c)

(d)

Figure 1. Measurement points on the field.

3 Methodology

3.1 Environment of the study

The selected area for measurements is the campus of the University of Ngaoundere, in Cameroon. This area is mainly characterised by randomly distributed trees and some small buildings as shown in Figure 1.a. To carry out measurements, three scenarios have been designed:

- Free space: a distance of 600m has been selected. At each interval of 50metres, the signal strength is measured. Figure 1.b shows the different points of measurement.
- Wooded area: the average height of trees is around 6,5metres. Fig.1.c shows the different points of measurement for this scenario.

Built-up area: houses have an average height of about 3.5metres. The different points of measurements are shown in Figure 1.d.

3.2 Description of the hardware and software tools

The access point used as a transmitter in all scenarios is manufactured by Alpha Network and complies with the IEEE 802.11n standard. it offers a bandwidth of 150Mbps and operates at 2.4GHz. The transmit power is 30dBm with an antenna gain of 12 dBi. Two wireless USB dongles are used as receivers. Both are compliant to the IEEE 802.11n, 802.11b and 802.11g standards. They are operating at 2.4GHz but providing different bandwidth (150Mbps and 300Mbps). A USB GPS is used to determine the different points for measurements. The characteristics of hardware is provided in Table 2.

We used the free and open source software Vistumbler in version 9.8 for wireless signal strength measurement. It enables live Google Earth tracking of access points and supports GPS integration.

Table 2. Characteristics of hardware

Type of equipment	Characteristics
Access Point	Manufacturer: Alpha Network, Model: N2, Power: 30 dBm, Antenna: 12 dBi
Wireless USB dongle 1	Manufacturer: Dodocool, Rate: 300 Mbps, Chipset: Realtek 8191, Driver: RTL8188SU/8191FEB28, Antenna: 2 dBi
Wireless USB dongle 2	Manufacturer: Dodocool, Rate: 150 Mbps, Chipset: Ralink RT5370, Driver: RALINK23FEB, Antenna: 2 dBi
USB GPS	Manufacturer: Navilock, Model: NL-464US 60122, Sensibility: -159, dBm

3.3 Data collection and analysis

We used the Single Marker Measurements [11] as positioning mode. The antennae of the USB dongles are oriented towards the sky throughout the measurement campaign. Ten measurements are performed at each selected point with the help of Vistumbler. Retrieved data are recorded in predefined Datasheets.

Data analysis is performed in five steps. First, we calculate the mean loss of signal. Second, we determine the fitting curve by using the least squares method. Third, we evaluate the path loss exponent n. Fourth, we calculate the σ parameter of Log Normal Shadowing Model. Finally, we compare the predicted by the selected models with the mean loss of the signal measured in the field.

The path loss exponent is determined using the least squares method. The attenuation of each type of obstacle required by the Liechty model is considered as the difference between the mean loss of the signal obtained before and after the obstacle.

The value of σ is determined using (9):

$$\sigma^2 = \frac{\sum_i (\overline{P_m} - \overline{P_r})^2}{k} \quad 1 \le i \le k \tag{9}$$

With:

$\overline{P_m}$: Power of the measured received signal

$\overline{P_r}$: Strength of the estimated received signal

k : Number of measurement points

4 Results interpretation

Table 3 provides empirical values of the model parameters. The mean error and the standard deviation are used as indicators to compare the predicted values to the measured ones.

Table 3. Empirical values of the model parameters

		300Mbps / 150Mbps	
Environment	**Obstruction (dB)**	**n**	**σ**
Free Space	0	1.950 / 2.186	1.12 / 1.50
Wooded area	0.37	3.407 / 3.656	2.41 / 2.61
Built-up area	0.41	4.799 / 4.996	2.10 / 2.48

4.1 Result analysis in free spaces

A total number of 100 measurements have been performed in free space. Figure 2 shows the pathloss of the signal for both USB wireless receivers (300Mbps and 150Mbps) depending on the distance Transmitter – Receiver (TR). Obviously, the mean loss of the signal increases as the TR distance increases. However, the 300Mbps wireless USB receiver provides lower losses than the 150Mbps one. Consequently, the maximal distance at which the signal is still useful (90dB) is about 350metres and 320metres for respectively 300Mbps and 150Mbps receiver.

