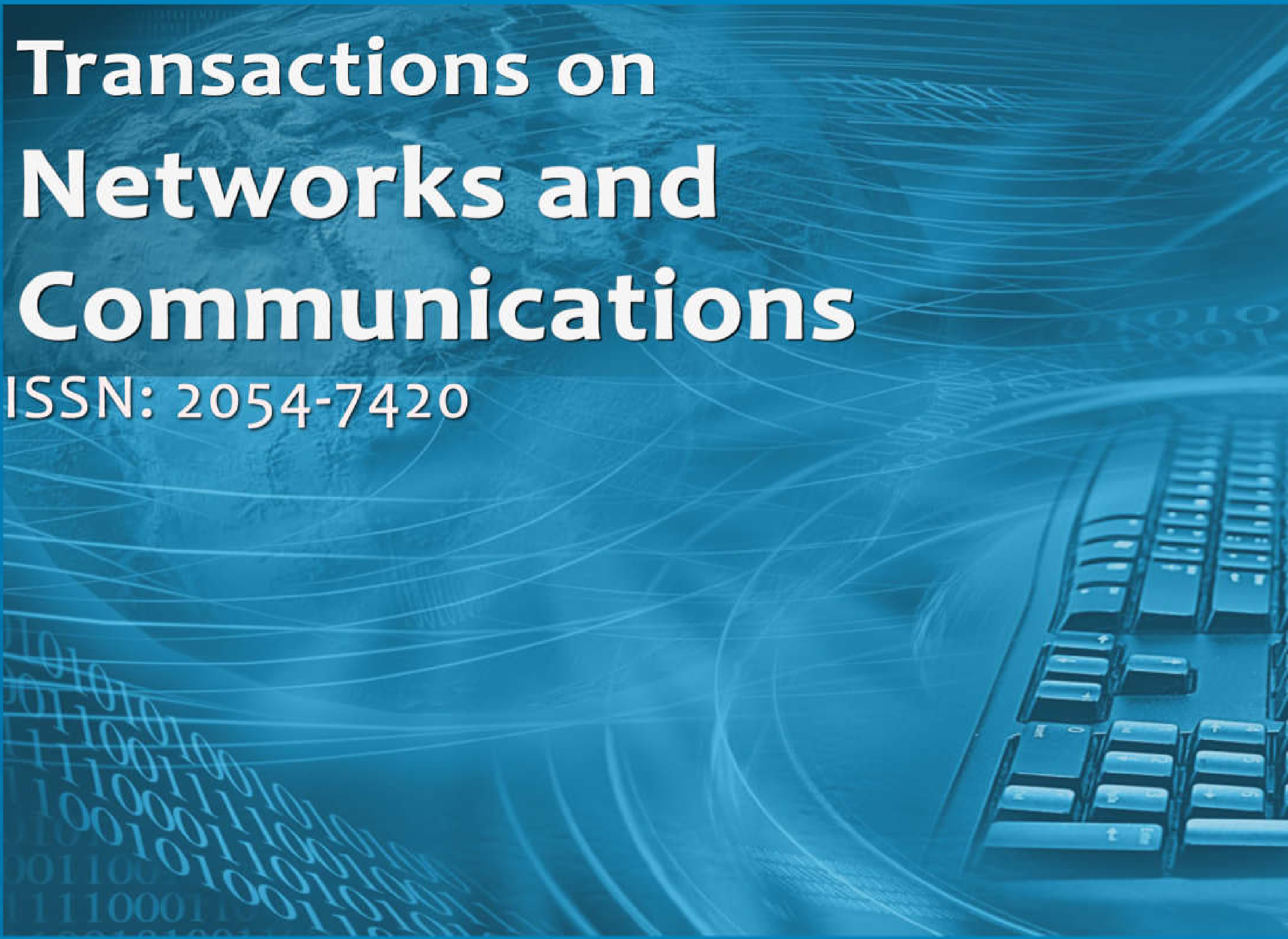
Volume 5 No. 4, August 2017



Transactions on Networks and Communications

ISSN: 2054-7420

Society for Sciences and Education - United Kingdom

TABLE OF CONTENTS

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 5, No. 4
ISSN: 2054 -7420

ZRSA Algorithm

Dr. Mohamed Alzaabi
alzaabi123@gmail.com

ABSTRACT

Today, RSA algorithm is one of the encryption algorithm that is used widely. However, with the advances in computer power it is becoming susceptible to be cracked. It is become a necessity to develop a new algorithm that can withstand future processing capacity. Zaabi RSA (ZRSA) is a new algorithm that is designed to remedy RSA weaknesses.

Keywords: ZRSA; RSA; Algorithm, Cryptography; RSA; Encryption; Decryption; Key Generation

1 Introduction

Cryptography is part of the art of protecting information. The modern uses for cryptography are encrypting and decrypting emails, credit cards and other confidential data.

Cryptography is known as "symmetric key system" that generate one secret key to be shared between the sender and the receiver and "public key system" (asymmetric) that generates two keys, a public key which is known by all and a private (secret) key which is known by the receiver.

2 Encryption Techniques

While doing encryption with the help of computers, there are two main approaches. These two approaches are symmetric encryption and asymmetric encryption systems

2.1 Symmetric Encryption

Symmetric encryption, also known as secret key cryptography is based on using the same key to encrypt and decrypt a message.

This technique requires great care in key distribution because the same key both encrypts and decrypts the message [1]. The problem is how to distribute keys and the solution varies. One can get along using a key when you physically meet or remotely using any type of media. Here, it is appropriate to select a media other than the one used for the encrypted information.

2.2 Asymmetric Encryption

The form of cryptosystem that uses encryption and decryption with two separate keys is called asymmetric encryption. One of the keys is called public key and the decryption key is called private key [2].

Asymmetric encryption is thus based on a user to use different keys to encrypt and decrypt a data set. The technique is also called Public Key Cryptography.

DOI: 10.14738/tnc.54.3460
Publication Date: 23rd July 2017
URL: http://dx.doi.org/10.14738/tnc.54.3460

Each user has a key pair consisting of a public and a private key. The public key can be made available to other users via a database. The private key, as the name suggests, is available only to its owner and the user should not give it to anyone else.

Asymmetric encryption algorithms based on the use of a reverse approach are called trap door encryption [3]. A trapdoor is a process that is easy to implement in one direction, but very difficult to do if one is to go back in the other direction. The asymmetric encryption algorithms that are considered safe today use all the trap doors that are taken from mathematics.

An example of an asymmetric encryption algorithm is the RSA algorithm, to be described in section 5.1.1.2.1. It is a block cipher algorithm that divides the plaintext into blocks that are encrypted separately. The size of the blocks depends on the length of the key that is used.

It is an asymmetric encryption algorithm because it uses public and private keys. Therefore, a user must first generate public and private keys to be able to make use of the RSA algorithm. The keys are calculated according to a certain pattern to be discussed in this section.

When a message is encrypted with asymmetric technology, it is done in the following way. Subscriber A wants to send an encrypted message to subscriber B, A encrypts the message with B's public key. When B receives the message, he can decrypt it with his private key.

The RSA algorithm easily multiplies two large prime numbers p and q to power n [4] but much more difficult to factor n to p and q.

3 RSA Algorithm

RSA algorithm is an asymmetric encryption algorithm that was developed in 1977. The algorithm gets its name from its three creators' surnames, Rivest, Shamir and Adelman. It is a block cipher algorithm, divides the plaintext into blocks that are encrypted separately. The size of the blocks depends on how big the key is used.

The RSA algorithm is an asymmetric encryption algorithm that uses public and private keys. Therefore, a user must first create a public and a private key to be able to make use of the RSA algorithm. The keys are calculated according to a certain pattern.

The working Principle and the key establishment process of RSA is described as follows. The RSA Algorithm revolves around the three-basic step wise procedures and they are classified accordingly as Key Generation, Encryption and Decryption.

3.1 Key Generation

There are generally two types of keys in RSA [5]. They are classified into public and private keys. The public key can be disclosed to anyone but the private key is confidential and never disclosed as it’s primarily very important in decrypting the data encoded using the RSA system. The keys are generated from the below factors.

Initially the user opts for two distinct prime numbers p and q. For security purposes, the integers, p and q, should be chosen at random, and should be the same bit-length. Prime integers can be efficiently found using a primality test.

The next stage involves in computing and finding the value of "n" which is equal to "n=p*q"

The third iteration process is using moduli for the asymmetric cryptographic process.

The next step is computing the values of "phi" which is ɸ= (p-1) (q-1)

The Fifth step involves in choosing an integer e such that 1 < E < ɸ, (E, ɸ) = 1

We then find the value of the d = E^{-1} Mod ɸ(n); i.e. d is the multiplicative inverse of E mod ɸ, and d is kept as a private exponent encryption secret.

3.2 Encryption

Assume a subscriber sender at "X", transmitting their public key (n, E) to another subscriber sender at "Y", the destination user keeps the private key as secret and does not disclose any information to the system. The text message is encrypted and sent to the destination as a cipher text to the user at the receiving end:

"C = M^E Mod n".

This is shown in Figure 3.2.1.

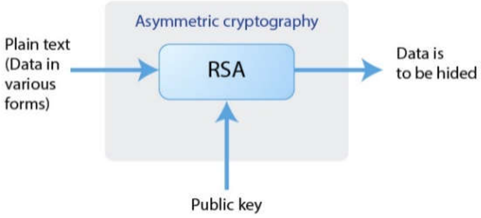


Figure 1: Encryption Using Public Key

3.3 Decryption

The Cipher text can be transformed into the original text through a recovering technique using the factor d of the cryptographic component.

The Message which is obtained can be transformed into "M = Cd Mod n”.

Generally, the length of the bit string of n should be 512 bits at least. The decryption using public at block level key is depicted Figure 3.3.1

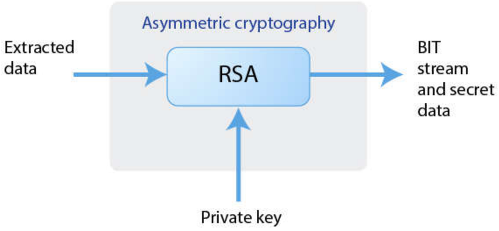


Figure 2: Decryption Using Public Key

3.4 Modified RSA with Two Random Numbers

Modified RSA with two random numbers is similar to RSA apart from using z1 and z2 to replace z.

z_1 = x * y and z_2 = x * y * a * b. a and b are considered two random numbers. With this method, z_2 is linearly bigger than z. Hence, the modified RSA is safer than the original RSA as the de factorization would require longer time. The paper provided an example to prove its accuracy.

3.5 Modified RSA with Three Prime Numbers

Paper published by Patidar [6] listed a new RSA algorithm with three prime numbers, instead of the usual two prime numbers.

The three prime numbers are:

p, q & r.

ϕ and n are calculated as follows:

$n = p * q * r$

$\phi(n) = (p-1) * (q-1) * (r-1)$

The two conditions GCD (E, ϕ(n)) =1 and 1<E< ϕ(n) are applied to this method, which is similar to the original RSA. The sender encrypts the message using the following standard formula:

$C=M^E \text{ Mod } n$

While the receiver computes the private key d using:

$d = E^{-1} \text{ Mod } \phi(n)$

The value of d would allow the receiver to decrypt the message using the following standard formula:

$M=C^d \text{ Mod } n$.

So, by inducing three prime numbers, this method has boosted the value of n. Since the number is bigger, this technique has met the same aim for the algorithm that uses two random numbers explained above.

3.6 Possible ways to attack RSA

The possible ways for attackers to obtain the private key d, is to view all parameters associated with the RSA algorithm and identify their weaknesses and strength.

The knowledge of the RSA algorithm is readily available to attackers/hackers. The parameters that are associated with the RSA algorithm are n, E, C and d. As disclosed above in section 3.5, n is public key and d is secret key. Worst case scenario, all other information of RSA is public, i.e. attackers may be able to get the values of n, E and C. Figure 3.6.1 exposes RSA security issues.

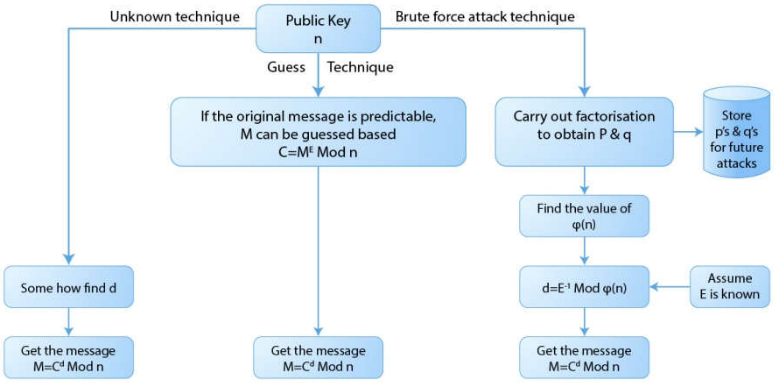


Figure 3: RSA security issues

Finding d is the first technique, the Unknown Technique, which may lead to decrypting the message and publishing it in the public domain.

With the second technique, if the message is predictable, the attacker makes a guess at the original message, by giving a value to M. Therefore, M^E Mod n can be calculated. If the result matches the encrypted version, the iterative process is to be stopped, otherwise, another guess is given and the process starts again. This technique pushed computer manufacturers to append random padding to make the message unpredictable.

The third technique is Brute Force Attack. Here n is factorised to reach p & q. Hence, ϕ can be calculated. Using the Extended Euclidian algorithm, d can be calculated and the message can be decrypted. If n is large this technique could take a long time.

Figure 3.6.2 defines n as small, medium and large numbers relative to the size of bits.

n is considered as:	**If:**
Small	$n < 2^{136}$
Medium	$2^{136} \leq n < 2^{512}$
Large	$2^{512} < n$

Figure 4: n is defined as small, medium or large depending on the size of bits

It has been reported that, if n is small or medium, RSA has been factorised [7]. As computing power is continually increasing, it is only a matter of time for large n to be factorised. Hence, a new RSA algorithm has to be developed to make the factorisation impossible or near impossible.

4 ZRSA

The two new RSAs listed above have provided techniques to show that the value of n is increasing linearly with the introduction of new parameters. The increase in value n may make the factorisation difficult but not different from the original RSA, at least for the time being. However, time within the last two/three decades has proven that computing power is increasing continually. So, it is a matter of short time and hackers would be able to crack RSA and the above suggested RSAs, with ease.

