CHAPTER 3

OPTIMIZATION OF BATCH REKEYING INTERVAL FOR NEXT GENERATION NETWORK

3.1 CONTENT MANAGEMENT NETWORK OF IPTV

Next generation network is a packet based network to provide telecommunication services and QoS enabled transport technologies. The most interesting feature of NGN is that the service related functionalities are completely independent from the underlying architecture. IPTV will be a killer application for the next generation networks. IPTV supports voice, data and video integrated into a single stream and calls as triple pay package.

IPTV is a convergence of digital television and Internet. In IPTV the digital video services are provided by IP over a network structure. IPTV affords various unique services like broadcast live TV, Video on Demand (VoD), peer-to peer (file sharing) services, Network Digital Video Recording (NDVR) and customized programs and advertisements. With the help of push metaphor technique, all the contents are pushed to the subscriber’s instrument in traditional TV services, which resembles simplex communication (Xiaojun Hei et al 2008). However, the infrastructure behind IPTV is based on the personal choices, combining push and pull, depending on subscriber’s needs and interests through IP network. Because of the enhanced services, IPTV employs two-way interactive communication between the service provider and the subscriber. Example of two-way communication is streaming control
functions such as pause, forward, rewind, and so on, which lack in traditional cable television services.

The technologies behind IPTV are the mixing of digital media and IP protocols. Original analog signals are digitized into strings of 0s and 1s that are then compressed to become manageable for transmission. In most IPTV systems, the primary underlying protocols to transmit data over a managed network is IGMP and the Realtime Transport Protocol (RTP) or Real Time Streaming Protocol (RTSP). The functional architecture of IPTV broadcasting system is shown in Figure 3.1. The application server for an IPTV is service provider who offers triple pay package that is voice data and video into a single bundle (Fengdan Wan 2008). The set top box is the digital component present in the server side which has special functions like video on demand, content decryption subsystem, content decoder subsystem and broadcast application that initiates the push message service from the subscriber’s TV (Fu-Kuan et al 1999). The most important functionalities are abstracted in the middleware. The middleware architecture consists of the following subsystems

1. Prerogative system
2. Head end control
3. Content delivery platform
4. Electronic subscriber program
5. Multicast Session manager
6. Asset manager

The prerogative system authorizes the subscriber’s viewing list and coordinates with the multicast session manager to ensure the paid subscribers are able to watch the programs without any interruption. The electronic
subscriber program authenticates the subscriber and creates a multicast session based on the subscription and the multicast session initiated will be maintained by the multicast session manager (DongHyun Je et al 2013). The asset manager maintains various value added special services by the provider.

**Figure 3.1 Functional architecture of IPTV systems**

In common IPTV services, multiple video channels are grouped together in to bundles. The video channels are generated by one or more
Video Servers (VS) and broadcasted to group of authorized subscribers. (Pinto & Ricardo 2009). The broadcast server performs the major security related functions. The detailed architecture of the broadcast server called as head end of IPTV is shown in Figure 3.2.

![Figure 3.2 Internal structure of broadcast server of IPTV](image)

The requirements identified for this broadcast system are

1. Individual user access control
2. Support for legacy end-systems
3. Transparent operation over existing networks and network equipments
4. Low network resources consumption
5. Support for rapid switching between video channels

6. Scalability

The foremost requirement in maintaining the security of the IPTV broadcast system is the confidentiality of the cryptographic materials used by the service provider (Yu-Lun Huang et al 2004). The end device at the subscriber’s side should use less computational and communication power. In order to satisfy this requirement, the algorithm used for maintaining group secrecy should involve minimum computation and communication overhead, and should be able to accommodate high number of subscribers. The major focus of the research is maintaining secure distribution of multimedia contents to its subscribers by allowing zero delay for channel zapping.

Secure IP multicast assures transmission of IP packets securely to group of receivers. It can be an ideal solution for IPTV broadcasting. However, secure IP multicast has the following problems to be addressed when used in IPTV scenarios

1. Large number of subscribers with differentiated access rights.

2. Variety of subscribers’ viewing preferences to be maintained by the server.

3. Subscribers are to be allowed rapid channel zapping with zero delay.

Scalable secure group communication is an active research area and has much standard mechanism in existence. These techniques aim at securing the data sent to the multicast group with equal access rights but not for bundles of video channels. For instance, subscriber A may be interested in generic and sports bundle while subscriber B wants to view only generic bundle. The group privacy preserving mechanism should be able to employ
different level of encryption with different set of keys. As an access control mechanism, group key management scheme can be employed (Inshil Doh et al 2012). IPTV systems can broadcast the multimedia contents securely by grouping subscribers of specific bundle and assigning a common shared secret known as session key, this can ensure the transmission of contents securely only to the paid subscribers (Mihaljevic 2004).