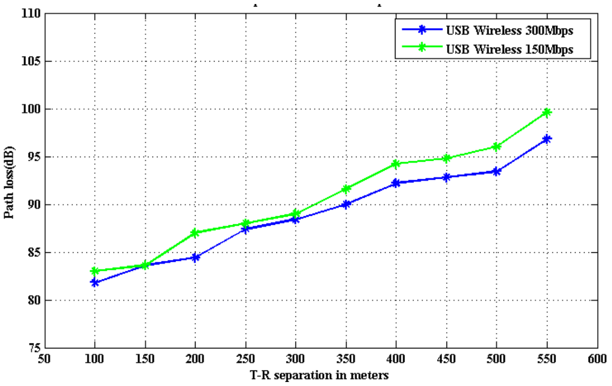
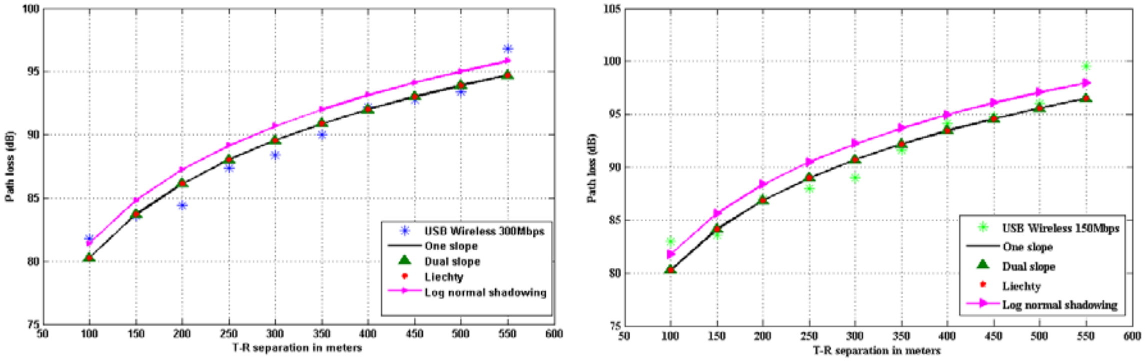


Fig. 2. Mean losses in free space

Comparison between models is summarised in Table 4. From Figure 3, One Slope, Dual Slope and Liechty models are quite close to the measured values. Because of their large errors, results from the Partitioned model could not be plotted.

From the experimental results provided in Table 4, One Slope, Dual Slope, and Liechty models are the best in free space. In fact, they provide a mean error of 0.90dB and a standard deviation of 1.12dB for the 300Mbps receiver, and a mean error of 1.12dB and a standard deviation of 1.50dB for the 150Mbps receiver.



a) 300Mbps wireless receiver

b) 150Mbps wireless receiver

Fig. 3. Comparison of the predicted values in free space.e

Table 4. Experimental results in free space

Models	300Mbps / 150Mbps	
	Mean Error (dB)	Standard Deviation (dB)
One Slope	0.90 / 1.12	1.12 / 1.50
Dual Slope	0.90 / 1.12	1.12 / 1.50
Lietchy	0.90 / 1.12	1.12 / 1.50
Log Normal	1.54 / 1.71	1.68 / 1.85
Partitioned	141.03 / 103.83	142.89 / 104.06

4.2 Result analysis in a wooded area

A total number of 200 measurements have been performed in wooded area. Figure 4 shows the pathloss of the signal for both USB wireless receivers (300Mbps and 150Mbps) depending on the TR distance. Figure 4 shows an increasing saw tooth curve, in contrary to the fairly linear curve observed in free space. The main reason is the presence of obstacles which do not ease the prediction of the path loss at any point. However, as it is the case in free space, the 300Mbps wireless USB receiver provides lower losses than the 150Mbps one. In addition, the maximal distance at which the signal is still useful (90dB) is about 182metres and 177metres for respectively 300Mbps and 150Mbps receiver.

