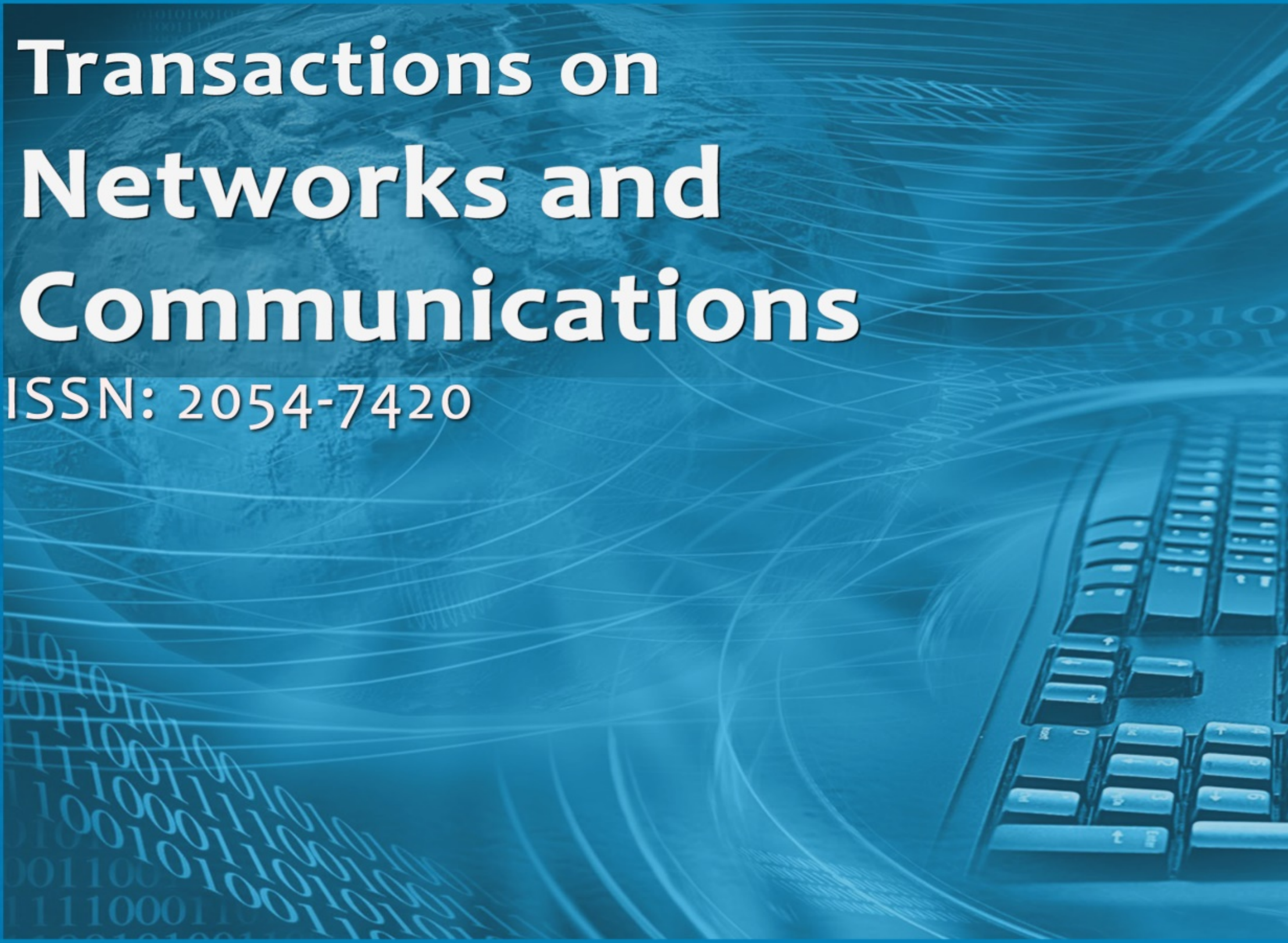
Volume 2. Issue 6 - December 2014



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.

TNC Transactions on Networks and Communications

Society for Science and Education
United Kingdom

Volume 2, Issue 6
ISSN: 2054 - 7420

Effect of Varying Packet Interval Time on Multihop Routing Protocol in Wireless Sensor Network

[1]Namita Sharma and [2]Parveen Kakkar

Dept of CSE, DAV Institute of Engg & Technology, Jalandhar, India

[1]snamita65@yahoo.com; [2]parveen.daviet@gmail.com

ABSTRACT

The distinguishing characteristics of sensor networks are that they are basically infrastructure less, self configured wireless networks used to monitor environmental conditions such as temperature, sounds etc. which and communicate with each other using radio signals. The sensor nodes contain non chargeable batteries and they in several rounds of data transmission they soon get drained out of energy. Depending upon the given conditions, the sensor nodes in the random network may be hundreds or thousands in number .Each individual node has its own sensing and computing devices along with the radio transceivers and power components. In this paper, effect of changing the packet interval time on the energy consumption, throughput of the proposed multihop routing protocol is analyzed. It is a substantial fact that change in the packet interval time when the packets are to be transmitted to further levels of hierarchy has a relation with the latency and load balancing of the network. The simulation results show that with the variation of the packet interval time, the throughput attains a peak value at a certain point, then decreases alternately whereas the energy consumed by the protocol remains constant for a particular time interval but the value increases as the packet transmission time interval increases.

KEYWORDS:- Packet interval variance, Proposed multihop routing protocol, throughput, energy consumption.

1 Introduction

Sensor networks have emerged as a promising tool for monitoring the physical worlds, utilizing self-organizing networks of battery-powered wireless sensors that can sense, process and communicate. They consist of small low power nodes with sensing, Computational and wireless communications capabilities that can be deployed randomly or deterministically in an area from which the users wish to collect data. Typically, wireless sensor networks contain hundreds or thousands of sensor nodes that are generally identical. These sensor nodes have the ability to communicate either among each other or directly to a base station (BS). The sensor network is highly distributed and the nodes are lightweight. Intuitively, a greater number of sensors will enable sensing over a larger area.[1] The main features of WSNs are scalability with respect to the number of nodes in the network, self-organization, self-healing, energy efficiency, a sufficient degree of connectivity among nodes, low-complexity, low cost and size of nodes.

As the batteries of the sensor nodes are not chargeable, the need is to make the methods of data transmission so effective that the data should be able to be routed to the intended base station as quickly as possible thus minimizing delays and negating all kinds of the packet drops, routing

DOI: 10.14738/tnc.26.615
Publication Date: 14th November 2014
URL: http://dx.doi.org/10.14738/tnc.26.615

overheads etc along with make the design of the routing protocol energy efficient. In addition to it, WSNs face few other challenges as well. A fundamental challenge to these small networks is that wireless sensor networks are power constrained networks i.e. the wireless sensor node can only be set with a limited power supply usually less than 1.2 V. In some situations, recharge/refill of power resources (battery) might be impossible. So we can say that lifetime of a sensor node is totally dependant on battery lifetime. In a multihop adhoc sensor network, each node plays the double role of data originator and data router. The improper working of a few nodes can cause considerable topological changes and might require rerouting of packets and reorganization of the network. The main task of a sensor node in a sensor field is to detect events, perform quick local data processing, and then transmit the data [2]

The are many applications of WSNs to the real world like environmental monitoring , health care , positioning and tracking , to logistic, localization, and so on but everywhere the longevity of such networks has been a source of concern as parameters such as QoS , maximum data transmission for any ideal network or a specific application cannot be compromised. Once data has been made available to the CHs, the next task is to route that data either using single hop manner or in multihop manner so that it could reach Base station. The results for homogeneous networks are better than heterogeneous networks but this fact is also true that the inclusion of certain heterogeneous nodes in the homogeneous environments can further improve the lifetime of the wireless sensor network. [3] Small periodic data packets are the most common workload in sensor networks, but certain cases arise where larger transfers are needed. Therefore, the larger the packet interval, the more the latency is increased. However, the latency decreases with decreasing packet intervals [4] .Also if the packets are transmitted to the nodes up in the hierarchy by the sensor nodes after certain periodically increased intervals, the network traffic load can be managed effectively although in this process there may be slight increase in the energy consumption of the network but overall efficiency would increase considerably.

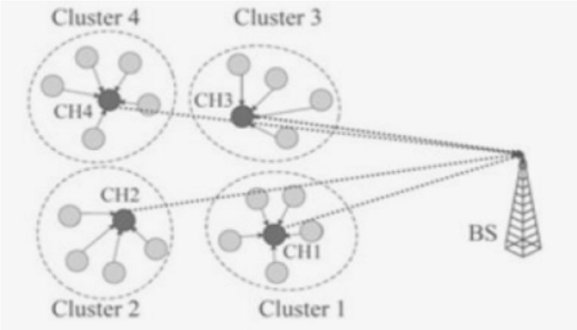


Figure 1: Structure of WSN in LEACH Protocol [5]

The basic structure of Wireless Sensor network consists of normal sensor nodes which are grouped to certain Cluster head nodes which are elected randomly in every round of transmission of data from source to sink. The cluster head nodes are the highest energy nodes in each cluster in a particular round of data transmission. After data aggregation is done by the cluster head nodes, the data is sent to the base station either using flat routing, single hop or multihop routing. But over the years energy efficiency, load balancing, extension of network lifetime many algorithms, techniques and protocols have been developed like LEACH(Low Energy Adaptive Clustering Hierarchy) Protocol [5] followed by HEED[6], A-LEACH[7],C-LEACH[8] etc. The distribution of the paper goes as second section discusses the related work, third section discussed the proposed technique, fourth section tells the result analysis and discussion finally the fifth and sixth sections describe the conclusion and future scope of the proposed technique.

2 Related Work

Multihop-LEACH can further improved by increasing probability of cluster head and vice cluster head. With a varying probability of clustering, it is clear that more cluster heads in a network results in better connectivity. We can still minimize the energy consumption and extend the network life time by improving the clustering technique[1] The concept of the multihop routing was discussed and implemented in multihop LEACH protocol[9] In this protocol, Multihop-LEACH uses both inter cluster as well as intra cluster communication. The power usage, latency and success rate in Multihop-LEACH can further improved by increasing probability of clustering .Then another multihop routing protocol[10] was proposed which gave the concept of introduction of gateway nodes in the network at the next level to the cluster head nodes. The total number of the gateway nodes was about 10% of the total number of sensor nodes in the network these would act as intermediaries for the transmission of data from the cluster heads to the base station. They followed a constraint that no two gateway nodes would transfer data to the gateway at the same time, rather the gateway nodes would provide them certain set time slots during which they would transmit data to them. At the given instant of time in the network if one gateway node is not free for transmission of data to the cluster head node, it would not waste time waiting for it turn, rather it would check the availability of other gateway nodes, which so ever is free, it would select it and transfer the data to it for transmission to the base station. Thus with increase of one more hop in the network, there is considerable extension in the network lifetime as compared to the single hop routing protocol.

Yet another protocol named Assisted LEACH[7] focuses on network lifetime goes down when both data aggregation and routing are carried out by Cluster Heads alone which can be eradicated by usage of Helper Nodes for Routing and Cluster Heads for Data Aggregation. It reduced the overhead for route formulation to base station by electing next hop at each Helper Node using the Received Signal Strength values of beckon signal from base station already available at helper nodes during Helper Node Selection phase. The concept of Helper Nodes in Assisted LEACH (A-LEACH) protocol has improved the lifetime of the network by distributing the minimized energy dissipation throughout the nodes.

2.1 Effect of Packet Interval on Performance of Network

In a wireless sensor network packet size has the direct effect on reliability and performance of communication between wireless nodes, so there is need to have an optimal packet size for wireless sensor networks. In fact, if a packet transmission fails, the sender has to wait for a random back off period before resuming the packet transmission. However, this period is computed independently from the channel coherence time. Therefore if the channel conditions during retries are still the same or worse, successive failures occur and latency is increased. Network performance would improve if the packet interval depends on the time coherence of the channel. If the interval is too small compared to the coherence time, packet error rate will be high when channel conditions are bad and vice-versa. The packet interval management may also involve the application layer. There is an impact of changing the packet interval on the network performance. The packet transmission time can be tuned in order to optimize the packet delivery ratio. [11]

Sensor networks are predicated using low- power RF transceivers in a multi-hop fashion. Multiple short hops can be more energy-efficient than one single hop over a long range link. Poor cumulative packet delivery performance across multiple hops may degrade performance of data transport and expend significant energy. The traffic pattern is very simple in wireless sensor network as each node sends roughly k packets per second with an exponentially distributed inter- packet interval (to avoid synchronization). The traffic load on the network can be adjusted by changing the average load. Periodically, each node broadcasts a packet so that all nodes can construct their neighbor lists. Nodes log received packets in the node's memory [12].

To guarantee the real time performance of the nodes, each data packet is constrained in a time interval in which it must be sent to the destination node. If time expires, the data packet has to be discarded. Once node failure or congestion occurs, large amounts of data packets will be discarded, which may cause disastrous consequences. Consequently, it is more significant and challenging to provide both real-time and fault tolerance characteristics in WSN routing protocol. Failure nodes are treated as an empty area (VOID), and data packets are sent to the sink node via bypass. However, these methods do not predict network congestion in advance, and the remaining transmission time of the data packet is only used for checking the validity of the data packets. When a data packet cannot be transmitted to the next hop node, it will be automatically discarded at once, which wastes the transmission energy. Moreover, the upper stream node cannot receive the feedback information from the current node and thus affect the subsequent transmission. Each node utilizes the remaining transmission time of the data packets and the state of the forwarding candidate node set to dynamically select the next hop. Once node failure, network congestion or void region occurs, the transmission mode will switch to jumping mode, which aims at reducing the transmission time delay and ensuring the data packets to be sent to the destination node within the specified time limit [13].

The sensor network application scenarios and network traffic characteristics differ significantly from conventional computer networks. Typically data is sent periodically in short packets. To achieve fairness and energy efficient transmission through a multihop network, they design an adaptive rate control protocol to that is optimized for n-to-1 data reporting and multihop networking .To let the receiver sleep for most of the time when the channel is idle, nodes periodically wake up and check for activity on the channel. If the channel is idle, the receiver goes back to sleep. Otherwise, the receiver stays on and continues to listen until the packet is received. Packets are sent with long preambles to match the channel check period. Each time the node wakes up, it turns on the radio and checks for activity. If activity is detected, the node powers up and stays awake for the time required to receive the incoming packet. After reception, the node returns to sleep. If no packet is received (a false positive), a timeout forces the node back to sleep. Data that can be delivered by each protocol and the cost of delivering that data. In all tests where we mention “packet size”, we are referring to the size of the data payload only, not the header information. [14]

Channel utilization is a traditional metric for MAC protocols that illustrates protocol efficiency. High channel utilization is critical for delivering a large number of packets in a short amount of time .In sensor networks, quickly transferring bulk data typically occurs in network reprogramming or extracting logged sensor data. By minimizing the time to send packets, we can also reduce the network contention.These services implement the appropriate hidden terminal support for their workloads. For example, after sending a multihop message, all nodes in the cell should refrain from transmitting until one packet time has elapsed to allow the parent to retransmit up the tree as proven to be more efficient than control messages for multihop traffic. By allowing the service to decide, many costly control message exchanges are eliminated. Although the network is homogeneous, we can exploit that the base station runs with a different duty cycle (since it is always on) than the data collection network. Instead of sending packets with long preambles, nodes one hop away send packets with only an 8 byte preamble to the base station. Note that Lpreamble is reduced from the long LPL preamble to only 8 bytes. The node can return to sleep for the check interval after receiving a packet or can perform early rejection much quicker than packets sent with the long preambles. [14]

3 Proposed Protocol

The proposed scheme consists of assumptions, radio propagation model, algorithm which are discussed as:

3.1 Assumptions of proposed Protocol

For the proposed protocol [3] some basic assumptions are made which are as under:-

1) These nodes are Mobile and homogeneous in nature.
2) Base station is far away from the network and is fixed.
3) Every sensor node is capable of communicating with every other sensor node scattered randomly in the network and to the Base Station if needed.

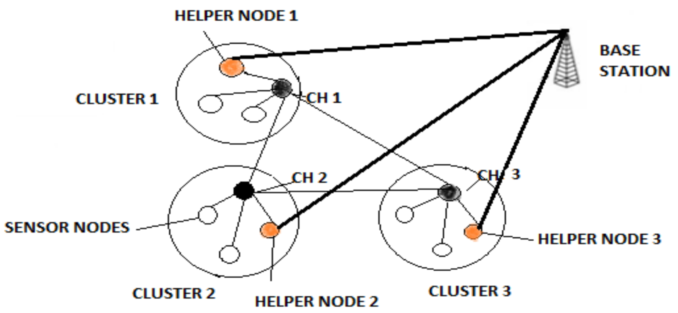


Figure 2: Diagram of the Proposed Scheme [3]

3.2 Radio Energy Dissipation Model

For the proposed protocol, the first order radio model is used for energy dissipation in communication [3], where radio dissipates Eelec = 50 nano Joule / bit to drive the transmitter and the transmit-amplifier dissipates εelec =100 pico Joule/ bit/m2. To save energy, when required the radio can be turned on or off. Also the radio spends the minimum energy required to reach the destination. The energy consumed for data transmission of k bits packet is calculated from the Eq. (1).

E TX (k,d) = E elec * k + ε elec * k*d 2 (1)[3]

and to receive this message, the radio expends energy is shown in Eq. (2):

E Rx (k) = E RX-elec (k) (2)[3]

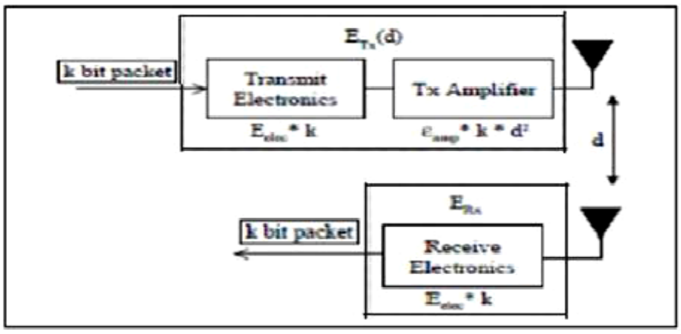


Figure 3: Radio Dissipation Model [3]

3.3 Algorithm of Proposed Protocol

The main goal of the approach is to extend network lifetime of the network [3]. For this reason, cluster head selection is mainly based on the residual energy of each node .The highest energy node that is if the remaining battery power is high then that node will become CH and the least mobility node will become a CH. Distance of a node from the cluster centroid. The BS calculates the distance of each node to its cluster centroid. The lesser distance node from the BS to itself will have the higher probability to become a CH. The network initialization phase starts after the sensor nodes are randomly distributed in the application area. The base station broadcasts a “HELLO” message to all the nodes in the network to ensure that the network is alive.

The algorithm starts with randomly selects a starting node that has not been visited and it retrieves all neighbor nodes which is density reachable from starting node with respect to Eps and MinPts. Here Eps is a radius of the cluster and MinPts is a minimum nodes required inside the cluster. If the number of neighbors is greater than or equal to MinPts then the cluster is formed as Let the distance between two sets of nodes S1 and S2 be defined as dist (S1, S2) = min {dist (p,q) | P Ɛ S1, q Ɛ S2} and further the nodes with the highest energy are selected as cluster heads by the sensor nodes to which “ADVERTISEMENT” message is broadcasted by the CH and all the sensor nodes which join the cluster reply back with “ACK” message. The next phase deals with the selection of the cluster heads for each cluster. After the clusters are formed, the Base station should decide whether or not the node becomes a cluster head for the current round. To find that, the value of energy are computed for all the nodes in the network for each round. The node which has highest residual energy is elected the cluster head for the specific round.

Once the clusters are created and the CH issues a TDMA schedule to all the other sensor nodes in the clusters during which they need to transmit data to their Cluster heads. Base Station constantly observes the residual energy and Mobility of the existing CH. If it is below the threshold value then it select another CH based on same conditions, described earlier. Finally the CH should be checked out the routing path. If the routing path residual energy goes below the threshold or any node fails, BS selects another path and sends the routing path to the respective CH. So, the base station calculated the distance of all nodes in the network to itself using RSSI value [3] which is calculated with the help of two ray ground model

$$P_r(d) = \frac{P_t * G_t * G_r * h_t^2 * h_r^2}{d^4 L} \qquad (3)[3]$$

Where P_r: Power received at distance d P_t: Transmitted signal power G_t: Transmitter gain (1.0 for all antennas) G_r: Receiver gain (1.0 for all antennas) d: Distance from the transmitter L: Path loss (1.0 for all antennas) ht: Transmitter antenna height (1.5 m for all antennas) hr: Receiver antenna height (1.5 m for all antennas)

The data aggregated by all the cluster heads are sent to the helper nodes. The helper nodes are those which have second highest energy left in them at the end of each round. Sometimes there might be a situation when there is no such helper node left inside the cluster as it too has been drained out of its energy so in that case the cluster head would search for some other available nearby helper node in some other cluster to which data can be transmitted. The cluster heads enter into sleep mode once they transmit data to the helper nodes so that their energies are saved. At a given time, all the cluster head nodes send data to the helper nodes using multihop routing. Further the helper nodes are informed of the shortest path calculated by the base station along which the data is transmitted again by multihop routing[3].Thus this protocol would enhance the performance as well as improve the lifetime of wireless sensor network.

4 Result Analysis and Discussion

The simulation scenario consists of 50 sensor nodes deployed in the network field of size 1300m*1000 m in the wireless sensor network. All the simulations have been performed using NS2.The results have been obtained at the end of seven rounds of the network at simulation time = 30 sec for both the protocols. The blue line shows the results of the proposed protocol.The main objective of simulation is to analyze the effect on proposed multihop routing protocol by varying packet interval.

Table 1: Simulation Parameters

Simulator	Ns-2.35
Simulation time	30 sec
Channel Type	Wireless
No of nodes	50
Topology	1300m *1000m
Radio Propagation model	Two way ground
Communication Model	Bi direction
Transmission Range	250m
Interface Queue Type	Queue/Drop Tail/Pri Queue
Initial energy	100 Joules
Antenna Type	Omni Antenna
Traffic Type	CBR
Packet Size	256 bytes

4.1 Performance Metrics

The performance analysis of the proposed protocol is done by analyzing the results of Proposed Protocol by using some of the performance metrics such as:

- **Throughput**: It is the measure of the number of bits of data packets that are transmitted from source to destination in given time. It is always less than 1. The formula of measuring throughput is

$$\frac{Number\ \ of\ \ bytes\ \ received}{Time\ \ in\ \ milliseconds} \qquad (4)[3]$$

Generally it is measured in Kb/sec or Bytes/sec. For the protocol aiming to enhance the throughput of the network , it is must that the packet drop rate, jitters , routing overheads and congestion or packet loss should be as less as possible otherwise lower value of throughput would decrease the data packets delivery from the source to the destination.

- **Average Energy Consumption (Ea):** The average energy consumption is calculated across the entire topology. It measures the average difference between the initial level of energy and the final level of energy that is left in each node.

Let E_i = the initial energy level of a node, E_f = the final energy level of a node and N = number of nodes in the simulation. Then

$$E_a = \sum_{k=1}^{n} (E_{ik} - E_{fk}) / N \qquad (5)[3]$$

This metric is an important because the energy level the network uses is proportional to the network's lifetime. The lower the energy consumption the longer is the network's lifespan. Thus the ideal value for average energy consumed by the protocol should be as less as possible otherwise if the protocol would consume more energy after every round then it would become difficult to increase the lifetime of the network . The exact formula for calculation of average energy is inbuilt in NS2.

Here two cases are to be analyzed to understand the effect that the packet interval has on overall efficiency of the protocol by initially setting the packet interval at value of one second and then at three seconds and further understanding the impact on throughput and energy consumption .

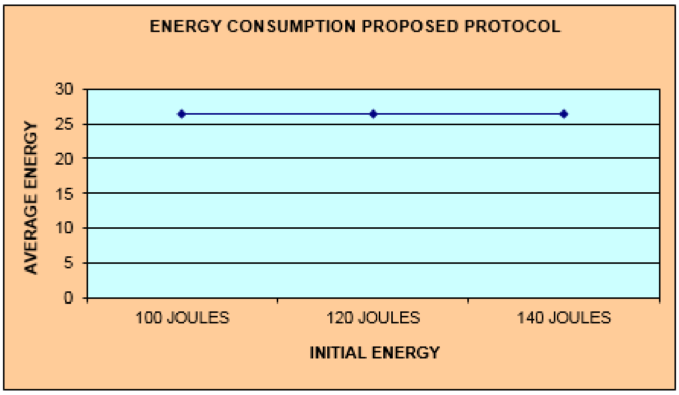


Figure 4: Graph showing Proposed protocol based on energy consumption when packet interval is 1 second.

The graph clearly shows that on running the simulation for seven rounds, when packet size is kept at 256 bytes and the initial energy is varied from 100 Joules to 140 Joules, the average energy consumed by the proposed protocol remains constant at value of 26 Joules. This clearly indicates that with the increase in the packet interval time (transmission), the protocol remains unaffected on the grounds of energy consumption while routing packet from source to sink round by round.

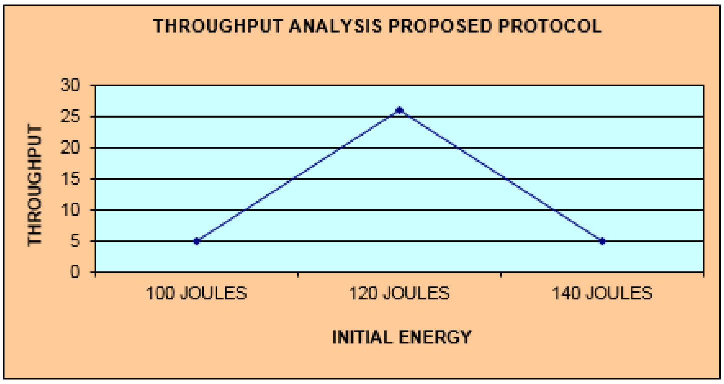


Figure 5: Graph showing Proposed protocol based on throughput when packet interval is 1 second

The graph clearly shows that on running the simulation for seven rounds, when packet size is kept at 256 bytes and the initial energy is varied from 100 Joules to 140 Joules, the throughput of the proposed protocol attains peak value at 120 Joules initial energy and declines for 140 Joules which means that on increasing the value of packet interval the proposed protocol starts consuming more energy although at certain point of time the throughput of the protocol increases and decreases alternately.