The suggested alternative to the original RSA algorithm in this project is referred to as ZRSA. ZRSA uses three prime numbers and three random numbers which is the main difference with the original RSA algorithm. The third random number is expressed as:

$$z = e^{\text{1st Random Number}} + e^{\text{2nd Random Number}}$$

The new parameters within ZRSA affect the three components of the generated public and private keys (E, n) & (d, n), i.e. E, n & d. ZRSA Algorithm revolves around the three basic stepwise procedures and they are classified accordingly as Key Generation, Encryption and Decryption.

4.1 Key Generation

There are generally two types of keys in ZRSA, they are classified into public and private keys. The public key can be disclosed to anyone but the private key is confidential and never disclosed as it is primarily very important in decrypting the data encoded using the RSA system. The keys are generated from the below factors.

Initially, the user for ZRSA opts for three distinct prime numbers p, q and r. For security purposes, the integers, p, q and r, should be chosen at random, and should be of the same bit-length. Prime integers can be efficiently found out using a primality test. The user for ZRSA opts for two distinct random numbers a and b. The third random number labelled as z is expressed as:

$$z = e^{\text{1st Random Number}} + e^{\text{2nd Random Number}}$$

The next stage involves computing the value of "n" which is equal to "n=p*q*z"

The third iteration process is using moduli for the asymmetric cryptographic process.

The next step is computing the values of "phi" which is $\phi(n) = (p-1)(q-1)(z-1)$

The fifth step involves choosing an integer E such that $1 < E < \phi$, provided GCD $(E, \phi(n)) = 1$

The sixth step is to find the value of $d = E^{-1} \text{ Mod } \phi(n)$. From the formula d is the multiplicative inverse of E Mod ϕ. d is kept as the private exponent encryption secret.

To find the modular inverse with respect to the 'ϕ', d which is one element of the ZRSA private key, the following set of equations has to be used:

$$d = E^{-1} \text{ Mod } \phi$$

or

$$d * E = 1 \text{ Mod } \phi$$

The extended Euclidian algorithm has to be used to calculate the value of 'd', as follows:

GCD $(\phi, E) = \phi x + Ey$, where GCD $(\phi, E) = 1$ in ZRSA, and

$y = d$ if $d < \phi$ and $d \neq$ negative integer.

How to find the inverse of ϕ Mod E?

If GCD = $x*E + y*\phi$, $E = n$ and $\phi = m$, GCD in ZRSA is = 1.

Hence:

$$1 = x*E + y*\phi$$

If $E = r_{i-1}$ and $\phi = r_{i-2}$, use:

$r_{i-2} = q_i r_{i-1} + r_i$, q_i is quotient and r_i is the reminder ---- (1)

Equation (1) is used to generate the quotients, q_i's.

$d = x_{i-2} – (x_{i-1} * q_{i-2}) * \text{ Mod } \phi$, where $x^{-2} = 0$, $x^{-1} = 0$, $q^{-2} = 0$ & $q^{-1} = 0$

$r_{i-2} = q_i * r_{i-1} + r_i$

The above set of equations has been formalised in Excel file to produce n, E and d.

Hence, the public key and the private key are:

URL:http://dx.doi.org/10.14738/tnc.54.3460

Public Key = (n, E) and

Private Key = (n, d)

4.2 Encryption

There are conditions to ZRSA, prior to transmitting and receiving, which have to be fulfilled to ensure security. These conditions are similar to the standard RSA conditions. The first condition is that a sender, subscriber (SS1), has to transmit a public key (n,E) to the receiver, the other subscriber (SS2). The second condition is that the destination user, SS2 keeps the private key unit d as secret and does not disclose it to the WiMAX system. The text message M is padded, encrypted and sent to the destination as a cipher text, C to the user at the receiving end. The encryption formula is:

C = ME Mod n

The conceptual idea of ZRSA encryption is shown in Figure 4.2.1.

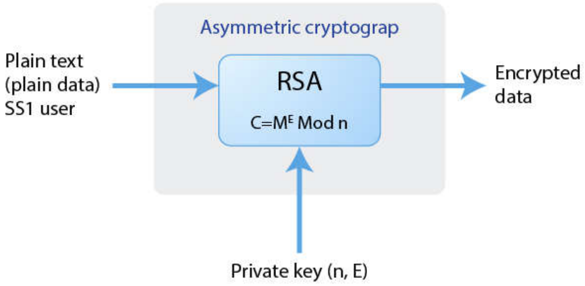


Figure 5: Encryption Using Public Key

4.3 Decryption

The Cipher text can be transformed into the original text using the factor d of the cryptographic component. The decryption formula is:

M = C^d Mod n

M is transformed into plain text with a specified padding. The conceptual idea of ZRSA decryption is shown in Figure 4.3.1**.**

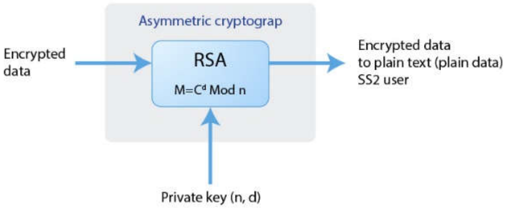


Figure 6: Decryption Using Private Key

5 Modified ZRSA

Again, the modified ZRSA is based on three basic steps: Key Generation, Encryption and Decryption. All of the three steps are described below.

5.1 Key Generation

The values of n's and ϕ's are worked out as follows:

Set of n’s equations:

$n_p = q*z$

$n_q = p*z$

$n_z = p*q$

Note that n_p is related to the combination of prime’s q & z, n_q is related to the combination of prime’s p & r and n_r is related to the combination of prime’s p & q. other combinations, such as p & q, q & z and z & p or even p & q & z, will lead to similar results.

Set of ϕ’s equations:

$\phi_p = (q-1)\ (z-1)$

$\phi_q = (p-1)\ (z-1)$

$\phi_r = (p-1)\ (q-1)$

ZRSA generates two types of keys. They are classified into public and private keys. The public key can be disclosed to anyone but the private key is confidential and never disclosed as it is primarily very important in decrypting the data encoded using the RSA system. The keys are generated from the below factors.

Initially, the user for ZRSA opts for three distinct prime numbers p, q and z. For higher security purposes, the integers, p, q and z, should be chosen at random and the same bit-length. Prime integers can be efficiently found out using a primality test. The user for ZRSA opts for two distinct random numbers, namely a & b. The third random number labelled as z expressed as the multiplication of the first two random numbers, a & b. As the random numbers will increase the value of n, within this section, they have been eliminated from the following example.

The next stage involves computing the value of "n_p, n_q & n_z “.

The third iteration process is using moduli for the asymmetric cryptographic process.

The next step is computing the values of "phi’s" which are” ϕ_p, ϕ_q & ϕ_z “.

The fifth step involves choosing an integer E such that $1 < E < \phi$, GCD $(E, \phi(n)) = 1$

The sixth step is to find the value of $d = E^{-1}$ Mod $\phi(n)$. From the formula, d is the multiplicative inverse of E Mod ϕ. d is kept as the private exponent encryption secret.

To find the modular inverse with respect to 'ϕ', d, the following set of equations has to be used:

$d_p * E_p = 1$ Mod ϕ_p

$d_q * E_q = 1$ Mod ϕ_q

$d_z * E_z = 1$ Mod ϕ_z

The extended Euclidian algorithm has to be used to calculate the value of 'd', as follows:

GCD $(\phi, E) = \phi x + Ey$, where GCD $(\phi, E) = 1$ in ZRSA, and

$y = d$ if $d < \phi$ and $d \neq$ negative integer.

How to find the inverse of ϕ mod E for p, q & z primes?

If GCD = E*x + ϕ*y, E = n and ϕ = m, GCD is equal 1.

Hence:

1 = E*x + ϕ*y

If E = r_{i-1} and ϕ = r_{i-2}, use the following iterative equations:

$r_{i-2} = q_i\, r_{i-1} + r_i$ q_i is quotient and r_i is the remainder

d = $x_{i-2} – (x_{i-1} * q_{i-2})$ * Mod ϕ, and

$r_{i-2} = q_i\, r_{i-1} + r_i$

The above set of equations has produced n, E and d. Hence, the public key and the private keys for primes p, q and z are:

p

Public Key = (n_p, E_p) and

Private Key = (n_p, d_p)

q

Public Key = (n_q, E_q) and

Private Key = (n_q, d_q)

r

Public Key = (n_z, E_z) and

Private Key = (n_z, d_z)

ZRSA public and private keys can be summarised as two set of keys:

Public key (E_p, E_q, E_z, n_p, n_q, n_z)

Private key (d_p, d_q, d_z, n_p, n_q, n_z)

p, q & z keys are used for successive letters repeatedly.

5.2 Encryption and Decryption

The encryption and decryption keys for:

p

To be applied to the first letter of the encrypted/decrypted message:

Encryption $C = M^{Ep} \pmod{n_p}$

Decryption $M = C^{dp} \pmod{n_p}$

q

To be applied to the second letter of the encrypted/decrypted message:

Encryption $C = M^{Eq} \pmod{n_q}$

Decryption $M = C^{dq} \pmod{n_q}$

z

To be applied to the third letter of the encrypted/decrypted message:

Encryption $C = M^{Ez} \pmod{n_z}$

Decryption $M = C^{dz} \pmod{n_z}$

6 Conclusion

With the continuous advances in computing power, RSA algorithm is becoming subject to be cracked soon. However, the new cryptography algorithm ZRSA will ensure that cracking is not achievable in the near future since it is almost impossible to break ZRSA algorithm due to the way its secret keys are calculated.

REFERENCES

[1]. Delfs, H. &. K. H., 2007. *Introduction to cryptography: principles and applications*. Canada: Springer. .

[2]. Ferguson, N. & Schneier, B., 2003. *Introduction to cryptography: principles and applications*. NY: Wiley.

[3]. Diffie, W. & Hellman, M., 1976. *New Directions in Cryptography*. *IEEE TRANSACTIONS ON INFORMATION THEORY,* VOL. IT-22(6), pp. 644-654.

[4]. Li, Y., Z, Y. & Nui, W., 2010. *A Method of Privacy Preserving in Mobile Wireless Environments.* s.l., 7th International Conference on In Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing (UIC/ATC), IEEE.

[5]. Ferguson, N. & Schneier, B., 2003. *Introduction to cryptography: principles and applications*. NY: Wiley.

[6]. Patidar, R. a. B. R., 2013. *Modified RSA Cryptosystem Based on Offline Storage and Prime Number*. *2013 IEEE International Conference on Computational Intelligence and Computing Research.*

[7]. Al-Hamami, A. H. &. A. I. A., 2012. *Enhanced Method for RSA Cryptosystem Algorithm*. *2012 International Conference on Advanced Computer Science Applications and Technologies,* pp. 402-408.

Society for Science and Education
United Kingdom

TNC Transactions on Networks and Communications

Volume 5, No. 4
ISSN: 2054 -7420

Instilling QoS in Wireless Sensor Networks

[1]Kamil Samara, [2]Hossein Hosseini
[1]School Department of Computer Science, Illinois Wesleyan University, United States
[2]Department of Electrical Engineering and Computer Science, University of Wisconsin - Milwaukee, United States;
ksamara@iwu.edu; hosseini@uwm.edu

ABSTRACT

Wireless Sensor Networks (WSNs) have been a desired choice for monitoring and automatic control of remote and unreachable objects and environments due to their low cost. However, such deployment requires quality-of service (QoS) techniques to assure reliable performance. Furthermore, provision of QoS in WSNs is a challenging task due to hardware limitations. A cross-layer approach is a promising option where information from different layers can be used to make QoS decisions. In this paper, we present a routing protocol where information from the application layer is used to make differentiated routing decisions based on data packets classifications. In our case, data packets are classified into: normal, urgent, and critical. Based on this classification each data packet class is treated differently by storing each data packet class in a designated buffer. Different buffers will have different routing priority decided by the protocol designer.

Keywords: wireless sensor networks; cross-layer optimization; QoS; routing.

1 Introduction

Recent advances in technology especially in electronics and communications allowed the emerge of WSNs. A WSN is a collection of sensor nodes.

Sensor nodes have sensing capabilities which make them a suitable solution for sensing and collection data from different environments [1].

Sensor nodes are low-cost nodes with limited computing power, scarce memory, low bandwidth, and most importantly limited energy source. Sensor nodes are usually operated by batteries, which in most cases are not rechargeable or easy to replace [2].

WSNs have a wide range of applications in different fields, but the most common use for WSNs is monitoring. The WSN is usually deployed over a region or structure to monitor one or more phenomenon then reporting the reading collected, by its sensors, to the base station which can convey the aggregated data to a human operator or a central controller for further processing.

A WSN is a collection of hundreds or thousands of wireless sensor nodes that are often deployed in remote areas as shown in Fig. 1, whose job is to collect data wirelessly and deliver it to a base station. Each node contains a sensing component, processor, communication, and storage components [3].

DOI: 10.14738/tnc.54.3544
Publication Date: 13th Aug 2017
URL: http://dx.doi.org/10.14738/tnc.54.3544

Some common examples of WSNs applications are military, where sensor nodes are used to detect enemy movement, traffic control, where sensor nodes collect data about car jams, or in healthcare sector for monitoring patient's conditions, and in many more applications [1].

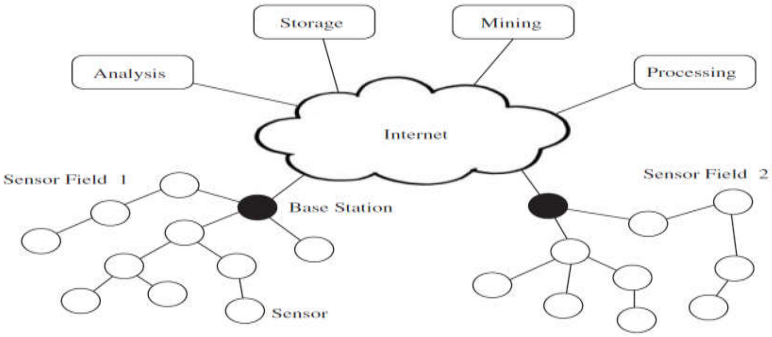


Figure 1 Wireless Sensor Network

Besides resource limitations in WSN nodes, WSNs are usually deployed in unattended and harsh environments implementing crucial applications. These factors emphasize the importance of QoS in WSNs [4].

The special nature of WSNs, mainly the limited energy source in addition to low computation and memory capabilities, makes maintaining QoS through Integrated Services approach unapplicable in WSNs. WSN nodes do not have sufficient resources to establish end-to-end flow connections and manage the information needed for these connections [3].

This work is an extension of our earlier work presented in [5] where all traffic was treated the same without any kind of classification.

In this paper, we are proposing a protocol that provides QoS features through implementing differentiated services, where packets are classified as critical, urgent, and normal. Based on this classification, different packets are assigned different priorities and resources. This results in a more reliable delivery and less delay for urgent and critical packets.