3.2 NEED FOR ORI SCHEME

To ensure group secrecy, the cryptographic keying material is to be changed for change in group membership, called as group rekeying. Group rekeying raises scalability problems in large dynamic groups. IPTV has huge subscribers in its list and also the user’s community of IPTV may join and leave at any time and also may change the channels most frequently. Numerous architectures, algorithms, and protocols have been proposed in the literature to cope with this scalability problem. Related work on optimizing rekeying performance mostly concentrates on minimizing the number of required cryptographic operations and thus the length of the rekeying message.

The basic requirement in the secure multicasting of IPTV is that no subscribers outside the target set can decrypt the message. An accepted strategy to reduce rekeying costs utilizes batch processing of rekeying requests, which are summed up during a rekeying interval (Sangho Lee & Jong Kim 2011). However the length of rekeying interval is not specified clearly. It mainly depends on the type of application. The main research focus in batch rekeying is identification of rekeying interval (Cho et al 2008). Short rekeying interval does not provide much benefit resembles the performance of individual rekeying (Lee et al 2007). Also long rekeying interval instigates a delay in joining or leaving and violation of access policy (Wee hock Desmond Ng & Zhilli Sun 2006). Identification of rekeying interval has not been
addressed satisfactorily and the relationship between the batch rekeying interval and the environmental conditions (i.e. the arrival rate of join or leave requests) remains to be investigated. The proposed optimal rekeying interval (ORI) scheme identifies the optimal interval size for rekeying in IPTV applications mathematically.

3.3 DESCRIPTION OF THE MODEL

The subscribers of the IPTV broadcasting network are divided into various groups. The subscribers of the channel randomly join and leave which initiates a rekeying request to the broadcast server (BS). Since rekeying is a pure random process, it is modeled as Poisson process and the rekey server accumulates the entire rekeying request for a specific time period. The subscriber’s arrival or departure is deemed as rekeying request

The rekeying requests are accumulated and processed as a batch of size k. These characteristics of the rekeying server paves path for applying queuing theory to predict the optimal rekeying interval.

3.3.1 Basics of Queuing Theory

To describe a queueing system the input and output process should be specified clearly (Giovanni Giambene 2005). The input process to the queueing system is usually called as arrival process, which could be single or multiple at a given point of time. Multiple arrivals are called as bulk arrivals. Since the subscribers present in the systems do not affect the arrival process, a probability distribution function can describe rate of arrival at anytime. The output process of a queueing system is named as service process. Similar to arrival, service is also portrayed by the probability function (Ho Woo Lee & Santosh Kumar 2007). Figure 3.3 shows the basic elements of a queueing system.
A special notation, called Kendall's notation, is used to describe a queueing system. The notation has the form

\[ A/B/C/K \text{/Queue discipline/system capacity/population} \]  

Where

- \( A \) describes the interarrival time distribution
- \( B \) denotes the service time distribution
- \( C \) specifies the number of servers that can range from 1 to \( n \)
- \( K \) the size of the system capacity (including the servers).

Based on the nature of arrival and service process \( A \) and \( B \) can be

1. Memory less and interarrival times are exponentially distributed (\( M \) )
2. Deterministic distribution (\( D \)).
3. Erlang distribution with shape parameter \( k \) (\( E_k \)).
4. General Distribution (\( GI \))

The next characteristic of a queue is the number of parallel servers. Queue discipline can be First Come First Served (FCFS), Last Come First Served (LCFS), Service in Random Order (SIRO), General queue Discipline (\( GD \)).
System capacity denotes the maximum allowable number of customers in the system including customers who are waiting and customers who are in service (Avishai Mandelbaum & Petar Momcilovic 2012). The last parameter gives the size of the population from which customers are drawn. Unless the number of potential customers is of the same order of magnitude as the number of servers, the population size is considered as infinite.

The time between two successive arrivals are called as interarrival time (Allen 1978). Interarrival time is usually considered as exponential distribution with the rate $\lambda$. The main advantage of exponential distribution is that the property of arrival rate at time $t-1$ doesn’t affect arrival rate of $t$, which is popularly known as memory less property.

Based on the number of servers used, queuing model can be classified as

1. Single server queuing model ($M/M/1/FCFS$, $M/Er/1/FCFS$, $G/G/1/FCFS$)
2. Multi server queuing model ($M/M/S/FCFS$)

3.3.2 System Setup

The subscriber’s arrival and departure is a pure random process which is coined as rekeying request. Since the subscriber may join or leave at any point of time and also cannot be predicted by the system, the rekeying request is modeled as Poisson process with the parameter $\lambda$. The rekeying request is processed by the rekey server with the service rate $\mu$. Since the current processing of rekey request does not depend on the previous service, the rekey server is assumed to be exponentially distributed. The threshold to start the rekey server is $k$ called as batch size.
The architecture of the rekey server present in the broadcast server is shown in the Figure 3.4.