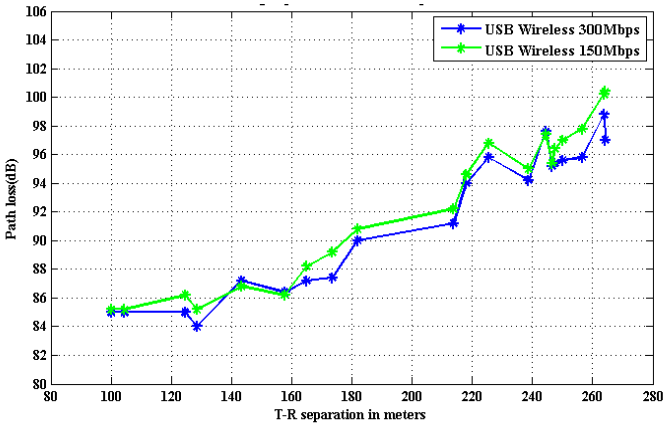
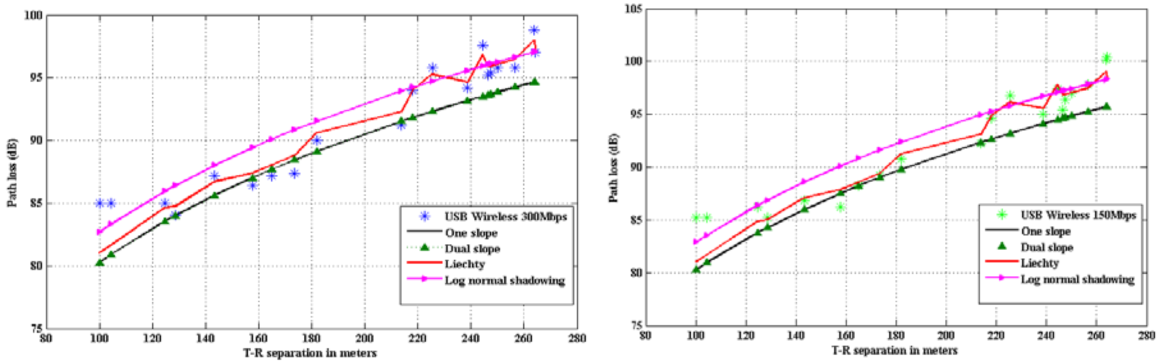


Fig. 4. Mean losses in wooded area.

From Figure 5, the predicted values from the Liechty model are quite close to the measured ones.



a) 300Mbps wireless receiver

b) 150Mbps wireless receiver

Fig. 5. Comparison of the predicted values in the wooded area.

According to the experimental results provided in Table 5, the Liechty model provides the best results. In fact, this model provides a mean error of 0.94dB and a standard deviation of 1.35dB for the 300Mbps receiver; and a mean error of 1.04dB and a standard deviation of 1.50dB for the 150Mbps receiver.

Table 5. Experimental results in wooded area

Models	300Mbps / 150Mbps	
	Mean Error (dB)	Standard Deviation (dB)
One Slope	1.73 / 2.02	2.10 / 2.48
Dual Slope	1.73 / 2.02	2.10 / 2.48
Lietchy	1.60 / 1.27	2.03 / 1.78
Log Normal	1.97 / 1.85	2.49 / 2.30
Partitioned	109.71 / 108.48	109.93 / 108.69

4.3 Result analysis in built-up areas

A total number of 200 measurements have been performed in built-up area. Figure 6 shows the pathloss of the signal for both USB wireless receivers (300Mbps and 150Mbps) depending on the TR distance. Because of the presence of obstacles, Figure 6 also shows a non-monotonic increasing curve. As expected, since it is the case in free space and wooded area, the 300Mbps wireless USB receiver provides lower losses than the 150Mbps one. From Figure 7, the predicted values from the Liechty model are quite close to the measured ones. In contrary to previous scenarios, the maximal distance at which the signal is still useful (90dB) is the same and about 165metres, irrespectively the type of receiver.