The rest of the paper is organized as follows: In section II we will give a general background about QoS at the network layer. Section III will briefly touch on some of the approaches for achieving QoS in WSNs. Our proposed protocol is described in section IV. Section V is dedicated to simulation results and finally conclusions are presented in section VI.

2 QoS Background

QoS is the overall performance experienced by users when using a networking system. To be able to measure the quality of service, several characteristics are usually measured, such as error rates, bit rate, throughput, delay, availability, and more [6].

Approaches to QoS can be categorized into two main architectures, Integrated Services (IntServ), and Differentiated Services (DiffServ). Differentiated Services is a coarse-grained QoS system. DiffServ provides QoS by classifying network traffic and providing different service according to the packet traffic class [6].

Modern networks carry different classes of data, including voice, video, and text. Each class has its own QoS needs.

DiffServ classifies packets and then routers on the path from the sender to the receiver to implement a per-hop behavior that manages each traffic class differently preferring higher-priority packets [6].

Considering the nature of WSNs, the IntServ approach is not applicable to WSNs. WSN nodes do not have sufficient resources to establish end-to-end connections and manage the information needed for these connections.

Although the QoS requirements differ in WSNs according to the network application, WSN nodes work collectively to achieve the application goals that make the DiffServ a better option to use with WSNs [4].

The most important points to be considered when designing a QoS system for WSNs are as follows:

- QoS must be integrated into all the network layers
- QoS parameters must be decided based on the WSN application
- Resources constraints must be considered

3 QoS Approaches

Routing protocols that ensure shorter paths can increase QoS in the network, because as shown in [7] each node increase on the path between the source and the destination increases the average packet loss ratio by approximately 5-10 %.

Network layer can increase reliability, which is an important QoS aspect, by enforcing multipath routing. WSNs usually have high node density, so the possibility of having more than one path between the source and the destination is high. According to [8 - 10] the delivery ratio on a 14-hop path can be increased from 50% to 75% if there is a second disjoint path.

RAP [11], and SPEED [12] all use geographic forwarding (GF), in which nodes forward packets to their one-hop neighbor that is closer to the sink. This will ensure faster delivery and shorter paths. Multipath Multi-SPEED (MMSPEED) [13] also uses GF but adds the feature of multipath.

JiTS (Just-in-Time Scheduling) [14] is a network layer protocol for soft real-time packet delivery. JiTS orders packets in a forwarding queue based on their transmission time. Transmission time is calculated by multiplying the average one-hop delay by the number of hops. When a packet’s transmission time is reached, it is dequeued from the queue head.

4 Proposed Protocol

This section will be dedicated to describing the design of our proposed protocol.

Routing decisions are made by collecting data about all the available paths from the sensing node to the sink and then use these collected data to decide the best path according to our criteria.

The collected data are total energy on the path, number of nodes on the path (Hop Count), and the lowest energy level of a node on the path to be able to identify critical nodes. Critical nodes are nodes which have been used more than others due to their location, which results in energy drainage.

In our protocol, we classify packets as critical, urgent, and normal as shown Fig. 2. Urgent packets are packets that need to be delivered as fast as possible like real time video packets so that no delay in video streaming is caused. Critical packets are packets that hold sensitive data where reliable delivery is very crucial to the user like enemy movement and data may not be reproducible once is lost. Normal packets are just regular traffic.

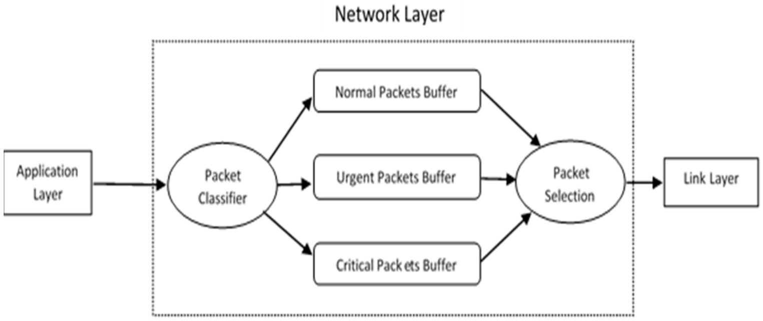


Figure 2 Packets Classification

After the routing paths are established, packets are forwarded per their class. In our protocol, urgent packets are given the highest priority so they are forwarded first, then critical packets, and lastly normal packets. This will result in less delivery delay and highest throughput for urgent and critical packets.

To ensure reliable delivery of critical packets, fault-tolerance is used where sink node will enforce two paths instead of one to deliver critical packets.

The steps of our proposed routing algorithm are shown in Fig. 4 and are described below:

4.1 Interests Propagation

An interest (also called query) is a packet generated by the application layer based on a data request submitted by the network operator. An interest packet will hold the requested data and data class. Data class is decided by the operator and it will be either normal, urgent, or critical.

Interests are flooded through the sensor network. For each active task, the sink will broadcast an interest message shown in Fig. 3 to all its neighbors. Each node that receives an interest message will also broadcast it to all its neighbors.

Query ID	Source Address	Destination Address	Time to Live	Requested Data	Data Class

Figure 3 Interest Packet

Every node maintains an interest cache. Each item in the cache corresponds to a distinct interest. Two interests are distinguished by the ID field.

Interest entries in the cache do not contain information about the sink, but just about the immediately previous hop. Also identical interests are aggregated into a single entry.

When a node receives an interest, it checks to see if the interest exists in the cache. If no matching entry exists the node creates an interest entry. This entry has a single gradient (a gradient specifies a direction in which to send events) pointing toward the neighbor from which the interest was received. If an interest entry does exists, but no gradient for the sender of the interest, the node adds a gradient with the specified value.

Finally, if both an entry and a gradient do exist, the node simply updates the attribute fields if they are different. When an interest's entry has expired, the interest's entry is removed from the interests' cache. Not all received interests are resent. A node may suppress a received interest if it recently resent a matching interest.

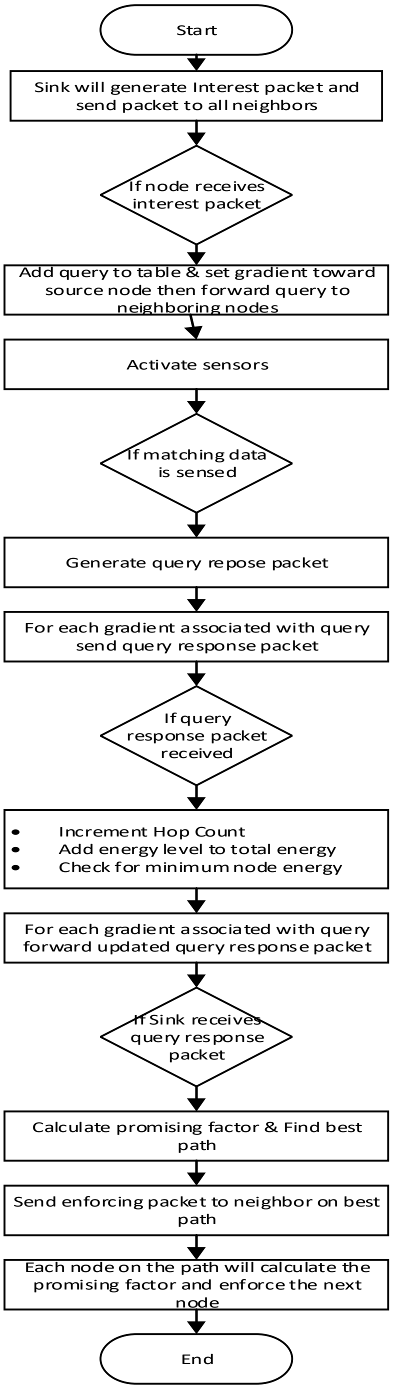

Start
Sink will generate Interest packet and send packet to all neighbors
If node receives interest packet
Add query to table & set gradient toward source node then forward query to neighboring nodes
Activate sensors
If matching data is sensed
Generate query repose packet
For each gradient associated with query send query response packet
If query response packet received
Increment Hop Count
Add energy level to total energy
Check for minimum node energy
For each gradient associated with query forward updated query response packet
If Sink receives query response packet
Calculate promising factor & Find best path
Send enforcing packet to neighbor on best path
Each node on the path will calculate the promising factor and enforce the next node
End

Figure 4 Flow chart of algorithm steps

4.2 Query Response

This stage is which allows us to collect the needed data about each available path and then use these data in enforcing the best path according to our criteria.

As shown in Fig. 5, we are collecting total energy on the path, the number of nodes on the path (Hop Count), and the lowest energy level of a node on the path to be able to identify critical nodes. The set of collected data could be different from one application to another according to what is important to the application in use.

For example, an application that is concerned about fast delivery could collect data about nodes' buffer size to avoid congested nodes.

When a node has at least one active interest, the node will switch on its sensors and start sensing for the requested data.

If the sensing node senses data that matches the requested data by the interest, it will generate a Query Response Packet and send a copy of it to all the gradients associated with the interest.

The base station will receive the Query Response Packet through multiple paths. Then the base station can choose the shortest path and reinforce the source sensors to use the chosen path by using enforcing packets.

Forwarding nodes, on the other hand, could receive the same Query Response Packet flooded by the sensing node from multiple neighbors, but it will only forward one of them.

Query ID	Source Address	Destination Address	Hop Count	Total Energy	Lowest Energy

Figure 5 Query Response Packet

4.3 Reinforcing Paths

When the base station starts receiving Query Response Packets in the reply of an interest that was propagated earlier, it will receive the packets through multiple paths. This is due to the source node sending the Query Response Packet to all the nodes from which it received the interest propagation packet.

Each Query Response Packet received by the base station will hold hop count, total energy, and lowest energy fields about the path it took. Based on that information the base station will choose the best path.

Fig. 6 provides an illustration of the Interest propagation, gradients setup, and data delivery stages.

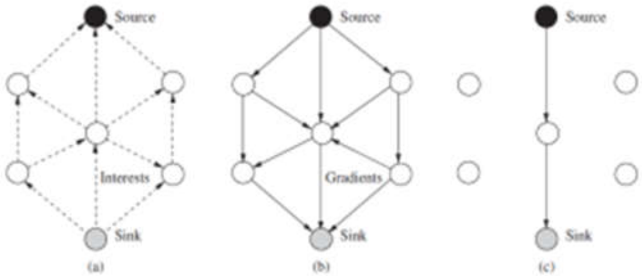


Figure 6 (a)Interest propagation (b)gradients setup (c)data delivery

Choosing the best path is done by calculating the promising factor (PF) for each path from the source (sensing node) to the destination (sink node) using the following formula:

$$\mathrm{PF\ s \rightarrow d} = \frac{\alpha(\mathrm{TE}) * \beta(\mathrm{LE})}{\gamma(\mathrm{HC})} \tag{1}$$

Where:

TE: Path Total Energy Ratio

LE: Path Lowest Energy Level Ratio

HC: Path Hop Count Ratio

α, β, ϒ : are the weights given to each factor in the equation (default value for all weight is 1)

The term TE in equation (1) will basically give preference to paths with higher energy, which is an indicator of the health level of a path. The LE will help us avoid critical nodes, which has been used more than others and their energy level dropped significantly. The term HC will give preference to shorter paths over longer ones which will result in faster delivery.

The Path Total Energy Ratio (TE) is calculated using the following formula:

$$\mathrm{TE\ for\ path\ s \rightarrow d} = \frac{Path\ s{\rightarrow}d\ Total\ Energy}{\sum_{All\ Paths\ s\rightarrow d} Total\ Energy} * 100 \tag{2}$$

Where:

$$Path\ s \rightarrow d\ Total\ Energy = \sum_{All\ node\ on\ Path\ s\rightarrow d} Node\ Energy\ Level \tag{3}$$

The Path Lowest Energy Level Ratio (LE) is calculated the following formula:

$$\mathrm{LE\ for\ Path\ s \rightarrow d} = \frac{Path\ s{\rightarrow}d\ Lowest\ Energy\ Level}{\sum_{All\ Paths\ s\rightarrow d} Lowest\ Energy\ Level} * 100 \tag{4}$$

Where:

$$Path\ s \rightarrow d\ Lowest\ Energy\ Level = \min_{All\ nodes\ on\ Path\ s\rightarrow d} (Node\ Energy\ Level) \tag{5}$$

The Path Hop Count Ratio (HC) is calculated the following formula:

$$\mathrm{HC\ for\ Path\ s \rightarrow d}\ = \frac{Path\ s{\rightarrow}d\ Hop\ Count}{\sum_{All\ Paths\ s\rightarrow d} Hop\ Count} * 100 \tag{6}$$

Where Hop Count is the number of nodes on path s → d

After choosing the best path, the base station will send an enforcing packet to the neighboring node that forwarded the Query Response Packet which resulted in the highest promising factor. In turn each forwarding node on the path of the enforcing packet will forward the enforcing packet to the node it received the Query Response Packet from. The forwarding node will use the interests table to know which node to forward the enforcing packet to.

In case of critical data, base station will enforce two paths instead of one by sending send an enforcing packet to two neighboring nodes with the two highest promising factors.

4.4 Data Propagation

After the reinforcing phase is done, the source nodes know which neighboring node to use to forward the data packets. Every time it senses a matching data to the interest requested data it will generate a data packet and forward the data packet towards the base station using the enforced node.

In addition, priority queueing is used to send packets, where every node along the path will store the received packet in its designated queue according to its priority. Then, packets in the urgent buffer are forwarded first, if there are no packets left in the urgent buffer then packets in critical buffer are forwarded and lastly packets in the normal buffer are forwarded. The forwarding process will continue until the data packet reaches the base station.

This process will continue until the "time to live" field associated with the interest becomes zero. Then this interest will be removed from the table of interests in the source node, and no more data packets will be generated in response to this interest.

5 Simulation and Results

The To evaluate the performance of our protocol we implemented it using Castalia simulator. Castalia simulator is a framework that can be used on top of OMET++ to simulate WSNs, Body Area Networks (BAN) and generally networks of low-power embedded devices. Castalia is an open source simulator which allows researchers to develop and implement their own protocols [15]. Simulation parameters shown in table I.

To compare performance each simulation experiment was conducted twice. Once with differentiated services activated and second without differentiated services being activated; we call the second option single service because all packets' classes are provided with the same service (stored in the same buffer).