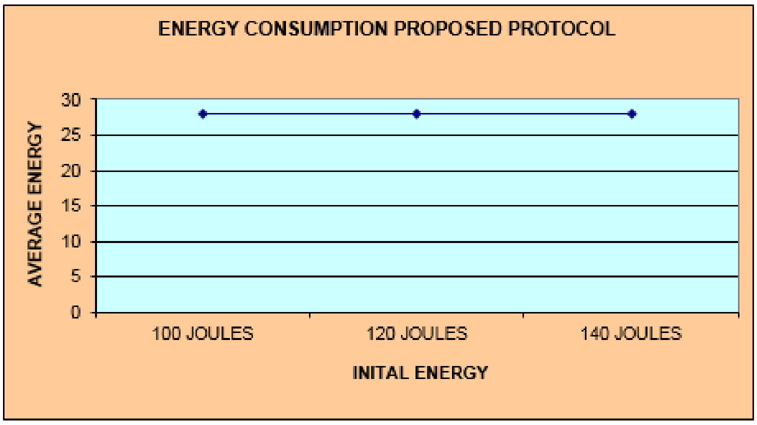


Figure 6: Graph showing proposed protocol based on energy consumption when packet interval is 3 second

The graph clearly shows that on running the simulation for seven rounds, when packet size is kept at 256 bytes and the initial energy is varied from 100 Joules to 140 Joules, the average energy consumed by the proposed protocol remains constant at value of 28 Joules. This clearly indicates that with the increase in the packet interval time (transmission), the protocol remains unaffected on the grounds of energy consumption while routing packet from source to sink round by round but the net energy consumption is slightly increased after every interval.

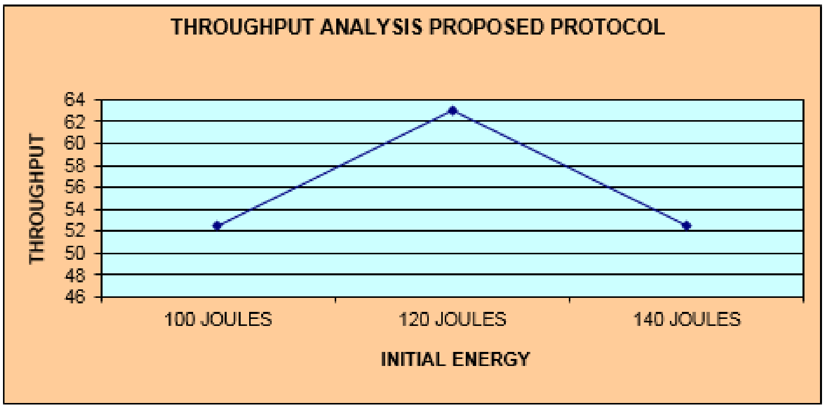


Figure 7: Graph showing proposed protocol based on throughput when packet interval is 3 second

The graph clearly shows that on running the simulation for seven rounds, when packet size is kept at 256 bytes and the initial energy is varied from 100 Joules to 140 Joules, the throughput of the proposed protocol attains peak value at 120 Joules initial energy and declines for 140 Joules which means that on increasing the value of packet interval the proposed protocol start consuming more energy although at certain point of time the throughput of the protocol increases considerably as compared to the value at packet interval time of one second but then decreases afterwards.

Table 2: Summary of Results: PACKET INTERVAL = 1 second

Initial Energy	**Proposed Scheme**	
	Throughput	Average Energy
100 Joules	5 Kb/s	26.5 Joules
120 Joules	26 Kb/s	26.5 Joules
140 Joules	5 Kb/s	26.5 Joules

Table 3: Summary of results: packet interval = 3 second

Initial Energy	**Proposed Scheme**	
	Throughput	Average Energy
100 Joules	52.5 Kb/s	28 Joules
120 Joules	63 Kb/s	28 Joules
140 Joules	52.5 Kb/s	28 Joules

5 Conclusion

The above results provide an insight to the fact that by keeping the packet size constant and varying the packet interval while transmission of data over the network and analyzing its impact on the various performance metrics like average energy consumption and throughput, the inference thus drawn is that at lower packet interval time, when the data is transmitted to the base station, the protocol expends less energy but as the packet interval time increases, the energy consumption of the proposed protocol increases though it remains constant for a particular time interval and similarly the value of throughput increases and decreases for the proposed protocol at alternate intervals in accordance with the change in initial energy.

6 Future Scope

In future the effect on the various other performance metrics can be determined by simulating the proposed protocol on other simulators or by simulating it on other range of applications like VBR (Variable Bit Rate), VOIP etc. Apart from it, the environment can be varied from homogeneous to heterogeneous and its implications can be studied. In addition to it, this protocol can be evaluated by applying to certain specific case studies so that its implications can be interpreted in a wide scenario.

REFERENCES

[1]. J S Rauthan, S Mishra , An improved Cluster Based Multi-hop Routing in Self-Organizing Wireless Sensor Networks, International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 4, June - 2012 ISSN: 2278-0181

[2]. Mehjabeen leghari,Shazia Abassi,Dr Lachhman Das Dhomeja, Survey on Packet Size optimization techniques in Wireless Sensor Networks, Institute of Information and Communication Technology , University of Singh,Janshoro.

[3]. Namita Sharma, Impact of Varying Packet Size on multihop routing protocol in Wireless Sensor Networks, IJASCSE,Volume 3,Issue 9, 2014.

[4]. Dae-Suk Yoo, Seung Sik Choi, Medium Access Control with Dynamic Frame length in Wireless Senor Networks, Journal of Information Processing Systems,Volume 6, No.4, December 2014.

[5]. Wendi B. Heinzelman, Anantha P. Chandrakasan, and Hari Balakrishnan,An Application-Specific Protocol Architecture for Wireless Microsensor Networks, 660 IEEE Transactions on Wireless Communications, Vol. 1, No. 4, October 2002.

[6]. Amir Akhavan Kharazian1, Kamal Jamshidi and Mohammad Reza Khayyambashi,Adaptive Clustering in Wireless Sensor Network: considering nodes with lowest energy, International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC) Vol.3, No.2, April 2012

[7]. Sunkara Vinodh Kumar and Ajit Pal, Assisted-Leach (A-Leach) Energy Efficient Routing Protocol for Wireless Sensor Networks, International Journal of Computer and Communication Engineering, Vol. 2, No. 4, July 2013.

[8]. J.Gnanambigai, Dr.N.Rengarajan, K.Anbukkarasi, Leach and Its Descendant Protocols: A Survey, International Journal of Communication and Computer Technologies Volume 01 – No.3, Issue: 02 September 2012 ISSN Number: 2278-9723

[9]. S.Koteswararao, M.Sailaja, T.Madhu, Implementation of Multi-hop Cluster based Routing Protocol for Wireless Sensor Networks, International Journal of Computer Applications (0975 – 8887) Volume 59– No.8, December 2012

[10]. S. Taruna, Rekha Kumawat, G.N.Purohit, Multi-Hop Clustering Protocol using Gateway Nodes in Wireless Sensor Network International Journal of Wireless & Mobile Networks (IJWMN), Vol 4, No 4, August 2012.

[11]. Chiraz Chaabane, Alain Pegatoquet, Michel Auguin, Maher Ben Jemaa, A Joint Mobility Management Approach and Data Rate Adaptation Algorithm for IEEE 802.15.4/Zigbee Nodes, Scientific Research, Wireless Sensor Networks, 2014,6,27-34.

[12]. Jerry Zhao, Ramesh Govindan, Understanding Packet Delivery Performance in Dense Wireless Sensor Networks, ACM, SenSys, Nov 5-7,2003.

[13]. Guowei Wu, Chi Lin,Feng Xiq, Lin Yao, He Zhang,Bing Liu, Dynamical Jumping Real –Time Fault-Tolerant Routing Protocol for Wireless Sensor Networks, School of Electronics & Information,Dalian University of Technology.

[14]. Joseph Polastre,Jason Hill, David Culler, Versatile Low Power Media Access for Wireless Sensor Networks, SenSys-04, November 3-5, 2004,Baltimore, Maryland,USA.

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TNC Transactions on Networks and Communications

VOLUME 2, ISSUE 6
ISSN: 2054 - 7420

Cloud Services Usage Profile Based Intruder Detection and Prevention System: Intrusion Meter

[1]Dinesha H A and [2]Vinod Kumar Agrawal
PES Institute of Technology, Visvesvaraya Technological Univeristy, Belgaum, India;
[1]sridini@gmail.com; [2]vk.agarwal@pes.edu

ABSTRACT

With the emerging usage of cloud computing services, the misuse of possible vulnerabilities grows at the same speed. The distributed nature, on demand services, wide usage of the cloud computing makes it an attractive target for potential intruders. Intruders are the network security attackers intend to breach cloud security. Despite security issues delaying cloud adoption, cloud computing has already become an inescapable needs and ready industry solutions. Thus, security mechanisms to ensure its secure adoption are in demand. One security mechanism is intrusion detection and prevention systems (IDPS). IDPS have been used widely to detect malicious behaviors in network communication and hosts. Here, we focus on IDPS to defend against the cloud intruders. We propose a technique called cloud service usage profile based IDPS technique. This technique is to detect and prevent intruders in cloud service intrusion based on the cloud service usage profile. In turn, this usage profile helps to detect unusual usage and prevent intrusion.

Keywords: Cloud Computing, Cloud Usage Profile Based Technique, Intrusion Detection and Prevention Systems

1 Introduction

Cloud computing is an emerging paradigm that allows customers to obtain computing services and resources such as networks, servers, storage and applications. It provides services according to a pay-per-use business model [1]. Cloud computing has a high demand because it enables IT managers to provision services to users sooner and in a gainful way. Cloud computing technology has been facing some security issues. Cloud computing operational models, enabling technologies and its distributed nature, clouds are easy targets for intruders [2][3][4].

Many intrusion detection and information security approaches for securing cloud have been proposed and are in practice [5, 6–16]. In a recent research paper by, Rocha and Correia [17] presents how malicious insiders can steal confidential data. Anup gosh and chris greamo [18] has presented how malware effects cloud computing environment. A multi-agent based system for intrusion detection by Islam M.Hegazy et al [19] has described a framework for intrusion detection using agent based technology. Hisham A.Kholidy et.al [20] has proposed a framework for Intrusion Detection in cloud systems where IDS is deployed at all the nodes including database which should also be secured. An autonomous agent based incident detection system for cloud environments has proposed agent based model with sensors by monitoring business flows customer behavior can be predicted can determine DoS attacks [21].

DOI: 10.14738/tnc.26.590
Publication Date: 16th November 2014
URL: http://dx.doi.org/10.14738/tnc.26.590

In this paper, we present the identified existing intrusion attacks, existing intrusion detection and prevention techniques and drawbacks of existing IDPS solution for cloud intrusion attacks. We propose novel cloud service usage profile based intruder detection and prevention system to some of the cloud intrusion attacks. It detects and prevents intrusion based on their regular cloud service usage profiles. Usage profile may consist of many parameters like regular usage time, usage roles, usage privileges, usage logs and etc.

This paper organized as following manner. Section II, presents the study details on identified existing attacks and intruder detection and prevention system. Section III, describes the proposed cloud usage profile based intruder detection and prevention system, design details with analysis. Section IV, concludes the paper along with future enhancement.

2 Study on Existing Attacks, Intruder Detection and Prevention System

This section summaries a existing intrusion detection and prevention system. It illustrates several common intrusions and attacks, which causes availability, confidentiality and integrity issues to Cloud resources and services. Intrusion Detection System (IDS) are a proactive monitoring technology and protective mechanism in defending critical IT infrastructures from malicious behaviors. It may compromise sensitive data and critical applications through cyber attacks. IDS generally fall into two groups: signature based detection group and anomaly detection group [2]. Earlier, IDS can protect cloud based system from various types of attacks but it cannot identify suspicious activities in a cloud environment [4]. IDSs may be classified according to the source of data into: (i) Host-based IDS: Here, sensors that detect an intrusion are focused on a single host. (ii) Network-based IDS: sensors are focused on a network segment. (iii) Distributed IDS: It integrates both types of sensors. It can be categorized as Mobile Agent IDS, Grid based IDS and recently Cloud based IDS. Current IDSs have a many deficiencies which are listed in Table1 [22-30]. Intrusion thwarts their adoption in a cloud atmosphere. Cloud based Intrusion attacks are (i) Masquerade attacks (ii) Host-based attacks and (iii) Network-based attacks. Table 2 illustrates the identified existing intrusion attacks and its correspondence solutions [22-30].

Table 1: Existing IDS/IPS deficiency

IDS/IPS	Characteristics / Strengths	Limitations / Challenges
Signature based	Identifies intrusion by matching captured patterns with preconfigured knowledge base. High detection accuracy for previously known attacks. Low computational cost.	Cannot detect new or variant of known attacks. Knowledge base for matching should be crafted carefully. High false alarm rate for unknown attacks.
Anomaly detection	Uses statistical test on collected behavior to identify intrusion. Can lower the false alarm rate for unknown attacks.	Lot of time required to identify attacks. Detection accuracy is based on amount of collected behavior or features.
Hybrid Techniques	It is an efficient approach to classify rules accurately.	Computational cost is high.
HIDS	Identify intrusions by monitoring host’s file system, system calls or network events. No extra hardware required.	Need to install on each machine such as VMs, hypervisor or host machine. It can monitor attacks only on host where it is deployed.

NIDS	Identify intrusions by monitoring network traffic. Need to place only on underlying network. Can monitor multiple systems at a time.	Difficult to detect intrusions from encrypted traffic. It helps only for detecting external intruders. Difficult to detect network intrusions in virtual network.
Hypervisor based IDS	It allows user to monitor and analyze communications between VMs, between hypervisor and VM and within the hypervisor based virtual network.	New and difficult to understand.
DIDS	Uses characteristics of both NIDS and HIDS, and thus inherits benefits from both of them.	Central server may be overloaded and difficult to manage in centralized DIDS. High communication and computational cost.

Table 2: The identified existing intrusion attacks and its correspondence solutions

Sl No	Identified Existing Attacks	Description	Existing solution
1	Insider attack	Authorized cloud user or insiders may commit frauds and disclose information to others.	Signature based intrusion detection
2	Flooding attack	Attacker tries to flood victim by sending huge number of packets from innocent host in network. It leads to fake usage of cloud VMs.	Either signature based intrusion detection or anomaly based intrusion detection techniques can be used.
3	User to Root attacks	Attacker gets an access to legitimate user's account by sniffing password. In case of Cloud, attacker acquires access to valid user's instances which enables him/her for gaining root level access to VMs or host.	Initially anomaly based intrusion detection techniques can be used. Later signature based intrusion detection can be used. But it blocks the genuine user.
4	Port Scanning Attack	Attackers can find open ports and attack on services running on these ports. Network related details such as IP address, MAC address, router, gateway filtering, firewall rules etc. can be known through this attack. In Cloud scenario, attacker can attack offered services	Initially anomaly based intrusion detection techniques can be used. Later signature based intrusion detection can be used. But it blocks genuine ports.
5	Attacks on Virtual Machine (VM) or hypervisor	By compromising the lower layer hypervisor, attacker can gain control over installed VMs. Through these attacks, hackers can be able to compromise installed-hypervisor to gain control over the host.	Anomaly based intrusion detection techniques
6	Backdoor channel attacks	It is a passive attack which allows hackers to gain remote access to the infected node in order to compromise user confidentiality. In Cloud environment, attacker can get access and control Cloud user's resources through backdoor channel and make VM as Zombie to initiate DoS/DDoS attack.	Either signature based intrusion detection or anomaly based intrusion detection techniques can be used.

Some of the above attacks can also be controlled by firewall. Firewall protects the front access points of system and is treated as the first line of defence. Firewalls are used to deny or allow protocols, ports or IP addresses. It diverts incoming traffic according to predefined policy. Fig. 1 describes the firewall types and its characteristics summary [22].

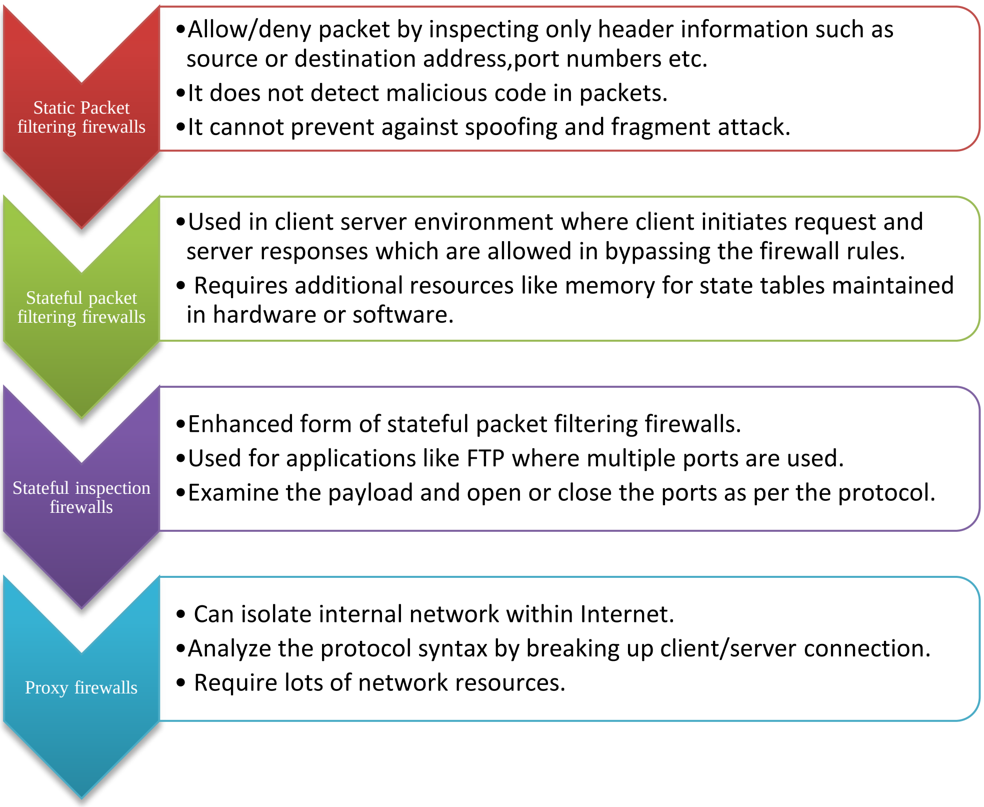

Static Packet filtering firewalls
•Allow/deny packet by inspecting only header information such as source or destination address,port numbers etc.
•It does not detect malicious code in packets.
•It cannot prevent against spoofing and fragment attack.
Stateful packet filtering firewalls
•Used in client server environment where client initiates request and server responses which are allowed in bypassing the firewall rules.
• Requires additional resources like memory for state tables maintained in hardware or software.
Stateful inspection firewalls
•Enhanced form of stateful packet filtering firewalls.
•Used for applications like FTP where multiple ports are used.
•Examine the payload and open or close the ports as per the protocol.
Proxy firewalls
• Can isolate internal network within Internet.
•Analyze the protocol syntax by breaking up client/server connection.
• Require lots of network resources.

Figure 1: Summary of Firewall

3 Proposed Cloud Usage Profile Based Intruder Detection and Prevention System

This section describes the cloud usage profiles based IDS/IPS. It also briefs how it detects intruder, gather intruder information using honey pot as s service system and prevents the intrusion.

Cloud usage profile based intruder detection and prevention system is a technique where in which it detects and prevents intruders based on the customer cloud usage profiles. Customer usage profiles prepared based on their regular usage and it may consists of many parameters like usage timing, duration, access privileges, type of accessed service, logs and etc. After successful service agreement between the customer and vendors, usage profile will be created based on the inputs received from customers. Customer usage profiles may vary from one organization to another. These profiles are very important while vendor providing the cloud service to customer. Every time, before providing services, it is going to check against those profiles. If usage profile and behavior of usage is varies then IPS authenticates internally by rising some questionnaire to the customer. If authenticates fails, IDS systems triggers alerts to vendor. It again forwards the service connection to honey pot system to gather confidential data through intelligent information gathering system. Fig 2 shows the architecture diagram of cloud usage profile based IDS/IPS.

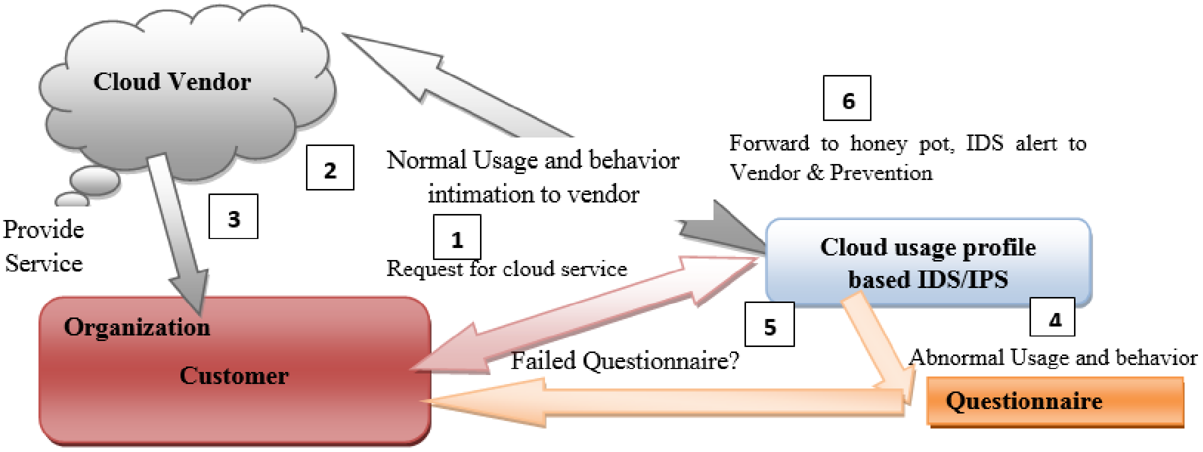

Cloud Vendor
2
Normal Usage and behavior intimation to vendor
6
Forward to honey pot, IDS alert to Vendor & Prevention
3
Provide Service
1
Request for cloud service
Cloud usage profile based IDS/IPS
5
4
Organization
Customer
Failed Questionnaire?
Abnormal Usage and behavior
Questionnaire

Figure 2: Activity flow diagram of cloud usage profile based IDS/IPS

3.1 Intrusion Detection System

Usage profile prepared based on the customer usage logs and summary. Usage profile contains the parameter like regular usage timings, roles, privileges, service types and etc. It prepares initially based on the service level agreement and later on their regular usage. It continuously update in profile database at vendor side. Every time during service usage, it observes the customer against profile. System will calculate security risk percentage based on the many parameters and it has its own threshold value. If usage crosses the security threshold, then system consider it has misuse and later access will have to face questionnaire. Questionnaires are the predefined question asks to the customer/hacker against the misusage. Questionnaires may be like customer logo, vision, start date, nick name and any movement noted during service level agreement. Once questionnaires are failed to answer, then system considers this as an intrusion attack and forward to honey pot. Intrusion attacks reports to vendor and stops the cloud service for some movement. It may even show danger alert for that system. Hence it prevents the intrusion.

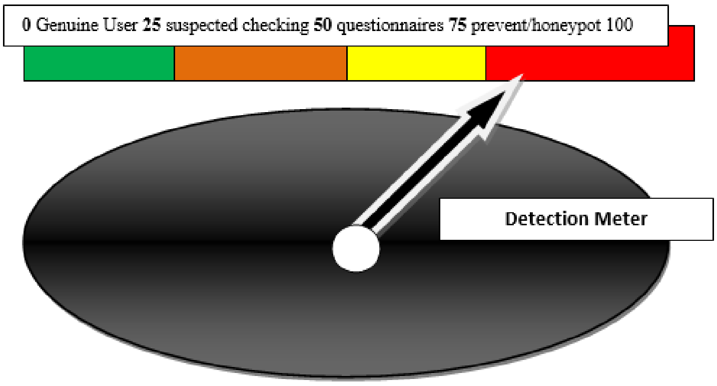

0 Genuine User 25 suspected checking 50 questionnaires 75 prevent/honeypot 100
Detection Meter

Figure 3: Intrusion Detection Meter

Table 3: Different States and it ranges

Sl No	Range Consider	State consider	Description
1	0-25	Genuine Usage State	We observe normal usage
2	25-50	Suspected State	We detect abnormal usage and check against usage profiles.
3	50-75	Questionnaire State	Usage check against predefined questions
4	75-100	Intruder/Prevention state	Honey pot to collect some details /Prevent the usage via active response/ Inform Vendor/Change attacked resource

3.2 Intrusion Detection Meter

This section briefs about the detection meter which we are proposing. As shown in fig 3, detection meter helps system in detecting the intrusion. Initially detection meter state will be in green state (0-25) which is safe usage state. With respect to usage, if we come across any variation or abnormality, detection put in orange state (i.e 25-50). If usage continue with the same state for some time, then detection meter push forward to yellow state (i.e 50 – 75) called questionnaire. Here user will face some predefined questionnaire (increases the authentication level of security) like, registered mobile number, company logo and etc. If it successfully answers, detection meter will move backward to one stage and again observe the usage, if it consider genuine then it moves to green state. If it not consider genuine, then it move forward to red to ask some more questionnaires. Finally it will reach red state (i.e 75-100) it is a honey pot, prevention and active response state. In this state, it collects the hacker details, signature and usage then it will forward to vendor intrusion prevention system. It performs active response and update/change the targeted resource.

3.3 Gathering Information using Honey Pot as a Service

This section briefs how honey pot useful in gathering hacker details. In cloud, one can setup the honey pot. Honey pot could be a virtual network / computer system/service which is expressly set up to attract and "trap" people who attempt to break through other people's computer systems. It could be designed with weak/no security and no confidential information to lure potential hackers. It can also have recording feature to observe the moment of hackers and to record entire actions. This may helps us to track the hacker, host and hacker signatures etc. It will also help to safe guard the actual confidential system. Fig 4 shows honey pot system in usage profile based IDS/IPS. As illustrates in figure 4 Honey pot as a service, Honey pot is a detection and response tool, rather than prevention. Honey pots cannot prevent a particular intrusion or spread of virus or worm, it purely collects information and detects attack patterns. Honey d detects and logs any connection to any UDP or TCP ports. It helps cloud vendor to add in block listed signature data base.

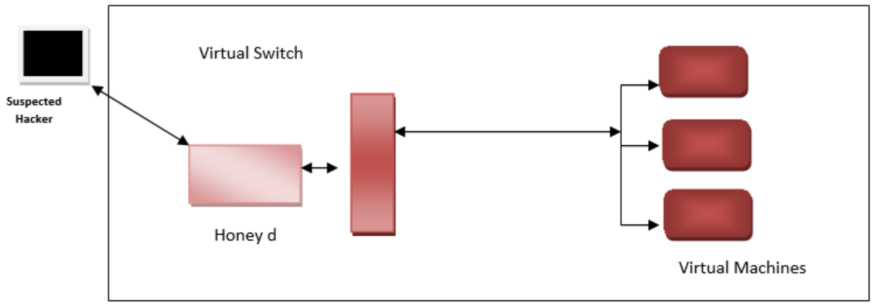


Figure 4: Honey pot as a service

3.4 Intrusion Prevention System

Usage profile based IPS will give active response to intruder/vendor by updating the policies and signatures. It also modifies the destination entity which was tried for attack. Cloud vendor can view the logs and records information given by honey pot recorded system to take safety action in future. Below example shows the usage profile based, IDS/IPS. Figure 5 shows the usage profile IDS/IPS action flow where based on customer usage observation against profile and limits detection and prevention action will be taken care using honey pot.