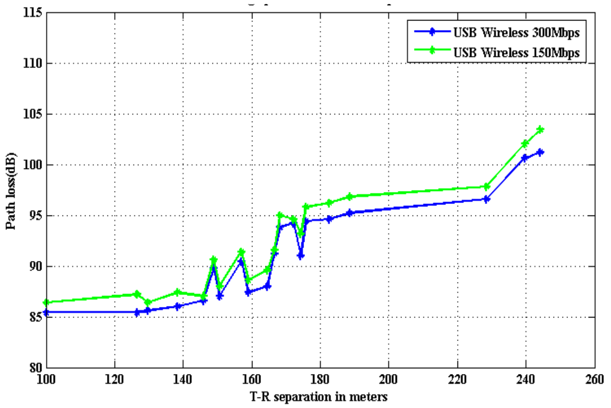
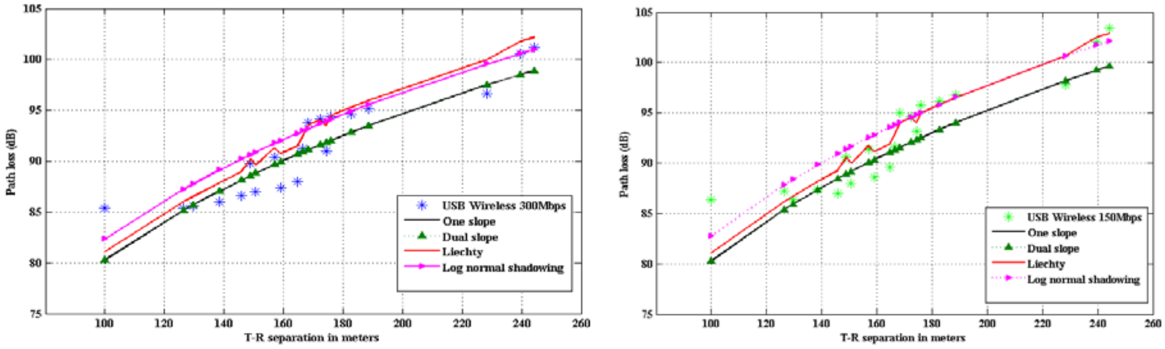


Fig. 6. Mean losses in a built-up area.

Table 6. Experimental results in built-up area

Models	300Mbps / 150Mbps	
	Mean Error (dB)	Standard Deviation (dB)
One Slope	1.73 / 2.02	2.10 / 2.48
Dual Slope	1.73 / 2.02	2.10 / 2.48
Lietchy	1.60 / 1.27	2.03 / 1.78
Log Normal	1.97 / 1.85	2.49 / 2.30
Partitioned	109.71 / 108.48	109.93 / 108.69



a) 300Mbps wireless receiver

b) 150Mbps wireless receiver

Fig. 7. Comparison of predicted values in a built-up area.

From the experimental results provided in Table 6, the Liechty model provides once more the best results. Indeed, this model provides a mean error of 1.60dB and a standard deviation of 2.03dB for the 300Mbps receiver; and a mean error of 1.27dB and a standard deviation of 1.78dB for the other one. new model.

5 Improving the Lietchy model in combined environment

The better results of the Liechty model are justified by the fact that the model considers the number of obstacles between transmitter and receiver as well as their attenuation on the signal. However, we assumed that the distance between the transmitter and each attenuation point may also influence the quality of the signal. Therefore, considering those distances during the prediction of the signal could improve the accuracy. But, since it is difficult to consider the distances of all those attenuation points, we take only into account the distance to the first obstacle. This point is called the breakpoint. We suppose that the greatest attenuation of the signal occurs at this point.

5.1 Path loss expressions

The new model is obtained by merging the parameters of the Dual slope and the Liechty models. We obtain equation (10):

$$PL_{new} = PL_{fs}(d_0) + \left\{ \begin{array}{c} 10n_1 \log_{10} d \\ 10n_2 \log_{10} \frac{d}{d_{bp}} + 10n_1 \log_{10} d_{bp} + \sum_i num_i * a_i \end{array} \right\} \begin{array}{l} (10.1) \\ (10.2) \end{array} \qquad (10)$$

With, $1m < d \le d_{bp}$ in (10.1) and $d > d_{bp}$ in (10.2).