Table 1. Simulation Parameters

Parameter Name	Value
Simulation Time	300 sec
X axis	40 – 180 meters
Y axis	40 – 180 meters
Number of Nodes	25- 256
Sink node	Node 0
Radio Type	CC2420
MAC Protocol	TMAC
% of Normal Packets	60%
% of Urgent Packets	30%
% of Critical Packets	10%

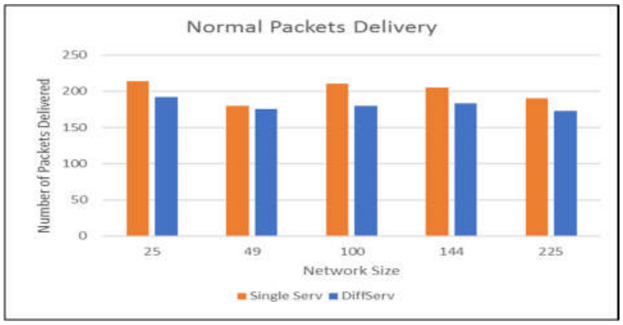
Two simulation experiments were conducted. The first simulation experiment focus was to compare the total number of packets delivered to the sink node. The second simulation experiment focus was to compare the average delay of packets.

5.1 Experiment One: Packets Delivery

The simulation was performed with different number of nodes ranging from 25 to 225 to prove that the same outcome will occur regardless of networks size.

Figures 7, 8, and 9 show the simulation results comparing the total number of delivered packets to the sink in both single service (Single Serv) in orange color and differentiated services (Diffserv) in blue color.

Fig. 7, compares the number of normal packets delivered to the sink in both DiffServ and Single Serv cases. While figures 8 and 9 compare the number urgent and critical packets delivered to the sink node respectively.



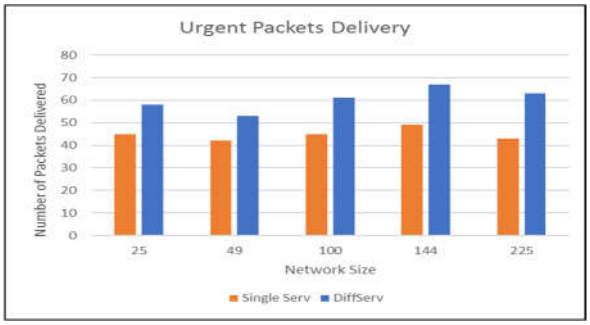


Figure 7 Normal Packets Delivery

Figure 8 Urgent Packets Delivery

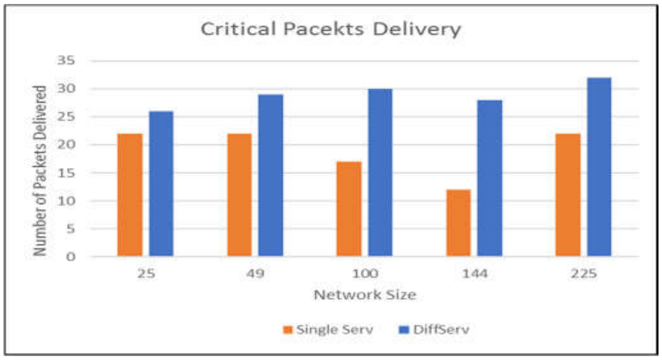


Figure 9 Critical Packets Delivery

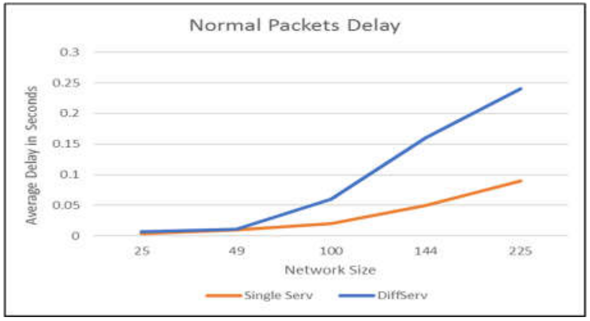
As shown in Fig. 7, the number of normal packets delivered to the sink node using Diffserv decreased because normal packets are being stored in their own queue which has lower priority than urgent and critical queues. On the other hand, figures 8 and 9 show the increase in the total packets delivered for urgent and critical packets when using Diffserv due to buffering these types of packets in separate queues from normal packets with higher priority.

After analyzing figures 7, 8, and 9 we can conclude that using differentiated services improved the packets' delivery rate for urgent and critical packets over the expense of normal packets.

5.2 Experiment Two: Average Delivery Delay

The simulation was performed with different number of nodes ranging from 25 to 225 to prove that the same outcome will occur regardless of networks size.

Figures 10, 11, and 12 show the simulation results comparing the average delivery delay in both single service (Single Serv) in orange color and differentiated services (Diffserv) in blue color.



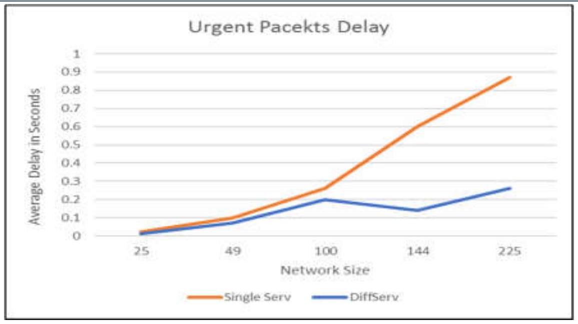


Figure 10 Normal Packets Average Delay

Figure 11 Urgent Packets Average Delay

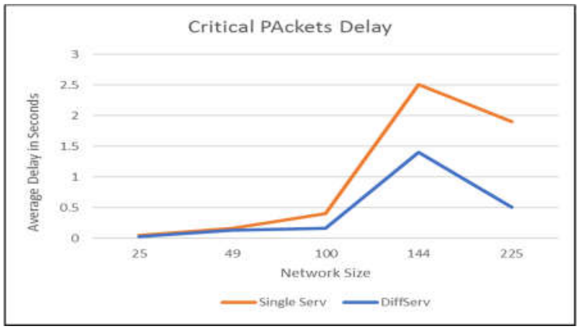


Figure 12 Critical Packets Average Delay

Fig. 10, compares the average delivery delay of normal packets delivered to the sink in both in both DiffServ and Single Serv cases. While figures 11 and 12 compare the average delivery delay of urgent and critical packets delivered to the sink node respectively.

As shown in Fig. 10, the average delivery delay of normal packets delivered to the sink node using Diffserv increased because normal packets are being stored in their own queue which has lower priority than urgent and critical queues. On the other hand, figures 11 and 12 show the decrease in the average delivery delay for urgent and critical packets when using Diffserv due to buffering these types of packets in separate queues from normal packets with higher priority.

After analyzing figures 10, 11, and 12 we can conclude that using differentiated services improved the packets' average delivery delay for urgent and critical packets over the expense of normal packets. We also, can notice how the improvement tends to get higher as the network size gets bigger. This is due to longer paths being used in larger networks.

6 Conclusions

QoS is a crucial feature in WSNs to ensure predictable performance in harsh environments. In this paper, we presented a routing protocol utilizing cross-layer communication where information from the application layer is used to take differentiated routing decisions by the network layer based on data packets classification. In our case, data packets are classified into: normal, urgent, and critical. Each data class is stored in its own designated routing buffer. Urgent packets buffer has the highest priority for transmission, then critical packets buffer, and lastly the normal packets buffer.

The performance of the differentiated services model was compared with the single service model through simulation experiments. Two experiments were conducted. The first experiment compared packets delivery, the second experiment compared packets' average delivery delay. From these experimental results, it was shown that the differentiated services model performed better than the single service model in all aspects by increasing the packets delivery for urgent and critical packets and decreasing their delivery delays.

REFERENCES

[1] Nikolaos A. Pantazis, S.A.N.a.D.D.V., "Energy-Efficient Routing Protocols in WirelessSensor Networks: A Survey". Communications Surveys & Tutorials, IEEE 2012. 15(2): p. 551 - 591.

[2] Md. Atiqur Rahman, S.A., Md. Ileas Pramanik, Md. Ferdous Rahman. A "Survey on Energy Efficient Routing Techniques in Wireless Sensor Networks". ICACT, 2013. p. 200 - 205.

[3] Dr. Geoff V Merret, D.Y.K.T., "Wireless Sensor Networks: Application-Centric Design". InTech, 2010.

[4] B. Bhuyan, H. Sarma, N. Sarma, A. Kar, R. Mall, *Quality of Service (QoS) Provisions in Wireless Sensor Networks and Related Challenges*, Wireless Sensor Network, 2010, Vol. 2, pp. 861-868

[5] Kamil Samara, Hossein Hosseini, "Aware Diffusion: A Semi-Holistic Routing Protocol for Wireless Sensor Networks", Wireless Sensor Network Journal, Vol.8 No.3, March 2016, pp. 37-49

[6] J. Mbowe, G. Oreku, Quality of Service in Wireless Sensor Networks, Wireless Sensor Network, 2014, 6, pp. 19-26

[7] N. Ota, D. Hooks, P.Wright, D. Auslander, and T. Peffer., Application to Demand Response Energy Pricing, Proceedings of the First international conference on Embedded Networked Sensor Systems, November 05–07, 2003

[8] E. Felemban, C. Lee, and E. Ekici. MMSPEED: Multipath multi-SPEED protocol for QoS guarantee of reliability and timeliness in wireless sensor networks, IEEE Transitions on Mobile Computing, vol. 5, no. 6, pp. 738–754, 2006

[9] D. Ganesan et al. Highly resilient, energy efficient multipath routing in wireless sensor networks, Mobile Computing and Communications Review, vol. 5, no. 4, pp. 11–25, 2002

[10] X. Huang and Y. Fang. Multiconstrained QoS multipath routing in wireless sensor networks, Wireless Networks Journal, vol. 14, no. 4, pp. 465–478, 2007

[11]C. Lu et al., RAP: A Real-Time Communication Architecture for Large-Scale Wireless Sensor Networks, In Proceedings of the Eighth Real-Time and Embedded Technology and Applications Symposium, IEEE CS Press, Los Alamitos, CA, 2002, pp. 55 - 66

[12] T. He, J.A. Stankovic, C. Lu, and T.F. Abdelzaher. SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks, In Proceedings of International Conference on Distributed Computing Systems (ICDCS '03), Providence, RI, May 2003, pp. 46 – 55

[13] E. Felemban, C. Lee, and E. Ekici. MMSPEED: Multipath multi-SPEED protocol for QoS guarantee of reliability and timeliness in wireless sensor networks, IEEE Transitions on Mobile Computing, vol. 5, no. 6, pp. 738–754, 2006

[14] K. Liu, N. Abu-Ghazaleh, and K.D. Kang. JiTS: Just-in-Time Scheduling for Real-Time Sensor Data Dissemination. In Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM'06),Washington, DC, IEEE Computer Society, Silver Spring, MD, 2006, pp. 42–46

[15] Boulis, "Castalia user's manual", NICTA.

Society for Science and Education
United Kingdom

TNC Transactions on Networks and Communications

Volume 5, No. 4
ISSN: 2054 -7420

Advances in Artificial Intelligence
Are you sure, we are on the right track?

Emanuel Diamant
VIDIA-mant, Israel
emanl.245@gmail.com

ABSTRACT

Over the past decade, AI has made a remarkable progress. It is agreed that this is due to the recently revived Deep Learning technology. Deep Learning enables to process large amounts of data using simplified neuron networks that simulate the way in which the brain works. However, there is a different point of view, which posits that the brain is processing information, not data. This unresolved duality hampered AI progress for years. In this paper, I propose a notion of Integrated information that hopefully will resolve the problem. I consider integrated information as a coupling between two separate entities – physical information (that implies data processing) and semantic information (that provides physical information interpretation). In this regard, intelligence becomes a product of information processing. Extending further this line of thinking, it can be said that information processing does not require more a human brain for its implementation. Indeed, bacteria and amoebas exhibit intelligent behavior without any sign of a brain. That dramatically removes the need for AI systems to emulate the human brain complexity! The paper tries to explore this shift in AI systems design philosophy.

Keywords: Intelligence as data processing, Computational intelligence, Deep learning, Intelligence as information processing, Cognitive intelligence, Brainless intelligence.

1 Introduction

There is now a broad consensus that AI research is making impressive advances in many fields of applications. Reports about dramatic achievements reached in the last years are updated persistently, [1], [2]. The most recent one “2014 in Computing: Breakthroughs in Artificial Intelligence”, [2], published by MIT Technology Review, summaries the most important achievements of the past year**.** Some of them are worth to be mentioned:

Facebook’s new AI research group reports a major improvement in face-processing software; a technique called deep learning could help Facebook understand its users and their data better.

Researchers at Google have created brain-inspired software that can use complete sentences to accurately describe scenes shown in photos—a significant advance in the field of computer vision.

DOI: 10.14738/tnc.54.3562
Publication Date: 16th Aug 2017
URL: http://dx.doi.org/10.14738/tnc.54.3562

The search company Baidu, nicknamed "China's Google," also spent big on artificial intelligence. It set up a laboratory in Silicon Valley to expand its existing research into deep learning, and to compete with Google and others for talent.

After IBM's Watson defeat over champions of the sophisticated language game Jeopardy, IBM is now close to make a version of Watson's software help cancer doctors use genomic data to choose personalized treatment plans for patients.

A special part of MIT's review is devoted to a machine learning startup called DeepMind (recently purchased by Google for more than $600 million). DeepMind seeks to build artificial intelligence software that can learn when faced with almost any problem. A computer system developed by DeepMind has learned to play seven Atari video games, six better than previous computer systems and three better than human experts, with no foreknowledge of the games other than knowing that the goal is to maximize its score. To reach their goals, the DeepMind scientists built on and improved a set of techniques known as Deep learning.

According to Wikipedia's definition, Deep learning is a set of algorithms in machine learning that attempt to model high-level abstractions in data by using model architectures composed of multiple non-linear transformations, [3]. Deep learning software works by filtering data through a hierarchical, multilayered network of simulated neurons that are individually simple but can exhibit complex behavior when linked together.

Almost all mentioned above (in the MIT review) companies base their research on such or another version of Deep learning technique. However, only DeepMind proclaims that their mission is to "solve intelligence", [4]. Such a strong-minded statement looks a bit strange keeping in mind that the term "intelligence" is still not really defined in AI research.

Shane Legg, one of the DeepMind founders, wrote on this issue: "A fundamental problem in strong artificial intelligence is the lack of a clear and precise definition of intelligence itself. This makes it difficult to study the theoretical or empirical aspects of broadly intelligent machines", [5].

In a more earlier publication, "A Collection of Definitions of Intelligence", [6], Legg and Hutter have assembled a list of 70+ different definitions of intelligence proposed by various artificial intelligence researchers. There is no consensus among the items on the list. Such inconsistency and multiplicity of definitions is an unmistakable sign of philosophical immaturity and a lack of a will to keep the needed grade of universality and generalization, [7].