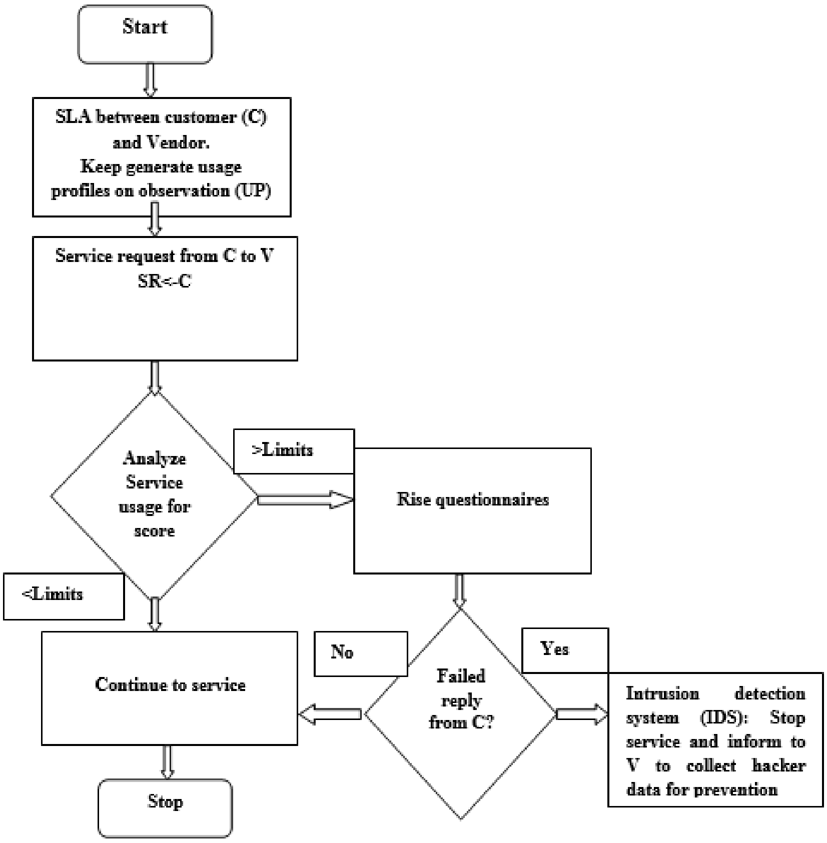

Start
SLA between customer (C) and Vendor. Keep generate usage profiles on observation (UP)
Service request from C to V SR<-C
Analyze Service usage for score
>Limits
Rise questionnaires
<Limits
Continue to service
No
Failed reply from C?
Yes
Intrusion detection system (IDS): Stop service and inform to V to collect hacker data for prevention
Stop

Figure 5: Usage profile based IDS and IPS system action flow

3.5 System Modeling using Petrinets

The system modeling is done using Petri nets, which are vividly portrayed in figure 6. Petri nets are a special form of bipartite directed graph represented by < P, T, In, Out> , in which Place (denoted as p) and Transitions (denoted as t) are disjoint sets of nodes, and In and Out are sets of edges. We carry out formal modeling for our system to precisely discover when the user can and cannot access the services of the cloud based usage profile. The model is explained as follows. As shown in figure 6, Cloud service accessed by customer represented by p1,p3,p14 and p15 places and t1, t2 and t3 transition. Intruder different states are represented in p6, p8, p9, p10 and p13 places and t6, t7, t8 and t9 transitions. Genuine state represented by p7 place. P16 places identify the intruder and trigger the different intruder states. Active responses done by p17 places and t10, t11 transitions. Figure 7, state diagrams of proposed system executes each service with this multistate model. Above model represents that it is a detection, prevention and active response technique immediately after the attack. Hence, It may be the solution for Insider attack, Flooding attack, User to Root attacks, Port Scanning Attack, Attacks on Virtual Machine and hypervisor backdoor channel attacks.

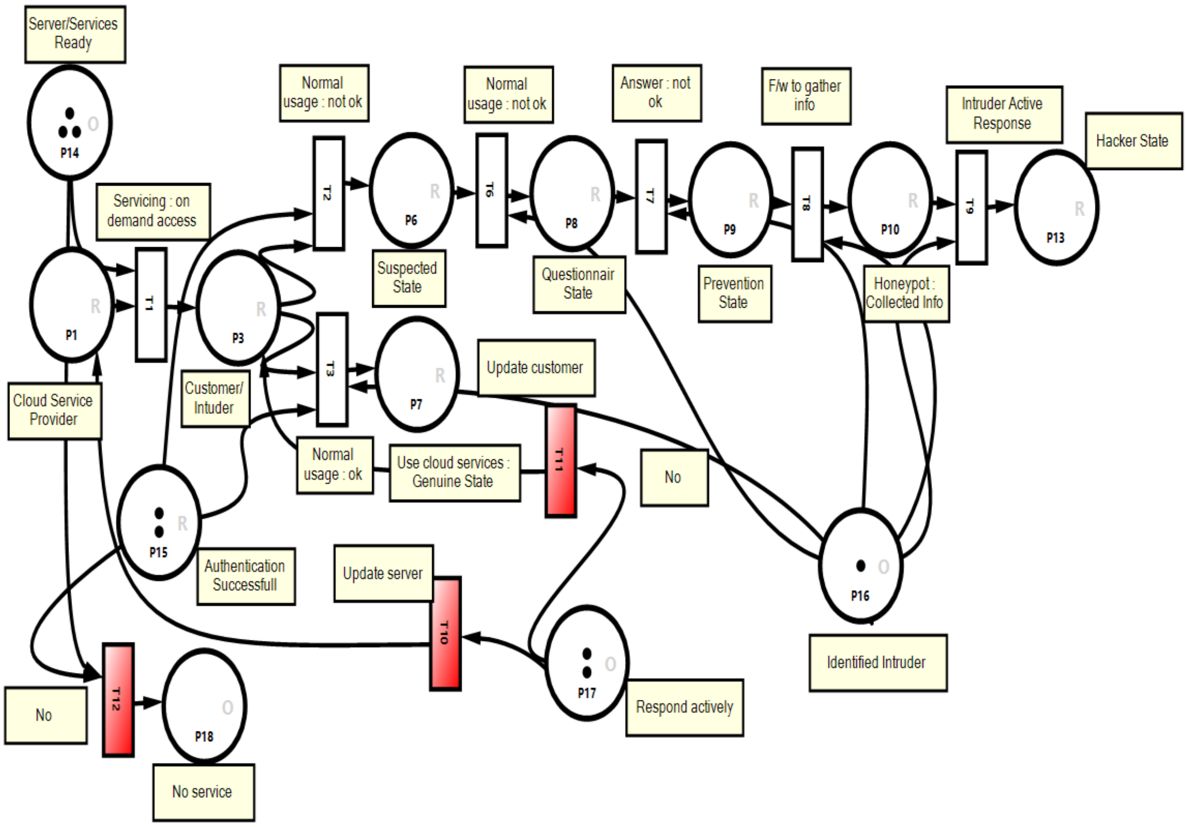


Figure 6: System modeling using Petri nets

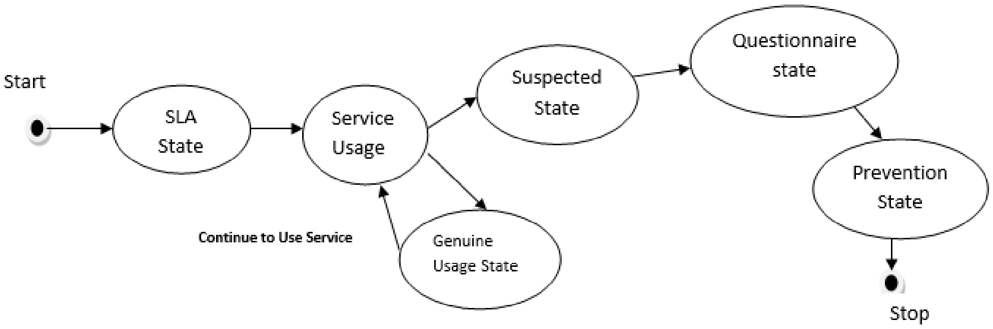


Figure 7: State diagram of the proposed system

4 Detailed Analysis of Proposed System

4.1 Assumption

Consider an IT industry as a customer who using cloud IaaS as usage service. Assuming that customer are using virtual machines for their software testing environment, proposed system prepare normal usage profiles. It is based on customer inputs/interaction during service level agreement and during regular usage. Recommended some of the usage profile parameters and corresponding percentage of value are described in below set.

Parameters P= {Location, Time, Device, Role, Mode, Duration, Running Apps, Settings} => {P1, P2, P3, P4, P5, P6, P7, P8}, Questionnaires Q= {Secrete Q&A, SMS, Email} => {Q1, Q2, Q3}

Table 4: Multi state parameter and meaning

Sl No	Parameters	Meaning	Violation	Meter Value
		Genuine State :By default state		
1	Location	What are the locations that were used for access?	sudden change in location (out of region, country etc)	25
2	Time	What are the normal usage timings?	unexpected change in time(midnight, after business hours, holidays etc)	25
3	Device	What are the devices that were using to avail VMs?	abrupt change in device (workstations, servers, mobiles, hacker specialized equipments and etc)	25
		Suspected State start from here (if above any one occur meter =25)		>=25
4	Role	What are likely access role for particular VMs?	rapid change in role(admin role, VM deletion role, VM deployment role and etc)	25
5	Mode	What is the operating mode normally?	gradual change in operating mode (Delete file, download, copy, use external devices and etc)	25
6	Duration	How long he used to have service normally?	swift change in usage time(long usage, repeated short intervals usages and etc)	25
7	Running Apps	What are the apps normally running?	hasty change in apps usage(accessing apps which are not required like BIOS setup, Control panels, and other admin apps)	25
8	Settings	What are the system settings normally updating?	Try to change the VM settings (hardware, advanced, security and etc)	25

Table 5: Recommended parameter for each state

		Questionnaire state start from here (meter =50)		>=50
Sl No	**Question**	**Meaning**	**Violation**	**Meter Value**
1	Secrete Q&A	Secrete question and answer used during password settings?	Not able to answer	25
2	SMS	Send SMS to registered mobile for one time password/secrete code too jump to normal usage?	Invalid Entry	25
3	Email	Send emails to register account with specific code to be entered?	Invalid value	25
		Prevention State :Honey pot start collecting the user details / device for further report (meter=75)		>=75

Note: Parameter and meter value can be changed depends on vendor security perspective.

Refer to this table4; we simplify genuine state 3 parameters, suspected state 5 parameters and 3 questionnaires as G [3], S [5] and Q [3] respectively.

{'0' Violation/Failure to answer {1 non violation/success to answer then in G[3] array if any one index got 0 then state move to S[5]. Similarly S[5] array if any one index got 0 then state move to Q[3]. Similar way if Q[3] array if any one index got 0 then the proposed system meter reach 75 and consider user as intruder. We can derive a Deterministic Finite Automata as shown in figure 8. Table 5, shows the transition table of derived DFA.

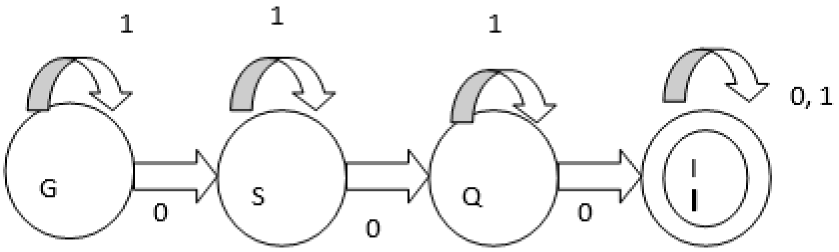


Figure 8: Deterministic Finite Automata to detect intruder'd'

Table 6: Transition table

States	0 Violation/ Un Answering	1 Non Violation/ Answering
G	S	G
S	Q	S
Q	I	Q
I	I	I

Regular Expression is as follows: Intruder -> d-> (a+b+c+) +

States = {G, S, Q, I} Transition {0, 1} Final state = I,

If input value is 000 then DFA push the state to final state known as I (Intruder). Below table represents the possible value with corresponding meaning.

Table 7: Finite Automata Value and Its Meaning (- Represents No Transition)

DFA Input	Meaning
000	Intruder, Honey pot state
1_ _	Safe state
100,101,110, 011,010	Invalid inputs
01_	Genuine user accessing from different way
001	Misbehavior of genuine user, can be consider as internal attack
0_ _, 00_ ,	Intruder stops their action abruptly

Proposed system can be solution for any attacks since it detect and prevent during service usage (suspected state). Drawback of proposed system damage which cause before detection cannot be avoided.

5 Conclusion and Future Enhancement

In the future work, Cloud computing has many benefits and more customer usage demand. It gives cost benefits by providing ready infrastructure and effective resource management. However, security is the main issue which needs to be resolved on priority basis. Intrusion detection and prevention systems are available in the literature. Specific to cloud security and intrusion, effective technique requires on high priority basis.

Cloud usage profile based intruder detection and prevention system prepares the usage profiles and check cloud customer usage against usage profiles. In turn, it report and prevents the intruder using intrusion detection meter, questionnaires and vendor reporting mechanisms. Hence it may be the solution for Insider attack, Flooding attack, User to Root attacks, Port Scanning Attack, Attacks on Virtual Machine and hypervisor backdoor channel attacks.

ACKNOWLEDGMENT

Our sincere thanks to Prof. K N B Murthy, Principal and Prof. Shylaja S S, HOD, Department of Information Science and Engineering, PESIT, Bangalore, for their constant encouragement.

REFERENCES

[1]. C. B. Westphall and F. R. Lamin. SLA Perspective in Security Management for Cloud Computing. In Proc. of the Int. Conf. on Networking and Services (ICNS), 2010. Pp. 212-217.

[2]. Hisham A. Kholidy, Fabrizio Baiardi CIDS: A framework for Intrusion Detection in Cloud Systems, 2012 Ninth International Conference on Information Technology- New Generations, 978-0-7695-4654-4/12 $26.00 © 2012,pp 379-385.

[3]. Karen Scarfone and Peter Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)", National Institute of Standards and Technology(NIST), Special Publication 800-94, Feb. 2007.

[4]. J.H. Lee, M.W. Park, J.H. Eom, T.M. Chung, "Multi-level Intrusion Detection System and Log Management in Cloud Computing", In *13th International Conference on Advanced Communication Technology*, pp.552-555, 2011.

[5]. H. Jin, G. Xiang, D. Zou et al., "A VMM-based intrusion prevention system in cloud computing environment," The Journal of Supercomputing, pp. 1–19, 2011

[6]. T. Udaya, V. Vijay, and A. Naveen, "Intrusion detection techniques for infrastructure as a service cloud," in Proceedings of the 9th IEEE International Conference on Dependable, Autonomic and Secure Computing, IEEE Computer Society, pp. 744–751, Sydney, Australia, 2011.

[7]. W. Cong, W. Qian, R. Kui, and L. Wenjing, "Ensuring data storage security in cloud computing," in Proceedings of the 17th International Workshop on Quality of Service (IWQoS '09), pp. 1–9, July 2009

[8]. J. Arshad, P. Townend, and J. Xu, "An automatic intrusion diagnosis approach for clouds," International Journal of Automation and Computing, vol. 8, pp. 286–296, 2011.

[9]. P. Angin, B. Bhargava, R. Ranchal et al., "An entity-centric approach for privacy and identity management in cloud computing," in Proceedings of the 29th IEEE Symposium on Reliable Distributed Systems (SRDS '10), pp. 177–183, November 2010.

[10]. Bharadwaja, S. Weiqing, M. Niamat, and S. Fangyang, “Collabra: a xen hypervisor based collaborative intrusion detection system,” in Proceedings of the 8th International Conference on Information Technology: New Generations (ITNG '11), pp. 695–700, Las Vegas, Nev, USA, 2011.

[11]. Borisaniya, A. Patel, D. Patel et al., “Incorporating honeypot for intrusion detection in cloud infrastructure,” in Trust Management VI, vol. 374, pp. 84–96, Springer, Boston, Mass, USA, 2012.

[12]. L. Flavio and P. Roberto Di, “Secure virtualization for cloud computing,” Journal of Network and Computer Applications, vol. 34, no. 4, pp. 1113–1122, 2011.

[13]. Gupta, S. Horrow, and A. Sardana, “IDS based defense for cloud based mobile infrastructure as a service,” in Proceedings of the 8th IEEE World Congress on Services (SERVICES), pp. 199–202, Honalulu, Hawaii, USA, 2012.

[14]. R. Ranchal, B. Bhargava, L. B. Othmane et al., “Protection of identity information in cloud computing without trusted third party,” in Proceedings of the 29th IEEE Symposium on Reliable Distributed Systems (SRDS '10), pp. 368–372, November 2010.

[15]. A. S. Ibrahim, J. Hamlyn-Harris, J. Grundy, and M. Almorsy, “CloudSec: a security monitoring appliance for Virtual Machines in the IaaS cloud model,” in Proceedings of the 5th International Conference on Network and System Security (NSS '11), pp. 113–120, 2011.

[16]. T. J. Arshad and J. Xu, “A novel intrusion severity analysis approach for Clouds,” Future Generation Computer Systems, vol. 28, pp. 965–1154, 2011.

[17]. F.Rocha,M. Correia,2011,Lucy in the sky without diamonds: Stealing confidential data in the cloud.

[18]. Anup ghosh, Chrish greamo, page 79-82, 2011, “Sandboxing and Virtualization”, Security and privacy,IEEE.

[19]. Islam M. Hegazy, Taha Al-Arif, Zaki.,T. Fayed, and Hossam M. Faheem ,Oct-Nov 2003,”Multi-agent based system for intrusion Detection” ,Conference Proceedings of ISDA03, IEEE.

[20]. Hisham A. Kholidy, Fabrizio Baiardi, 2012 CIDS: “A Framework for Intrusion and Detection in cloud Systems”, 9th International Conference on Inform- ation Technology- New Generations,IEEE.

[21]. Frank Doelitzscher∗, Christoph Reich∗, MartinKnahl and Nathan Clarke, p197-204, 2011,”An autonomous agent based incident detection system for cloud environments”, 3rd IEEE International Conference

[22]. Modi, C., Patel, D., Patel, H., Borisaniya, B., Patel, A. & Rajarajan. (2012). A survey of intrusion detection techniques in Cloud. Journal of Network and Computer Applications, doi: 10.1016/j.jnca.2012.05.003

[23]. C. B. W. C. M. W. K. M. VIEIRA, A. SCHULTER, “Intrusion detection techniques in grid and cloud computing environment,” *IEEE IT Professional Magazine*, 2010.

[24]. S. Roschke, C. Feng, and C. Meinel, “An Extensible and Virtualization Compatible IDS Management Architecture,” *Fifth International Conference on Information Assurance and Security*, vol. 2, 2009, pp.130-134.

[25]. A.bakshi, and B. Yogesh, “Securing Cloud from DDOS Attacks Using Intrusion Detection System in Virtual Machine,” Second International Conference on Communication Software and Networks, 2010, pp. 260-264

[26]. L. Fagui Liu, S. Xiang Su, and L. Wenqianl, "The Design and Application of Xen-based Host System Firewall and its Extension," in The 2009 International Conference on Electronic Computer Technology, 2009, pp. 392-395.

[27]. C. C. Lo, C. C. Huang, and J. Ku, “Cooperative Intrusion Detection System Framework for Cloud Computing Networks,” First IEEE International Conference on Ubi-Media Computing, 2008, pp. 280-284.

[28]. K. A. B. A. V. Dastjerdi, and S. G. H. Tabatabaei, “Distributed intrusion detection in clouds using mobile agents,” in Third International Conference on Advanced Engineering Computing and Applications in Sciences, 2009. ADVCOMP ’09, 2009, pp. 175 – 180.

[29]. Y. Guan, and J. Bao, “A CP Intrusion Detection Strategy on Cloud Computing,” In International Symposium on Web Information Systems and Applications (WISA), pp. 84–87, 2009.

[30]. C. Mazzariello, R. Bifulco, and R. Canonoco, “Integrating a network IDS into an Open source Cloud computing,” Sixth International conference on Information Assurance and Security (IAS), 2010, pp. 265-270.

TNC Transactions on Networks and Communications

Society for Science and Education United Kingdom

Volume 2, Issue 6
ISSN: 2054 - 7420

Optimization of WSNs Flooding Rates by Khalimsky Topology

[1]Mahmoud Mezghani, [2]Riyadh Gargouri and [3]Mahmoud Abdellaoui
[1]National Engineering School of Sfax (ENIS), Sfax University, Tunisia;
[2]Department of Sciences Taibah University, Saudi Arabia;
[3]Engineering School of Electronics and Communication of Sfax (ENET'COM), Sfax University, Tunisia;
[1]mahmoud.mezghani@gmail.com; [2]tn_riadh_30@yahoo.fr; [3]mahmoudabdellaoui4@gmail.com

ABSTRACT

In this paper, we proposed a new method of deploying and building an organized architecture of gateway nodes in a Wireless Sensors Network (WSN) formed also by randomly deployed sensors arranged in clusters. This method, based on the Khalimsky theory, reduces the energy consumption and the flooding rates of the conventional flooding algorithm.

Our solution allowed to build a hierarchical topology reduces the number of communication links between nodes while maintaining the adjacency list to minimize the data redundancy.

It allows calculating an optimal set of forwarding gateway nodes to route data packets between a source and a destination. This set includes all optimal paths using the less number of nodes. Our simulation study shows that the Khalimsky theory reduces considerably the flooding rates and the energy consumption.

Our solution can be applied in the fields needing prefixed gateway nodes and randomly deployed sensors which use urgent data with real-time aspect such as the security and the monitoring of industrial zones and home building.

Keywords: Wireless Sensor Network; WSN; Flooding; Khalimsky Topology; Flooding Khalimsky Algorithm; Flooding Rates; Energy Consumption; WSN Deployment.

1 Introduction

Technological advances in recent years have enabled the development of new types of sensors equipped with wireless communication system that can be configured to form autonomous networks called WSNs. These WSNs are used in various domains such as home automation, health care, military domain and environment monitoring. In the network, the sensors nodes cooperate between them to treat and to convey the collected data towards the base station which acts according to programmed scenarios. These sensors are characterized by a small memory size, a low capacity for treatment and limited energy resource [1, 2].

Generally, dissipated energy by a sensor is the total of energies used to gather, treat and send/receive data. Most of this energy is used by the radio module in sending/receiving information. The main focus of WSN research has been on energy efficiency. The objective is the development of topologies and routing protocols allowing the reduction of the dissipated energy by the radio component. We should note that the effectiveness of a routing protocol depends on the architecture of sensors nodes deployment and their operating processes. For that, flooding is very used for the WSN architecture construction. It identifies relationships between nodes and their

DOI: 10.14738/tnc.26.598
Publication Date: 16th November 2014
URL: http://dx.doi.org/10.14738/tnc.26.598

positions to calculate and maintain the routing tables. It consumes a lot of energy and needs to be refined and optimized to eliminate unnecessary data redundancy [3, 4].

Accordingly, we devote our research tasks to adapt scientific advances research in the fields of mathematics such as Khalimsky theory and the multi-agents systems to carry out a sensors deployment method for WSNs reducing the flooding rates and minimizing the energy consumption. Our idea consists to create a WSN formed by a set of fixed sensors used as gateway nodes based on the Khalimsky topology and a set of clusters containing member nodes to gather information. The Khalimsky topology permits to convey the flooding process using optimal paths between gateway nodes and the sink when a node transmits a data packet. The multi agents system permits to assign the jobs for each node in the clusters and in the set of the gateway nodes to complete a task of collecting and transmitting information between a source and a destination. Our objective is to minimize the flooding rates by the limiting of the flooding in the set of the gateways, based on the Khalimsky architecture, using only those which form optimal paths.

In this paper, we present Khalimsky topology used to build the architecture of deployed gateways in a WSN. This topology provides a hierarchical architecture in which each sensor node is capable to convey data to their destination in an optimal set of nodes minimizing the flooding rates. In the second time, we present the evaluation results of this method compared with conventional flooding algorithm. This evaluation is carried out by the TOSSIM simulator and the PowerTOSSIM-Z module which allows to model power consumption by sensors nodes using the TinyOS2.1 operating system [5-7].

The remainder of this paper is organized as follows: section 2 presents an outline on WSN deployment architectures, types of routing protocols and flooding problems such energy consumption and information redundancies. Section 3 shows the Khalimsky topology characteristics and the auto-organization algorithm of sensor nodes. Section 4 shows the novel flooding algorithm according to Khalimsky theory. Section 5 presents the simulation and evaluation results of the developed algorithms. Section 6 presents some future works. Finally, section 7 gives some conclusions.

2 Related Work

Several research studies have appeared to optimize the energy consumption of nodes through the use of innovative conservation methods to improve network performance, including the maximization of its life. Since the radio module is the greediest component, the trend was to reduce the communications rates by optimizing the data routing to reduce the energy consumption. Consequently, several routing protocols have been developed. However, they cannot be applied appropriately to any application context. Depending on the architecture, the deployed sensors take place by a random or organized way. In the organized way, sensors must be deployed in predetermined positions where it is possible to program their activities and data packets routes. In the random way, the nodes are scattered on the collecting field in mass by several means such as throwing them by plane. In this mode, an auto-organization algorithm is necessary to improve the performance of the WSN [8].

In the literature, WSNs routing protocols can be classified in three main classes: flat, hierarchical and geographic network routing protocols.

In flat architecture, each node plays the same role and has the same functionality as other sensor nodes in the network. Sensors can send data to the base station directly using a high power, or hop by hop with a very low power. The most popular flat-based routing protocol is data-centric protocols like flooding. In this type of protocols, it is not necessary to have an addressing mechanism for sensor nodes. Data is propagated gradually to the interested neighbors requiring the data announced or in flood towards selected or all the neighbors with an important redundancy until reaching the base station [8-10]. The unnecessary flooded data packets, the not optimum paths, the data transmission time and the different ratios of energy consumption between sensor nodes are the most disadvantages in the dense flat networks.

In hierarchical architecture, the node representing the cluster-head transmits directly collected data from member-nodes to the base station, or via a multi-hop mode between the cluster-heads. The cluster-heads are generally a sensors more powerful than the member nodes or sensors whose battery level is the highest in the cluster. This type of protocol is the most adopted today. It minimizes more the energy consumption compared to flat architecture and it has two great advantages: scalability and the simple implementation of the aggregation mechanisms. Leach, TEEN and PEGASIS are the most popular hierarchical routing protocols for WSNs [8-10]. The data propagation between the cluster-heads and the sink uses multiple paths which are not necessary optimized. It requires an auto-organization algorithm to select cluster heads and to affect member nodes to the calculated cluster.

In a geographic routing protocol, a node is supposed to know its geographical position, its neighbors and the destination node. To transmit data to the sink, the nearest node destination, among the neighboring nodes, is selected as the next hop. Multiple geographic routing protocols like GPSR, SPEED, GEAR, GAF ... are proposed. In such protocol, the routing data is optimized and the cost to control the algorithm is reduced. However, the disadvantage of geographic-based routing protocols is the means used for nodes localization. The localization is ensured by the exchange of a big number of messages by flooding algorithm to discover nodes between them [8-10].

