# UPnP and Secure Group communication Technique for Zero-configuration Environment construction using Incremental Clustering

**Purna Chandra Sethi**

PhD Scholar Department of Master in Computer Application, Utkal University, Bhubaneswar, Odisha

**Abstract**

The scales of smart living are needed from small to large size applications. As the scale of the space increases, we can expect that the requirements for the two features zero-configuration and secure data communication channels are getting more important. The feature of zero-configuration reduces the cost to setup the network and secure data communication channels guarantee both the privacy and confidentiality of possible sensitive data transmitted in the network. In this paper, we integrated two technologies, UPnP and secure group communication techniques, to construct an almost zero-configuration secure environment for smart living spaces. A secure and flexible communication environment is constructed as follows. An UPnP controller is implemented to manage devices in the same administrative domain and hence these devices can be treated as members in the same communication group. Using generalized ring signature algorithm key can be managed for building both point-to-point and broadcast secure channels over the UPnP network.

**Keywords:** SUPnP, Group Communication, Zero-communication Environment, Generalized Ring Signatures, Incremental clustering.

## INTRODUCTION

Nodes in the same network are able to communicate with each other. Services and content displayed at the user end can be presented much more interactively and dynamically. To build a communication network that supporting intelligent devices, in addition to those basic design issues such as scalability, fault tolerance, and availability, there are two more design issues that must be taken into consideration [1]. The first is the ease of system configuration because it could greatly reduce the cost of maintenance. Algorithms for clustering data have been widely used for secure and faster data transmission [2]. Clustering algorithms [3, 4] can be classified as either hierarchical or iterative (partitional, density search, factor analytic or clumping and graph theoretic). Incremental clustering is done in an unsupervised environment and is stored in local computer such that it can be sent quickly between the nodes.

In a network system a large number of information are there for sharing operation. For example, deploying huge amount of information systems in a large organization or software company may require repeating similar setups on each device. An easy-to-configure system can thus reduce the cost of deployment. Second, as the information loaded on these intelligent devices can be customized to the users, the security and privacy of data that transmitted between the device and the network must be protected. According to the factors discussed above, we adopt the concept of the Universal Plug and Play (UPnP) Device Architecture [5] for the ease of device discovery and management. The UPnP architecture supports zero-configuration networking. Different ARQ protocols [6] can be referred for error control of data during data communication. When a device joins the network, it can be automatically discovered and integrated into the existing system [1]. The device then conveys its own capabilities to other devices and also receives the information about capabilities of other devices. With the benefits brought by the UPnP

architecture, service providers do not have to worry about the complicated network settings, and thus can concentrate more on the content. Based on the UPnP architecture, to provide secure communication channels, some aspects have to be taken into account.

In the proposed architecture, control messages are managed by a central control point. Since application layer services may require both unicast and multicast communication, the control point must have the ability to transform message for the two different secure channels. Furthermore, the introducing to generalized ring signature algorithm [7] for key management will not break the zero-configuration property of UPnP architecture. Thus, we proposed a secure-UPnP (SUPnP) framework to integrate both the UPnP architecture and secure communication channels [8].

The rest of this paper is organized as follows. Section 2 shows the complete system architecture within which the generalized ring signature algorithm will be implemented. Section 3 shows the details of the SUPnP protocol, which includes the node registration protocol, the construction of secure data communication channels and the message relaying protocol. Section 4 will contain the implementation of generalized ring signature algorithm in SUPnP framework. It also deals with the several aspects of the proposed framework and we finally conclude in Section 5.

## THE SYSTEM ARCHITECTURE

For the convenience of discussion, in this paper, we assume that devices are connected by a local area network (LAN). Practically, this assumption can be relaxed by the establishment of secure tunnels between devices or by appropriated configured routers for a MAN or WAN application. Fig-1 represents the system architecture SUPnP. A centralized control point device, abbreviated as the controller, is accounted for managing the whole system.