You can agree or disagree with the Ben Goertzel's statement about "philosophical immaturity and a lack of a will" in AI research, [7], but you cannot escape the feeling that without a clear understanding about what is "intelligence" any talks about impressive advancement toward it (DeepMind: our mission is to solve intelligence!, [4]) are simply groundless.

Bearing in mind that AI research is directed towards human intelligence simulation, it seems quite reasonable to inquire into human life sciences for suitable references for such a case. It turns out that human life sciences are coping with the same problem – they too see Intelligence as an undefinable entity. In fact, the polemic has divided the life science community for decades and controversies still rage over its exact definition.

To legitimate my assertion, I would like to draw readers' attention to three most authoritative publications in the field of human intelligence: A "Survey of expert opinion on intelligence" (1987), [8];

A "Report of debates at the meeting of the Board of Scientific Affairs of the American Psychological Association" (1996), [9]; and a review of "Contemporary Theories of Intelligence" (2013), [10].

There is a widely shared opinion that human intelligence cannot be defined as a single trait (Spearman's view (1904) on intelligence as a General Intelligence). Theories of Multiple Intelligences are steadily gaining mainstream attention, [10]. Intelligence becomes an umbrella term that embraces (integrates) a multitude of cognitive capabilities (to sense, to perceive and to interpret the surrounding environment; to recognize, to categorize, to generate plans and to solve problems; to predict future situations, to make decisions and select among alternatives; to learn, to memorize, and so on), which all together produce the effect of intelligence.

Another commonly shared conception is that human cognitive capabilities are all a product of human brain activity, [11]. More precisely, a product of the brain's information processing activity, [12].

If you get an impression that you are near the end of the search and a feel of relief and accomplishment had washed over you, do not lie down and relax – our journey is still not over. Does anybody know "What is information?" No, nobody knows what it is. (Despite of the fact that these days it is the most frequently used and widely adopted word).

The notion of "Information" was first introduced by Claude Shannon in his seminal paper "A Mathematical Theory of Communication" in 1948, [15]. Today, Stanford Encyclopedia of Philosophy offers (side by side with Shannon's definition of information) an extended list of other suggestions being considered: Fisher information, Kolmogorov complexity, Quantum Information, Information as a state of an agent, and Semantic Information (once developed by Bar-Hillel and Carnap), [13]. Again, as it was mentioned earlier, multiplicity of definitions is not a sign of well-being. What makes the difference between "intelligence" and "information" is that in the case of "intelligence" we have forced to cope with multiple definitions of multiple types of intelligence while in the case of "information" we have to cope with multiple definitions of a single entity – certainly an encouraging difference.

I think that the Introduction has already went out of bounds and it will be wise to start with the main part of our discussion.

2 What is information?

For the most of my life, I have earned my living as a computer vision professional busy with Remote sensing and Homeland security projects. The main goal of these endeavors was to understand what is going on in the observable field of view. That is, to reveal the information content of an acquired image. How to do this? –Nobody did not knew then. Nobody knows today. The common practice is to perform low-level image processing hoping in some way or another to reach high-level decisions concerned with image objects detection and recognition.

I tried to approach the problem differently. It took me a long time to do this, but at the year 2005, I have published my first definition of information. I will try to share with you my view on the subject, but for the sake of time and space saving I will provide here only a short excerpt from my once published (and mostly unknown) papers. Interested readers are invited to visit my website (http://www.vidia-mant.info), where more of such papers can be found and used for further elaboration of the topics relevant to this discourse.

Contrary to the widespread use of Shannon's Information Theory, my research relies on the Kolmogorov's definition of information, [14]. According to Kolmogorov, the definition of information can be expressed as follows: **"Information is a linguistic description of structures observable in a given data set".**

For the purposes of our discussion, digital image is a proper embodiment of what can be seen as a data set. It is a two-dimensional array of a finite number of elements, each of which has a particular location and value. These elements are regarded to as picture elements (also known as pels or pixels). It is taken for granted that an image is not a random collection of these picture elements, but, as a rule, the pixels are naturally grouped into specific assemblies called pixel clusters or structures. Pixels are grouped in these clusters due to the similarity in their physical properties (e.g., pixels' luminosity, color, brightness and as such). For that reason, I have proposed to call these structures **primary or physical data structures**.

In the eyes of an external observer, the primary data structures are further arranged into more larger and complex assemblies, which I propose to call **secondary data structures**. These secondary structures reflect human observer's view on the composition of primary data structures, and therefore they could be called **meaningful or semantic data structures**. While formation of primary data structures is guided by objective (natural, physical) properties of the data, ensuing formation of secondary structures is a subjective process guided by human habits and customs, mutual agreements and conventions between and among members of an observer group.

As it was already said, **Description of structures observable in a data set should be called "Information".** Following the explained above subdivision of the structures discernible in a given image (in a given data set), two types of information must be distinguished – **Physical Information and Semantic Information**. They are both language-based descriptions; however, physical information can be described with a variety of languages (recall that mathematics is also a language), while semantic information can be described only by using human natural language.

I will drop the explanation how physical and semantic information are interrelated and interact among them. Although that is a very important topic, interested readers would have to go to the website and find there the relevant papers, which explain the topic in more details. Here, I will continue with an overview of the primary points that will facilitate our understanding of the issue.

Every information description is a top-down evolving coarse-to-fine hierarchy of descriptions representing various levels of description complexity (various levels of description details). Physical information hierarchy is located at the lowest level of the semantic hierarchy. The process of sensor data interpretation is reified as a process of physical information extraction from the input data, followed by an attempt to associate the physical information at the input with physical information already retained at the lowest level of a semantic hierarchy. If such association is achieved, the input physical information becomes related (via the physical information retained in the system) with a relevant linguistic term, with a word that places the physical information in the context of a phrase, which provides the semantic interpretation of it. In such a way, the input physical information becomes named with an appropriate linguistic label and framed into a suitable linguistic phrase (and further – in a story, a tale, a narrative), which provides the desired meaning for the input physical information. (Again, more details can be found on the website).

3 Rethinking intelligence

As it follows from the above, intelligence has to be understood through cognition, cognition has to be understood through information processing, information processing has to be understood as physical and semantic information interaction and cooperation.

On neither of these steps, data or data processing are not considered as a part or a basis of an ongoing process. Indeed, data features are meaningless in our world perception (and judgment). We understand the meaning of a written word irrelevant to letters' font size or style. We recognize equally well a portrait of a known person on a huge size advertising billboard, on a magazine front page, or on a postage stamp – perceptive information is dimensionless. We grasp the meaning of a scene irrelevant to its illumination. We look on the old black-and-white photos and we do not perceive the lack of colors.

The same is true for voice perception and spoken utterance understanding – we understand what is being said irrelevantly to who is speaking (a man, women, or a child). Irrelevant to the volume levels of the speech (loudly or as a whisper). Blind people read Brail-style writings irrelevant to the size of the touch-code.

Semantic information processing has nothing to do with raw data and its features – raw data features were dissolved in physical information (which is later processed in the semantic information hierarchy). Despite all of this, the new wave of AI innovations relies totally on data processing. The MIT Technology Review, speaking about breakthroughs in AI in the year 2014, enumerates prevailing Deep Learning based developments (which are all data-processing ventures) and then describes the latest IBM's feat – "IBM Chip Processes Data Similar to the Way Your Brain Does"! (Really? Ask Google: "brain is an information processing" – 39900 hits, compare this with: "brain is a data processing" – 6 hit!).

As it follows from the preceding discussion, semantics is not a property of the data. Semantics is a property of a human observer that watches and scrutinizes the data. Semantic information is shared among the observer and other members of his community (and that is the common basis of their intelligence). By the way, this community does not have to embrace the whole mankind. This can be even a very small community of several people or so, which, nevertheless, were lucky to establish a common view on a particular subject and a common understanding of its meaning. Therefore, this particular (privet) knowledge cannot be acquired in any other way. (By Machine Learning, for example, by Deep Learning, or other tricks). **Semantic information should be only shared or granted**! There is no other way to incorporate it as the system's reference knowledge base (used for processing/interpreting physical information at the system's input). Therefore, common attempts to formalize semantics and to derive it from input data are definitely wrong.

The form in which semantic information has to be reified is a string of words, a piece of text, a story, a narrative. (That follows from semantic information definition already given above). If we accept this assumption, it will be reasonable to suppose that semantic information processing means some sort of language texts processing. (For humans it is, obviously, human natural language texts, but for plants or bacteria it will be a different kind of language – every living being possess its own intelligence reified as its ability to process semantic information, which is reified in some pra- or proto-language). What implications follow from the statement "semantic information processing means language text processing"? – I do not know (at least at this stage of my research). As to my knowledge, nobody else knows about this not more than I. (Despite there is a well-known research field of computational linguistics, however, the domain of its studies does not overlap with semantic information processing).

In this regard, it will be fair to mention that IBM's Watson designers and Kurzweil's group at Google are working hard on enabling computers to understand and even to speak in natural language. The goals look similar to my – to reach human level language texts processing. But there is a great and essential difference – both teams pinned all their hopes on suitable deep learning algorithms development.

Adoption of the idea that intelligence is an information processing (duty) naturally raise a question: Is the human brain the only proper means to facilitate this purpose? The answer is: **obviously not**! Bacteria and amoebas exhibit intelligent behavior (intentional external world interaction), which certainly requires information processing, but without any sign of a brain or a nervous system. The same is right for invertebrates, plants, animals, and even mammals. If that is right (and that is right!), why AI systems have to emulate human brain-based intelligence? For me the answer is straightforward: AI systems have to emulate information-processing abilities, not the complexity of a human brain!

In the same vein, it would be right to introduce a new notion for intelligence – Brainless intelligence. And to challenge AI designers with the goal of brainless intelligent devices development.

The time is right to introduce another novelty in the playground of intelligence notions: Computer is a data processing machine; therefore, data-processing-based intelligence has been once dubbed **Computational intelligence**. The advent of information-processing-based approach requires an appropriate new name. In my opinion, **Cognitive intelligence** will be the most suitable option. (Because cognition is a result of information processing and, at the same time, the footing base of intelligence).

It is worth to mention, that IBM and other leading R&D companies are planning to develop and introduce in the near future a computing device of tomorrow – the Cognitive Computer! Which will be capable to perform human like Big data volumes handling. A contradiction (common to all AI projects) is placed from the beginning in the foundations of the enterprise: "Cognitive" is juxtaposed with "Computing". "Cognitive", which implies information processing, and "Computing", which implies data processing! The two are signs of opposite actions. The two are incompatible! (But who cares!)

4 Conclusions

Artificial Intelligence was invented at the Dartmouth College meeting in the summer of 1956. Four brilliant scientists (J. McCarthy, M.L. Minsky, N. Rochester, and C.E. Shannon) have worked out and put into operation the idea of AI research. Despite of the famous fathers, AI research has never been skilful enough to reach its goals. The fathers have failed to assess the complexity of the task and mutual contradictions between its basic constituents. (Recall the story how Minsky proposed to hire a student to solve the problem of vision during the student's summer vocations).

Other examples are plentiful: It was assumed that the best-known manifestation of intelligence is human intelligence; therefore, AI's aim was defined to replicate human intelligence. It was also assumed that the brain is the core and the basis of intelligence. Both assumptions are incorrect – intelligence is an attribute of all living beings, and the existence of a brain is not obligatory for intelligence (bacteria and amoebas exhibit intelligent behavior without any sign of a brain).

The computational paradigm that emerged in the second half of the past century has been generally accepted as the prevalent paradigm of the contemporary science. For that reason, we have "computational intelligence" as well as "computational biology" or "computational linguistics". The brain has become regarded as a computing device, that is, a device aimed at number crunching and data manipulation. Therefore, even today, all AI algorithms are devised to process and to operate data. All breakthrough achievements of the last time are data-processing implements, (Deep Learning is assumed as the most prominent among them).

At the same time, it is generally accepted that the brain is an information-processing appliance. The contradiction between the two definitions can be explained by the peculiarities of scientific development in the past century. It is generally accepted that Intelligence of a living being is expressed in his behavior, that is, in his interaction and communication with the environment. Claude Shannon (one of the AI founders) in his seminal "Mathematical Theory of Communication" [15], has defined that what is being conveyed in a communication process is Information. Shannon was aware that this definition of information is applicable only to data communication, and has nothing to say about the meaning of the conveyed message. Shannon's Information Theory has become widespread and popular in almost all fields of scientific research, despite its shortfalls in message semantics handling. The results of this deficiency are well recognized (over the all AI history) – they have derailed AI research permanently and forever.

This paper is aimed to help the AI researchers to understand the roots of their permanent failures. I have introduced here a new definition of information that will certainly help to avoid in the future the prior mistakes.

REFERENCES

[1] Jjohn Markoff, The Rapid Advance of Artificial Intelligence, The New York Times Company, October 14, 2013, http://www.nytimes.com/2013/10/15/technology/the-rapid-advance-of-artificial-intelligence.html?pagewanted=all&_r=0

[2] Tom Simonite, 2014 in Computing: Breakthroughs in Artificial Intelligence, MIT Technology Review, December 29, 2014, http://www.technologyreview.com/news/533686/2014-in-computing-breakthroughs-in-artificial-intelligence/

[3] Deep learning, From Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Deep_learning

[4] DeepMind, http://deepmind.com/

[5] Shane Legg and Joel Veness, An Approximation of the Universal Intelligence Measure, http://xxx.tau.ac.il/pdf/1109.5951v1.pdf

[6] Shane Legg and Marcus Hutter, A Collection of Definitions of Intelligence, http://arxiv.org/pdf/0706.3639v1.pdf

[7] Ben Goertzel, Ten Years To The Singularity… CreateSpace Independent Publishing Platform, December 25, 2014, http://goertzel.org/TenYearsToTheSingularity.pdf

[8] M Snyderman and S Rothman, Survey of expert opinion on intelligence and aptitude testing, American Psychologist, 1987, http://kodu.ut.ee/~spihlap/snyderman@rothman.pdf

[9] Ulric Neisser, et al. Intelligence: Knowns and unknowns, American Psychologist, Vol 51(2), Feb 1996, pp. 77-101. http://psycnet.apa.org/?&fa=main.doiLanding&doi=10.1037/0003-066X.51.2.77

[10] James C. Kaufman , Scott Barry Kaufman , and Jonathan A. Plucker. Contemporary Theories of Intelligence, The Oxford Handbook of Cognitive Psychology, 2013. http://scottbarrykaufman.com/wp-content/uploads/2012/10/51_Reisberg_ch51.pdf

[11] Roberto Colom, et al. Human intelligence and brain networks, Dialogues in Clinical Neuroscience, vol. 12, No. 4, 2010. www.ncbi.nlm.nih.gov/pmc/artiwcles/PMC3181994/pdf/DialoguesClinNeurosci-12-489.pdf

[12] Richard E. Mayer, An information processing view of learning and cognition, Handbook of Psychology, vol. 7, Chapter 3, John Wiley & Sons, Inc., 2003. http://t5303.oceanikpsi.org/downloads/handbook_of_psychology_vol_.pdf#page=47

[13] Information. (2012). Stanford Encyclopedia of Philosophy, First published Fri Oct 26, 2012 http://plato.stanford.edu/entries/information/

[14] Kolmogorov, A. (1965). Three approaches to the quantitative definition of information, Problems of Information and Transmission, Vol. 1, No. 1, pp. 1-7, 1965. http://alexander.shen.free.fr/library/Kolmogorov65_Three-Approaches-to-Information.pdf

[15] Shannon, C., Weaver, W. (1949). The Mathematical Theory of Communication, University of Illinois Press, 1949. http://raley.english.ucsb.edu/wp-content/Engl800/Shannon-Weaver.pdf

Society for Science and Education
United Kingdom

TNC Transactions on Networks and Communications

Volume 5, No. 4
ISSN: 2054 -7420

Water Wave Optimization Algorithm based Congestion Control and Quality of Service Improvement in Wireless Sensor Networks

Mukhdeep Singh Manshahia
Punjabi University , Patiala
mukhdeep@gmail.com

ABSTRACT

Many researchers have implemented various machine learning algorithms and verify their results with the existing algorithms to control congestion in Wireless Sensor Networks. The major challenge lies in developing an algorithm which optimizes the value of the objective function on the basis of parameters like network throughput, residual energy and packet loss rate of the nodes in the network. An objective function based on these parameters is proposed in the present work. Water wave optimization algorithm is applied on the objective function and an optimum solution is obtained. The proposed approach is compared with the Congestion Detection and Avoidance algorithm (CODA) and Particle Swarm Optimization Algorithm (PSO). The proposed solution outperforms both algorithms on the basis of various performance parameters.