The flooding mechanism is used by several routing protocols for sensor nodes organization. It permits the routes setup, the neighboring discovery for each node and the links state updates. Such mechanism suffers from a huge amount of information redundancy due to the big number of exchanged auto-organization data packets. However, the reduction of the flooding rates is imperative in order to reduce data redundancy and to minimize energy consumption. For all these reasons, we devoted this paper to the communications energy efficiency in WSNs. The main objective of our work consists to develop a WSN auto-organization method optimizing the communication links between sensors and reducing the flooding rates. For that, we used the mathematical Khalimsky theory to construct a geographic-hierarchical topology [11-15]. This theory permits to calculate the optimized paths when a sensor need to forward data. Theses paths are formed by a succession of nodes belonging to an employed optimal set of gateways between a source and destination.

3 WSN Deployment according to Khalimsky Topology

In this section we present the Khalimsky theory and the developed algorithms for the WSN deployment and the flooding of data packets between the sensors and the sink.

3.1 Khalimsky theory definitions

In what follows, we present some terminology and definitions necessary for understanding the results of the Khalimsky topology.

Let $\mathcal{B} = \{\{2n+1\};\ n \in \mathbb{Z}\} \cup \{\{2n-1, 2n, 2n+1\}; n \in \mathbb{Z}\}$ be a family of subsets of the set of integers$\mathbb{Z}$. Then $\mathcal{B}$ is a base of a topology on $\mathbb{Z}$ called the Khalimsky topology and denoted by K. The Khalimsky plane is the Cartesian product of two Khalimsky topologies $(\mathbb{Z}, K) \times (\mathbb{Z}, K)$. This topology on $\mathbb{Z}^2$ is characterized as follows. A point at its two coordinates of the form $(2k; 2k')$ is closed point (they are shown in figure 1 by the black squares), if both coordinates are of the form $(2k+1; 2k'+1)$ point is an open point (shown in the figure 1 by the white squares). These types of points are called "*pure points*". Other points are called *"mixed points"* (presented by two lines). It is clearly that $(\mathbb{Z}^2, k)$ is connected space [11-15]. Figure 1 show a connectivity graph of the Khalimsky topology.

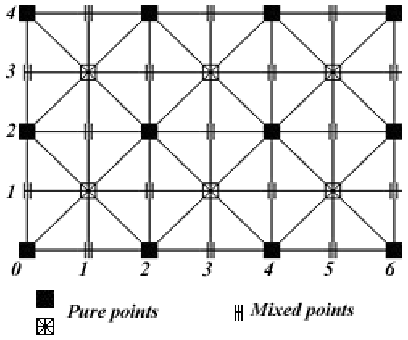


Figure 1: Connexity graph part of Khalimsky topology on $\mathbb{Z}^2$

According to the Khalimsky topology, we define:

- $S_{(X_S, Y_S)}$: Source point.
- $Q_{(X_Q, Y_Q)}$: Destination point.
- A_S^Q : A digital arc joining the points S and Q.
- M_S^Q : A minimal digital arc joining the points S and Q.
- $BL(S,Q) = \cup\, M_S^Q$: The whole of minimal digital arcs joining the points S and Q.
- $d_\infty(S,Q) = max(|X_Q - X_S|, |Y_Q - Y_S|)$: The norm in $\mathbb{Z}^2$

According to these definitions, we define:

- $S_{(X_S, Y_S)}$: Source sensor node.
- $Q_{(X_Q, Y_Q)}$: Destination sensor node.
- A_S^Q : A link regrouping a set of intermediate sensors joining two sensor nodes S and Q.
- M_S^Q : A minimal link joining the sensor nodes S and Q.
- $BL(S,Q) = \cup\, M_S^Q$: The whole of minimal link joining the sensor nodes S and Q.
- $d_\infty(S,Q)$: The number of sensors in the link joining nodes S and Q.

3.2 Khalimsky deployment algorithm

According to the previous definitions, our algorithm permits the deployment of the gateway sensors in the collecting field as the Khalimsky topology as shown in figure 2. The gateway nodes in this topology are either pure or mix. Pure nodes have 8 neighbors and mix nodes have only 4 neighbors. Mix nodes communicate with pure nodes only. The nodes are organized in levels. The sink $S_{(0,0)}$ belongs to the level *0* its neighbors belong to the level *1* and respectively until level *n*.

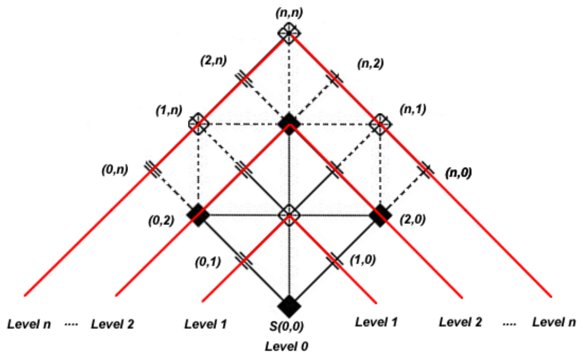


Figure 2: Sensors coordinates assignment in the Khalimsky topology

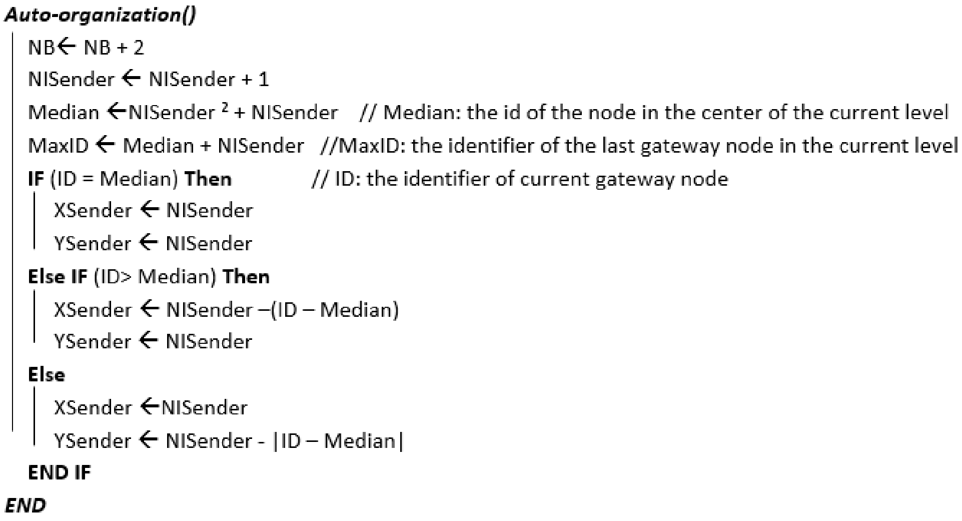
To create the topology, the sink $S_{(0,0)}$ diffuses a message towards all the nodes in its range to allow the coordinates between them. This message is a structure composed by five variables:

- IDSender : the identifier of the sender node,
- (XSender, YSender): the coordinates of the sender node,
- NISender : the level of the sender in the Khalimsky hierarchy,
- NB: the nodes number in the current level of the sender node.

The NB variable is calculated by the following (U_n) :

$$U_n = \begin{cases} U_0 = 1 \\ U_{n+1} = U_n + 2 \end{cases} \tag{1}$$

Each node receiving the message, run the auto-organization algorithm as shown in the following algorithm to calculate its coordinates of the auto-organization structure, and then diffuses them towards the nodes in its range.



```
Auto-organization()
  NB ← NB + 2
  NISender ← NISender + 1
  Median ← $NISender^2$ + NISender    // Median: the id of the node in the center of the current level
  MaxID ← Median + NISender   //MaxID: the identifier of the last gateway node in the current level
  IF (ID = Median) Then          // ID: the identifier of current gateway node
     XSender ← NISender
     YSender ← NISender
  Else IF (ID> Median) Then
     XSender ← NISender –(ID – Median)
     YSender ← NISender
  Else
     XSender ←NISender
     YSender ← NISender - |ID – Median|
  END IF
END
```

Algorithm 1: Gateway sensor nodes auto-organization algorithm

4 Khalimsky Flooding Algorithm

In order to avoid data redundancy and the problems due to the implosion and the overlapping caused by the conventional flooding approach, we developed a novel algorithm based on the Khalimsky topology which reduces the field of used nodes when a source sensor $Q_{(X_Q,Y_Q)}$ broadcasts data to the sink $S_{(0,0)}$. This field is defined by $BL(S,Q) = \cup M_S^Q$ the set of nodes forming all minimal links which connect the source node $Q_{(X_Q,Y_Q)}$ to the sink $S_{(0,0)}$. Figure 3 shows $BL(S,Q)$ of a nodes set used when a sensor $Q_{(X_Q,Y_Q)}$ broadcasts data to the sink $S_{(0,0)}$.

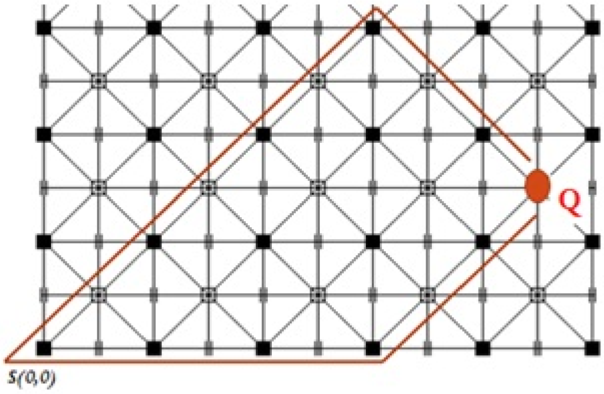


Figure 3: $\mathrm{BL}(\mathrm{S},\mathrm{Q})$ of nodes belonging minimal paths between $\mathrm{Q}_{(\mathrm{X_Q},\mathrm{Y_Q})}$ and $\mathrm{S}_{(0,0)}$

4.1 $BL(S,Q)$ Construction

Khalimsky topology distinguishes five cases to determine the lists of nodes to traverse to form the minimal paths separating two nodes $\mathrm{Q}_{(\mathrm{X_Q},\mathrm{Y_Q})}$ and $\mathrm{S}_{(0,0)}$ [11-15]. These nodes are in the zone I; II; III or IV; as well as, on the lines $\mathrm{D}_{(\mathrm{Y}=\mathrm{X})}$or $\mathrm{D}_{(\mathrm{Y}=-\mathrm{X})}$ as shown in figure 4.

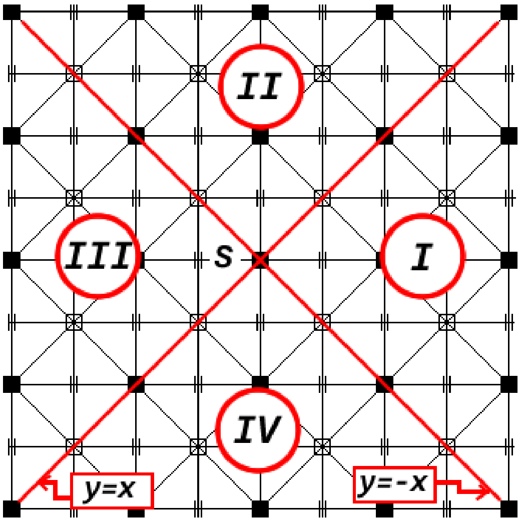


Figure 4: Minimal path zones in Khalimsky topology on $\mathbb{Z}^2$

In the following, P represents an intermediate sensor node in the optimal whole of sensors joining two nodes $S_{(X_S,Y_S)}$ and $\mathrm{Q}_{(\mathrm{X_Q},\mathrm{Y_Q})}$.

- **Case 1 :** If S is a pure node and $|X_Q - X_S| = |Y_Q - Y_S|$ then Q is a pure node and

$$\mathrm{P} \in \mathrm{BL}(\mathrm{S},\mathrm{Q}) \Leftrightarrow \begin{cases} |\mathrm{X_P} - \mathrm{X_S}| + |\mathrm{X_Q} - \mathrm{X_p}| = |\mathrm{X_Q} - \mathrm{X_S}| \\ |\mathrm{Y_P} - \mathrm{Y_S}| + |\mathrm{Y_Q} - \mathrm{Y_p}| = |\mathrm{Y_Q} - \mathrm{Y_S}| \end{cases} \qquad (2)$$

  The set of sensors joining nodes $\mathrm{S}_{(0,0)}$ and $\mathrm{Q}_{(\mathrm{X_Q},\mathrm{Y_Q})}$ are on the lines $\mathrm{D}_{(\mathrm{Y}=\mathrm{X})}$or $\mathrm{D}_{(\mathrm{Y}=-\mathrm{X})}$.

- **Case 2 :** If S is a mixed node and $|X_Q - X_S| = |Y_Q - Y_S|$ then Q is a mixed node and

$$\mathrm{P} \in \mathrm{BL}(\mathrm{S},\mathrm{Q}) \Leftrightarrow \mathrm{BL}(\mathrm{S},\mathrm{Q}) = \mathrm{BL}(\mathrm{S}_1,\mathrm{Q}_1) \cup \mathrm{BL}(\mathrm{S}_2,\mathrm{Q}_2) \cup \{\mathrm{S},\mathrm{Q}\} \qquad (3)$$

$$\text{With} \quad \mathrm{S}_1\left(\mathrm{X_S}, \mathrm{Y_S} - \frac{\mathrm{Y_S}-\mathrm{Y_Q}}{|\mathrm{Y_S}-\mathrm{Y_Q}|}\right),\ \mathrm{S}_2\left(\mathrm{X_S} - \frac{\mathrm{X_S}-\mathrm{X_Q}}{|\mathrm{X_S}-\mathrm{X_Q}|}, \mathrm{Y_S}\right),$$

$$\mathrm{Q}_1\left(\mathrm{X_Q} - \frac{\mathrm{X_Q}-\mathrm{X_S}}{|\mathrm{X_Q}-\mathrm{X_S}|}, \mathrm{Y_Q}\right), \mathrm{Q}_2\left(\mathrm{X_Q}, \mathrm{Y_Q} - \frac{\mathrm{Y_Q}-\mathrm{Y_S}}{|\mathrm{Y_S}-\mathrm{Y_Q}|}\right)$$

When $S_{(X_S, Y_S)}$ and $Q_{(X_Q, Y_Q)}$ area mixed node, it's necessary to join two pure node from (S_1, Q_1) and (S_2, Q_2) to calculate$BL(S, Q)$. Then the set of nodes joining S and Q is the union of determined nodes between (S_1, Q_1) and(S_2, Q_2), and the nodes S and Q as shown in figure 5.

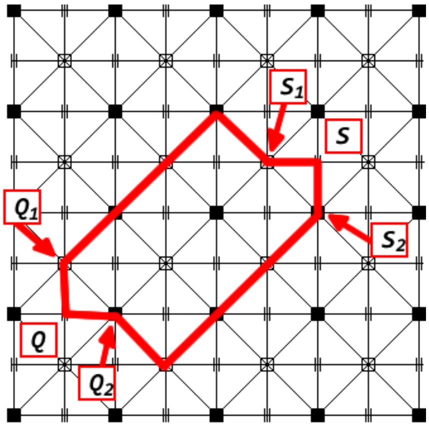


Figure 5: BL (S, Q) when S and Q are mixed nodes

- **<u>Case 3 :</u>** If S and Q are pure nodes and $|X_S - X_Q| \neq |Y_S - Y_Q|$, then there is two cases :
  a. If $|X_S - X_Q| > |Y_S - Y_Q|$ then

$$\mathrm{P} \in \mathrm{BL(S,Q)} \Leftrightarrow \begin{cases} |\mathrm{X_P - X_S}| + |\mathrm{X_Q - X_p}| = |\mathrm{X_Q - X_S}| \\ |\mathrm{Y_P - Y_S}| \leq |\mathrm{X_P - X_S}| \\ |\mathrm{Y_Q - Y_p}| \leq |\mathrm{X_P - X_Q}| \end{cases} \tag{4}$$

  In this case, the nodes joining S and Q are in zone I or III as shown in figure 4.

  b. If $|X_S - X_Q| < |Y_S - Y_Q|$ then

$$\mathrm{P} \in \mathrm{BL(S,Q)} \Leftrightarrow \begin{cases} |\mathrm{Y_P - Y_S}| + |\mathrm{Y_Q - Y_p}| = |\mathrm{Y_Q - Y_S}| \\ |\mathrm{X_P - X_S}| \leq |\mathrm{Y_P - Y_S}| \\ |\mathrm{X_Q - X_p}| \leq |\mathrm{Y_P - Y_Q}| \end{cases} \tag{5}$$

  In this case, the nodes joining S and Q are in zone II or IV as shown in figure 4.

- **<u>Case 4 :</u>** If S and Q are not both pure nodes or mixed nodes and $|X_S - X_Q| \neq |Y_S - Y_Q|$, then there is two cases :
  a. S is a pure node and Q is a mixed node and $|X_S - X_Q| > |Y_S - Y_Q|$, then

$$\mathrm{P \in BL(S,Q) \Leftrightarrow\ P\ \in BL(S,Q') \cup \{Q\}} \tag{6}$$

with $\mathrm{Q'}\left(\mathrm{X_Q} - \frac{\mathrm{X_Q - X_S}}{|\mathrm{X_Q - X_S}|}, \mathrm{Y_Q}\right)$ a pure node as shown in figure 6.

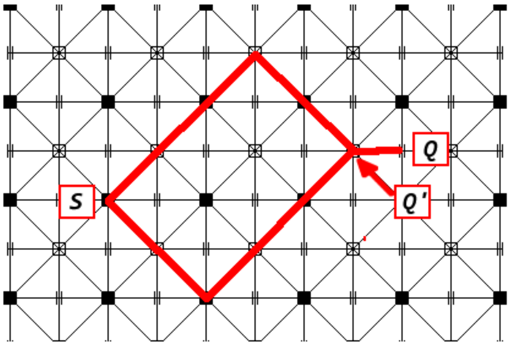


Figure 6: $\mathbf{BL(S,Q)}$ when S is a pure node and Q is a mixed node and $|X_S - X_Q| > |Y_S - Y_Q|$

  b. S is a pure node and Q is a mixed node and $|X_S - X_Q| < |Y_S - Y_Q|$, then

$$\mathrm{P \in BL(S,Q) \Leftrightarrow\ P\ \in BL(S,Q') \cup \{Q\}} \tag{7}$$

With $\mathrm{Q}'\left(\mathrm{X_Q}, \mathrm{Y_Q} - \frac{\mathrm{Y_Q}-\mathrm{Y_S}}{|\mathrm{Y_S}-\mathrm{Y_Q}|}\right)$ a pure node as shown in figure 7.

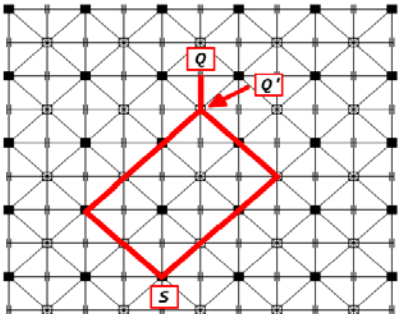


Figure 7: $\mathbf{BL(S,Q)}$ when S is a pure node and Q is a mixed node and $|X_S - X_Q| < |Y_S - Y_Q|$

- **Case 5 :** If S and Q are both mixed nodes and $|X_S - X_Q| \neq |Y_S - Y_Q|$, then there is two cases:
  a. If $|X_S - X_Q| > |Y_S - Y_Q|$, then

$$\mathrm{P} \in \mathrm{BL(S,Q)} \Leftrightarrow \mathrm{P} \in \mathrm{BL(S',Q')} \cup \{\mathrm{S},\mathrm{Q}\} \qquad (8)$$

With $\mathrm{S}'\left(\mathrm{X_S} - \frac{\mathrm{X_S}-\mathrm{X_Q}}{|\mathrm{X_S}-\mathrm{X_Q}|}, \mathrm{Y_S}\right)$ and $\mathrm{Q}'\left(\mathrm{X_Q} - \frac{\mathrm{X_Q}-\mathrm{X_S}}{|\mathrm{X_S}-\mathrm{X_Q}|}, \mathrm{Y_Q}\right)$ are pure nodes as shown in figure 8.

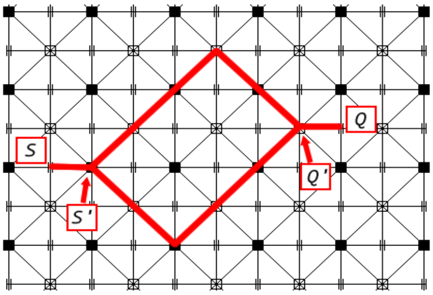


Figure 8: $\mathbf{BL(S,Q)}$ when S and Q are both mixed nodes and $|X_S - X_Q| > |Y_S - Y_Q|$

  b. If $|X_S - X_Q| < |Y_S - Y_Q|$, then

$$\mathrm{P} \in \mathrm{BL(S,Q)} \Leftrightarrow \mathrm{P} \in \mathrm{BL(S',Q')} \cup \{\mathrm{S},\mathrm{Q}\} \qquad (9)$$

With $\mathrm{S}'\left(\mathrm{X_S}, \mathrm{Y_S} - \frac{\mathrm{Y_Q}-\mathrm{Y_S}}{|\mathrm{Y_S}-\mathrm{Y_Q}|}\right)$ and $\mathrm{Q}'\left(\mathrm{X_Q}, \mathrm{Y_Q} - \frac{\mathrm{Y_Q}-\mathrm{Y_S}}{|\mathrm{Y_S}-\mathrm{Y_Q}|}\right)$ are pure nodes as shown in figure 9.

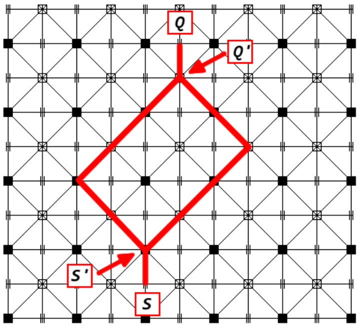


Figure 8: $\mathbf{BL(S,Q)}$ when S and Q are both mixed nodes and $|X_S - X_Q| < |Y_S - Y_Q|$

4.2 Khalimsky flooding algorithm

The Khalimsky flooding algorithm determines the relay nodes progressively for transmitting data packets from a source gateway node $\mathrm{Q}_{(\mathrm{X_Q},\mathrm{Y_Q})}$ to the sink node $\mathrm{S}_{(0,0)}$ in$\mathrm{BL(S,Q)}$. Each node receiving a package, diffuses it once towards the nodes which it has calculated their coordinates. For each node in $\mathrm{BL}(S,Q)$, there is one, two or three gateway nodes to forward data in each step according to the Khalimsky theory as shown in the figure 9.

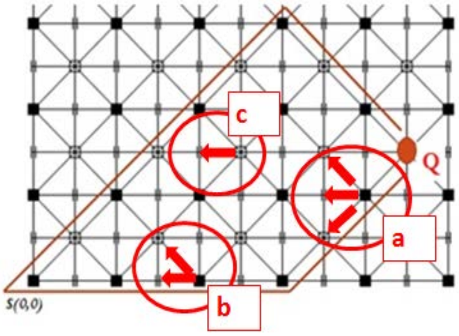
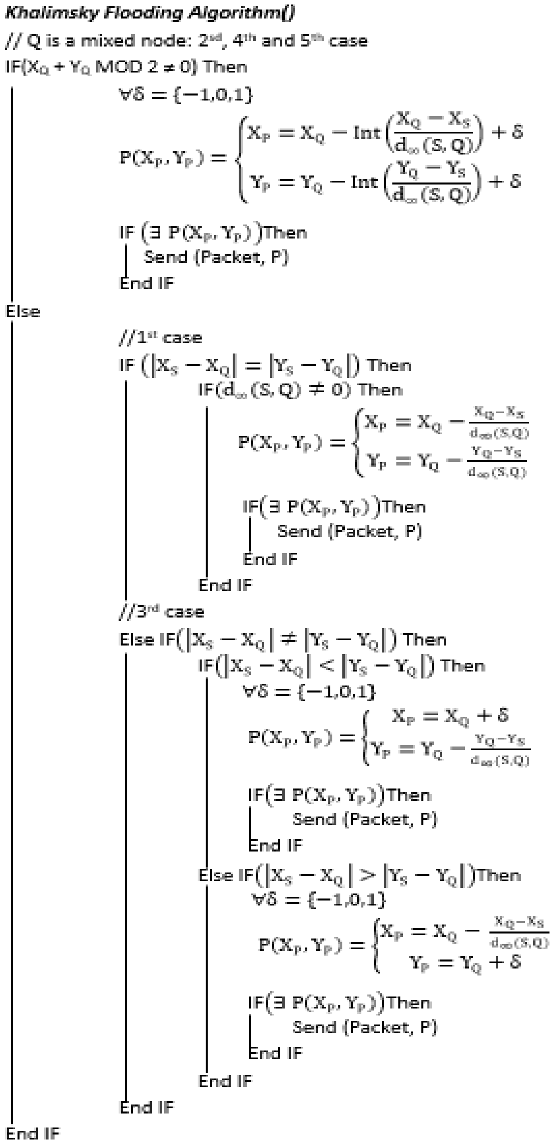


Figure 9: Khalimsky flooding cases of forwarding packets in $\mathrm{BL}(S,Q)$; (a): three gateway nodes; (b): two gateway nodes; (c): one gateway node

According to the five cases of the Khalimsky theory, in the previous subsection, we have developed the following flooding algorithm which permits to calculate the $\mathrm{BL}(S,Q)$ and the gateway nodes, hope by hope, in the process of data transmission.