Where:

$PL_{fs}(d_0)$: Signal path loss in free space at distance d$_0$ (m)

d_0 : Distance of reference (m)

d : Distance between the transmitter and the receiver (m)

d_{bp} : Distance from the transmitter to the first obstacle (m)

num_i : Number of obstructions of type i

a_i : Attenuation of obstructions of type i (dB)

n_1 : Path loss exponent of the environment $d \le d_{bp}$

n_2 : Path loss exponent of the environment $d > d_{bp}$

5.2 Accuracy analysis of the model

The accuracy of the new model is analyzed by considering the measurements of the signal in both LOS and NLOS. Only wooded and built-up areas are considered. The consideration of both LOS and NLOS is materialized by using different path loss exponents, as it is the case in the new model: n_1 in LOS and n_2 in NLOS. The accuracy is evaluated in NLOS environment and in combined environment (LOS and NLOS) environment.

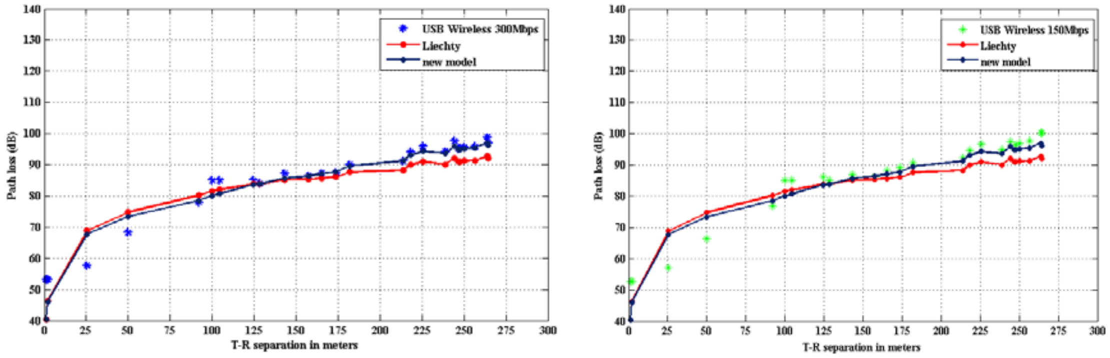
5.2.1 Wooded area

Table 7 presents the results of the accuracy analysis in the wooded area. From Table 7, even if both models provide similar results, Liechty model is more precise in NLOS environment. However, the new model provides better predictions in combined environment. In fact, the new model provides in combined environment a mean error of 2.33dB and 2.67dB respectively for 300Mbps and 150Mbps USB receivers;

which are lower compared to 4.22dB and 4.00dB provided by Liechty model. Likewise, the new model also provided in combined environment a standard deviation of 4.02dB and 4.21dB respectively for 300Mbps and 150Mbps receivers; which are also lower compared to 5.12dB and 5.09dB provided by Liechty model. Figure 8 and 9 give a graphical comparison between both models and the measured data in wooded area, respectively for combined (LOS and NLOS) and NLOS environments.

Table 7. Results of the accuracy analysis in the wooded area

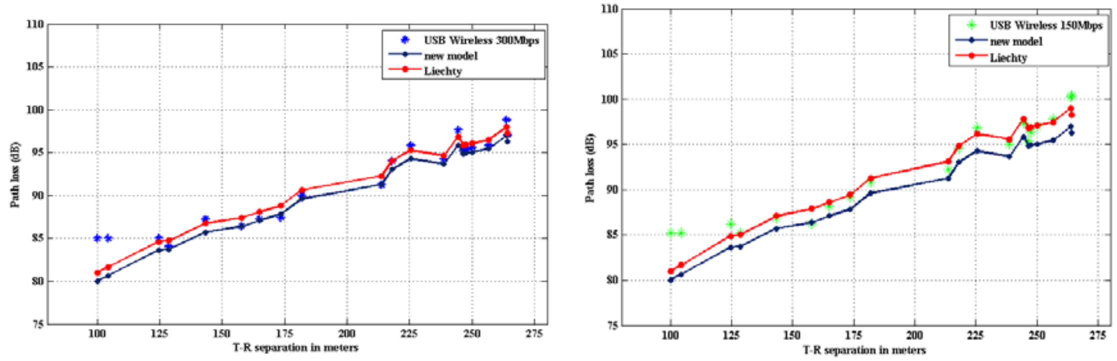
Models	LOS and NLOS (300Mbps/150Mbps)		NLOS only (300Mbps/150Mbps)	
	Mean Error (dB)	Std Deviation (dB)	Mean Error (dB)	Std Deviation (dB)
Liechty	4.22 / 4.00	5.12 / 5.09	0.94 / 1.04	1.35 / 1.50
New model	2.33 / 2.67	4.02 / 4.21	1.13 / 2.02	1.73 / 2.41