Keywords: Water wave optimization algorithm, Congestion control, Wireless Sensor Networks

1 Introduction

Wireless sensor networks (WSNs) are the sensors deployed in physical environment which are controlled by a central receiving unit called Base Station. Base Station is responsible for the collection and processing of data collected from various different types of sensors [1, 2]. The sensors deployed in the network are either of the same type or of different types depending on the application [3]. Open-air environments need many sensor nodes to cover a large area whereas fewer sensor nodes are needed in indoor environments in a sensor network [4, 5]. High number sensors are deployed in the industrial environment to enhance coverage, fetch live data and take precise decisions [6]. Lots of potential applications of wireless sensor networks make wireless sensor networks a fast growing market [7, 8].

Network lifespan and congestion are two basic performance issues in sensing networks. The energy of the wireless sensor nodes is spent during transmission of data from sensor to sink, sensing the particular environment and processing of data [9]. Congestion can occur while transmitting the data from sensor to sink node. Network congestion can shortened the lifetime of sensor nodes due to packet loss and retransmissions. Therefore minimizing congestion and the energy consumption is the key demands among the sensor network protocols and algorithms.

DOI: 10.14738/tnc.54.3567
Publication Date: 20th Aug 2017
URL: http://dx.doi.org/10.14738/tnc.54.3567

2 Literature Survey

Motdhare et al. [10] provided a congestion control mechanism for WSN using a mobile sink to avoid congestion and lifetime maximization. They have analyzed the impact of mobile sink in congestion reduction and increasing lifetime of the sensor network by considering various parameters and compared it with static sink. Simulation results have shown that proposed technique is effective for congestion avoidance and better lifetime of WSN.

Antoniou et al. [11] proposed swarm Intelligence inspired congestion control mechanism for WSNs. They have provided congestion avoidance and control mechanism by mimicking the obstacle avoidance behavior of bird flock. Packets are considered as flocks who are trying to avoid the congested regions while moving towards sink. At each node, a packet interacts with the neighboring nodes and each packet got attracted magnetically towards sink. Evaluation results have shown that proposed approach perform better in load balancing, robustness against failing nodes and scalability.

Luha et al. [12] proposed Redundancy Aware Hierarchical Tree Alternative Path (RAHTAP) algorithm to control congestion in WSNs. RAHTAP have managed to control congestion in WSNs using redundancy detection and elimination whenever congestion occurs in the network. Elimination of the duplicate packets at each sensor nodes has increased throughput and efficiency. Simulation results have shown that the proposed method is better in confronting over-burden circumstances in Wireless Sensor Networks.

Wei-qiang et al. [13] proposed penalty function-based optimal congestion control (POCC) algorithm with the notion of link's interference set. Each link transmits the information on congestion state to its interference set periodically. Simulation results have proved that POCC is effective in efficient and fair resource allocation.

Raghunathan et al. [14] presented that the results related to stability and fairness of TCP-AIMD cannot be used in wireless networks due to broadcast nature of wireless networks. Wireless congestion control is complicated issue because of interference like self-interference and inters flow interference. In wireless channel, Packet transmission by neighbouring nodes may not be simultaneously due to interference. Ns-2 based simulation results have indicated oscillatory behaviour.

Zhao et al. [15] proposed energy effective congestion control for wireless ad hoc network. They have used source node rate control to control congestion and intermediate node power allocation to eliminate the bottleneck. Dual decomposition has been proposed to decompose the problem into intermediate node power allocation and source flow control. Numerical simulation results have shown that the proposed approach increases network performance and reduces the energy consumption of nodes.

3 Problem Formulation

A fitness function is considered based on various parameters like network throughput, residual energy and packet loss rate to control congestion in WSN [16, 17]:

$$\boldsymbol{Fitness\ Function}\ (f_j) = \sum_{i=1}^{N} (w_1 * \tau_i + w_2 * (1 - P_{L_i}) + w_3 * E_i + w_4 * d_{j,i}) \qquad (1)$$

Where, i is the iteration which ranges from 1 to N (total number of nodes),

w_1, w_2, w_3 and w_4 are the weights supplied to the algorithm,

τ is the throughput of the network,

P_L is the normalized Packet lost rate in the network

$d_{j,i}$ is the distance between node i and j and

E is the residual energy of each node in the network.

4 Proposed Methodology

In the proposed approach, Water wave optimization algorithm is implemented to solve congestion problem in WSN. Water Wave Optimization algorithm is based on the shallow water wave models for solving optimization problems [18]. It is studied as when a wave travels from deep water to shallow water, its wave height increases and its wavelength decreases and vice versa. Without losing generality, suppose we have a maximization problem with objective function f. In Water Wave Optimization, the solution space X is analogous to the seabed area, and the fitness of a point $x \in X$ is measured inversely by its seabed depth: the shorter the distance to the still water level, the higher will be the fitness $f(x)$. During the problem-solving process, we consider three types of operations on the waves: Propagation, Refraction, and Breaking [18].

4.1 The Water Wave Algorithm

```
$Randomly\ initializes\ a\ population\ P\ of\ n\ waves\ (solutions)$;
$\boldsymbol{while}\ stop\ criterion\ is\ not\ satisfied\ \boldsymbol{do}$
    $\boldsymbol{for\ each\ x} \in \boldsymbol{P\ do}$
        $Propagate\ x\ to\ a\ new\ x'$;
        $\boldsymbol{if}\ f(x') > f(x)\ \boldsymbol{then}$
          $\boldsymbol{if}\ f(x') > f(x^*)\ \boldsymbol{then}$
            $Break\ x'$ ;
            $Update\ x^*\ with\ x'$ ;
          $Replace\ x\ with\ x'$ ;
        $\boldsymbol{else}$
            $Decrease\ x.h\ by\ one$;
         $\boldsymbol{if}\ x.h > 0\ \boldsymbol{then}$
        $Refract\ x\ to\ a\ new\ x'$ ;
    $Update\ the\ wavelengths$;
$\boldsymbol{return\ x^*}$
```

Water wave algorithm is applied to detect and reduce congestion in WSN. The objective is to maximize the throughput of network by optimizing the objective function.

Three types of operations are performed on a wave: Propagation, Refraction and Breaking. In deep water, the depth of the sea or the distance of the seabed from the surface is considered as the residual energy of the node.

This also represents the nodes which are less used or which have larger energy as compared to every other node in the network. So, our route consists of those nodes. The congestion will be high near the high wave i.e. around shallow water and these nodes are considered as congested nodes. So the packet loss rate will be high in these nodes. By putting these values in the objective function, we have tried to find the optimal value of the route and thus try to maximize the throughput of the network.

Figure 2 shows the flow diagram of the proposed approach of using the refraction, breaking and propagation techniques for congestion control in wireless sensor networks.

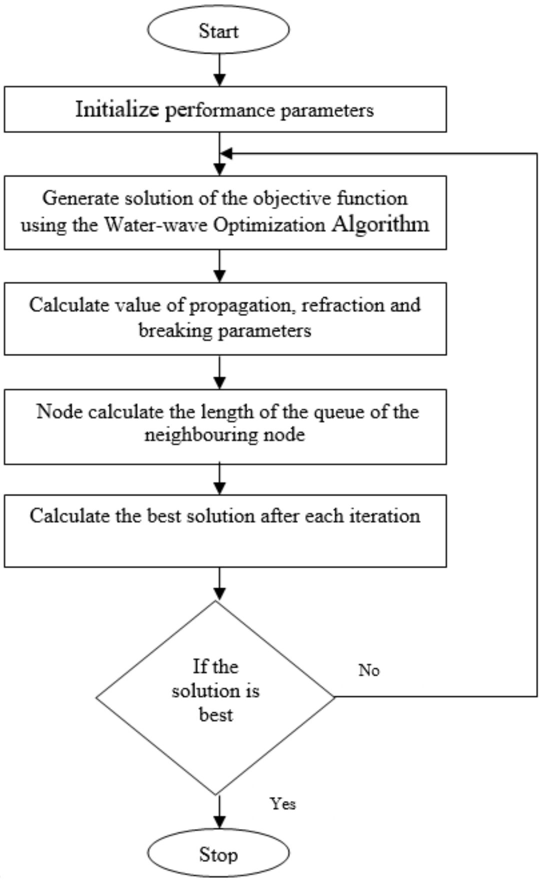

Start
Initialize performance parameters
Generate solution of the objective function using the Water-wave Optimization Algorithm
Calculate value of propagation, refraction and breaking parameters
Node calculate the length of the queue of the neighbouring node
Calculate the best solution after each iteration
If the solution is best
No
Yes
Stop

Figure. 2 Flow Diagram of the Proposed Methodology

We are comparing our proposed approach with Congestion Detection and Avoidance (CODA) [19] and particle swarm optimization (PSO) based congestion control approach [20, 21, 22].

5 Simulation Environment

We have placed 50 nodes in a 1000*1000 size grid using Network Simulator (NS)-2.35. Each node in the network is configured to wireless channel and Omni-directional antenna type. MAC layer is used with standard IEEE 802.11 and the radio model has been adopted [23, 24, 25]. Table 1 shows the simulation parameters used in the simulation.

Table 1: Simulation Parameters

Parameter	Value
No. of Nodes	50
X dimension	1000
Y dimension	1000
Grid Area	1000*1000
Mac Protocol	IEEE 802.11
Propagation Model	Two-Ray Ground model
Transmission Range	200m (approx.)
Source Traffic	512 B
Antenna type	Omni-Antenna
Channel type	Wireless Channel
Routing Protocol	AODV
Simulation time	60
Interface Queue type	Prequeue

6 Results And Discussions

Queue Length of each node in the network is inversely proportional to the number of hops in the network and with increase in number of hops, queue length decreases as shown in figure 3. The packets flowing in the network can adapt alternate routes and thereby decreases the congestion in the network.

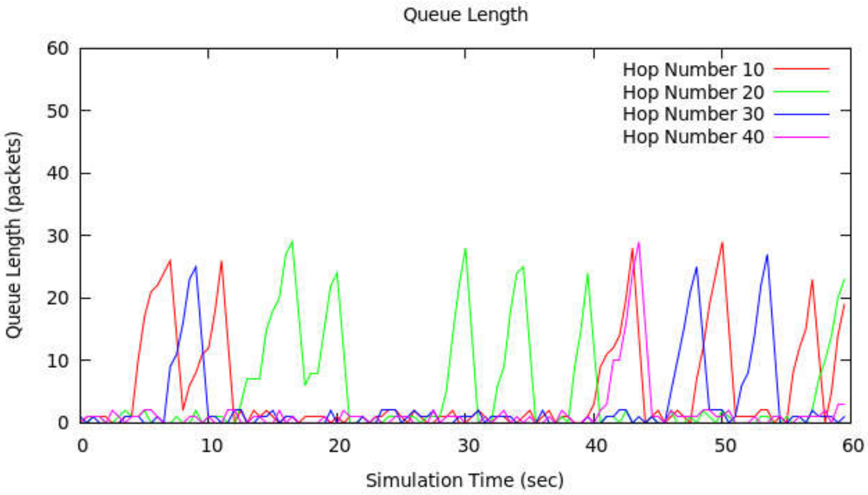


Figure. 3 Graph between Simulation Time and Queue Length for Hop count 10, 20, 30 and 40.

Figure 4 shows that the value of fitness function decreases with increase in number of hops in the network and tends towards its optimal value. The fitness function has various parameters like throughput, packet loss, energy consumed etc.