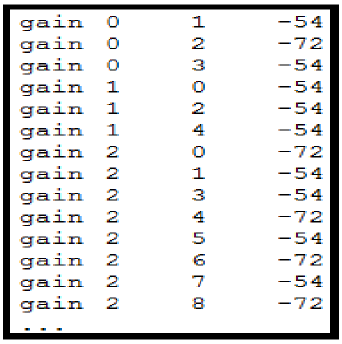


```
Khalimsky Flooding Algorithm()
// Q is a mixed node: 2sd, 4th and 5th case
IF($X_Q + Y_Q$ MOD 2 ≠ 0) Then
        $\forall\delta = \{-1,0,1\}$
        $P(X_P,Y_P) = \begin{cases} X_P = X_Q - \mathrm{Int}\left(\frac{X_Q - X_S}{d_\infty(S,Q)}\right) + \delta \\ Y_P = Y_Q - \mathrm{Int}\left(\frac{Y_Q - Y_S}{d_\infty(S,Q)}\right) + \delta \end{cases}$
        IF ($\exists P(X_P,Y_P)$)Then
            Send (Packet, P)
        End IF
Else
        //1st case
        IF ($|X_S - X_Q| = |Y_S - Y_Q|$) Then
            IF($d_\infty(S,Q) \neq 0$) Then
                $P(X_P,Y_P) = \begin{cases} X_P = X_Q - \frac{X_Q - X_S}{d_\infty(S,Q)} \\ Y_P = Y_Q - \frac{Y_Q - Y_S}{d_\infty(S,Q)} \end{cases}$
                IF($\exists P(X_P,Y_P)$)Then
                    Send (Packet, P)
                End IF
            End IF
        //3rd case
        Else IF($|X_S - X_Q| \neq |Y_S - Y_Q|$) Then
            IF($|X_S - X_Q| < |Y_S - Y_Q|$) Then
                $\forall\delta = \{-1,0,1\}$
                $P(X_P,Y_P) = \begin{cases} X_P = X_Q + \delta \\ Y_P = Y_Q - \frac{Y_Q - Y_S}{d_\infty(S,Q)} \end{cases}$
                IF($\exists P(X_P,Y_P)$)Then
                    Send (Packet, P)
                End IF
            Else IF($|X_S - X_Q| > |Y_S - Y_Q|$)Then
                $\forall\delta = \{-1,0,1\}$
                $P(X_P,Y_P) = \begin{cases} X_P = X_Q - \frac{X_Q - X_S}{d_\infty(S,Q)} \\ Y_P = Y_Q + \delta \end{cases}$
                IF($\exists P(X_P,Y_P)$)Then
                    Send (Packet, P)
                End IF
            End IF
        End IF
End IF
```

Algorithm 2: Khalimsky flooding algorithm

5 Simulation results

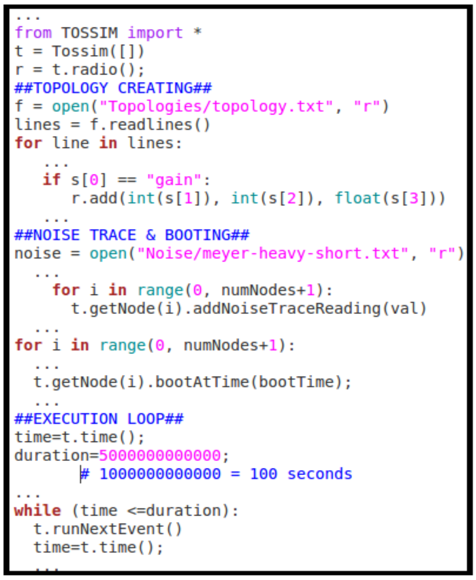
To evaluate the performance of the Khalimsky theory applied on the WSNs area, we used the TOSSIM simulator. We simulated a WSN composed of 50 MicaZ sensors where the energy consumption is modeled by the PowerTOSSIM-Z plugin, compatible with TinyOS2.1, for each node [5-7]. The sensors are deployed hierarchically and uniformly according to the Khalimsky architecture in a sector of the plane, with a sink placed at the center of the sector as shown in figure 2. This topology is defined in a specific file that we called ''topology.txt'' in TOSSIM to configure the links and adjacency lists between the simulated sensors as shown in figure 10. This file defines the gain in the 4thcolumn between the source node in the 2edcolumn and the destination node in the 3rdcolumn.



```
gain  0    1    -54
gain  0    2    -72
gain  0    3    -54
gain  1    0    -54
gain  1    2    -54
gain  1    4    -54
gain  2    0    -72
gain  2    1    -54
gain  2    3    -54
gain  2    4    -72
gain  2    5    -54
gain  2    6    -72
gain  2    7    -54
gain  2    8    -72
...
```

Figure 10: extract of topology file configuration in TOSSIM simulator

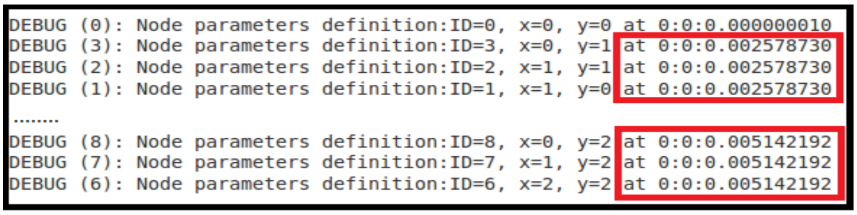
To launch the simulation we have developed a program file written in the Python language. This program is used to load entities, representing the sensors in the computer memory as the Khalimsky topology file previously created, to apply noise model and to boot sensors starting with the Sink. Figure 11 shows an extract of the Python program that we used to run the simulation. This extract contains three parts. The first part allows the loading of the topology file previously created. The second part concerns the application of the noise model and the start sensor nodes. The third section determines the time of simulation and run each program of sensors.



```
...
from TOSSIM import *
t = Tossim([])
r = t.radio();
##TOPOLOGY CREATING##
f = open("Topologies/topology.txt", "r")
lines = f.readlines()
for line in lines:
   ...
   if s[0] == "gain":
      r.add(int(s[1]), int(s[2]), float(s[3]))
   ...
##NOISE TRACE & BOOTING##
noise = open("Noise/meyer-heavy-short.txt", "r")
  ...
    for i in range(0, numNodes+1):
      t.getNode(i).addNoiseTraceReading(val)
  ...
for i in range(0, numNodes+1):
  ...
  t.getNode(i).bootAtTime(bootTime);
  ...
##EXECUTION LOOP##
time=t.time();
duration=5000000000000;
      # 1000000000000 = 100 seconds
...
while (time <=duration):
  t.runNextEvent()
  time=t.time();
  ...
```

Figure 11: Extract of Python program running the simulation

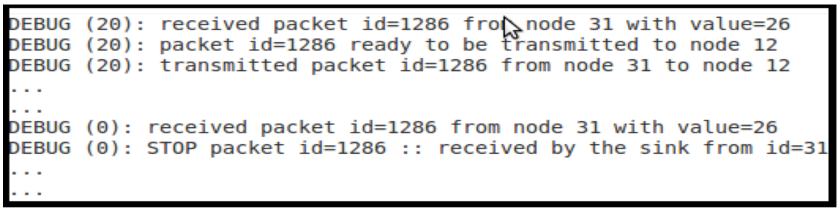
When TOSSIM run simulation, each booted sensor node run the auto-organization algorithm that we developed in algorithm 1 to calculate its coordinates. Figure 12 shows a part of the result of the execution of this algorithm for several sensors. It shows the sensor ID in TOSSIM, the calculated coordinates and the moment of reaching.



```
DEBUG (0): Node parameters definition:ID=0, x=0, y=0 at 0:0:0.000000010
DEBUG (3): Node parameters definition:ID=3, x=0, y=1 at 0:0:0.002578730
DEBUG (2): Node parameters definition:ID=2, x=1, y=1 at 0:0:0.002578730
DEBUG (1): Node parameters definition:ID=1, x=1, y=0 at 0:0:0.002578730
........
DEBUG (8): Node parameters definition:ID=8, x=0, y=2 at 0:0:0.005142192
DEBUG (7): Node parameters definition:ID=7, x=1, y=2 at 0:0:0.005142192
DEBUG (6): Node parameters definition:ID=6, x=2, y=2 at 0:0:0.005142192
```

Figure 12: Auto-organization algorithm results

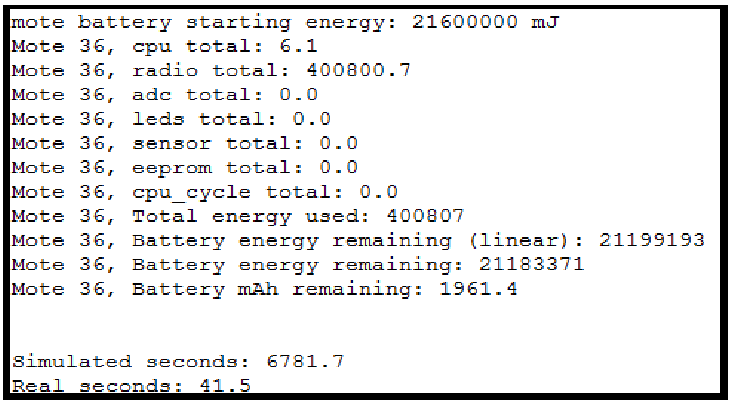
The last part of the Python program consists to run the program of each simulated sensor according to the Khalimsky theory. Each sensor has a timer that organizes these tasks. These tasks are called evenements. The main activities of a sensor are the data collecting from its environment and the routing of created and received packets. Figure 13 shows a part of the execution result of the routed packet by Khalimsky flooding algorithm from the node 31 via the node 20 to the Sink. The node 20, after calculating the destination node selects the node 12 to pass the received packet. It is the case 1 in Khalimsky flooding algorithm since the node 20 is a pure node and its coordinates are x= y = 4.



```
DEBUG (20): received packet id=1286 from node 31 with value=26
DEBUG (20): packet id=1286 ready to be transmitted to node 12
DEBUG (20): transmitted packet id=1286 from node 31 to node 12
...
...
DEBUG (0): received packet id=1286 from node 31 with value=26
DEBUG (0): STOP packet id=1286 :: received by the sink from id=31
...
...
```

Figure 13: Result part of Khalimsky flooding algorithm simulation

Well TOSSIM simulator that can simulate a WSN, it is not possible to model the energy consumption. PowerTOSSIM-Z is a plugin compatible with TinyOS2.1 used to model the energy consumption of each sensor. It uses the results provided by TOSSIM to determine the energy consumed by the sensor nodes. Figure 14 shows the total consumed energy by the node 36 and the simulated time. It shows also the energy consumed by the different components in the node when it's possible to ignore among them. All nodes have the same power 21600000mJ when starting the simulation.



```
mote battery starting energy: 21600000 mJ
Mote 36, cpu total: 6.1
Mote 36, radio total: 400800.7
Mote 36, adc total: 0.0
Mote 36, leds total: 0.0
Mote 36, sensor total: 0.0
Mote 36, eeprom total: 0.0
Mote 36, cpu_cycle total: 0.0
Mote 36, Total energy used: 400807
Mote 36, Battery energy remaining (linear): 21199193
Mote 36, Battery energy remaining: 21183371
Mote 36, Battery mAh remaining: 1961.4


Simulated seconds: 6781.7
Real seconds: 41.5
```

Figure 14: PowerTOSSIM-Z energy modeling results part

In the following, our evaluation is based on comparing the flooding rates and the energy consumption of conventional flooding algorithm versus Khalimsky flooding algorithm by the analysis of result files generated by TOSSIM simulator and PowerTOSSIM-Z as shown in Figure 14.

Figure 15 shows the flooding rates values of conventional flooding algorithm versus Khalimsky flooding algorithm. At the beginning the two algorithms have the same values of the flooding rates. After 11 hours, the conventional flooding algorithm attains 13242250 transmissions while the Khalimsky flooding algorithm makes 9189662 transmissions. After 23 hours, the conventional flooding algorithm attains 25098762 transmissions while the Khalimsky flooding algorithm makes 18434604 transmissions.

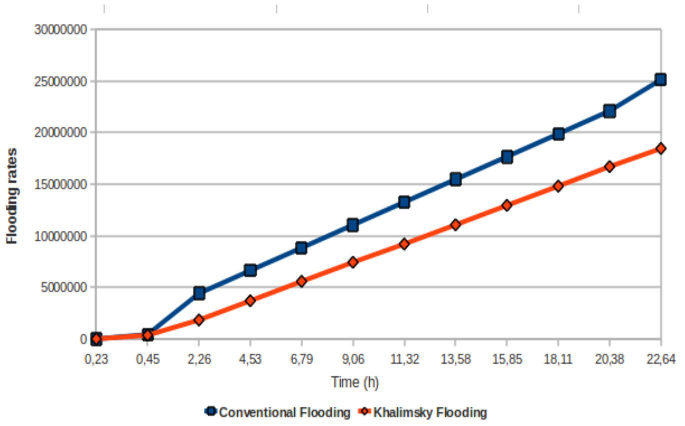


Figure 15: Flooding rates

According to the obtained results, we denote that the Khalimsky topology allowed reducing the flooding rates to about 31%.

Figure 16 shows the energy consumption difference between conventional flooding algorithm and Khalimsky flooding algorithm. The amount of dissipated energy applying the Khalimsky theory is very lower than the second algorithm. At the beginning all nodes have 21600000mJ. After 11 hours, in the conventional flooding algorithm the consumed energy attain 10000000mJ while in the Khalimsky flooding algorithm the value is 4000000mJ. After 23 hours, the conventional flooding algorithm makes 21000000mJ when the Khalimsky flooding algorithm does not exceed 6000000mJ.

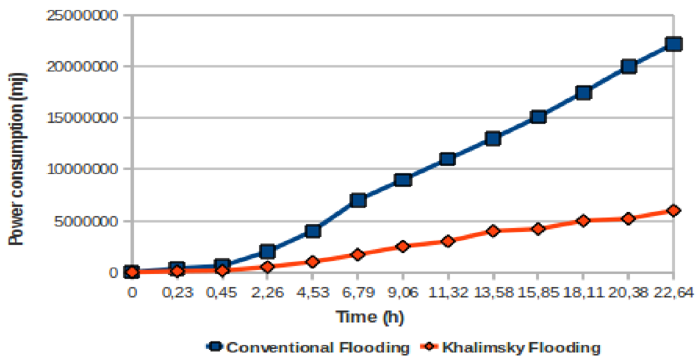


Figure 16: Flooding power consumption results

According to the power consumption obtained results, we denote that the energy consumption average gain obtained after 11 hours is about 50% but after 23 hours is about 75%.

We deduct from the two figures figure 15 and figure 16 an important gap between the red and the blue curves (representing the energy consumption and the flooding rates of Khalimsky flooding algorithm and the conventional flooding approach). The Khalimsky topology reduces the links between sensors. When a node sends a data packet by flooding, the number of used sensors determined by Khalimsky theory is limited by nodes forming optimal paths as shown in figure 3.

6 Advantages & Future Works

Our solution for deploying a WSN based on the topology of Khalimsky can be applied effectively in many areas such as domotic field and industrial field, for monitoring and building security of infrastructures and industrial areas against fires, gas leakage and pollution. In these areas, it is necessary to deploy fixed nodes acting as gateways to convey the data routing toward base station. These gateways provide the transfer of data collected by sensors members grouped into clusters to the base station. The collected data can be important and have a real-time aspect. In this case, the data must be transmitted to the base station in a safe and immediately manner. So with our flooding solution, based on the Khalimsky theory, it’s possible to perform the tasks in an optimal way with minimal data redundancy and the minimum consumed energy.

For future works, we have used the Khalimsky theory to develop a routing protocol for WSNs, based on the clustering method and the multi-agent systems. These WSNs may contain sensors randomly deployed and prefixed sensors acting as gateway. The randomly deployed sensors are grouped into clusters. Each cluster must be able to reach one or more gateways. Gateway sensors are deployed according to the Khalimsky topology. The simulation tests, in the process, showed encouraging results.

7 Conclusion

In this paper, we have been developed a novel hierarchical topology for WSNs based on mathematical Khalimsky theory. This method calculates a subset of nodes to use as gateways for rooting data between a source node and a destination node. This subset of these nodes includes all optimal paths. However, a diffused data packet by a node will never go through outside this subset. This implies the reduction of the flooding rates in a WSN. Thereafter, a considerable minimization of the energy consumption is denoted. This method was validated by a simulation of a WSN using TOSSIM and PowerTossim-Z in TinyOS2.1 that allows the energy consumption modeling. The Khalimsky protocol has a great capacity of energy conservation and of flooding rates reduction. Following these simulations, we clearly note the optimality and the effectiveness of the Khalimsky flooding algorithm compared to the conventional flooding algorithm in term of energy consumption and flooding rates.

REFERENCES

[1]. M. Mezghani, G. Ellouze, A. Grati, I. Bouabidi, M. Abdellaoui, *Multitasks-Generic platform via WSN*. International Journal of Distributed and Parallel Systems (IJDPS), Vol.2, No.4, July 2011, p. 54-67.

[2]. A. Nayak, I. Stojmenovic,*Wireless Sensor and Actuator Networks: algorithms and protocols for scalable coordination and data communication*. WILEY series, 2010, chapter 4 and 5, p. 95-152.

[3]. G. Anastasi, M. Conti, M. Di Francesco, A. Passarella,*Energy conservation in wireless sensor networks: A survey*, Ad Hoc Networks,7 (2009), p. 537-568.

[4]. Y. Youssef, *Routage pour la gestion de l'energie dans les réseaux de capteurs sans fil*. These de doctorat, Haute Alsace-France, Juillet 2010.

[5]. M. Mezghani, O. Mezghani, H. Rekik, M. Abdellaoui, *TinyOS2.1 with nesC, TOSSIM and PowerTOSSIM-Z Emulation and Simulation Environments to Networked Domotic Embedded Systems.*12th International conference on Sciences and Techniques of Automatic control and computer engineering (STA2011), December 18-20, 2011, Sousse, Tunisia.

[6]. E. Perla , A. O. Cathain, R. S. Carbajo , *PowerTOSSIM-z: Realistic Energy Modelling for Wireless Sensor Network Environments.* Proceedings of the 3nd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks, October 31, 2008, Vancouver, BC, Canada, p. 35-42.

[7]. P. Levis, D. Gay, *TinyOS programming.* Cambridge University Press, 2009, p. 3-105.

[8]. M. Hadjila, M. Fehman, *A comparative study of the wireless sensor networks routing protocols scalability.* International Journal of Distributed and Parallel Systems (IJDPS), Vol.2, No.4, July 2011, p. 26-33.

[9]. K. Beydoun, *Conception d'un protocole de routage hiérarchique pour les réseaux de capteurs.* Thèse de doctorat, Franche-Comte-France, Décembre 2012.

[10]. M. Lehsaini, *Diffusion et couverture basées sur le clustering dans les réseaux de capteurs : aplication à la domotique*. Thèse de doctorat, Franche-Comte-France, Juillet 2009.

[11]. E. Bouacida, *The jordan curve theorem in the Khalimsky plane*, Applied General Topology 9, 252 (2008).

[12]. U. Eckhardt, L.J. Latecki, *Topologies for the digital spaces Z2 and Z3*. Comput. Vision Image Understanding, 90 (2003), p. 295-312.

[13]. E.D. Khalimsky, *On topologies of generalized segments.* Soviet Math. Dokl., 10 (1999), p. 1508-1511.

[14]. E.D. Khalimsky, R. Kopperman, P.R. Meyer, *Boundaries in digital planes*. J. Appl. Math. Stoch. Anal. 3 (1990), p. 27-55.

[15]. E. Khalimsky, R. Kopperman, P. R. Meyer, *Computer graphics and connected topologies on finite ordered sets*.Topol. Appl. 36, 1 (1990).

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TNC Transactions on Networks and Communications

VOLUME 2, ISSUE 6
ISSN: 2054 - 7420

Multipath Discovery Algorithms for VoD Streaming In Wireless Mesh Network

[1]Praful C Ramteke, [2]V.S.Jadhav and [3]Raju Wadekar
[1&2]Maharashtra Institute of Technology, Pune, India
[3]University of Lancaster, United Kingdom
[1]praful.c.ramteke@gmail.com, [2]vinod_jadhav@yahoo.com, [3]govindraoraju@yahoo.com

ABSTRACT

Transmission and routing of video data over wireless network is a challenging task because of wireless interferences. To improve the performance of video on demand transmission over wireless networks multipath algorithms are used. IPD/S (Iterative path discovery/selection) PPD/S (Parallel Path discovery/selection) are two algorithms which is used for discovering maximum number of edge disjoint paths from source to destination, for each VoD request by considering the effects of wireless interferences. In this paper performance evaluation of these multipath discovery algorithms for VoD (Video on demand) streaming in wireless mesh network is presented. These algorithms are evaluated on the bases of Number of Path discovers, Packet drop ratio and delay. Simulation result shows that PPD/S works batter as compared to IPD/S because it's able to discover more paths than IPD/S under same circumstances.

Keywords—Multisource video streaming, IPD, PPD wireless mesh network and multipath routing

1 Introduction

Video streaming is gaining popularity among mobile users recently. It has been forecasted that mobile video data traffic will account for 70% of total mobile data traffic by 2016[1]. One of the most used application in internet services recently is video on demand application (VoD). In this VoD application a video is provided by a respected server to the demanding users. Most of the VoD applications use peer to peer technology to increase the VoD performance. Considering video which have been watched recently by users is store in their local storage. In peer to peer technology when a user wants to watch a new video, she or he primarily discovers which peer user have buffered the video and then streams the video from both peer client and servers through multi-path. A multi-path multi-source video on demand streaming has achieve a great success in wired network but somehow remains a challenging in case of wireless network due to several wireless interferences.

As wireless networking technology is attracting more interest in research industry and building community networks due to low cost infrastructure. A community network is composed of many mesh routers where each mesh router established connectivity with its neighboring router. When a user request for a video it can stream the video from two sources 1) the server or peer that have buffered the video. 2) From the other users who have access to the required video.

Video streaming is a challenging task due to high bit rate, delay and loss sensitivity. However it is assume that by streaming video from multiple paths we can improve the performance of video streaming in wireless network. There are several peer-to-peer architecture has been proposed. [2][3][4]. Path selection is depend on two factors i.e. Path disjointness and Hop count

DOI: 10.14738/tnc.26.523
Publication Date: 30th November 2014
URL: http://dx.doi.org/10.14738/tnc.26.523

1.1 Path Disjointness

There are several path discovering algorithms and they are discovering several paths sharing common nodes or even links. Such set of paths are less reliable because transmission failure in any of shared nodes causes loss of data streams most probably packet loss. One of the major problems in multi-source VoD applications is discovery of multiple independent path and path selection form the selected paths. The independent paths are defined as the path which have are not depending on other paths i.e. edge-disjoint or vertex-disjoint paths. In edge disjoint paths no two paths with share a same link and therefore the possibility of link failure is minimized. Vertex disjoint is stronger than edge disjointness, because it provide guarantee that node failure will affect mostly one path. In wireless mesh network it's very difficult to find independent paths due to the wireless interference. Another disadvantage of shared nodes is reduction in paths capacity, since the they aggregate traffic from several paths. This may cause self interference. Hop count has impact on two quality metric- end to end delay and path reliability as the number of hop increases it increases the end to end delay. [7]

In this paper evaluation of two multipath discovery algorithms is presented. One of the major problems in multi-source VoD applications is discovery of multiple independent path and path selection form the selected paths. The independent paths are defined as the path which have are not depending on other paths i.e. edge-disjoint or vertex-disjoint paths. In edge disjoint paths no two paths with share a same link and therefore the possibility of link failure is minimized. Vertex disjoint is stronger than edge disjointness, because it provide guarantee that node failure will affect mostly one path. In wireless mesh network it's very difficult to find independent paths due to the wireless interference. All we know about wireless mesh network is that this network is designed to share among multiple users so it provides advantage while using VoD application in this network.

2 Multisource Video on Demand in WMN

In multi-source video on demand streaming a single required video is stream from more than one source in general in conventional video streaming video is only available through server in this case of multi source video on demand streaming video the required video is not only provided by the respected server but also by the other means of source also (from fig.1) i.e. other users. For some popular videos the VoD performance can be improved by P2P technology. In this technology whenever the user request for a video it first registers with server so that server keeps the list of visited users. In future when new user request for the same video server can provide list of previous users so that new user can not only get a video stream from the server but also from the previous users of the video.

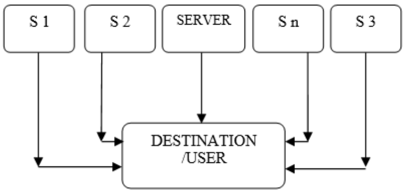


Figure 1: Multi-source VoD in wireless mesh Network

3 Multi Path Discovery Algorithms

In wireless mesh networks, the mobility of mesh router is minimal. For example in community networks most of the time routers are fixed on roofs of houses in addition to it due to overhead of dynamic channel switching a static channel allocation strategies are used in which channel are often changed. This make possible for each router to collect the global information about its network including channel assignment and position of each router.

There are several algorithms present which is used for multipath discovery in this paper we are dealing with IPD/S and PPD/S i.e. iterative path discovery/selection and parallel path discovery/selection algorithms.

3.1 Iterative Path Discovery/Selection (IPD/S)

The Iterative Path Discovery algorithm (IPD/S) finds paths one by one from the senders to the receiver. In each iteration, one path from a sender to the receiver, and then update the topology accordingly. This process continues until no new paths can be found from the remaining topology.

In Iterative path discovery algorithm paths from all source to single destination is discovered in a step by step manner. (Form fig.2)When a destination or user demand for the video then the request is first send to the respected server from the server a user collect the information about other source that are able to provide the desired video to the destination. After gathering all the information about all the available sources the Iterative path discovery starts working. Here IPD/S select a single source and then find the path towards destination after reaching at destination it updated the topology and again starts the search this time by considering other source and the search is going on till the last available source. Each time after discovering the path it is necessary to update the topology which guarantees the edge disjointedness of path.

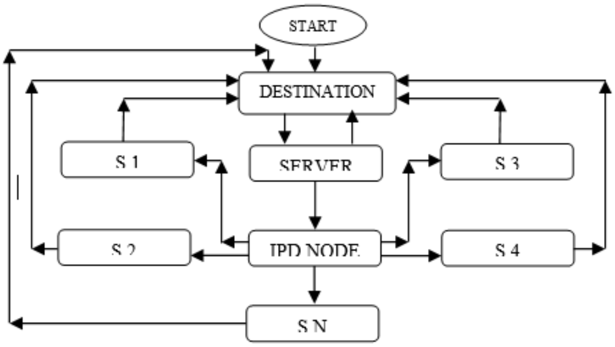


Figure 2: Block Diagram of IPD/S

3.1.1 Path Selection

Selection of path is done on the basis of minimum total interference, so that the selected path will render fewer changes in the remaining topology and thus leaves more flexibility for finding more paths afterwards.

Let S be the set of senders and T be the initial topology. Let S' be the set of remaining senders, for which no paths found yet, and T' be the remaining topology. Initially, $S' = S$ and $T' = T$. In this step, for each $s \in S'$, First find a minimum $WCETT$ path p from s to r in T' if such a path exists and $WCETT(p) \leq \gamma \cdot wT\ (s, r)$, where $wT\ (s, r)$ is the $WCETT$ value of the optimal path from s to r in T and γ is a constant to control the quality of each path. Denote the resulting set of paths as P, after gating the resulting set of paths next step is to decide which path to select from set of paths P.
The total interference of p in network is denoted by $IFT'\ (p)$, the interfere with any edge in p with respect to the number of edges in $T'$is given as

$$IFT'\ (p) = |\ \{e' \mid e' \in E' \wedge \exists e \in p : Interfere(e, e')\}\ |$$

$$(e, e') \text{ is true if } e = e'$$

In VoD applications, it's not only about finding more disjoint paths from source to destination but also need to guarantee the quality of each path with respect to packet loss, delay and throughput. There are several metric present for finding good routes between single source to destination in wireless network for example The WCETT metric [9] is widely used in multichannel multiinterface wireless mesh networks. It not only considers packet loss and delay, but also accounts for channel diversity in each path so as to reduce intraflow interference. Here we are using this metric to evaluate the quality of each selected path it not only considers packet loss and delay, but also accounts for channel diversity in each path so as to reduce intraflow interference. Therefore, by using this metric to evaluate the quality of each selected path. while keeping the WCETT value of each path below a certain threshold. The WCETT metric of a path can be calculated as follows:

$$\text{WCETT} = (1\text{-}k) * \sum_{i=1}^{n} ETT_i + k \; * max_{1\le j\le c} X_j$$

Where ETT_i is expected transmission time of a packet on the link i and

X_j : - is the sum of transmission times of hops on channel j

3.1.2 Topology Update

Once a path has been selected, it needs to update accordingly. Updating the topology guarantees the edge disjointedness of paths, so that the paths found later will not overlap with the paths previously found. In addition to this it needs to consider interference on topology and guarantee the level of independency among the final set of selected paths.