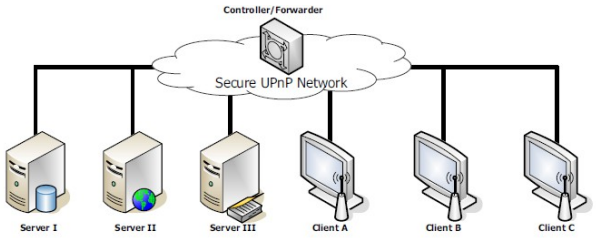



Figure 1:The SUPnP System Architecture

Once a UPnP-compatible device wants to join the network, it sends an IGMP join message to introduce itself and receives IGMP messages sent from the controller. The control point and the devices communication can be represented as specified in fig-2.

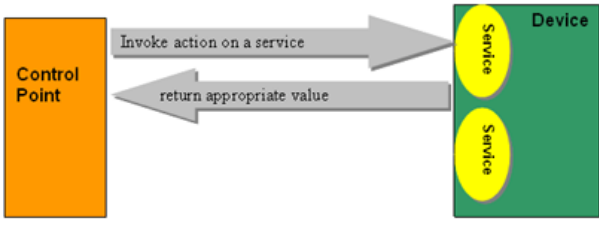



Figure 2: Communication between Control point and Devices

During process execution, a new device may join the system. After the device has joined into the system, it will obtain its own keys through the SUPnP node registration protocol, which will be explained in detail in section 3.1. On completing the registration procedure, services on the device will be advertised to other existing devices. The control point invokes action on a service for appropriate device communication and the device will reply appropriate information based on the request done. Devices except the controller in the proposed system are categorized into client devices and server ones. Under UPnP network, it is unnecessary for the devices to have the knowledge of other devices network settings (such as IP addresses, domain names, service ports, etc.). Once a request message is generated by a client device, it will be sent to the controller through the secure unicast channel. The controller will analyze the request and then broadcast it to the back-end server devices through the secure broadcast channel. On receipt of a request, each server device parses the request header and decides whether it should process the request or not.

When a server replies, the controller will forward the replied messages to the original client device. According to the demand of different service types, a client request may require one or more responses from the server devices. The detail of secure channel construction for client devices, the server devices, and the message relaying service will be further explained in Section 3.

## THE DESIGN OF SPnP

In this section, we explain how the secure UPnP environment is built in detail. Fig-3 shows the protocol architecture of the SUPnP design. The design of the SUPnP network is a layered design. As the under layer follows the UPnP basic device definition [8], the SUPnP protocol is able to coexist with any other UPnP devices.

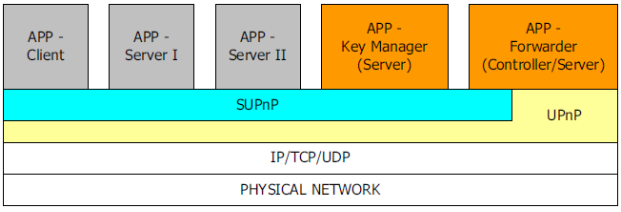



Figure - 3: The UPnP Architecture Environment

Most of the smart devices are built on top of the SUPnP layer. These devices can be classified into two categories, namely the client and the server devices. By definition, the major work of a client device is to interact with the environments and make requests to server devices. On the contrast, server devices are in charge of answering requests from clients. Beside generic clients and servers, we have two special components in the SUPnP network. One is the “key manager”, which runs as a server device and otherwise the “forwarder”, which is also run as a server cooperating with an UPnP controller. The key manager is responsible to maintain the relationship of devices in the SUPnP network. The node registration protocol can be represented as given in fig - 4.