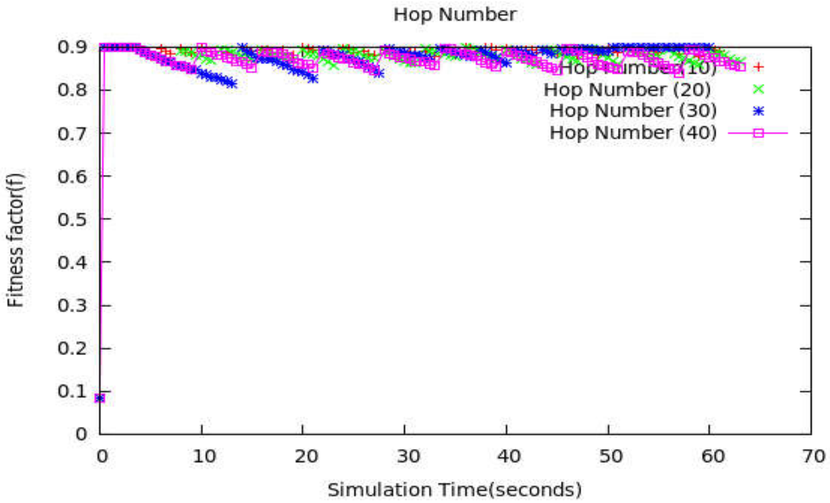


Figure.4 Graph between Simulation Time and Fitness Factor

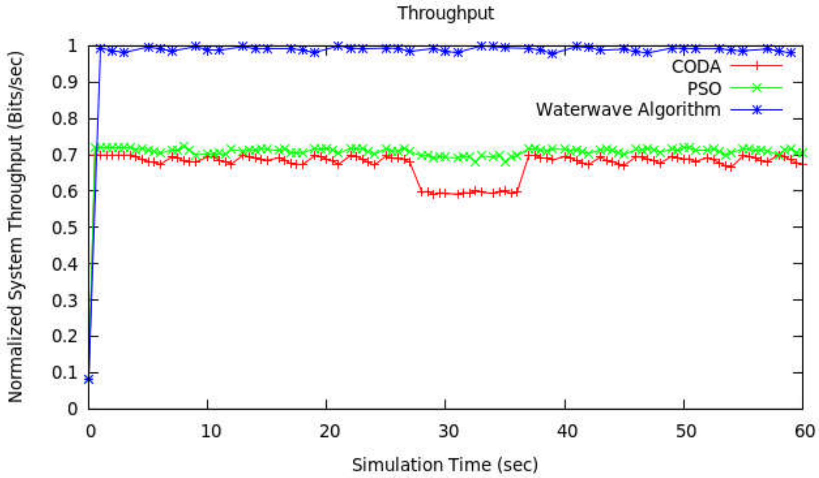


Figure. 5 Throughput of the Network

Figure 5 displays the graph between simulation time and network throughput. The graph compares CODA, PSO and the proposed approach. Water Wave algorithm outperforms the other two algorithms on throughput. Figure 6 displays the graph between the Simulation Time and the Network Lifetime of the nodes in the network. Network Lifetime of nodes in case of water wave optimization algorithm is greater than CODA and PSO because energy consumption of water wave optimization algorithm is low as compared to other algorithms.

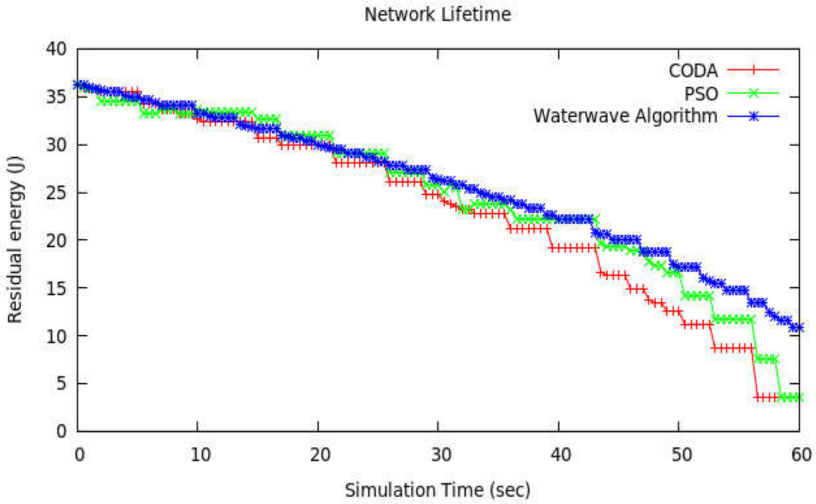


Figure.6 Network Lifetime of Nodes in Network

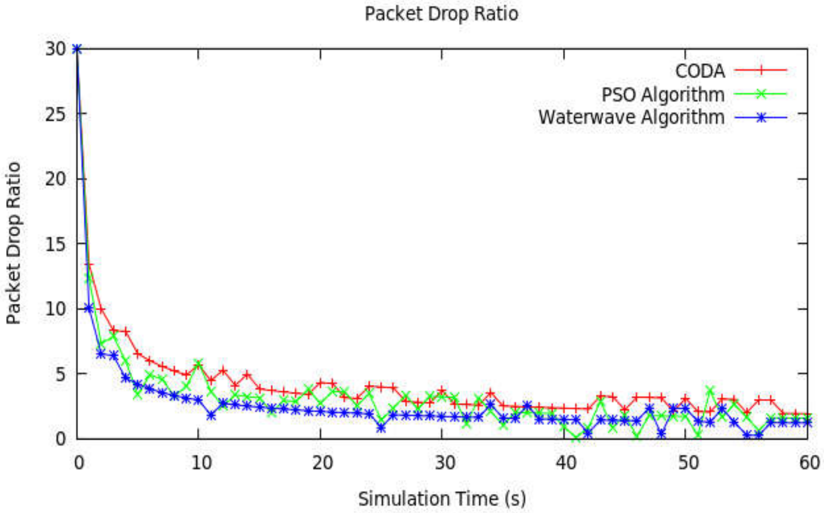


Figure. 7 Packet Drop Ratio of Nodes in Network

Figure 7 shows the packet drop ratio in the network. The water wave optimization algorithm is better as compared to the CODA algorithm and PSO algorithm is performing similar to the water wave algorithm in terms of packets drop ratio. The graphs are designed with hop count 40.

7 Conclusion and Future Scope

The results have proved that after implementation of water wave optimization algorithm, Queue length of each node decreases by increasing the number of hops. This proves that the traffic in the network is distributed among all the nodes in the network rather than concentrating on fewer ones. The network lifetime of nodes in the network is higher as compared to the CODA and PSO. Since the residual energy of each node in case of Water-wave algorithm is higher than CODA and PSO. Throughput of the network is also higher in case of Water-wave algorithm and is almost constant around 1. Packet drop ratio is higher in CODA and remains almost same in case of PSO algorithm as compared to the water wave optimization algorithm. Thus water wave optimization algorithm outperforms all other algorithms discussed in this work on quality of service parameters. In future other computational intelligence techniques will be implemented and compared with the proposed approach.

REFERENCES

[1] Kandris, D., Tsioumas, P.; Tzes, A., Nikolakopoulos, G., & Vergados, D. (2009). Power conservation through energy efficient routing in wireless sensor networks. Sensors, 9, 7320–7342.

[2] Jin, Z., Ping, Y., Wang, Z., Ping, L., & Guang, L. (2009). A survey on position-based routing algorithms in wireless sensor networks. Algorithms, 2, 158–182.

[3] Alam Bhuiyan and et al. , Energy and bandwidth-efficient wireless sensor networks for monitoring high-frequency events, in: Proceeding of 10th Annual IEEE Communications Society Conference on.Sensor, Mesh and Ad Hoc Communications and Networks (SECON), IEEE, 2013, pp. 194 – 202.

[4] Yick, J., Mukherjee, B., & Ghosal, D. (2008). Wireless sensor network survey, Computer Networks, 52(12), 2292–2330.

[5] Rawat, P., Kamal Deep Singh, Hakima C., & Jean Marie Bonnin. (2014). Wireless sensor networks: a survey on recent developments and potential synergies. The Journal of Supercomputing, 68(1), 1-48.

[6] Lo, Shou-Chih, Jhih-Siao Gao, & Chih-Cheng Tseng. (2003). A water-wave broadcast scheme for emergency messages in VANET. Wireless personal communications, 71(1), 217-241.

[7] I.F. Akyildiz, W. Su, ,Y. Sankarasubramaniam, & E. Cayirci. (2002). Wireless sensor networks: a survey. Computer Networks, 38(4), 393–422.

[8] Jennifer Yick, Biswanath Mukherjee, & Dipak Ghosal. (2008), Wireless sensor network survey, Computer Networks, 52(12), 2292–2330.

[9] Stefanos A. Nikolidakis, Dionisis Kandris, Dimitrios D. Vergados, & Christos Douligeris. (2013). Energy Efficient Routing in Wireless Sensor Networks Through Balanced Clustering. Algorithms, 6, 29-42.

[10] S. Motdhare. (2015). Congestion Control in Wireless Sensor Networks: Mobile Sink Approach. International Journal of Science and Research, 4(1), 2561-2565.

[11] P. Antoniou, A. Pitsillides, T. Blackwell, A. Engelbrecht, & L. Michael. (2011). Congestion Control in Wireless Sensor Networks based on Bird Flocking Behavior. Computer Networks, 57(5), 1167–1191.

[12] A.K. Luha, T. Vengattraman, & M. Sathya. (2014). RAHTAP Algorithm for Congestion Control in Wireless Sensor Network. International Journal of Advanced Research in Computer and Communication Engineering, 3(4), 6250-6255.

[13] X. Wei-qiang, & W. Tie-jun. (2006). Optimal congestion control algorithm for ad hoc networks: Penalty function-based approach. Journal of Zhejiang university SCIENCE A, 7(12), 2110-2117.

[14] V. Raghunathan, & P.R. Kumar. (2007). Counterexample in Congestion Control of Wireless Networks. Performance Evaluation, 64(5), 399-418.

[15] C. Zhao, Y. Luo, F. Chen, J. Zhang, & R. Wang. (2014). Energy Effective Congestion Control for Multicast with Network Coding in Wireless Ad Hoc Network. Mathematical Problems in Engineering, 2014.

[16] Manshahia, M.S., Dave, M. and Singh, S.B., Bio Inspired Congestion Control Mechanism for Wireless Sensor Networks, in: Proceedings of 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Madurai, India, December, 2015.

[17] Manshahia, M.S., Dave, M., & Singh, S.B. (2015). Congestion Control in Wireless Sensor Networks Based on Bioluminescent Firefly Behavior. Wireless Sensor Networks, 7, 149-156.

[18] Yu-Jun Zheng. (2015). Water wave optimization: A New Nature-inspired Metaheuristic. Computers & Operations Research, 55, 1–11.

[19] Wan. C.Y, Eisenman. S. B., & Campbell. A. T, CODA: Congestion Detection and Avoidance in Sensor Networks, in: Proceedings of the 1st international conference on Embedded networked sensor systems, SenSys '03 , Los Angeles, 2003, pp.266 - 279 .

[20] Antoniou P., Pitsillides A., Blackwell T., Engelbrecht A., & Michael L. (2013). Congestion Control in Wireless Sensor Networks based on Bird Flocking Behaviour Congestion. Computer Networks, 57(5), 1167-1191.

[21] R. C. Eberhart, J. A Kennedy, New optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro machine and Human Science, Nagoya, Japan, 1995, pp: 39-43.

[22] Kennedy, J., & Eberhart, R.C. (1995). Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, 1995, pp. 1942-1948.

[23] Manshahia, M.S., Dave, M. and Singh, S.B., Firefly algorithm based clustering technique for Wireless Sensor Networks, in: Proceedings of International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 23-25 March 2016.

[24] Akkaya K. & Younis M. (2005). A Survey of Routing Protocols in Wireless Sensor Networks. Ad Hoc Networks, 3(3), 325-349.

[25] Manshahia, M.S., Dave, M., & Singh, S.B.(2016). Improved Bat Algorithm Based Energy Efficient Congestion Control Scheme for Wireless Sensor Networks. Wireless Sensor Network, 8, 229- 241.

Society for Science and Education
United Kingdom

TNC Transactions on Networks and Communications

Volume 5, No. 4
ISSN: 2054 -7420

Setting an Optimization Problem of Distributing Traffic Across Multiple Independent Paths

Mohammad Alhihi
Philadelphia University, Amman, Jordan
Malhihi@philadelphia.edu.jo,alhihimohammad25@gmail.com

ABSTRACT

Communication network development is considered as an urgent need for the world. The proposed work in this paper is to solve the problem of the traffic engineering, using the multi-path routing mechanisms by applying a developed algorithm includes a multi-criteria optimization procedure to define the optimal number of paths. The values of the maximum flow and multi-path delay are used as partial criteria; however, a set of particular criteria can be extended when needed to ensure the required quality of service.

Keywords: Routing Protocols, QoS, TN, hybrid network, traffic engineering, traffic distribution, path delay, the Dijkstra's algorithm

1 Introduction

The paper presents the optimization procedure providing traffic distribution in a critical mode of network operation based on [1-15]. In [2-4], network coding technique was applied to obtain the optimum transmission algorithm based on a minimum number of transmitted packets and hence obtaining the shortest path to be shorter than the traditional optimization algorithm.

The benefit of the proposed procedure is implementing the recursive space-time operations of information flows redistribution, resulting to ensure the network functioning stability, accordingly, the network reaches a state where the traffic is approximately evenly distributed over the corresponding routes, after which the normal dynamic routing procedure is started. The analysis of the developed method shows the existence of a limited number of the most significant routes, which depends on the connectivity of the network, and the amount of traffic that allows maximizing the compound index of the quality of service (QoS) such as in [5-8].

2 Optimization of Traffic Distribution

The single-product flow distribution between network node *s* to node *t* across *m* independent shortest paths is considered in this work. This problem is a special case of the multi-path routing problem considered in the literature [7-17]. The algorithm for solving such a generalized problem has a large computational complexity and the time of solution of such a problem cannot meet the requirements for the means of multipath routing. These requirements are applied to the routing algorithms traditional requirements, and its low computational complexity, rapid convergence and minimum volumes of created

DOI: 10.14738/tnc.54.3346
Publication Date: 20th Aug 2017
URL: http://dx.doi.org/10.14738/tnc.54.3346

service traffic were proposed [7-9]. In addition, the conversion factor *q*, which determines the degree of filling of the arc, is determined only for a particular case. There are no general assumptions on the choice of this factor, which narrows the possibilities of practical use of this generalized model.

The notation system adopted in streaming models is used, where the network structure *G(N,E)* is determined by the set of nodes *N={1,2…n}* and set of arcs $E \subset (N \times N)$. Let the arc flow $i_k, (k = 1, cardE)$ be f_k . For each arc *k*, a value of c_k is assigned to define the upper boundary of the flow along the arc *k*. Let the set of independent shortest paths from noted *s* to node *t*, ($s,t \in N$) to be P_n, where $P \in G$.

The flow Q specified value between a minimal cut (correct section) C^* of the fragment *P* of the network *G* satisfies the following condition in (1):

$$Q < C^*, \text{ Where } C^* = \sum_{i=1}^{m} \min\{c_j\},\ j = 1, r_j, \tag{1}$$

$j = \overline{1, m}$, *m=card*, and r_j - rank of *j*- path of the flow *Q* across *m* shortest paths where more than one solution (not trivial solution) can be obtained, so, the setting optimization problem becomes possible. The requirement (1) is satisfied at the stage of choosing the number of shortest paths *m* and constructing the set of these paths *P*.