Consider a label l on the edges of T' to record the interference from the already selected paths, where (e) counts how many selected paths are interfering with link e. At the start of the algorithm, (e) is initialized to 0. Assume p is the selected path from $T'(V,E')$ in the current iteration. We define the path interfering set of p in T', denoted by $IET'\ (p)$, as the set of edges in $(T' - p)$ that interfere with any edge in p.

$$IET'\ (p) = \{e' \mid e' \in (E' - p) \land \exists e \in p : Interfere(e, e')\}$$

By updating (e) for each edge $e \in IET'\ (p)$ by increasing 1, indicating that there is one more selected path p that interferes with e. If $l(e) > \alpha$, then by taking off e from T'.This is because if e has been used in one more path in the remaining topology, then e will interfere with more than α already selected paths, which violates the constraint in the $COMINP$ problem.

Constrained Maximum Independent Paths problem (COMINP) the maximum number of edge-disjoint paths from S to D, such that

1. MPI(P) $\le$ α where α is the threshold to control the level of independency between path in P
2. $WCETT_{(P)} \le$ µ for p ϵ P, where µ is the threshold to control the quality of each path

3.1.3 Parallel Path Discovery/ Selection (PPD/S)

In parallel path discovery algorithm paths from all source to single destination is discovered in a single step.

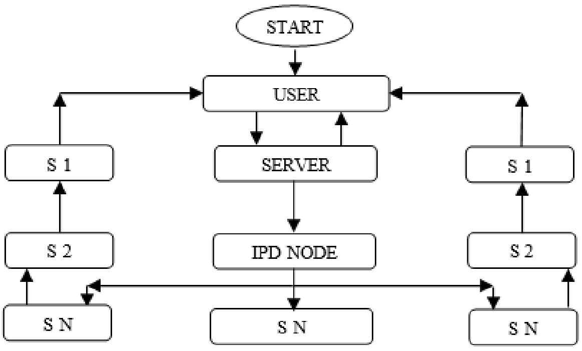


Figure 3: Block Diagram of PPD/S

Here as shown in Figure 3 when a destination or user demand for the video then the request is first send to the respected server from the server a user collect the information about other source that are able to provide the desired video to the destination.

After gathering all the information about all the available sources the Parallel path discovery starts working. PPD collects information about entire available source for the video transfer from the server and starts discovering the paths from each source to a destination. This is totally different from the IPD as in this algorithm all the paths are get discovered in a single step by this the time required to perform other operation is reduced. At the end of algorithm the information is get updated in a PPD node and the most appropriate path is get selected for the video transfer, while selecting the path the most important thing to consider is the paths need to be edge disjoint (i.e. the paths need to be independent of each other). In the Internet, the independent paths are usually defined as edge-disjoint or vertex-disjoint paths. In edge-disjoint paths, no two paths share a same link, and therefore any link failure will only affect one path.

4 Simulation Methodology

Simulation is performed by using NS2. To support multiple interfaces and multiple channels per node in simulation Hyacinth extension [5] has been used. During network transmission the frames are encapsulated into User Defined Protocol (UDP) packets and reconstructed at the receiver. In all the simulation 802.11 MAC protocol is been used with CBR (Constant Bit Rate) and UDP with the maximum packet size of 512 bit for multimedia data transport. A simulation is performed in 20, 40 and 60 nodes random topology within an area 500 × 500.There are four channels used in channel assignment. The channel allocation algorithm proposed in [6] is used to statistically assign channels for each interference in order to minimize the wireless interference within the network.

There are dependencies between the encoded video frames for ex. Any group of picture (GOP) is made up of three frames I, B and P. I-frames in GoP is required to decode all other frames P and B in a GoP and P-frames are required to decode all successive P as well as B-frames encoded with respect to these P-frames. So even if a frame has been received before deadline it is regarded as not constructed if its dependent frames are not reconstructed successfully.

Here we compare the following method of path discovery for multi-source video on demand streaming.

1. IPD/S: Used the Iterative path Discovery/selection algorithm to find the maximum number of edge disjoint independent path.
2. PPD/S: Used the Parallel Path discovery/Selection algorithm to find the maximum number of edge disjoint path.

5 Performance Evalutation

In this section result Based on NS2 simulation has been presented.

5.1 Number of Paths

For discovering number of paths in a network a single router is selected as a receiver in network and randomly designed n routers are selected as senders. While discovering a path by IPD/S and PPD/S edge-disjoint paths are only considered at the same time while selecting edge disjoint paths algorithms are performed for β=0 and β=1 i.e. at β=0 no path interfere with each other whereas at β=1 each link of any path interferes at most with one other path among the multiple path finally discover. The number of paths discovered by IPD/S and PPD/S under different number of senders are shown in figure [4] and [5].As we can observe PPD/S is able to discover more paths than IPD/S under the same constraints.

5.2 Packet drop Ratio

When a user streams video over multiple paths, it requires to determine the playout deadline that is the time the user waits before playing the video. There are two main reasons due to which packet drop occurs in a network. 1) When packet is loss due to interference in wireless transmission or collision. 2) The packet is received successfully but it’s received after deadline time.

5.3 Delay jitter

Network delay is an important design and performance characteristic of a computer network or a telecommunication network. The delay of a network specifies how long it takes for a bit of data to travel across the network from one node to another or end point and if the network reaches its maximum capacity further VoD request are blocked [8]

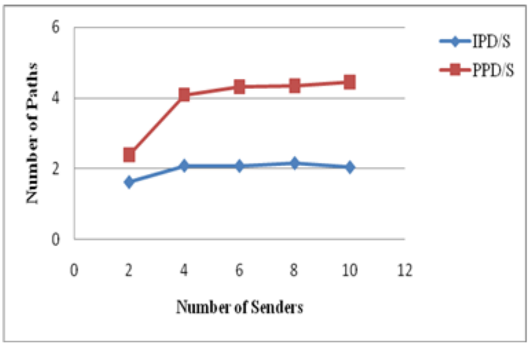


Figure 4: The Number of Paths Discovered (β=0)

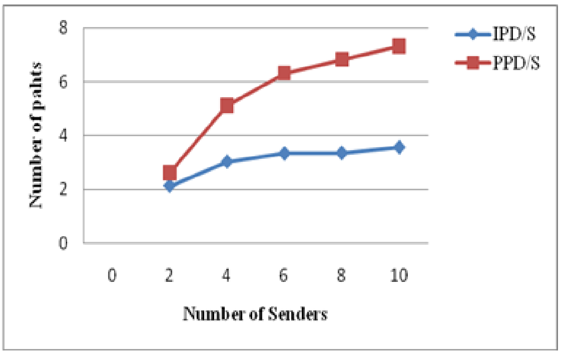


Figure 5: The Number of Paths Discovered (β=1)

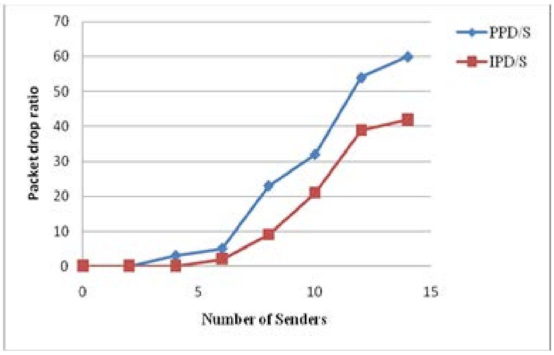


Figure 6: Packet drop ratio (payout deadline=10ms)

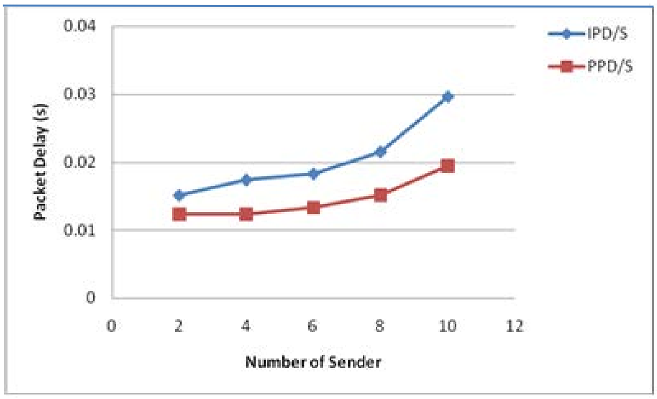


Figure 7: Packet Delay over number of senders

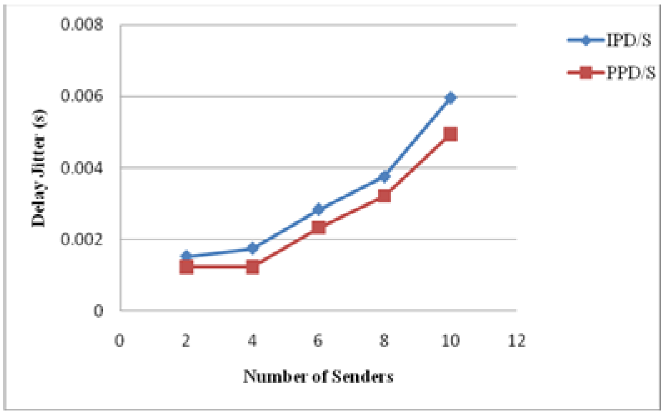


Figure 8: Delay Jitter over number of senders

6 Conclusion

In this paper two multipath discovery algorithms, IPD/S and PPD/S to find multiple independent paths from sender to receiver for each VoD request are studied. The simulations are performed on NS2 and evaluate the performance of algorithms. Simulation result has shown that PPD/S achieves better video streaming performance than IPD/S, because it is able to discover more paths than IPD/S under same conditions.

REFERENCES

[1]. Cisco, Cisco visual networking index: Global mobile data traffic forecast update, 2011-2016.

[2]. D. A. Tran, K. A. Hua, and T. T. Do, “A peer-to-peer architecture for media streaming,” IEEE J. Sel. Areas Commun., vol. 22, no. 1, pp.121–133, Jan. 2004.

[3]. T. Nguyen and A. Zakhor, “Multiple sender distributed video streaming,” IEEE Trans. Multimedia, vol. 6, no. 2, pp. 315–326, Apr.2004.

[4]. Y. Zhu, W. Zeng, H. Liu, Y. Guo, and S. Mathur, “Supporting video streaming services in infrastructure wireless mesh networks: Architecture and protocols,” in Proc. IEEE ICC, 2008, pp. 1850–1855.

[5]. T.-C. Chiueh, A. Raniwala, R. Krishnan, and K. Gopalan, “Hyacinth: An IEEE 802.11-based multi-channel wireless meshnetwork,” 2005 [Online] .Available: http://www.ecsl.cs.sunysb.edu/multichannel

[6]. A. P. Subramaniam, H. Gupta, and S. R. Das, “Minimum-interference channel assignment in multi-radio wireless mesh networks,” in Proc.IEEE SECON, 2007, pp. 481–490.

[7]. Lauris Cikovski and Ilmars Slaidins.”Analysis of wireless Ad-hoc Network Parameters for Efficient Multipath Video transfer.” In IEEE 2012

[8]. X. Zhu, S. Han, and B. Girod, “Congestion-aware rate allocation for multipath video streaming over ad hoc wireless networks,” in Proc IEEE ICIP, 2004, vol. 4, pp. 2547–2550

[9]. R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio multi-hop wireless mesh networks," Proc. ACM MobiCom, 2004, pp. 114–128.

[10]. J. Tang, G. Xue, and W. Zhang, "Interference-aware topology control and QoS routing in multi-channel wireless mesh networks," in Proc.ACM MobiHoc, 2005, pp. 68–77.

[11]. Y. Ding, Y. Yang, and L. Xiao, "Multi-path routing and rate allocation for multi-source video on-demand streaming in wireless mesh networks," in Proc. IEEE INFOCOM, 2011, pp. 2051–2059.

[12]. S. J. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in ad hoc networks," in Proc. IEEE ICC, 2001, vol. 10, pp.3201–3205.

[13]. M. Marina and S. Das, "On-demand multipath distance vector routing in ad hoc networks," in Proc. IEEE ICNP, 2001, pp. 14–23.

[14]. D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in Proc. Mobile Comput., 1996, pp. 153–181.

Society for Science and Education
United Kingdom

TNC Transactions on Networks and Communications

Volume 2, Issue 6
ISSN: 2054 - 7420

Multilevel Cryptography with Metadata and Lock Approach for Storing Data in Cloud

[1]Dinesha H A and [2]Vinod Kumar Agrawal
PES Institute of Technology, Visvesvaraya Technological Univeristy, Belgaum, India;
[1]sridini@gmail.com; [2]vk.agrawal@pes.edu;

ABSTRACT

Cryptography is a technique for secure communication. Cryptography main objectives are confidentiality, integrity, non-repudiation, availability and authentication. Cryptography is a well defined and used technique to secure sensitive data. It has been using in cloud computing technology by various cloud service provider. Customers across the world are looking for storage infrastructure to store huge amount of data securely. Hence they are opting on demand, ready available, internet based and maintenance free infrastructure known as cloud data storage as a service. Many potential vendors like Microsoft and Amazon providing this service to customer across the globe. But major challenge is customer trust on vendor. Vendor has to prove customer that their data is safe via cryptography, security breach penalty, policies and security agreement so on. However, it is difficult to gain customer confident on vendor security. Hence we are proposing customer end algorithm called multilevel cryptography for secure cloud data storage where customer performs multiple cryptography operations on their data before storing into a cloud. In this paper we present the multilevel cryptography algorithm for secure cloud data storage with design and analysis.

Keywords: Cloud service provider, Data storage as a Service, Multilevel Cryptography, Secure Cloud Data, and Secure Communication.

1 Introduction

Today data security are achieving through cryptography. Many encryption and decryption techniques are in place and ready to use in cloud computing technology. Kawser Wazed Nafi, Tonny Shekha Kar, Sayed Anisul Hoque [1] have proposed a method by implementing RSA algorithm to ensure the security of data in cloud computing. RSA algorithm used to encrypt the data to provide security so that only the authorized user can access it. It consists of Public-Key and Private-Key. Public-Key is known to all, whereas Private-Key is known only to the user who is authenticated. Once the data is encrypted with the Public-Key, it is possible to decrypt with the corresponding Private-Key only. Eman M.Mohamed, Hatem S. Abdelkader [2] explain the data security system implemented into cloud computing using RC4, RC6, MARS, AES, DES, 3DES, Two-Fish and Blowfish algorithm. The security architecture of the system is designed by using DES cipher block chaining, which eliminates the fraud that occurs today with stolen data. Kan Yang, Xiaohua Jia [3] describes the multilevel encryption for Ensuring Public Cloud. This technique applies multiple encryption algorithms for given plaintext. Follow the same level of decryption to convert back from cipher text. Cong wang, Qian wang, and Kui ren, Wenjing Lou describes how to ensure the data storage security in cloud computing [4]. In Cryptography and Network Security Principles and Practices [5],

DOI: 10.14738/tnc.26.591
Publication Date: 07th December 2014
URL: http://dx.doi.org/10.14738/tnc.26.591

mentioned many encryption technique and cryptography principles which can be adopted in cloud computing technology [5]. The following are amongst the most well known: i) DES: This is the 'Data Encryption Standard'. This is a cipher that operates on 64-bit blocks of data, using a 56-bit key. It is a 'private key' system. Ii) RSA: RSA is a public-key system designed by Rivest, Shamir, and Adleman. Iii) HASH: A 'hash algorithm' is used for computing a condensed representation of a fixed length message/file. This is sometimes known as a 'message digest', or a 'fingerprint'. iv) MD5: MD5 is a 128 bit message digest function. It was developed by Ron Rivest. AES

This is the Advanced Encryption Standard (using the Rijndael block cipher) approved by NIST. V) SHA-1: SHA-1 is a hashing algorithm similar in structure to MD5, but producing a digest of 160 bits (20 bytes). Vi) HMAC: HMAC is a hashing method that uses a key in conjunction with an algorithm such as MD5 or SHA-1. Thus one can refer to HMAC-MD5 and HMAC-SHA1 [5].

Many authentications also exist to ensure the cloud customer authentication while using cloud service they are i) Simple text password ii) Third party authentication iii) Graphical password iv) Biometric and v) 3D password object and etc are explained in [6][7][8][9]. We have presented multilevel authentication technique for accessing cloud services in [10].Some of the existing cloud authentication methods and techniques are described in [11] - [17].

The main challenge is though there are many techniques , method are available in cryptography and literature, what is the method to make sure customer that their data is keeping confidentially at cloud service provider end? What are the ways to gain customer faith on service vendor cryptography? As a solution, can vendors announce one service plan that customer is responsible for data confidentiality and vendor only responsibility is providing storage area with data disaster recovery. We are getting the motivation of the paper including shortcoming of the work carried out by various authors. In this paper, we present such service plan in details. We proposed a plan to customer to have a dedicated setup of software known as multilevel cloud cryptographer with customized option. It performs cryptography on sensitive data in multiple levels and in multiple ways before migrating customer data to service vendor infrastructure. Detailed information presented in remaining section of this paper. This paper is organized as following manner. Section II, presents the multilevel cryptography algorithms. Section III, describes the system design details of proposed algorithms Section IV presents the detailed analysis. Section V, concludes the paper along with future enhancement.

2 Multilevel Cryptography for Secure Cloud Data Storage

In this section we describe the proposed multilevel cryptography algorithm. This algorithm applied in customer side against their sensitive data. Before migrating to cloud vendor storage infrastructure customer performs data cryptography in multiple levels and in multiple ways. The levels and ways are decided by customer based on their data confidentiality and organizational structure. In this section we proposed three levels and three different ways on customer behalf. Proposed three levels are Chief Data officer, Cryptography Officer and Data Designers and its corresponding three ways are Data lock, Data encryption and Metadata respectively. Customer can customize levels and ways of cryptography. Customer is free to apply any techniques/methods available in literature as multiple levels way without knowing to service vendor or anybody. But first level and last level are recommended as important to have in their customized setup. Because its added different features in proposed cloud cryptography than encryptions alone. In algorithm1, we consider customer sensitive data as plaintext message m and migrated data as cipher text c.

Algorithm1: multilevel cryptography algorithm for migrating

```
Step 1:  plaintext message m processed in metadata to get jumbled message ‘mo’
  mo: jd (m)
Whereas ‘j’ refers to jumbled process, ‘d’ is data about the  jumbled  process
Step 2:  Data encryption for processed metadata and pack in file/folder f.
c=e (mo)
f= c1, c2, …cn
f=$\sum_{i=1}^{n} c$
Step 3 :Locking the data file/folder/package and send to cloud storage s
s: lock(f)
Therefore for migration formula is => lock ( f+ (e(jd (m))) obtained.
Algorithm2: multilevel cryptography algorithm for accessing the migrated data
Step 1: Unlocking the retrieved data file/folder/package from cloud storage
f: unlock(s)
Step 2:  Extract Folder/file and perform decryption to get back mo.
For i=1 to n
   c=Split (f)
mo =d (c)
Step 3:  processed in metadata to get plaintext message m from jumbled message ‘mo’
  m: j-d (mo)
Therefore for accession formula is => j-d(d (split - (unlock (s)))) obtained
Example
Let us take example and apply this algorithm for set of sensitive data i.e customer pin set
{2345, 4567, 5645}.
i) Below are the migration steps to get migrated data output
1. mo= jd jumbled with even first and odd next,  left to right i.e change data position to
2413 order
i.e = {3524, 5746,6554} d has order info i.e 2413 .
2. Apply any encryption algorithm we used simple substitution i.e add +2 to mo
 c= {5746, 7968, 8776}
3. Pack all, lock and place in file
lock (f= {574679688776 }) hence migrated data is locked 574679688776
ii) Below are the accession steps to get plain text from migrated data.
Split (Unlock (574679688776)) with size of 4
{5746, 7968, 8776}
Decrypt with -2  for spitted data {5746, 7968, 8776}
{3524, 5746,6554}
Reorder with jumbled inverse function i.e 2413-1
j-d({3524, 5746,6554})
=>{2345, 4567, 5645}
```

Proposed three levels and corresponding ways are in described in table 1 and corresponding architecture are represented in figure 1.

Table 1: Proposed multilevel cryptography

Level No	Level Name	Ways of Cryptography	Description
1.	Chief Data Officer	Data lock	Locks the data with one suitable authentication (password, biometric, Multi Level Authentication and etc)
2	Cryptography Officer	Data encryption	Responsible for data encryption with private. He is free to choose available encryption technique like RSA, DES, AES and etc.
3	Data Designer	Metadata	Responsible for data storage order/structure/pattern designs. Keep the data pattern of sensitive data.

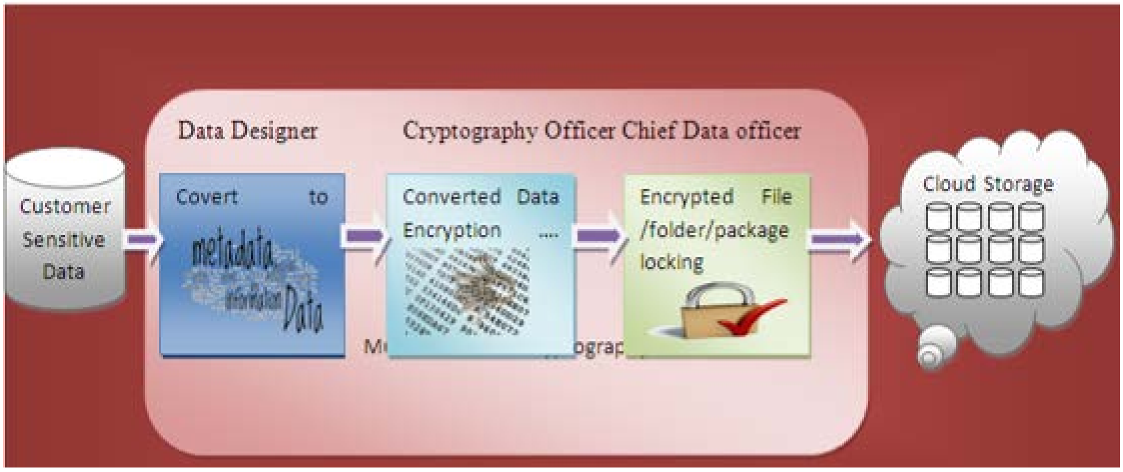

Data Designer
Cryptography Officer Chief Data officer
Customer Sensitive Data
Covert to
metadata
Data
Converted Data Encryption ....
Encrypted File /folder/package locking
Cloud Storage
M
ograp

Figure 1: Proposed multi-level cryptography architecture (before migration)

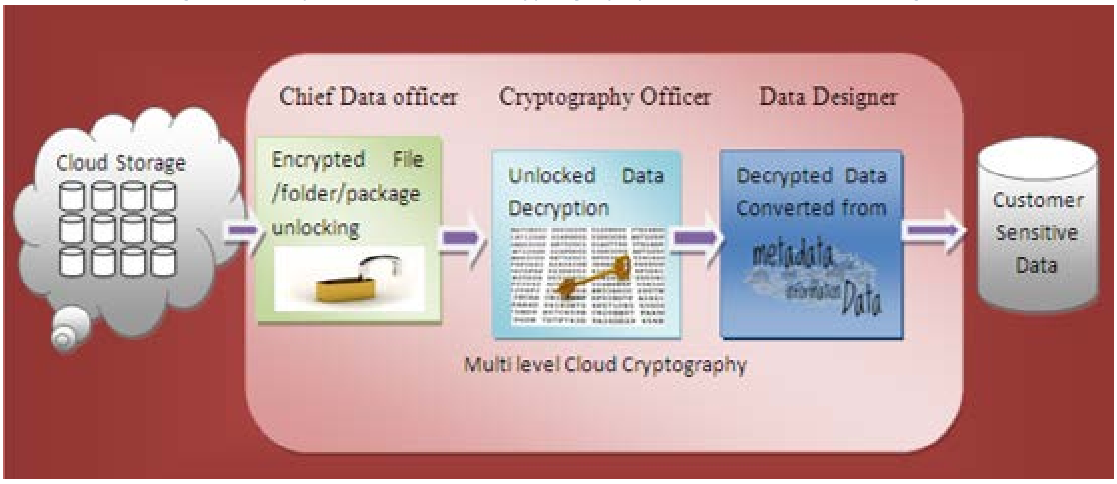


Figure 2: Proposed multi-level cryptography architecture (during accession)

Proposed architecture presents, sensitive data first converts into metadata by data designer. In second level converted data get processed with data encryption by Cryptography officer. This encryption algorithm can be multiple or single; it depends on customer and their confidentiality. Final step encrypted data file/folder/package locking done by chief data officer. Customer has to follow these multilevel steps before migrating data. Now, as a description for accessing the same data, customer has to follow the reverse steps as below figure 2. First, Chief Data officer has to unlock received data from cloud vendor. Cloud vendor may use cryptography technique or it just provides data storage with data disaster recovery. After unlocking, cryptography officer decrypt the unlocked files/folder/package. Finally data designed use metadata to take back original order/structure/pattern. Finally it reaches actual data base. Here, all the levels people are equally responsible and required in this process.

3 System Design

In this section we present the designs of the proposed multilevel cryptography using petri net theory. The system modeling is done using Petri nets, which are vividly portrayed in figure 6. Petri nets are a special form of bipartite directed graph represented by < P, T, In, Out> , in which Place (denoted as p) and Transitions (denoted as t) are disjoint sets of nodes, and In and Out are sets of edges. We carry out formal modeling for our system to precisely discover how user can migrate to cloud using multilevel cloud cryptography. How user can access the sensitive data back from migrated cloud. The model is explained as follows. Figure 3shown Petri net model before migrating into cloud storage. The places p1, p2, p3, p4, p5 and p10 represent the steps to convert from sensitive data to multilevel cryptography based migrated data. The transitions t1, t2, t3, t4 and t5 present the corresponding actions in it. The condition places p6, p7, p8, p9 and p11 has to be satisfied for successful migration.

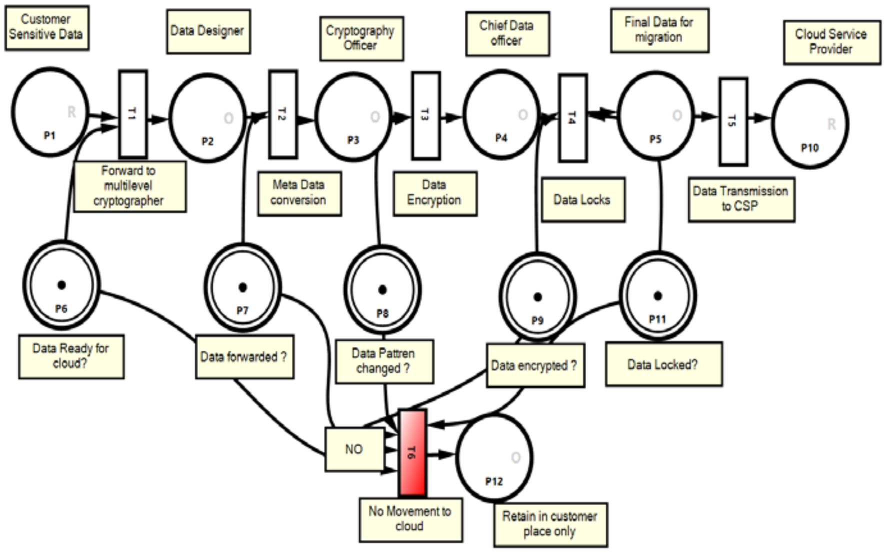


Figure 3: Petri net model before migrating into cloud storage

Figure 4 presents Petri net model for accessing the migrated data from cloud. The places p1, p2, p3, p4, p5 and p6 represents the steps need to follow before accessing the migrated data. The transition t1, t2, t3, t4 and t5 present the corresponding actions to achieve the same. The condition p7 and p8 has to satisfy for successful data reception.