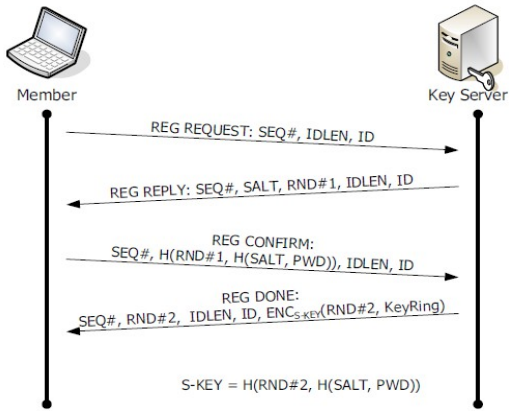



Figure - 4 : Node Registration Protocol

## ERROR CONTROL

When an error is detected in a message, the receiver sends a request to the transmitter to retransmit the faulted message or packet. The most popular retransmission scheme is known as Automatic-Repeat-Request (ARQ). Such schemes, where receiver asks transmitter to re-transmit if it detects an error, are known as reverse error correction techniques. There exist three popular ARQ techniques, as shown in fig - 5.

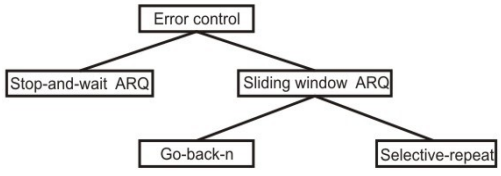



Figure – 5 :Error control techniques

A close comparison of all the ARQ protocols is done in the paper [2] and the result is that selective repeat ARQ protocol provides the efficient and error free data transmission. So, we can also follow the same approach during data transmission within a group communication. During data transmission, the authentication evaluation is required for security of information.

We intend to design an efficient authentication scheme, which is expected to provide a better security and support to Dieffie-Hellman key exchange protocol. Here, we summarize all these requirements to evaluate our new scheme as follows.

**User Anonymity:** The scheme should preserve user’s identity, namely, a server could not tell a user’s

identity. Once the connection between the user and the server has been established, the probability of the server to guess the user's identity is 1/*n*, where *n* is the number of ring members.

**Mutual Authentication:** The scheme should assure that not only can the server verify the legal users, but the users can also verify the server. As in authenticated protocols, mutual authentication is an important attribute; our scheme should also be in favor of it perfectly.

**Forward Security:** The scheme should satisfy forward security, namely, if the session key generated in *j* period has been leaked, the attacker can't forge any session key generated before *j* period. Therefore, the scheme should defeat some attacks like replay attack and so on.

**Reveal User's Identity:** The scheme should be able to reveal the user's identity, namely, after the protocol, if the user wants to reveal his own identity, he can reveal it to the server. In some cases, the server may not believe that the user is the real user, so it is necessary for the user to reveal his identity to the server after the protocol.

## PROPOSED WORK

Our work is based on [7]. In [9], the concept of ring signature was first proposed. Suppose that Alice wishes to generate a ring signature of a message m for a ring of n individuals A1, A2,..., An, where the signer Alice is $A^s$, $1 \le s \le n$. Denote $S=\{A^1, A^2,..., A^n\}$. Each $A^i \in S$ is called a ring member. The public key of $A^i$ is $P^i$ and the corresponding private key is $S^i$. In this paper, we will not distinguish between the ring member and its public key. Therefore, S will also be used to denote the set of public keys of all ring members.

The ring signature proposed by Rivest [9]. It is based on the RSA signature scheme. We call it RSA-based ring signature. The main idea of the RSA-based ring signature is illustrated in Fig – 6 .