a) 300Mbps wireless receiver

b) 150Mbps wireless receiver

Fig. 8. Accuracy of models in the wooded area (combined: LOS and NLOS).



a) 300Mbps wireless receiver

b) 150Mbps wireless receiver

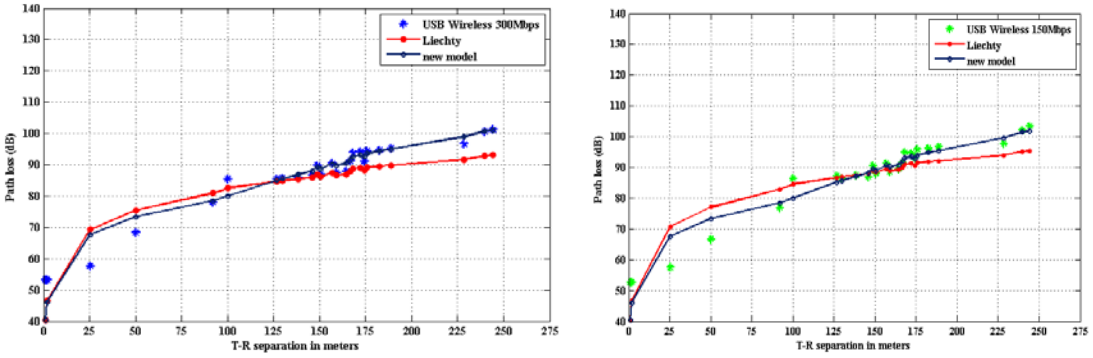
Fig. 9. Accuracy of models in the wooded area (NLOS).

5.2.2 Built-up area

Table 8 presents the results of the accuracy analysis in the built-up area. From Table 8, even if both models provide similar results, Liechty model provides better predictions in NLOS environment. But, the new model is more precise in combined environment as it was the case in wooded area. Actually, the new model provides in combined environment a mean error of 2.36dB and 2.61dB respectively for 300Mbps and 150Mbps receivers; which are low compared to 4.35dB and 4.00dB provided by Liechty model. Likewise, the new model also provided in combined environment a standard deviation of 4.01dB and 4.19dB respectively for 300Mbps and 150Mbps receivers; which are also low compared to 5.46dB and 5.45dB provided by Liechty model. Figure 10 and 11 give a graphical comparison between both models and the measured data in built-up area, respectively for combined (LOS and NLOS) and NLOS environments.

Table 8. Results of the accuracy analysis in the wooded area (LOS and NLOS)