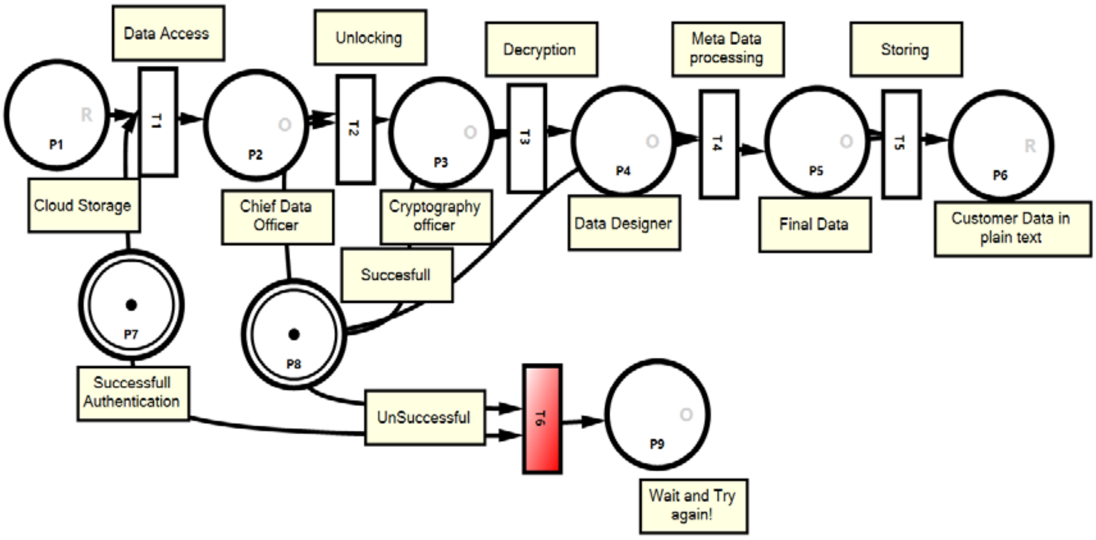


Figure 4: Petri net model for accessing the migrated data from cloud

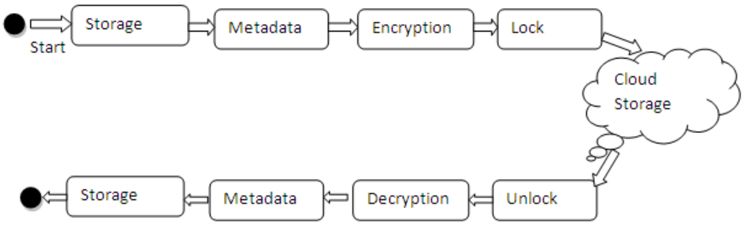


Figure 5: State Chart Diagram of migration and accession

Figure 5 shows the different states of the data to be stored in cloud before migrating and accessing the data from the cloud after migration. This can also be applying to save the local Virtual Machines related files. In cloud IaaS, VMs playing major rule. Private IaaS cloud setup having its own set of VMs that they use in regular business application. Backup of this VMs and its corresponding template files like .vmx, .vmdk and so on can be stored in cloud using this technique. A new optimized ranking algorithm.

4 System Analyze

In this section we analyze the proposed multilevel cryptography algorithm.

(a) Let's take the above derived migration process formula Cloud encryption Ce is => lock (f+ (e(jd (m))) and accession formula for Cloud decryption process Cd is M=> j-d(d (split - (unlock (s)))). Let us consider the number of levels is 3, hence to migrate and access levels L1, L2 and L3 and t L3, L2 and L1 has to be follow respectively.

Let us try to analyze probability of breaking this security from untrustworthy vendor or hacker H. If H breaks the confidentially then he should be success in L1(unlock key) , L2(decryption key & algorithm) and L3 (metadata details) levels. As shown in figure 6 , to breach confidentiality H should Success as SSS probability if we take sample space for 3 leves with Success S and Failure F outcomes.

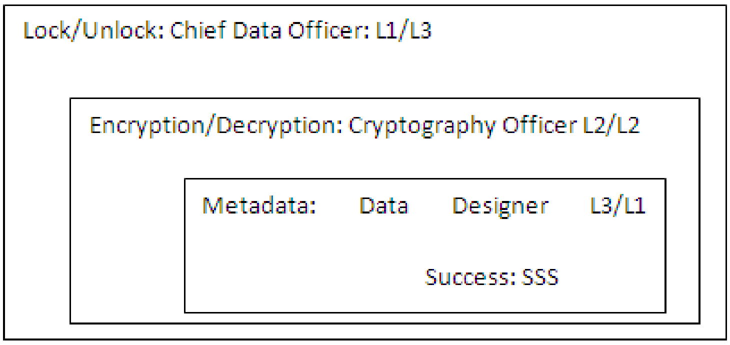

Lock/Unlock: Chief Data Officer: L1/L3
Encryption/Decryption: Cryptography Officer L2/L2
Metadata: Data Designer L3/L1
Success: SSS

Figure 6: Multilevel Cryptography Security Model

Probability of Hacking Event E is P(E)=P(SSS)= 1/8. Therefore (probability Theorem) for Inverse of Event P(E~)=1-P(E)= 1-1/8 =>7/8.

Consider the attack of n times and what is the probability of getting exactly three success SSS. Applying Probability Theorem 3.6 Given n Bernoulli trials with probability p of success on each experiment, the probability of exactly j successes is can be derived as

$$b(n, p, j) \quad = \binom{n}{j} p^j q^{n-j}$$

Where n=number of times attacks, j = number of success, p = probability of success in each try = 1/2=0.5 q=1-p.

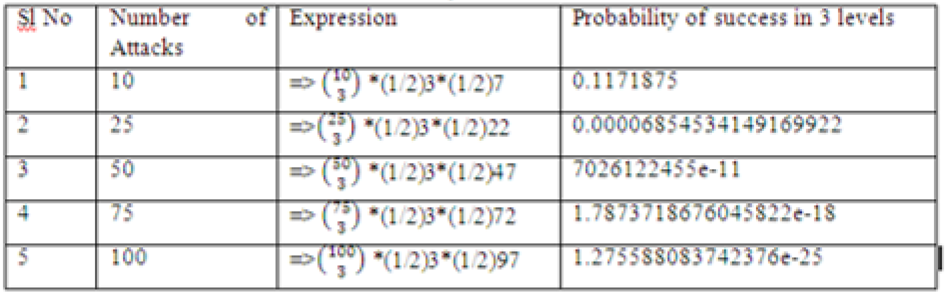
Case 1: If hacker attacks 10, 100 and 1000 times, what is the probability of success in all three levels?

b (10, 0.5,3)= $\binom{10}{3}$ *(1/2)3*(1/2)7 = 120*(1/8)*(1/128)

=120/1024 =15/128

=>120*0.125*0.0078125 =>15/128=>0.1171875

Table 2: Probability of success in 3 level



Sl No	Number of Attacks	Expression	Probability of success in 3 levels
1	10	=> $\binom{10}{3}$ *(1/2)3*(1/2)7	0.1171875
2	25	=>$\binom{25}{3}$ *(1/2)3*(1/2)22	0.00006854534149169922
3	50	=> $\binom{50}{3}$ *(1/2)3*(1/2)47	7026122455e-11
4	75	=> $\binom{75}{3}$ *(1/2)3*(1/2)72	1.7873718676045822e-18
5	100	=>$\binom{100}{3}$ *(1/2)3*(1/2)97	1.275588083742376e-25

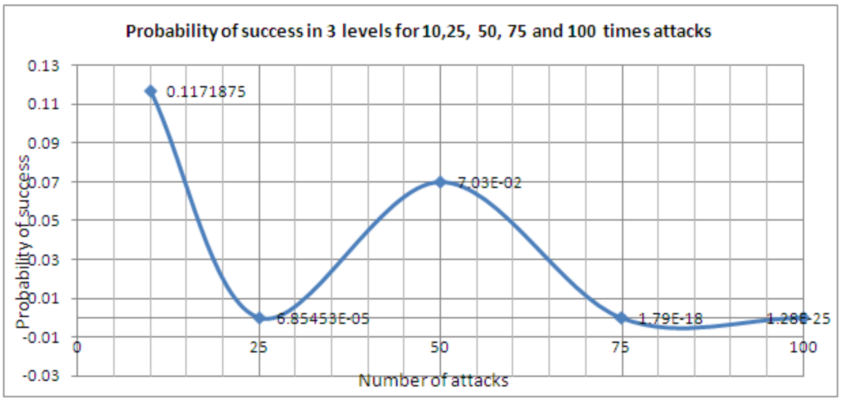


Figure 7: Graphical Representation: Security distribution for attacks n= {10, 25, 50, 75, 100} for 3 levels.

Table 3: Probability of success for 500 attacks in different levels of security

Sl No	Probability of success in levels (j)	Expression	500 Times Attacks (n=500)
1	3	=> $\binom{500}{3}$ *(1/2)3*(1/2)497	6.326314968353156e-144
2	4	=> $\binom{500}{4}$ *(1/2)4*(1/2)496	7.860446332904115e-142
3	5	=> $\binom{500}{5}$ *(1/2)5*(1/2)495	7.797562759063748e-140
4	6	=> $\binom{500}{6}$ *(1/2)6*(1/2)494	6.432989283101199e-138

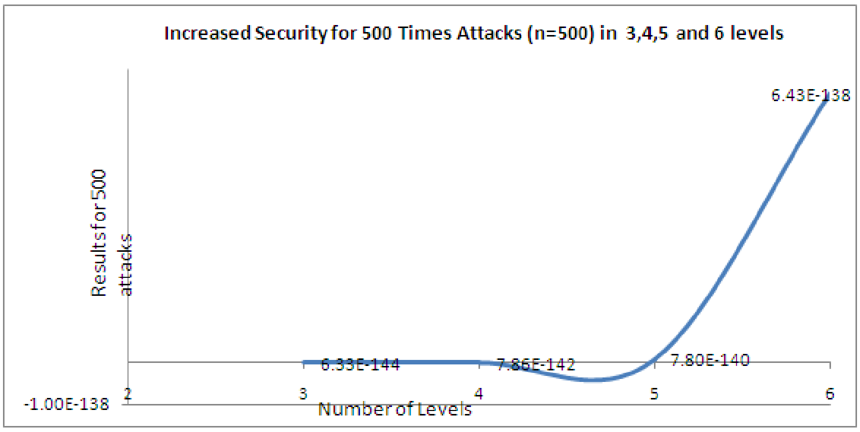


Figure 8: Graphical Representation Security Distribution for j= {3, 4, 5, 6} for 500 times attack

Different ways of attack could be man in the middle attack, phishing attack, multi tenancy attack, brute force attach, dictionary attack and so on. By providing multilevel security with multi way and multi man operation we can provide better security to cloud storage. As shown in above graph1, though the any number of attacks increases the security continues to be stable and security gets increases when the levels are increased. Though there are multilevel and each individual in the level have no burden of remembering many passwords and techniques. Hence it is not complex too.

5 Conclusion and Future Enhancement

Cryptography is a best technology to store customer data in cloud. Many algorithms and methods are exists to place the customer data in encryption format. However, it is difficult to gain customer confidence amount the service provider. We presents on method which helps customer to process their data in multiple levels and ways before migrating to cloud. Hence the customer no need worry about the service provider cryptography way. Both migration and accession steps are discussed in detail. Further we would like to implement this algorithm and make this as a customer side migration package known as multilevel cloud cryptographer software package.

ACKNOWLEDGMENT

Our sincere thanks to Prof. K N B Murthy, Principal and Prof. Shylaja S S, HOD, Department of Information Science and Engineering, PES University, Bangalore, for their constant encouragement.

REFERENCES

[1]. Kawser Wazed Nafi, Tonny Shekha Kar, Sayed Anisul Hoque, Dr. M. M. A Hashem, Newer User Authentication, File encryption and Distributed Server Based Cloud Computing security architecture, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No. 10, 2012, 181-185.

[2]. Eman M.Mohamed, Hatem S. Abdelkader, Enhanced Data Security Model for Cloud Computing, The 8th International Conference on INFOrmatics and Systems (INFOS2012) - 14-16 May Cloud and Mobile Computing Track, Faculty of Computers and Information - Cairo University, CC-12.

[3]. Kan Yang, Xiaohua Jia, An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 24, NO. 9, SEPTEMBER 2013, 1717-1726.

[4]. Cong wang, Qian wang, and Kui ren, Wenjing Lou,"Ensuring data storage security in cloud computing" at IEEE (8-1-4244-3876-1/09).

[5]. William, S., 2005. Cryptography and Network Security Principles and Practices. 4th Edn. PHI.

[6]. CA Technologies cloud authentication system http://www.ca.com/us/authentication-system.aspx

[7]. X. Suo, Y. Zhu, G. S. Owen, "Graphical passwords: A survey," in Proc. 21st Annual Computer Security Application Conf. Dec. 5–9, 2005, pp. 463–472.

[8]. S. Wiedenbeck, J. Waters, J.-C. Birget, A. Brodskiy, and N. Memon, “Authentication using graphical passwords: Basic results,” in Proc. Human-Compute. Interaction Int., Las Vegas, NV, Jul. 25–27, 2005.

[9]. Fawaz A. Alsulaiman and Abdulmotaleb El Saddik,“Three-Dimensional Password for More Secure Authentication,” IEEE, http://ieeexplore.ieee.org., Last Updated – 6 Feb 2008.

[10]. [10]Dinesha H A, Dr.V.K. Agrawal, Multi-level Authentication Technique for Accessing Cloud Services, IEEE conference, Dindigul.

[11]. Mostafa Hajivali , Faraz Fatemi Moghaddam , Maen T. Alrashdan , Abdualeem Z. M. Alothmani , Applying an Agent-Based User Authentication and Access Control Model for Cloud Servers, ICTC 2013, 978-1-4799-0698-7/13, 807-902,2013.

[12]. Laurent Hubert, Renaud Sirdey, Authentication and secured execution for the Infrastructure-as-a-Service layer of the Cloud Computing model, 2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 978-0-7695-5094-7, 291-296, 2013.

[13]. Ming-Huang Guo, Horng-Twu Liaw, Li-Lin Hsiao, Chih-Ta Yen, Authentication Using Graphical Password in Cloud, 177-181, 2013.

[14]. H. B. Tang*, Z. J. Zhu, Z. W. Gao, Y. Li, A SECURE BIOMETRIC-BASED AUTHENTICATION SCHEME USING SMART CARD,IEEE, 39-43,2013.

[15]. A. K. Das. “Analysis and improvement on an efficient biometric-based remote user authentication scheme using smart cards”, IET Information Security, 5 (3), pp. 145-151, 2011.

[16]. Wei Xie1, Lei Xie2, Chen Zhang1, Quan Zhang1, Chaojing Tang1, Cloud-based RFID Authentication, 2013 IEEE International Conference on RFID, 978-1-4673-5750-0/13,168-175, 2013.

[17]. Bernd Zwattendorfer, Arne Tauber, SECURE CLOUD AUTHENTICATION USING EIDS, Proceedings of IEEE CCIS2012, 978-1-4673-1857-0/12/, 397-401, 2012.

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TNC Transactions on Networks and Communications

VOLUME 2, ISSUE 6
ISSN: 2054 - 7420

Internet Economics of Distributed Systems

Hans W. Gottinger
STRATEC Munich Germany
gottingerhans@gmail.com

ABSTRACT

A macroscopic view of Internet-based distributed computer systems reveals the complexity of the organization and management of the resources and services they provide. The complexity arises from the system size (e.g. number of systems, number of users) and heterogeneity in applications (e.g. online transaction processing, e-commerce, multimedia, decision support, intelligent information search) and resources (CPU, memory, I/O bandwidth, network bandwidth and buffers, etc.) In a large distributed system, the set of systems, users and applications is continuously changing. In this paper we address some of the management issues of providing Quality of Service (QoS), pricing, and efficient allocation of resources (computational resources) in networks and systems facilitated through economic mechanism design.

Keywords: Internet Economics, Network Economy, Distributed Systems, Mechanism Design, Performance Management

'…the Internet is in its essence an economy, a game, an arena where agents act selfishly and are affected by everybody's decisions...' Hall, Nikolova and Papadimitriou, in Deng and Graham [11]

1 Introduction

A macroscopic view of decentralized (distributed) computer systems reveals the complexity of the organization and management of the resources and services they provide. The complexity arises from the system size (e.g. number of systems, number of users) and heterogeneity in applications (e.g. online transaction processing, e-commerce, multimedia, intelligent information search, auctions) and resources (CPU, memory, I/O bandwidth, network bandwidth and buffers, etc.)

The complexity of resource allocation is further increased by several factors. First, in many distributed systems, like the present day web, the resources are in fact owned by multiple organizations. Second, the satisfaction of users and the performance of applications is determined by the simultaneous application of multiple resources. For example, a multimedia server application requires I/O bandwidth to retrieve content, CPU time to execute server logic and protocols, and networking bandwidth to deliver the content to clients. The performance of applications may also be altered by trading resources. For example, a multimedia server application may perform better by releasing memory and acquiring higher CPU priority, resulting in smaller buffers for I/O and networking but improving the performance of the communication protocol execution (Gupta and Stahl[1]).

Finally, in a large distributed system, the set of systems, users and applications are continuously changing. In this paper we address some of the issues in managing Quality of Service (QoS) and pricing, and efficient allocation of resources (computational resources) in networks and systems. Resource allocation in networks relate to computational models of networks, as developed in the

DOI: 10.14738/tnc.26.847
Publication Date: 06th January 2015
URL: http://dx.doi.org/10.14738/tnc.26.847

works of Radner [2] ,Mount and Reiter [3] Mount and Reiter [4, Chap.4], van Zandt [5] . The economic features of Internet based network economies have been treated by Gottinger [6, Chap.9] to which we specifically refer. In this context they emanate from certain types of queueing systems, Kleinrock [7] and Wolff [8] on generalized networks.

The structure of this paper is as follows. Sec. 2 exhibits some broader design criteria on large scale networks which underlie the heterogeneity of Internet based resource allocation and use.

Also it shows the major components of an interface architecture with which an 'economically enhanced resource manager' (Macias *et al*. [9]) is confronted.

Sec. 3indicates more broadly the scope of mechanism design approaches that link economic modelling to computational resources, at the interface of economics, computer and management science.

Sec. 4 deals with a specific class of problems arising in mechanism design how in resource allocation processes pricing schemes have to be made 'incentive compatible'.

Sec. 5 relates to the basic structure of a data management economy which more recently features in major application areas as in grid computing, cloud computing, sponsored search auctions, broadcast protocols, and other areas like procurement auctions, spectrum auctions, communication networks, supply chain formation and social networks.

Strategic management issues emerging through resource provisioning and pricing are covered in Sec. 6, Conclusions follow in Sec. 7.

Some examples for service architectures relating to large scale distributed systems are sketched in the Appendix.

2 The Rationale of Economic Models in Networking

There are intrinsic interfaces between human information processing and networking that show the usefulness of economic modelling (as advanced early by Ferguson et.al [13]).

In designing resource allocation and control mechanisms in complex distributed systems and networks several goals need to be considered and could be traced in the literature in more detail, i.e. among others, Shenker *et al.* [10] ,Deng and Graham[11] , and Neumann *et al.* [12].

2.1 Decentralization*:*

In an economy, decentralization is provided by the fact that economic models consist of agents which selfishly attempt to achieve their goals. Suppose there are two types of economic agents: suppliers and consumers. A consumer attempts to optimize its individual performance criteria by obtaining the resources it requires, and is not concerned with system-wide performance. A supplier allocates its individual resources to consumers. A supplier's sole goal is to optimize its individual resources to consumers. A supplier's sole goal is to optimize its individual satisfaction (profit) derived from its choice of resource allocation to consumers.

2.2 Pricing and Performance:

Most economic models introduce money and pricing as the technique for coordinating the selfish behavior of agents. Each consumer is endowed with money that it uses to purchase required resources. Each supplier owns a set of resources, and charges consumers for the use of its resources. The supplier prices its resources based on the demand by the agents , and the available supply. Consumers buy resources or services such that the benefit they receive is maximized. Consumer-agents buy resources based on maximizing performance criteria. As a whole the system performance is determined by some combination of the individual performance criteria.

2.3 Administrative Domains*:*

Often large distributed systems and computer networks spread over several domains, the control of resources is shared by multiple organizations that own distinct parts of the network. In such an environment, each organization will have a set of services that it supports. Economic principles of pricing and competition provide several valuable insights into decentralized control mechanisms between the multiple organizations and efficient service provisioning.

2.4 Scalability*:*

A key issue in designing architectures for services in large computer networks and distributed systems is scalability. With the ever growing demand for new services, flexible service architectures that can scale to accommodate new services is needed. Economic models of competition provide, in a natural fashion, mechanisms for scaling services appropriately based on service demand and resource availability.

3 Mechanism Design Approaches

Network allocation and pricing could be looked at as part of mechanism design theory (Hurwicz and Reiter[15]) and in differential form by Williamson [16]. In a more economic historical context the justification for linking market mechanism to computational resource allocation may be attributed to the Austrian economist F.A.Hayek [17], so what we suggest an Internet based distributed system as a sort of Hayekian mechanism design. (This may fly into the face of many interventionistic Internet economists). More specific mechanism design approaches for distributed networks and grid-type systems are covered by Narahari *et al*. [18] and Neumann *et al*. [12] , see also Meinel and Tison [19]. In the context of computational resources, specifically, an algorithmic mechanism design uses a computational platform with an output specification and agents' preferences represented by utilities (Nisan[20]).

In its general form for distributed systems, the user can indicate the 'type' of transmission and the workstation in turn reports this type to the network. To ensure truthful revelation of preferences, the reporting and billing mechanism must be incentive compatible.

Most studies of resource allocation mechanisms have used a performance model of the resource, where the very concept of the resource is defined in terms of measurable qualities of the service such as utilization, throughput, response time (delay) and so on. Optimization of resource allocation is defined in terms of these measurable qualities, as a basis of performance management. One novelty introduced by the economic approach is to design a system which takes into account the diverse QoS requirements of users, and therefore use multiobjective (utilities) optimization techniques to characterize and compute optimum allocations. Economic modelling of computer and communication resource sharing uses a uniform paradigm described by two level modelling: QoS requirements as inputs into a performance model that is subject to economic optimization.

In the first step, one transforms QoS requirements of users to a performance (example: queueing service model). This model establishes quantifiable parameterization of resource allocation. For example, average delay QoS requirement, when based on a FIFO queueing model, is a function of resources, bandwidth and buffer, and user traffic demands. These parameters are then used to establish an economic optimization model. The question of whether the resource is a piece of hardware, a network link, a software resource such as a database or a server, or a virtual network entity such as a TCP/IP connection is not of primary importance. The first modeling transformation eliminates the details and captures the relevant behaviors and the optimization parameters.

A reasonable approach to follow evolves in the following sequence. Many users present QoS demands, which are translated into demands on resources based on a performance model. The suppliers compute the optimal allocations based on principles of economic optimization and market mechanisms. Once the optimization is done, the results provide inputs to mechanisms for QoS provisioning, such as scheduling of resources and admission of users in networks and load balancing in distributed systems.

3.1 Optimal allocation and QoS

We establish and solve a problem of allocating resources and providing services (QoS) to several classes of users at a single link (Gottinger [6], Chap. 9). The resources at the link are buffer space and bandwidth. The link (network provider) prices per unit buffer and bandwidth resources.

A simple example on the representation of QoS parameters is the bandwidth-buffer tradeoff. Bandwidth can be traded for buffer space and vice versa to provide the same QoS. If a bandwidth is scarce, then a resource pair that uses less bandwidth and more buffer space should be used. Resource pricing is targeted to exploit this tradeoff to achieve efficient utilization of the available resources. The pricing concept for a scarce resource is well-known in economics, but in the context of exploiting the bandwidth-buffer tradeoff, Low and Varaiya [21]used non-linear optimization theory to determine centralized optimal shadow prices in large networks. With respect to large scale application, however, the complex optimization process limits the frequency of pricing updates, which causes inaccurate information about available resources. In order to make pricing in the context of a buffer-bandwidth tradeoff more adjustable and flexible it should be based on decentralized pricing procedures according to competitive bidding in large markets where prices will be optimal prices if the markets are efficient. This would also allow flexible pricing which results in accurate representation of available resources in that prices are updated as the instance connect request arrives. The subsequent procedure is based on distributed pricing as a more feasible alternative to optimal pricing.

Here are the steps involved to invoke an incentive compatible pricing scheme based on QoS needs.

The consumers (user traffic classes), via economic agents, buy resources such that their QoS needs are satisfied. The network provider prices resources based on demand from the consumers. The ingredients are as follows:

- **Economic models**: use competitive economic models, of the type as outlined by Scarf [14], to determine the resource partitions between user traffic classes, which compete to obtain buffer and bandwidth resources from the switch suppliers.
- **Optimal allocations using economic principles**: look for Pareto optimal allocations that satisfy QoS needs of agents. Agents represent QoS via utility functions which capture the multiple performance objectives.
- **Pricing based on QoS**: compute equilibrium prices (or approximate prices) based on the QoS demands of consumers. Prices are set such that the market demand and supply are met. Prices help in determining the cost of providing a service. (In practical application this may be a hard task to do.)

- **Priorities**: using the economic framework, show a simple way to support priority service among the user-classes (or agents).
- **Decentralization**: show a natural separation between the interactions of the user-classes (represented by agents) and the network switch suppliers. The interaction is purely competitive and market based. This decentralization promotes scalable network system design.

3.2 Scheduling and pricing mechanisms

Consider a dynamic system where sessions arrive and leave a traffic class, and demand fluctuates over time. In such a setting, we investigate practical mechanisms, such as packet level scheduling to provide bandwidth and buffer guarantees, admission control mechanisms to provide class QoS guarantees, practical pricing to capture the changing demand, and charging mechanisms for user sessions within a class.

- Scheduling algorithms for class based QoS provisioning: provide novel scheduling mechanisms, which allocate bandwidth and buffer for meeting the demand from traffic classes. The scheduling mechanism allocates bandwidth, which is computed from the economic optimization.
- Admission Region and Control: compute the admission control region of the agents on the economic model. Due to the natural separation between those who control the admission of sessions into the traffic class, the admission region can be determined.
- Propose simple pricing models which capture the changing demand, and are easy to implement. Propose extended QoS based charging mechanisms for sessions in a class with applications to charging in ATM Networks and Integrated Services Internet.