The computation of ring signature can be defined using two algorithms:

ring-sign(m,S):

1. Choose a key. The signer $A^s$ first computes the symmetric key k as follows: k=h(m)

2. Pick a random glue value. The signer picks an initialization value $v \in \{0,1\}^b$ uniformly in random.

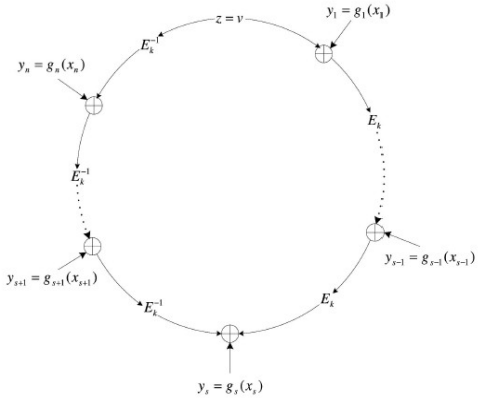



Figure – 6 : Ring Signature

3. Pick random $x^i$. As picks random $x^i$ for all the other ring members $1 \le s \le n$, $i \ne s$ uniformly and independently from $\{0,1\}^b$ and computes

$$y^i = g^i(x^i)$$

4. Solve for $y^s$. $A^s$ solves the following ring equations for $y^s$:

$$C^{k,v}(y^1, y^2, \ldots, y^n) = v.$$

Equivalently, we can solve $y^s$ as follows:

$$y_s = E_k(y_{s-1} \oplus E_k(\cdots \oplus E_k(y_1 \oplus v)))$$
$$\oplus E_k^{-1}(y_{s+1} \oplus E_k(\cdots \oplus E_k^{-1}(y_n \oplus E_k^{-1}(v))))$$

5. Invert $y^s$ using $A^s$ trapdoor permutation. $A^s$'s uses her knowledge of the trapdoor to invert $g^s$ on $y^s$ to obtain $x^s$:

$$x^s = g^{s^{-1}}(y^s).$$

6. Output the ring signature. The signature on the message m is defined to be

$$?\square\square S ; v ; x^1, x^2, \ldots, x^n\square$$

**Ring-verify***(m;□?□S):*

A verifier can check an alleged signature on message m as follows:

1. **Apply the trapdoor permutations.** For i=1, 2, ... , n, the verifier computes

$$y^i = g^i(x^i).$$

2. **Obtain k.** The verifier hashes the message m:

$$k = h(m).$$

3. **Verify the ring equation.** The verifier checks that the $y^i$'s satisfy

$$C^{k,v}(y^1, y^2, \ldots, y^n) = v.$$

If the ring equation is satisfied, the verifier Accepts the ring signature as valid. Otherwise, the verifier Rejects.

Database records are considered or referred as objects. The fitness function in GA is represented in the following way. It is defined using root mean square algorithm which is defined as:

$$f(new^{obj}) = \left( \sum_{i=0}^{n} (meta^{obj}(i) - new^{obj}(i)^2 \right)^{1/2} \quad (1)$$

Where, *n* is number of attributes in database,

*meta*$^{obj}$ is object from meta-file

*new*$^{obj}$ is object to be added

The key idea of clustering is that for each element of a cluster is in neighborhood of a given range (gR), i.e. the cardinality of the neighborhood has to exceed some threshold (tR). If the *f(new*$^{obj}$*)* is greater than threshold tR, then it is selected. Otherwise, *f(new*$^{obj}$*)* is returned back and MAX_THRESHOLD is returned. MAX_THRESHOLD is any largest decimal value.

Algorithmic Settings for GA - The GA operators selection, crossover, mutation and Pc (Crossover Probability) and Pm (Mutation Probability) are listed in Table 1. Initial Population of GA is meta_file. Individuals (meta objects properties) goes through fitness calculation (as per equation 1). After fitness calculation the GA uses operators Crossover, Reproduction and mutation with parameters setting shown in Table 1. The GA undergoes 50 generations.

The after GAs execution, fittest object's CLUSTER_ID will be Identity (ID) for new object.

TABLE 1: PARAMETER SETTING FOR GA

| Setting Type | Value |
|---|---|
| Encoding Scheme | Binary encoding |
| Population size | 200 |
| Evolution generation | 50 |
| Selection | Roulette Wheel |
| Crossover | One point |
| Mutation | Uniform |
| Pc | 0.6 |
| Pm | 0.01 |
| Elitism | Yes |
| Generations | 50 |

**THE ALGORITHM ICGA**

**Terminologies:**

*Object* is referred as a database record with attributes. *meta_file* is meta file having information about clusters. For security, the information will be transferred with a ring topology with a secure digital signature that will be transferred between the nodes within the network using the different ARQ protocols. *gR* is range or threshold.