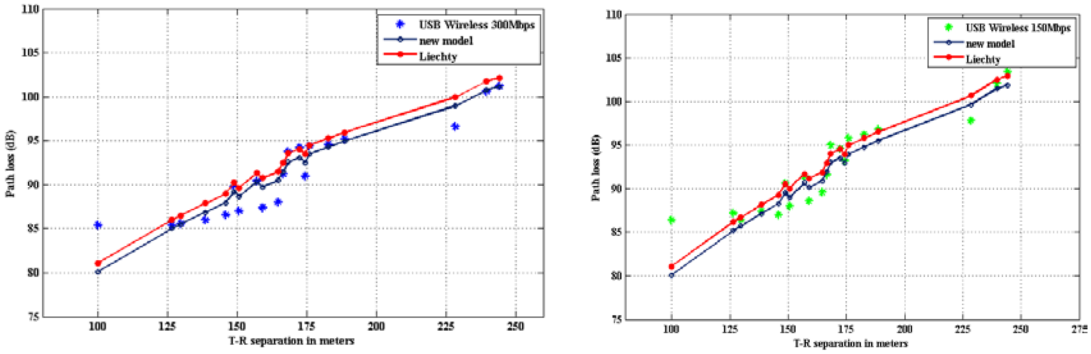
Models	LOS and NLOS (300Mbps/150Mbps)		NLOS only (300Mbps/150Mbps)	
	Mean Error (dB)	Std Deviation (dB)	Mean Error (dB)	Std Deviation (dB)
Liechty	4.35 / 4.00	5.46 / 5.45	1.60 / 1.27	2.03 / 1.78
New model	2.36 / 2.61	4.01 / 4.19	1.73 / 1.41	2.15 / 1.88



a) 300Mbps wireless receiver

b) 150Mbps wireless receiver

Fig. 10. Accuracy of models in the built-up area (combined: LOS and NLOS).



a) 300Mbps wireless receiver

b) 150Mbps wireless receiver

Fig. 11. Accuracy of models in the built-up area (NLOS).

6 Conclusion

The aim of this work was to determine a precise empirical path loss model for rural regions at 2.4GHz. We defined three scenarios tied to rural regions and we conducted a measurement campaign. We compared five selected models and we found that Liechty model is the more precise one. But Liechty model outperforms others models in separated environments. Further, we improved this model in combined environment (considering LOS and NLOS) by considering the distance to the first breakpoint. We obtained a better prediction model with a mean error of 2.33dB/2.67dB and a standard deviation of 4.02dB/4.21dB in worst case in wooded area and a mean error of 2.36dB/2.61dB and a standard deviation of 4.01dB/4.19dB in worst case in built space, respectively for 300Mbps/150Mbps USB receivers. A non-neglecting observation is the fact that standard deviation and the mean error in path loss prediction increase for receiver with low data rate. It would be interesting to study this variation in other to propose more realistic models.

Since we usually observe more than one breakpoint between the transmitter and the receiver, it can be interesting to study the impact of other breakpoints on the precision of the path loss model and to consider those points without making the model more complex. A last perspective is to implement the path loss model in network simulator in order to predict more accurately the signal path loss in likewise rural regions.

REFERENCES

[1]. Fendji, Ebongue Kedieng, J.L.: Rethinking Network Connectivity in Rural Communities in Cameroon. Presented at the May 17 (2015).

[2]. Isong, B., Dladlu, N., Magogodi, T.: Computer Network and Information Security. Comput. Netw. Inf. Secur. 11, 14–22 (2016).

[3]. Fendji Kedieng Ebongue, J.L., Nlong, J.M.: Rural Wireless Mesh Network: A Design Methodology. Int. J. Commun. Netw. Syst. Sci. 8, 1–9 (2015).

[4]. Egli, J.: Radio Propagation above 40 MC over Irregular Terrain. Proc. IRE. 45, 1383–1391 (1957).

[5]. OKUMURA, Y.: Field strength and its variability in VHF and UHF land-mobile radio service. Rev. Electr. Commun. Lab. 16, 825–873 (1968).

[6]. Hata, M.: Empirical formula for propagation loss in land mobile radio services. IEEE Trans. Veh. Technol. 29, 317–325 (1980).

[7]. Erceg, V., Greenstein, L.J., Tjandra, S., Parkoff, S.R., Gupta, A., Kulic, B., Julius, A., Jastrzab, R.: An empirically-based path loss model for wireless channels in suburban environments. In: IEEE GLOBECOM 1998 (Cat. NO. 98CH36250). pp. 922–927. IEEE.

[8]. Medeisis, A., Kajackas, A.: On the use of the universal Okumura-Hata propagation prediction model in rural areas. In: VTC2000-Spring. 2000 IEEE 51st Vehicular Technology Conference Proceedings (Cat. No.00CH37026). pp. 1815–1818. IEEE.