3.3 Network and Server Economies

Consider first a network economy, of many parallel routes or links, where several agents (representing user classes) compete for resources from several suppliers, where each supplier represents a route (or a path) between a source and destination. Agents buy resources from suppliers based on the QoS requirements of the class they represent. Suppliers price resources, independently, based on demand from the agents. The suppliers connect consumers to information providers, who are at the destination; the flow of information is from information providers to the consumers. This formulates and solves problems of resource allocation and pricing in such an environment.

Then consider a server economy in a distributed system. Again, we use a similar model of interaction between agents and suppliers (servers). The servers sell computational resources such as processing rate and memory to the agents for a price. The prices of resources are set independently by each server based on QoS demand from the agents. Agents represent user classes such as transactions in database servers or sessions for Web servers that have QoS requirements such as response time. Examples are given in Gottinger [22].

3.4 Server Economy: architecture for interaction

Consider a large scale distributed information system with many consumers and suppliers. Suppliers are content providers such as web servers, digital library servers, multimedia database and transaction servers. Consumers request for and access information objects from the various suppliers and pay a certain fee or no fee at all for the services rendered.

Consider that third party suppliers provide information about suppliers to consumers in order to let consumers find and choose the right set of suppliers.

3.4.1 Access and dissemination:

Consumers query third-party providers for information about the suppliers, such as services offered and the cost (price). Likewise, suppliers advertise their services and the costs via the third party providers in order to attract consumers. Consumers prefer an easy and simple way to query for supplier information, and suppliers prefer to advertise information securely and quickly across many regions or domains. For example, consider a user who wishes to view a multimedia object (such as a video movie). The user would like to know about the suppliers of this object, and the cost of retrieval of this object from each supplier.

3.4.2 Performance requirements:

Users wish to have good response time for their search results once the queries are submitted. However, there is a tradeoff. For more information about services offered, advanced searching mechanisms are needed , but at the cost of increased response time. In other words, users could have preferences over quality of search information and response time. For example , users might want to know the service costs in order to view a specific information object. In large networks , there could be many suppliers of this object, and users may not want to wait forever to know about all the suppliers and their prices. Instead, they would prefer to get as much information as possible within a certain period of time (response time).

From the above example, in order to let many consumers find suppliers, a scalable decentralized architecture is needed for information storage, access and updates.

Naming of services and service attributes of suppliers becomes a challenging issue when hundreds of suppliers spread across the globe. A simple naming scheme to connect consumers, across the Internet, with information about suppliers is essential. The naming scheme must be extensible for new suppliers who come into existence. A name registration mechanism for new suppliers and a de-registration mechanism (automatic) to remove non-existent suppliers is required. In addition, naming must be hierarchical, domain based (physical or spatial domains) for scalability and uniqueness. Inter-operability with respect to naming across domains is an additional challenging issue not covered in this paper.

The format of information storage must be simple enough to handle many consumer requests quickly within and across physical domains. For better functionality and more information, a complex format of information storage is necessary, but at the cost of reduced performance. For example, a consumer, in addition to current service cost, might want to know more information such as the cost of the same service during peak and off-peak hours, the history of a supplier, its services, and its reputation, in order to make a decision. This information has to be gathered when requested. In addition, the storage formats must be inter-operable across domains.

3.4.3 Performance:

A good response time is important to make sure consumers get the information they demand about suppliers within a reasonable time period, so that decision-making by consumers is done in a timely fashion. In addition, the design of the right architectures for information storage and dissemination is necessary for a large scale market economy to function efficiently. Using the previous example, consumers and suppliers would prefer an efficient architecture to query for and post information.

Consumers would prefer good response time in obtaining the information, and suppliers prefer a secure and fast update mechanism to provide up-to-date information about their services.

3.4.4 Security:

In transferring information and updating information at the bulletin boards (name servers) is crucial for efficient market operation and smooth interaction between consumers and suppliers. For this the third party suppliers (naming services) have to provide authentication and authorization services to make sure honest suppliers are the ones updating information about their services.

4 Allocation and Pricing Models

In economic models, there are two main ways to allocate resources among the competing agents. One of them is the exchange based economy and the other is the price based economy. In the exchange based economy, each agent is initially endowed with some amounts of the resources. They exchange resources until the marginal rate of substitution of the resources is the same for all the agents. The agents trade resources in the direction of increasing utility (for maximal preference). That is, two agents will agree on an exchange of resources (e.g. CPU for memory) which results in an improved utility for both agents. The Pareto optimal allocation is achieved when no further, mutually beneficial, resource exchanges can occur. Formally, an allocation of resources is Pareto Optimal when the utility derived by the competing economic-agents is at the maximum. Any deviation from this allocation could cause one or more economic agents to have a lower utility (which means the agents will be dissatisfied).

In a price based system, the resources are priced based on the demand, supply and the wealth in the economic system. The allocations are done based on the following mechanisms. Each agent is endowed with some wealth. Each agent computes the demand from the utility function and the budget constraint. The aggregate demand from all the agents is sent to the suppliers who then compute the new resource prices. If the demand for a resource is greater than its supply, the supplier raises the price of the resource. If there is surplus supply, the price is decreased. The agents again compute their demands given the current prices and present the demand to the suppliers. This process continues iteratively until the equilibrium price is achieved where demand equals the supply.

Bidding and auctioning resources is another form of resource allocation based on prices. There are several auctioning mechanisms such as the Sealed Bid Auction, Dutch Auction, and English Auction. The basic philosophy behind auctions and bidding is that the highest bidder (or in the Vickrey auction the second highest bidder) always gets the resources, and the current price for a resource is determined by the bid prices.

4.1 Allocation Principles

What are the general allocation principles? Can economic models give insight into the allocation mechanisms that can cause the computer system to reach equilibrium? Can these principles be used practically to evolve the computer system in a way that price equilibrium can be achieved? Even devoting the entire WINE 2007 proceedings to those issues, with active participation of K. Arrow, H. Scarf and C. Papadimitriou (in Deng and Graham [11]), still many practical issues of implementation haven’t been yet finally resolved .

5 The Data Management Economy

Unlike the flow control and load balancing economies where users maximize an utility function to compute the required allocation, this economy considers data migration , replication and pricing strategies for a data management economy as evidenced by large scale e-commerce facilitated through new platforms in grid computing, cloud computing and related application areas (Kushida et al. [23]). The problem of data migration, storage and replication is formulated in an economic setting. Transactions that enter the system for service are charged by the processors for read and write access to data objects. Processors also lease resources to other processors to make profit using the revenue they earn.

The distributed system consists of N processing nodes connected via links. Each processor P_i ($i \in [1,N]$) has rate r_i at which it can process operations on local data. A link e_{ij} connects processor P_i to P_j. There are M data object denoted by D_1 , D_2, ..., D_M . $S(D_i)$ defines the size of D_i in bytes. The economy treats these as abstract data objects. In a real system, they could correspond to relations, tuples, files, records or any other data structure. The data management problem is to minimize the mean transaction response time with the following as control variables.

- Number of copies of data object
- Assignment of copies to processing nodes
- Pricing strategies of suppliers

In the data management economy there are four types of agents. The consumers are transactions, and the suppliers are data object managers, local data agents and processors as through cloud computing. The economy functions in the following way . Each transaction T that arrives has an allocation of money M_T. Transactions pay to access data at a processor P_i. Data access is provided by the processor by leasing copies of data objects from data object managers. The local data agents act as an intermediary between a processor P_i and the object managers (remote). Two economic factors cause the data management economy to adapt the number of read copies of each object D_j to the read/write ratio. These are:

- The total revenue that all processors earn by selling Read(D_j) decreases as the initially set
- price of the agents given its wealth p_w increases
- The read lease price for D_j increases linearly with the number of copies c(j)
- The data management economy uses decentralized decision making to compute the number of read copies of each object. The business strategies of the processors are decoupled, and P_i uses only local information to estimate its revenue. The economy adapts itself to any read/write ratio without any external intervention. The economy is not completely self tuning, however, there is a subtle interaction between the following factors: (i) lease price function, (ii) transaction arrival rates and (iii) transaction arrival rates.

6 Strategic Internet Management Issues

6.1 Universal Access

A primary concern in regulating universal access to the Internet, next to security, had been the issue of pricing its services, the maintaining of competition among providers and strengthening incentives for private investment into the network infrastructure. Possible options emerged in identifying the issues toward a workable model:

1) charging by access to telecommunications capacity, e.g., flat rate pricing and keeping distance independent pricing

2) consider network externalities in the economics and growth of networks
3) introduce usage-based linear prices
4) introduce usage-based nonlinear prices

The evolution of Internet pricing poses interesting problems. Flat-rate pricing has been one of the factors that promoted the Internet to expand at a dramatic rate. It has enabled low-cost dissemination, beta-testing and refinement of new tools and applications. The strength of many of these tools is their ability to operate in and rationalize a distributed and heterogeneous structure making it easier to identify and retrieve information sources. The increased demand that has arisen due to the power and new resources these tools have brought to the Internet (and in view of lagging a corresponding capacity expansion due to advanced fiber-optic technology) is likely to create more gridlock and a need for a new pricing model. This despite new regulatory proposals on "net neutrality" emerging, usage based pricing and service charges or more specific content pricing should make the Internet attractive to many new users and also incentivize innovation driven product development on the net. One paradox of usage based pricing is that its implementation may actually cost more on a transaction basis than the underlying cost of transport. Therefore, it very much depends on network accounting capabilities as a critical implementation tool.

6.2 Congestion Problems:

A natural response by shifting resources to expand technology will be expensive and not necessarily a satisfactory solution in the long run. Some proposals rely on voluntary efforts to control congestion. Others have suggested that we essentially have to deal with the problem of overgrazing the commons, e.g. by overusing a generally accessible communication network. A few proposals would require users to indicate the priority they want each of the sessions to receive, and for routers to be programmed to maintain multiple queues for each priority class. If priority class is linked to the value the users attach to it, one could devise schemes of priority pricing. This is where application of mechanism design could help. At congested routers, packets are prioritized based on bids. In line with the design of a Vickrey auction, in order to make the scheme incentive compatible, users are not charged the price they bid, but rather are charged the bid of the lowest priority packet that is admitted to the network. It is well-known that this mechanism provides the right incentives for truthful revelation. Such a scheme has a number of desirable characteristics. In particular, not only do those users with the highest cost of delay get served first, but the prices also send the right signals for capacity expansion in a competitive market for network services. If all of the congestion revenues are reinvested in new capacity, then capacity will be expanded to the point where the marginal value is equal to its marginal cost. More recently, game-theoretic approaches adopt a unified view even for two-sided markets (Ackermann et al. in [11])

6.3 Quality-of-Service Characteristics:

With the Internet we observe a single QoS: "best effort packet service". Packets are transported first come, first serve with no guarantee of success.

Some packets may experience severe delays, while others may be dropped and never arrive.

Different kinds of data place different demands on network services. Email and file transfer requires 100 percent accuracy, but can easily tolerate delay. Real-time voice broadcasts require much higher bandwidth than file transfers and can tolerate minor delays but cannot tolerate significant distortions. Real-time video broadcasts or video telephony over VOIP have very low tolerance for delay and distortion. Because of these different requirements, network allocation algorithms should be designed to treat different types of traffic differently but the user must truthfully indicate which type of traffic (s) he is preferring, and this would only happen through incentive compatible pricing schemes.

QoS can be affected by various factors , both quantitative (network latency, CPU performance,…) and qualitative, among the latter could proliferate reputation systems that hinge on trust and belief in a certain QoS level being achieved, resulting in a service level arrangement (SLA) comprising service reliability and user satisfaction (Anandasivam and

Neumann in[12].

6.4 Internet and Telecommunications Regulation:

In contrast to traditional telecommunications services Internet transport itself is currently unregulated but services provided over telecommunication carriers are not. This principle has never been consistently applied to telephone companies since their services over fixed telephone lines also used to be regulated.

There have been increasing demands, sometimes supported by established telecommunication carriers that similar regulatory requirements should apply to the Internet. One particular claim is “universal access” to Internet services, that is, the provision of basic Internet access to all citizens at a very low price or even for free. What is a basic service, and should its provision be subsidized? For example, should there be an appropriate access subsidy for primary and secondary schools? A related question is whether the government should provide some data network services as public goods.

A particular interesting question concerns the interaction between pricing schemes and market structure for telecommunications services. If competing Internet service providers offer only connection pricing, inducing increasing congestion, would other service providers be able to attract high value “multimedia” users by charging usage prices but offering effective congestion control? On the other hand, would a flat rate connection price provider be able to undercut usage-price providers by capturing a large share of baseload customers who would prefer to pay for congestion with delay rather than with a fee. Could this develop into a fragmented market with different Internets? These developments may have profound impacts to shape a future telecommunications industry which may be taken over by different structured layers of the Internet.

7 Discussion

In this paper we focus on applications of mechanism design to resource management problems in distributed systems and computer networks. These concepts are used to develop effective market based control mechanisms, and to show that the allocation of resources are Pareto optimal. The emphasis here is on management implications given the economics of the Internet.

We follow novel methodologies of decentralized control of resources, and pricing of resources based on varying, increasingly complex QoS demands of users. We bring together economic models and performance models of computer systems into one framework to solve problems of resource allocation and efficient QoS provisioning matching large-scale e-commerce applications. The methods can be applied to pricing services in ATM networks and (wireless) Integrated Services Internet of the future. We address some of the drawbacks to this form of modelling where several agents have to use market mechanisms to decide where to obtain service (which supplier?). If the demand for a resource varies substantially over short periods of time, then the actual prices of the resources will also vary causing several side effects such as indefinite migration of consumers between suppliers. This might potentially result in degradation of system performance where the resources are being underutilized due to the bad decisions (caused by poor market mechanisms) made by the users in choosing the suppliers. As in real economies, the resources in a computer system may not easily be substitutable. The future work is to design robust market mechanisms and rationalized pricing schemes which can handle surges in demand and variability, and can give price guarantees to consumers over longer periods of time some of which have been discussed by Spulber and Yoo ([24], Chap.12). Another drawback is that resources in a computer system are indivisible resulting in non-smooth utility functions which may yield sub-optimal allocations, and potential computational overhead.

In addition to models for QoS and pricing in computer networks, we are also working towards designing and building distributed systems using market based mechanisms to provide QoS and charge users either in a commercial environment or in a private controlled environment by allocating quotas via fictitious money (charging and accounting) by central administrators.

In summary, economic based management is useful for implementing and operating internet-type systems. The Internet currently connects hundreds of millions of users and thousands of sites. Several services exist on many of these sites, notably the World Wide Web (WWW) which provides access to various information sources distributed across the Internet. Many more services (multimedia applications, commercial transactions) are to be supported in the Internet. To access this large number of services, agents have to share limited network bandwidth and server capacities (processing speeds). Such large-scale networks require decentralized mechanisms to control access to services. Economic/managerial concepts such as pricing and competition can provide some solutions to reduce the complexity of service provisioning and decentralize the access mechanisms to the resources.

8 Conclusions

We explore name service architectures for disseminating information about suppliers and their services to consumers, and look at the main properties of these architectures.

We use analytical models to compute the expected response time for consumers to access for information in each architecture. We compare the three architectures in terms of performance, security and flexibility. The economic models of networks and systems, and the corresponding mechanisms described previously can use the framework mentioned to allocate resources and provision services in a real environment.

REFERENCES

[1] Gupta A, Stahl DO. An Economic Approach to Networked Computing with Priority Classes. Cambridge, Ma.: MIT Press 1995

[2] Radner R .The Organization of Decentralized Information Processing. Econometrica 1993; 62: 1109-1146.

[3] Mount KR, Reiter S. On Modeling Computing with Human Agents. Center for Math. Studies in Economics and Management Science. Northwestern Univ., Evanston, Ill. 1994 , No.1080

[4] Mount KR, Reiter S. Computation and Complexity in Economic Behavior and Organization. Cambridge: Cambridge Univ. Press 2002

[5] Van Zandt T. The Scheduling and Organization of Periodic Associative Computation: Efficient Networks. Rev Ec Design 1998; 3: 93-127

[6] Gottinger HW. Strategic Economics in Network Industries. New York: NovaScience 2010

[7] Kleinrock L. Queueing Systems Vol. 2. New York : Wiley 1976

[8] Wolff RW. Stochastic Modeling and the Theory of Queues. Englewood Cliffs, N.J.: Prentice Hall 1989

[9] Macias M, Smith G, Rana O, Guitart J, Torres J. Enforcing Service Level Agreements using Economically Enhanced Resource Manager. In: [12] ; 109-127

[10] Shenker S, Feigenbaum, J. and M. Schapiro. Distributed Algorithmic Mechanism Design, in N.Nisan,T.Roughgarden,E.Tardos and V.V. Vazirani, eds. , Algorithmic Game Theory, Cambridge: Cambridge Univ. Press 2007

[11] Deng X, Graham FC. Internet and Network Economics. Third International Workshop. WINE 2007, San Diego, Berlin, New York: Springer 2007

[12] Neumann D, Baker M, Altmann J, Rana OF, eds. Economic Models and Algorithms for Distributed Systems. Basel: Birkhaeuser 2010

[13] Ferguson, D.F., Nikolaou,C. , Sairamesh,J. and Y.Yemini, "Economic Models for Allocating Resesources in Computer Systems",in S. Clearwater, ed., Market-Based Control: A Paradigm for Distributed Resource Allocation, Singapore: World Scientific, 1995

[14] Scarf H .The Computation of Economic Equilibria. Cowles Commission Monograph.New Haven and London: Yale Univ. Press 1973

[15] Hurwicz L, Reiter S. Designing Economic Mechanisms. PB ed., Cambridge: Cambridge Univ. Press 2006

[16] Williamson SR. Communication in Mechanism Design, A Differential Approach.Cambridge: Cambridge Univ. Press 2008

[17] Hayek FA. The Use of Knowledge in Society. Am Ec Rev 1945; 35:519-530

[18] Narahari Y, Garg D, Narayanam R, Prakash H. Game Theoretic Problems in Network Economics and Mechanism Design Solutions. London: Springer 2009

[19] Meinel C , Tison S eds. STACS 99, 16 Annual Symposion Theoretical Aspects of Computer Science, Trier, Germany, March ,Berlin: Springer 1999

[20] Nisan N. Algorithms for Selfish Agents, Mechanism Design for Distributed Computation, in 16 Annual Symposion Theoretical Aspects of Computer Science, Trier, Germany, March, Berlin: Springer 1999Pages. 1-15

[21] Low S, Varaiya P A. New Approach to Service Provisioning in ATM Networks. IEEE Trans. Networking 1993; 1: Nov. 1, 7-14

[22] Gottinger HW. Quality of Services for Queueing Networks of the Internet , iBusiness 2013; 5: Sept. 1-12

[23] Kushida KE, Murray J , Zysman J Diffusing the Fog: Cloud Computing and Implications for Public Policy , Berkeley Roundtable on the International Economy (BRIE),BRIE Working Paper 197, 2011; March 11

[24] Spulber DF, Yoo CS. Networks in Telecommunications, Economics and Law. Cambridge: Cambridge Univ. Press 2009

APPENDIX

Service Architectures for the Internet Economy

In designing market based frameworks for distributed systems one would like to look at corresponding architectures which let consumers find information about suppliers and their services, and let suppliers advertise QoS information about the services they offer and the corresponding costs.

Consider a large scale distributed information system with many consumers and suppliers. Suppliers are content providers such as web servers, digital library servers, multimedia database and transaction servers. Consumers request for and access information objects from the various suppliers and pay a certain fee or no fee at all for the services rendered. Consider that third party suppliers provide information about suppliers to consumers in order to let consumers find and choose the right set of suppliers.

Access and dissemination: consumers query third-party providers for information about the suppliers, such as services offered and the cost (price). Likewise, suppliers advertise their services and the costs via the third party providers in order to attract consumers. Consumers prefer an easy and simple way to query for supplier information, and suppliers prefer to advertise information securely and quickly across many regions or domains. For example, consider a user who wishes to view a multimedia object (such as a video movie). The user would like to know about the suppliers of this object, and the cost of retrieval of this object from each supplier.

Performance requirements: users wish to have good response time for their search results once the queries are submitted. However, there is a tradeoff. For more information about services offered, advanced searching mechanisms are needed, but at the cost of increased response time. In other words, users could have preferences over quality of search information and response time. For example, users might want to know the service costs in order to view a specific information object. In large networks, there could be many suppliers of this object, and users may not want to wait forever to know about all the suppliers and their prices. Instead, they would prefer to get as much information as possible within a certain period of time (response time).

From the above example, in order to let many consumers find suppliers, a scalable decentralized architecture is needed for information storage, access and updates.

Naming of services and service attributes of suppliers becomes a challenging issue when hundreds of suppliers spread across the globe. A simple naming scheme to connect consumers, across the internet, with information about suppliers is essential. The naming scheme must be extensible for new suppliers who come into existence. A name registration mechanism for new suppliers and a de-registration mechanism (automatic) to remove non-existent suppliers is required. In addition, naming must be hierarchical, domain based (physical or spatial domains) for scalability and uniqueness. Inter-operability with respect to naming across domains is an additional challenging issue not covered in this paper.

The format of information storage must be simple enough to handle many consumer requests quickly within and across physical domains. For better functionality and more information, a complex format of information storage is necessary, but at the cost of reduced performance. For example, a consumer, in addition to current service cost, might want to know more information such as the cost of the same service during peak and off-peak hours, the history of a supplier, its services, and its reputation, in order to make a decision. This information has to be gathered when requested. In addition, the storage formats must be inter-operable across domains.

Performance: a good response time is important to make sure consumers get the information they demand about suppliers within a reasonable time period, so that decision-making by consumers is done in a timely fashion. In addition, the design of the right architectures for information storage and dissemination is necessary for a large scale market economy to function efficiently. Using the previous example, consumers and suppliers would prefer an efficient architecture to query for and post information. Consumers would prefer good response time in obtaining the information, and suppliers prefer a secure and fast update mechanism to provide up-to-date information about their services.

Security in transferring information and updating information at the bulletin boards (name servers) is crucial for efficient market operation and smooth interaction between consumers and suppliers. For this the third party suppliers (naming services) have to provide authentication and authorization services to make sure honest suppliers are the ones updating information about their services.

Architecture Models. For our architecture and design, we choose the existing, operational Internet Domain Name Service (DNS) for reasons of scalability, simplicity, efficiency and performance, and for its distributed architecture. DNS has a simple hierarchical structure for uniquely naming internet hosts across the globe. DNS uses this naming in finding information about hosts located anywhere in the Internet. The naming space is divided among administrative domains, and within each domain, the naming is done independently.

DNS is a simple distributed architecture for storing information about hosts in the Internet. The name service has a database that keeps several resource records (RRs) for each host, indexed by the host domain name. One such RR is the IP address of a host indexed by the hostname. The RR is used commonly for mapping domain names (hostnames) to IP addresses for networking between hosts (example: email)

In addition to this widely used RR, there are several other types of RRs which store more information about a host, and its characteristics. The Internet is divided into domains. Each domain is controlled by a primary name server (NS) and some secondary name servers which replicate the primary NS database for better response time.

Within the DNS naming tree we can add any number of service nodes, which have RRs for storing IP addresses and RRs for service parameter information which is stored in the TXT record of the node. For each server, the TXT RR describes in a simple way (string), the service attribute value pairs.

Within the new DNS functionality and naming schemes, the customer can submit complex queries which can be based on attributes and other information. A customer could also ask information about services in other domains or zones. This means that the DNS engine has to query other name servers for information regarding the services. This querying can be done in a recursive fashion between the primary name servers to obtain information from other domains, similar to the way it is done for IP addresses of hosts in other domains.

We explore three architectures to store and retrieve information about various suppliers. The architectures are designed using the functionality offered by the Internet Domain Naming Service.

Centralized Read-Write (RW) Architecture

Each supplier (host) is registered at the primary NS, which maintains the whole database (DB) of supplier information in the RRs. The TXT RR stores information about services offered by suppliers and its service attributes. Each supplier updates DB securely at NS using Public Key Methods. NS contains information about each supplier. Consumers, via the Web, query NS for service information about each supplier.

Centralized Transfer-Access (TA) Architecture

Each supplier is a primary of its local domain. Each supplier keeps its information local (in the DB). This way the information is updated locally by the supplier and is secure. Suppliers belong to a global primary DNS (NS).

Decentralized Index Based (IB) Architecture

Each supplier maintains its own DB. The DB contains the services offered and prices , and the time periods where prices are fixed and the expiry dates. Each supplier is registered at the primary NS for the domain. The registration of the supplier is done in a secure fashion. A Registration Server exists and authenticates, using private and public key techniques, the digital signature of each host. The IP address of each supplier is stored in the primary name server. Also, the primary NS maintains a list of IP addresses for each service that is being offered in that domain.

Specialized Features in Centralized and Decentralized Models

The resource records of the node services show that www, video, gopher, ftp … are the services offered in this domain.One can use these keywords and find more information about the specific services , and suppliers offering these services and their corresponding service attributes.WWW based access to supplier information: consumers have an access to the supplier information via the world-wide-web interface. All the consumers see is a list of categories of services offered or a simple keyword based search, where the keywords should match with the services being offered in a domain. For example, a user can click on Netscape and obtain all the information about services offered in a domain. Once this is done , a user can pick a specific service and ask for the list of suppliers that offer this service. The requests are submitted via the cgi-bin interface of the www. The responses come back in a form that can be viewed by the Web browser.

Performance Model for RW, TA and IB Architectures

Centralized RW: We assume a simple model to study the performance. The model is based on M/M/1 with (two classes of traffic) queueing system. Read requests from consumers in a domain arrive at the primary at a certain Poisson rate λ_r , and update requests or updates arrive at the primary at a rate λ_w which is also a Poisson distribution. The average service rate of the read request is μ_r which is exponentially distributed, and the average service rate of the update requests is μ_w. Let C be the processing rate of the primary name server. Then the average delay in queueing and service for each request (whether read or write) is Delay and μ_{NS}

Centralized TA: In this model the primary NS services customer queries (all the load). In the simple model, the name server spends some time ion answering queries, and periodically polls the suppliers for information or any updates. We model such a system as an M/M/1 queueing system user queries for reads and writes and at a certain rate the secondaries transmit to the primary, and we assume that the rate has a Poisson distribution model.

Distributed Index Based Access: The primary name server acts as a simple router of requests to the suppliers, who respond with the final answers. Customers query the primary NS, and get a list of suppliers offering a service . They then query each supplier in that list and get more information about their services.

User read requests are first processed at the primary and then routed to the suppliers for more information. The overall request rate remains the same as in previous models. This model is distributed, as the processing of a query is done by suppliers. Therefore, the response time will be lower on an average to the customers compared to the other architectures.

Comparison of Response Time

Model 1 has a lower response time compared to model 2. This is because in model 2, the primary NS spends some time polling for update information from the suppliers. For model 3, we consider that the read requests are split evenly among the suppliers, likewise we consider that the update frequency is the same for each supplier, for the sake of simplicity. As expected model 3 gives a better response time