*new_object* is object to be inserted

**Algorithm** ICGA (*meta_file*, *gR, new_object*)

1. Initialize GA and all its parameters.
2. Get population from *meta_file*.
3. Calculate fitness of each individual
4. Apply GA operators to population
5. Generate new set of population.
6. If (No. of Generations > MAX_GEN)
   Goto Step 3
7. Else

a. Get fitness value of fittest individual after MAX_GEN generations

b. If (FINAL_FITNESS = MAX_THRESHOLD) then

i. Get new CLUSTER_ID (One greater than highest present ID )

ii. Create new entry in *meta_file* with attributes of *new_object.*

c. Else

i. Get CLUSTER_ID of the fittest individual

ii. Set that CLUSTER_ID to new object.

8. Return.

This is how ICGA works. The clustering will be done basing on the time of accessing, such that the frequently used information will be maintained in a single cluster and can be accessed in minimum time due to separate cluster.

**ALGORITHM FOR ICGA IMPLEMENTED USING SELECTIVE REPEAT ARQ PROTOCOL**

The digital signature is considered as a private key that is generated by the local computer and will be sent over the network between a pair of nodes.

Let, MAX_SEQ = 7 and should be in the form of $2^n$-1

NR_BUFS = ((MAX_SEQ + 1)/2)

$S^f$ = S = 0 and $S^l$ = 3 for current application

Time_out = x

ack = false

no_ack = false

1. Send the S bit to the receiver end and wait for *ack* signal from the receiver end.
2. If *ack* is not received at appropriate time i.e. Time_out take place then Set S = S+1 and send the S bit information to the receiver end and wait for *ack* signal.
3. If *ack* is received at the sender end at appropriate time (Time_out) then the sliding window moves S bits.

   Set, $S^f = S^f + S$

   $S^l = S^l + S$

   S = 0

   And apply the process from Step-1.
4. Else apply Step-2
5. If $S > S^l$ then reset the sending process from beginning considering no frame is transferred.
6. If *no_ack* signal is send from the receiver end within the Time_out, then

   Set S = S – 1

   and resend the S bit information to the receiver end.
7. Apply the process until the end of information.

Due to the Selective-Repeat-ARQ protocol implemented, the ICGA provides more accurate information with minimum transmission time. Unnecessarily, same information needs not to be transferred multiple times which reduces the complexity.

## DATA SET COMPARISON

### *A.* Implementation and Experimental Result

The experiment is based on the Clickstream data concept. Clickstream data is a natural by-product of a user accessing World Wide Web (WWW) pages, and refers to the sequence of pages visited and the time these pages were viewed. Clickstream data is to Internet marketers and advertisers. An instance of real clickstream records is the MSNBC dataset, which describes the page visits of users who visited msnbc.com on a single day. There are 989,818 users and only 17 distinct items, because these items are recorded at the level of URL category, not at page level, which greatly reduces the dimensionality. The 17 categories are tabulated with their category number.

| Front page | 1 |
|---|---|
| News | 2 |
| Tech | 3 |
| Local | 4 |
| Opinion | 5 |
| On-air | 6 |
| Misc | 7 |
| Weather | 8 |
| Health | 9 |
| Living | 10 |
| Business | 11 |
| Sports | 12 |
| Summary | 13 |
| Bbs | 14 |
| Travel | 15 |
| msn-news | 16 |
| msn-sports | 17 |

The sample sequences for the data set will be:

1 1

2

3 2 2 4 2 2 2 3 3

5

1

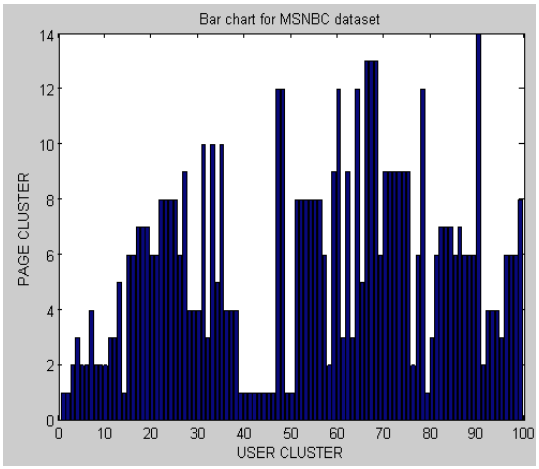
6

6 7 7 7 6 6 8 8 8 8

6 9 4 4 4 10 3 10 5 10 4 4 4

1 1 1 1 1 1 1 1

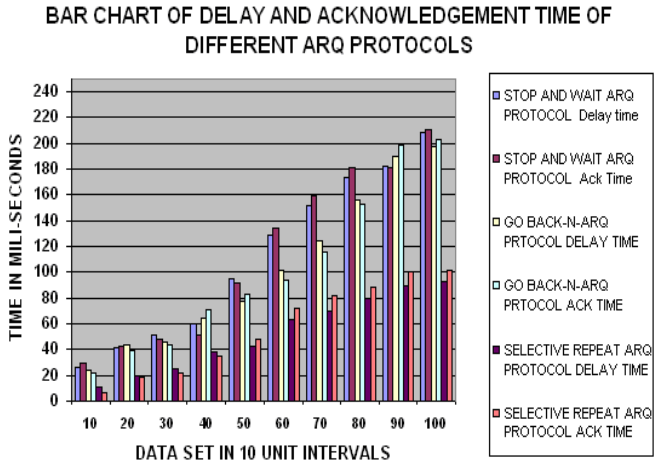
Each row describes the hits of a single user. For example, the first user hits "frontpage" twice, and the second user hits "news" once. Similarly third user hits “Tech” page once, “News” twice, then “Local” page once and then again “News” page twice. Finally, it accesses “Tech” page twice. The operation can be represented as:

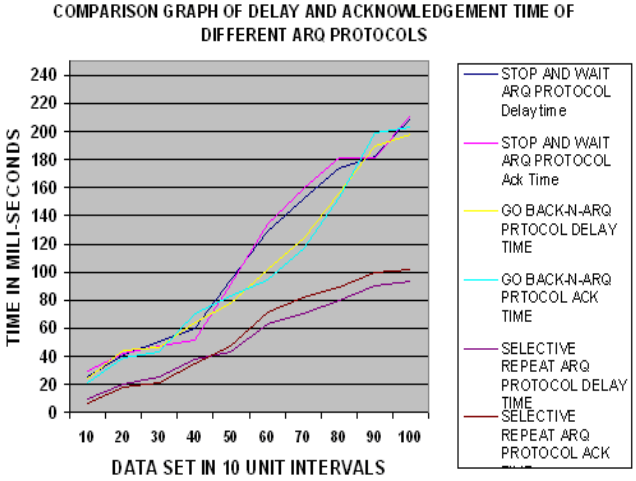



*B. Experimental Output*

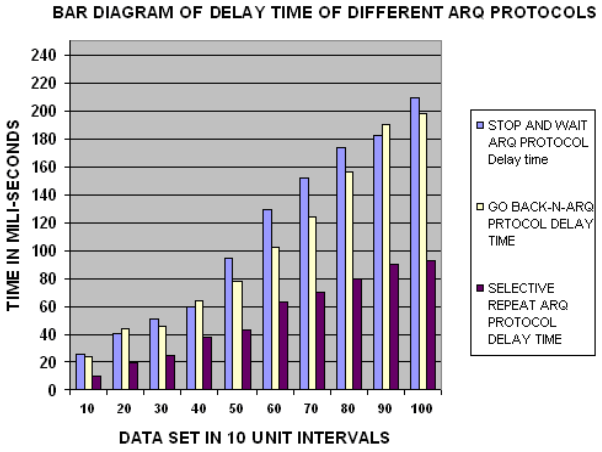
The algorithm is applied using C++ and the delay and acknowledgement time is calculated by the result of the clustered data sets. The experimental result is given int the table 8.1.