One of the requirements for the flow distribution over m independent paths is the minimization of the transmission time of the flow *Q* from *s* to *t*, which will be determined by the maximum transmission time of the corresponding parts of the flow *Q* across the *m* paths. Minimization can be achieved by equalizing the transmission time along all paths. Let the value of the flow routed along the *i*- path to be x_i , $i = \overline{1, m}$, $c_i = \min\{c_i\}$, $j = \overline{1, r_i}$, $h_i = \dfrac{c_i}{\sum_{j=1}^{m} c_j}$. Then the problem of optimal distribution of the flow of the value *Q* across the *m* paths with given minimum capacities c_i , can be formulated as a mini to max (minimax) problem:

$$\max\left|\frac{x_i}{c_i} - \sum_{j=1}^{m} h_i x_i\right| \to \min_{x \in \Omega} \tag{2}$$

Taking into account that those values of the given flow $\dfrac{x_i}{c_i}$, that are less than the weighted average value, are not necessary to minimize, and hence formula (2) is transformed to the (3):

$$\max\left(\frac{x_i}{c_i} - \sum_{j=1}^{m} h_i x_i\right) \to \min_{x \in \Omega} \tag{3}$$

Then the minimax problem (3) can be reduced to the linear programming problem:

$$z(x) = \sum_{C=1}^{m} \frac{c_i}{\sum_{j=1}^{m} c_j} \cdot x_i \to \max \tag{4}$$

under the following constraints in (5), (6), (7), and (8):

$$\sum_{i=1}^{m} x_i = Q \tag{5}$$

$$x_i \sum_{\substack{j=1 \\ j \neq i}}^{m} c_j - c_i \sum_{\substack{j=1 \\ j \neq i}}^{m} x_j \geq 0 \tag{6}$$

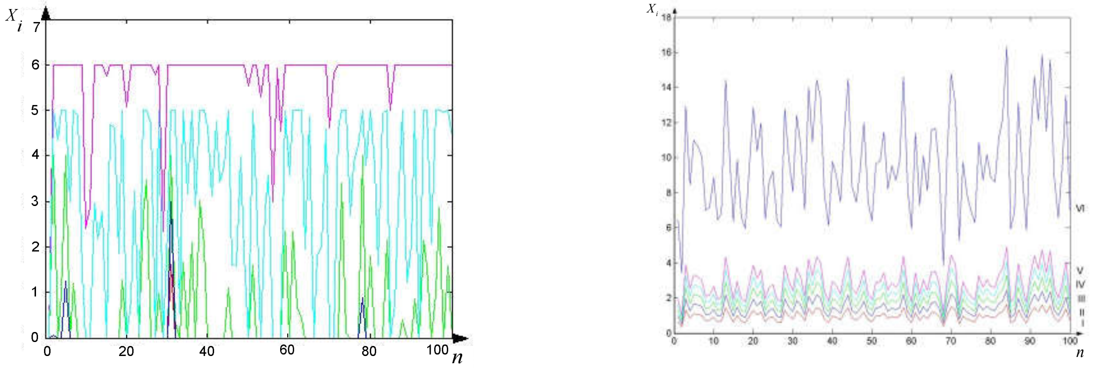
$$x_i \leq a_i c_i \tag{7}$$

$$x_i \geq 0 \tag{8}$$

The constraint (5) is the requirement to distribute the entire flow, the constraint (6) is the requirement that the distribution to be proportional, the constraint (7) is the requirement to prevent the overloads in the "bottlenecks" of the *i*-path, and the coefficient a_i determines the maximum load of the *i*- path.

3 Modeling Results Analysis

The modeling algorithm was carried out for a network section in which there are 5 independent shortest paths from the source to the destination using Matlab simulation. As a load model in the node *s,* a traffic source with a normal distribution and M.O. equal to 10 was used. To analyze the proposed algorithm, in the node *s*, the flow was distributed without optimization procedures and with the use of optimization (fig. 1 a and b) respectively.



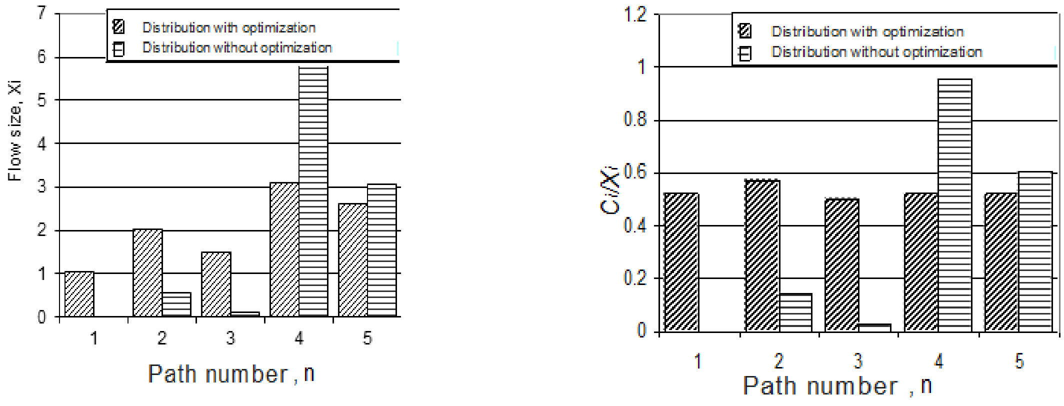
(a)

(b)

Figure. 1: Distribution of load across the paths without optimization (a), and with optimization (b).

The (fig.1.a and b) shows the distribution in time of the incoming flow along the multiple paths. When distributed without optimization (fig. 1-(a) the maximum capacity path has the highest priority and is "filled" completely. The "remainder" of the flow is routed along the path that has the highest priority among the remaining paths. Path priorities are determined on the basis of the throughput of the path. As can be seen from the (fig.1-a), path having the maximum bandwidth 6 is loaded completely, then the next path having the maximum value of 5, etc. Thus, when using such a load distribution scheme, some paths will be overloaded, while others will be idle. The fig. 1-(b) shows that when using optimization; all paths are loaded more compactly and values x_i are proportional to the path capacity.

The fig. 2-(a) shows the average value of the flow X_i along the *i*- path, so, can be seen, in the case of optimal load distribution, the average path loading is 52% (fig. 2-(a)), and the maximum value of loading is 58% (fig. 3). In contrast to this, when using load distribution without optimization several paths are loaded completely, while the others are idle (fig.2 (a) and (b)).



(a)

(b)

Figure. 2: The path number n with and without optimization for the flow size Xi (a), and Ci /Xi (b).

The Figure 3 shows the dependence $M\left[\max\limits_i\left\{\frac{x_i}{c_i}\right\}\right]$ on the value *n* when using optimization. Expectation value in this case is 0.58, i.e. 58% of the value $\max\limits_i\left\{\frac{x_i}{c_i}\right\}$. The value of this parameter when using the load distribution without optimization is 1.

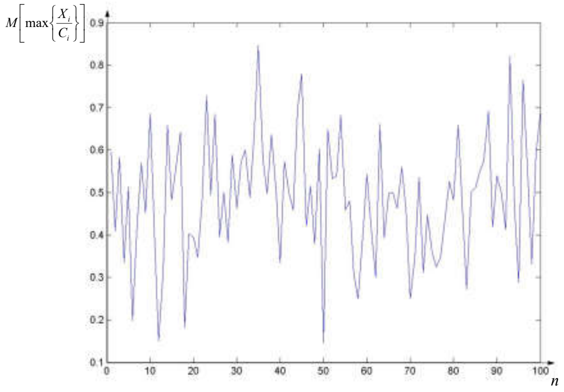


Figure. 3: Expectation value of the parameter

4 The Proposed Algorithm Computational Complexity Estimation

When solving the optimization problem; it is necessary to analyze the feasibility of the proposed routing algorithm in real time.

The real time is determined by the properties of the processes taking place in a certain separately taken system, therefore it is necessary to determine the time interval in which the routing task at the operation stage must be solved. As such a time interval, it is advisable to choose the time during which alternative paths for traffic transmission must be selected.

The multi-path routing algorithm proposed in the paper uses the methods of finding the shortest paths on a graph, solving the problem of multi-criteria optimization and mathematical programming for finding the paths and load distribution. The amount of calculations for finding the shortest paths using the Dijkstra's algorithm is known to be $O(N^2)$, where *N* is the number of vertices of the graph [13-14]. Then, assuming that the optimal number of independent shortest paths is equal to the connectivity of the graph of the network *S* (the worst case), the computation volume of this part of the problem can be estimated by the following formula:

$$E_1 \approx N^2 \cdot S \tag{9}$$

The amount of computation for solving the multi-criteria optimization problem is determined by the following procedures:

1) Calculation of utility functions according to formula

$$X_m^0 = \arg\underset{x \in X}{extr} K_m(x), m = \overline{1,n} \tag{10}$$

2) Finding the optimal solution for the given importance of particular criteria.

The assumption that $cardM = S$, where *M* is a set of shortest paths, *S* is the network graph connectivity, the calculation of the utility function for 2 criteria for S variants will be $2 \cdot 3 \cdot S$ operations. The construction of an additive criterion and finding the maximum will constitute $4 \cdot S$ operations. Then the amount of computation in solving the multi-criteria optimization problem can be estimated using the following formula:

$$E_2 \approx 10 \cdot S \tag{11}$$

To distribute the flow, in a given network, the linear programming problem was solved. As indicated by experts in the field of mathematical programming, the simplex method is considered to be the best procedure for solving linear programming problems [11]. There are many modifications of this method, however, according to experts; none of the versions of the simplex method gives a tangible gain of the time of calculations. It is shown in [12] that in the general case the linear programming problem is *NP*-complete. However, in [11-17] it is shown that when solving applied problems, such as finding the maximum flow, etc., the simplex method has a polynomial complexity. The computational efficiency of the simplex method according to [11] can be estimated using the following two parameters: number of iterations (admissible basic solutions needed to achieve the optimal solution of the problem) and the total machine time required to solve the problem. The experience of solving a large number of practical problems shows that the upper bound of the iterations in the solution of the linear programming problems in standard form with m constraints and n variables it is possible to consider the value 2(m+n).

$$E_3 \approx 2(n+m)^2 \cdot (5m+3) \tag{12}$$

For the problem under consideration, the number of constraints is three times greater than the number of unknowns, then

$$E_3 \approx 180n^3 + 54n^2 \tag{13}$$

The multi-path routing problem time can be estimated using the following expression:

$$T = (E_1 + E_2 + E_3) \cdot t_{on} \approx \left[180 \cdot \mathrm{S}^3 + 54S^2 + S(10+n^2)\right] \cdot t_{on}, \tag{14}$$

Table 1 shows the calculation of time for solving the multi-path routing problem for networks of various sizes and connectivity for a device with $t_{on} = 10^{-9} c$.

Table 1: Calculation of the computational complexity of the algorithm for various networks.

Number of network vertices, *N*	Network connectivity, *S*	Problem time, *T*
30	5	$2{,}85 \cdot 10^{-5} c$
30	20	$1{,}5 \cdot 10^{-3} c$
50	10	$2{,}1 \cdot 10^{-4} c$
50	30	$5{,}0 \cdot 10^{-3} c$
100	20	$1{,}02 \cdot 10^{-3} c$
100	50	$2{,}3 \cdot 10^{-2} c$

From Table1; it is clear that the problem time does not exceed the allowable value of 50 ms, which makes it possible to use the proposed algorithm in MPLS-TE networks.

5 Conclusion:

The proposed algorithm in this paper allows to significantly reduce the load on individual sections of the network and to perform load optimization, distributing traffic along the optimal number of paths. As shown by the simulation modeling, in the particular proposed case, the use of the proposed multipath routing scheme allows achieving a reduction of up to 40% of the total load on single paths by redistributing the traffic.

REFERENCES

[1] Alhihi M. Practical Routing Protocol Models to Improve Network Performance and Adequacy. Journal of Computer and Communications. 2017 Apr 6;5(06):114.

[2] Alhihi M. Network Coding for Wireless Sensor Network Cluster over Rayleigh Fading Channel: Finite State Markov Chain. International Journal of Communications, Network and System Sciences. 2017 Jan 17;10(01):1.

[3] El-Hihi M, Attar H, Solyman AA, Stankovic L. Network Coding Cooperation Performance Analysis in Wireless Network over a Lossy Channel, M Users and a Destination Scenario. Communications and Network. 2016 Sep 22;8(04):257.

[4] Attar, H., Stankovic, L., Alhihi, M. and Ameen, A., 2014, May. Deterministic network coding over Long Term Evaluation Advance communication system. In *Digital Information and Communication Technology and it's Applications (DICTAP), 2014 Fourth International Conference on* (pp. 56-61). IEEE.

[5] Hung, Tran Cong, and Ly Quoc Hung. "ENERGY CONSUMPTION IMPROVEMENT OF TRADITIONAL CLUSTERING METHOD IN WIRELESS SENSOR NETWORK." *network* 6: 7.(2016)

[6] Birmpilis, Stavros, and Timotheos Aslanidis. "A Critical Improvement On Open Shop Scheduling Algorithm For Routing In Interconnection Networks." *arXiv preprint arXiv:1702.08236* (2017).

[7] Awduche D., Requirements for traffic engineering over MPLS// RFC2702, 1999.

[8] Moy J., "OSPF Version 2", RFC-2328, Internet Engineering Task Force, April 1998.

[9] Cisco Systems Inc Rukovodstvo po tekhnologiyam obyedinennykh setey , 4-e izdanie.: Per. s angl. – M.: Izdatelskiy dom «Vilyams», 2005. – 1040c.

[10] Singkh M., Titli A. Sistemy: dekompozitsiia, optimizatsia I upravlenie. M.: Mashinostroenie, 1986. 494 p.

[11] Rekleitis G., Reyvidran A., Regsdel K. Optimizatsiia v tekhnike, M.:Mis, 1986, 455c.

[12] Maynika E., Algoritmi optimizatsii na setyakh I graphakh, Per. s angl. M.:Mir., 1981. – 324 p.

[13] Sun,Liang,and Wenying Chen."Development and application of a multi-stage CCUS source–sink matching model." Applied Energy 185 (2017): 1424-1432.

[14] Dinitz, Yefim, and Rotem Itzhak. "Hybrid Bellman–Ford–Dijkstra algorithm." *Journal of Discrete Algorithms* (2017).

[15] Attar, Hani. "Data Combination over Physical Layer Using Network Coding with PUM Turbo Codes." *Journal of Computer and Communications* 5.06 (2017): 32.

[16] Kim, Hyung-Sin, et al. "Load balancing under heavy traffic in RPL routing protocol for low power and lossy networks." *IEEE Transactions on Mobile Computing* 16.4 (2017): 964-979.

[17] Menon, Varun G., P. M. Joe Prathap, and A. Vijay. "Eliminating Redundant Relaying of Data Packets for Efficient Opportunistic Routing in Dynamic Wireless Ad Hoc Networks." *Asian Journal of Information Technology* 15.20 (2016): 3991-3994.