[9]. Rakesh, J., A., W.V., Dalal, U.: A Survey of Mobile WiMAX IEEE 802.16m Standard. (2010).

[10]. Akl, R.G., Tummala, D., Li, X.: Indoor Propagation Modeling at 2.4 GHZ for IEEE 802.11 Networks. Sixth Int. Assoc. Sci. Technol. Dev. Int. Multi-Conference Wirel. Opt. Commun. 2006, Banff, Alberta, Canada. (2006).

[11]. Liechty, L.C.: Path Loss Measurements and Model Analysis of a 2.4 GHz Wireless Network in an Outdoor Environment. (2007).

[12]. Yenke, B.O., Tala, D.C.M., Louis, J., Fendji, E.K.: Extended Probabilistic Cost Model (EPCM): A Framework for Cost Estimation of Wireless Network Deployment in Rural Areas. Inf. Eng. Electron. Bus. 1, 1–9 (2017).

[13]. Fendji Kedieng Ebongue, J.L., Mafai, N., Nlong, J.M., Ebongue, J.L.F.K., Nelson, M., Nlong, J.M.: Empirical path loss models for 802.11n wireless networks at 2.4 GHz in rural regions. In: Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST. pp. 53–63 (2015).

[14]. Tan, S.Y., Tan, H.S.: Propagation model for microcellular communications applied to path loss measurements in Ottawa city streets. IEEE Trans. Veh. Technol. 44, 313–317 (1995).

[15]. Sun, Q., Tan, S.Y., Teh, K.C.: Analytical Formulae for Path Loss Prediction in Urban Street Grid Microcellular Environments. IEEE Trans. Veh. Technol. 54, 1251–1258 (2005).

[16]. Taflove, A., Hagness, S.C.: Computational electrodynamics : the finite-difference time-domain method. Artech House (2005).

[17]. Rana, M.M., Mohan, A.S.: Segmented-Locally-One-Dimensional-FDTD Method for EM Propagation Inside Large Complex Tunnel Environments. IEEE Trans. Magn. 48, 223–226 (2012).

[18]. Friis, H.T.: A Note on a Simple Transmission Formula. Proc. IRE. 34, 254–256 (1946).

[19]. Seidel, S.Y., Rappaport, T.S.: 914 MHz path loss prediction models for indoor wireless communications in multifloored buildings. IEEE Trans. Antennas Propag. 40, 207–217 (1992).

[20]. Andrade, C.B., Hoefel, R.P.F.: IEEE 802.11 WLANs: A comparison on indoor coverage models. In: CCECE 2010. pp. 1–6. IEEE (2010).

[21]. Cichon, D.J., Kürner, T.: Digital mobile radio towards future generation systems: Cost 231 final report. COST Eur. Coop. F. Sci. Tech. Res. 231, (1993).

[22]. Green, D.B., Obaidat, A.S.: An accurate line of sight propagation performance model for ad-hoc 802.11 wireless LAN (WLAN) devices. In: 2002 IEEE International Conference on Communications. Conference Proceedings. ICC 2002 (Cat. No.02CH37333). pp. 3424–3428. IEEE.

[23]. ERCEG, V.: Channel Models for Fixed Wireless Applications. IEEE 802.16.3c-01/29rl. (2001).

[24]. Brown, D. and M. S. Gregory: A report on technology independent methodology for the modelling, simulation, and empirical verification of wireless communication system performance in noise and interference limited systems operation on frequencies between 30 and1500 mhz. Technical report, Telecommunication Industry Association (TIA) TR8 Working Group 8.8, May 1997.

[25]. National Institute of Standards and Technology: Hata and CCIR Formulas IST. http://w3.antd.nist.gov/wctg/manet/calcmodels_r1.pdf, visited: March 2018.

[26]. European Radio Communications Committee: The analysis of the coexistence of FWS cells in the 3.4 - 3.8 GHz bands. ERC Report, European Conference of Postal and Telecommunications Administrations (CEPT), May 2003.

DOI: 10.14738/tnc.51.2796
Publication Date: 09th March 2017
URL: http://dx.doi.org/10.14738/tnc.51.2796