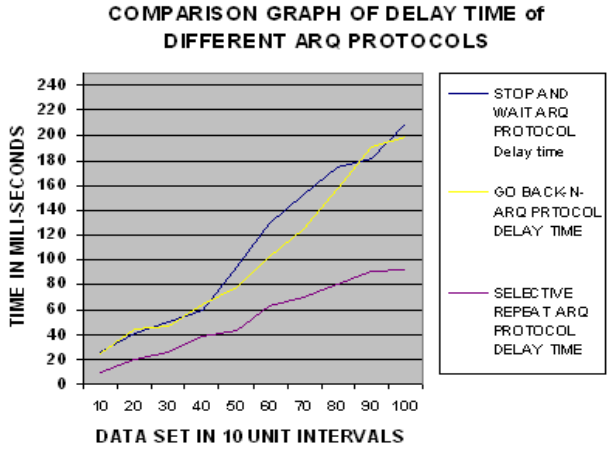
The corresponding graph for acknowledgment time and delay time can be represented as:



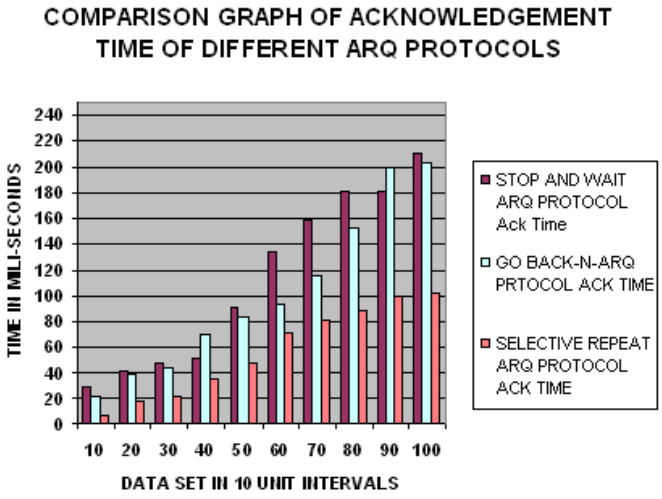



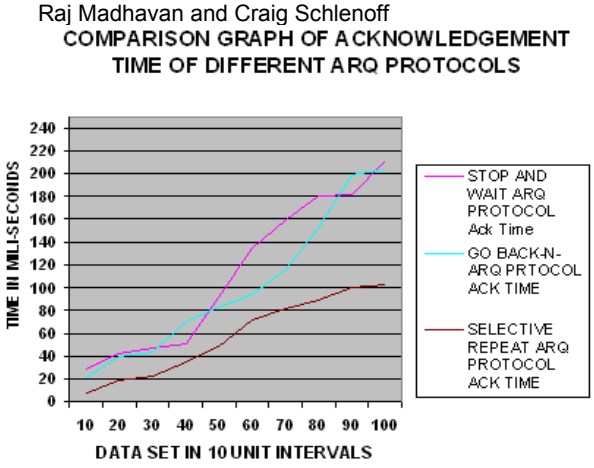

The bar chart and graph for the delay time of different ARQ protocols are:




The bar chart and graph for the acknowledgement time of different ARQ protocols are:






The above data comparison provides a concept that for large dataset, Selective Repeat ARQ protocol provides less complexity and accuracy for data transmission. Stop and Wait ARQ provide less efficiency and takes more time for data transmission as compared to Selective Repeat ARQ protocol and less time as compared to Go- Back-N ARQ protocol.

## LIMITATIONS

The msnbc data set is static in nature and the algorithm is applied for clustering of the static data set. The present algorithm does not deals with a dynamic implementaion. The dynamic implementaion can also be implemented simlarly. The clustering of data set is considered according to the page number grouping. So if the data set is dynamic in nature then the data can be clustered accordingly.

## CONCLUSIONS

Data warehouses provide a great deal of opportunities for performing data mining tasks such as classification and clustering. Typically, updates are collected and applied to the data warehouse periodically. In this paper, we presented the new approach incremental clustering using genetic algorithm (ICGA) with the implementation of Selective Repeat ARQ protocol for mining in a data warehousing environment. During network congestion it increases the maximum chance of the receiving of the mined information. ICGA requires distance function and, therefore, it is applicable to any database containing data from a metric space.

Table 8.1. (Experimental Result of Delay time and Acknowledgement time of diff ARQ protocols for msnbc dataset )

| SIZE OF DATASET | STOP AND WAIT ARQ PROTOCOL | | GO BACK-N-ARQ PROTOCOL | | SELECTIVE REPEAT ARQ PROTOCOL | |
|---|---|---|---|---|---|---|
| | Delay time (in milisec) | Ack Time (in milisec) | Delay time (in milisec) | Ack Time (in milisec) | Delay time (in milisec) | Ack Time (in milisec) |
| 10 | 26 | 29.257019 | 24 | 21.63385 | 10 | 7.069092 |
| 20 | 41 | 42.106812 | 44 | 39.294312 | 20 | 18.628284 |
| 30 | 51 | 47.553101 | 46 | 43.735596 | 25 | 21.92218 |
| 40 | 60 | 51.188599 | 64 | 70.79541 | 38 | 35.231506 |
| 50 | 95 | 91.18634 | 78 | 83.155334 | 43 | 47.794678 |
| 60 | 129 | 134.469849 | 102 | 94.004639 | 63 | 71.675659 |
| 70 | 152 | 159.183289 | 124 | 116.169678 | 70 | 81.445923 |
| 80 | 174 | 180.92041 | 156 | 152.485718 | 80 | 88.74231 |
| 90 | 182 | 181.551086 | 190 | 198.965149 | 90 | 100.041321 |
| 100 | 209 | 211.034729 | 198 | 203.33197 | 93 | 102.041321 |

## REFERENCES

[1] S. Mittra. Iolus: a framework for scalable secure mul- ticasting. In *Proceedings of the Conference on Appli- cations, Technologies, Architectures, and Protocols for Computer Communication*, pages 277-288. ACM SIG-COMM, 1997.

[2] P. C. Sethi, C. Dash: *High Impact Event Processing using Incremental Clustering in Unsupervised Feature Space through Genetic algorithm by Selective Repeat ARQ protocol*: ICCCT- 2nd IEEE Conference – 2011, pp. 310-315..

[3] R.C. Dubes and A.K. Jain. *Clustering methodologies in exploratory data analysis*, Adv. Compute., 19, pp.113–228, 1980

[4] Atul Kamble, *Incremental Clustering in Data Mining*

*using Genetic Algorithm*, International Journal of Computer Theory and Engineering, Vol. 2, No. 3, June, 2010. 1793-8201

[5] UPnP device architecture version 1.0.1. UPnP Forum, Dec. 2003

[6] Data Communications and Networking, 3rd Edition. by Behrouz A. Forouzan, McGraw-Hill Companies, Inc., 2007G. page 278-280.

[7] Jian Ren, Member, IEEE, and Lein Harn: Generalized Ring Signatures, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 5, NO. 3, JULY-SEPTEMBER 2008

[8] S. Lawrence. UPnP basic device definition version 1.0. UPnP Forum, Dec. 2002.

[9] R.L. Rivest, A. Shamir, and Y. Tauman, "How to Leak a Secret", Advances in Cryptology—ASIACRYPT, 2001