**Figure 3.4 Architecture of rekey server**

3.3.3 Markov Chain

A stochastic process \( \{X(n), n \in \mathbb{N}\} \) is a sequence of stages in which the outcome at any stage can be represented by a probability function (Poul E Heegaard & Kishor S Trivedi 2009). A Markov process or a Markov chain is a stochastic process with the following properties:
• The number of possible outcomes or states is finite.

• The present outcome at any stage depends only on the outcome of the immediate previous stage.

• The probabilities are constant over time.

\[ P(X_n = i_n \mid X_0 = i_0, \ldots, X_{n-1} = i_{n-1}) = P(X_n = i_n \mid X_{n-1} = i_{n-1}) \quad (3.2) \]

In Markov property, the future predictions depend only on the current state of the system (Bini Dario et al 2005). In continuous time Markov Process, the time is perturbed by exponentially distributed holding times in each state while the succession of states visited still follows a discrete time Markov chain.

3.3.4 Birth Death Process

The birth-death process is a special case of continuous time Markov process, where the state symbolizes the current amount of inhabitants and the transitions are limited to birth and death (DU Keliang et al 2004). When a birth occurs, the process goes from state \( i \) to state \( i + 1 \). Similarly when a death occurs, the process goes from state \( i \) to state \( i - 1 \). It is assumed that the birth and death events are independent of each other (By Iain Macphee et al 2007).

The subscriber’s statistical deed can be portrayed by an embedded Markov chain. Usage of Markov chain has basic advantage of predicting member’s rekeying request based on its previous states. In this rekeying model whenever the batch rekeying algorithm is executed and the new session key is communicated and it is equated to death process. Similarly the collection of rekeying request accumulated in one batch initiates the birth
process. The markov chain model of rekeying request with the rate $\lambda$ and the service with the rate $\mu$ is illustrated in Figure 3.5.

![Markov chain model of request and service](image)

**Figure 3.5 Markov chain model of request and service**

### 3.3.5 Performance Measures

In general, queuing model may represent static or dynamic system, the values of the performance measures may vary with time. The system is said to be in steady state when all transient behavior has ended, the system has settled down, and the values of the performance measures are independent of time (Gunter Bloch et al 1998). The system is then said to be in statistical equilibrium, i.e., the rate at which the rekeying request enters into the system is equal to the rate at which rekeying request are solved by generating and communicating the new session key. Such a system is called as stable system.

The relevant performance measures used in the analysis of identification of rekeying algorithm using queuing model for broadcasting the contents of IPTV are

1. The distribution of the waiting time and the sojourn time of a batch of rekeying request. The sojourn time is the waiting time plus the service time.
2. The distribution of the number of total rekeying request in the system (including or excluding the one or those in service).

3. The distribution of the amount of work in the system. That is the sum of service times of the waiting rekeying request to be solved and the residual service time of the rekeying request in service.

4. The distribution of the busy period of the rekey server. This is a period of time during which the rekey server generates and communicates the new session key to existing subscribers.

The various mean performance representation of queuing system with respect to group key management is shown in Table 3.1.

**Table 3.1 Performance measures used in queuing model**

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>average number of subscribers present in the queuing system</td>
</tr>
<tr>
<td>L_q</td>
<td>average number of rekeying request waiting in line to be served</td>
</tr>
<tr>
<td>L_s</td>
<td>average number of subscriber in currently being processed</td>
</tr>
<tr>
<td>W</td>
<td>average time a subscriber spends in the system</td>
</tr>
<tr>
<td>W_q</td>
<td>average time rekeying request spends in line</td>
</tr>
<tr>
<td>W_s</td>
<td>average time a subscriber spends in service</td>
</tr>
<tr>
<td>( \lambda )</td>
<td>Arrival rate</td>
</tr>
<tr>
<td>( \mu )</td>
<td>Service rate</td>
</tr>
<tr>
<td>( \rho )</td>
<td>Traffic intensity</td>
</tr>
<tr>
<td>( T_s )</td>
<td>Mean service time for each arrival</td>
</tr>
<tr>
<td>( N )</td>
<td>No of servers</td>
</tr>
</tbody>
</table>
3.3.6 Little’s Law

Little’s law gives a very imperative relation between the average number of rekeying request in the system, the mean sojourn time and the average number of subscribers entering and exiting from the system per unit time (Fralix Brian & Riano 2010). The Little’s law states that the average number of rekeying request in a system over a predefined interval is equal to the average arrival rate, multiplied by the average time in the system. All the averages are considered as steady state averages. For any queuing system in which a steady-state distribution exists, the following relations hold

\[ L = \lambda \cdot W \]  \hspace{1cm} (3.3)

\[ L_q = \lambda \cdot W_q \]  \hspace{1cm} (3.4)

\[ L_s = \lambda \cdot W_s \]  \hspace{1cm} (3.5)

In the steady state conditions L is calculated as

\[ L = \frac{\rho}{(1-\rho)} \]  \hspace{1cm} (3.6)

Where \( \rho = \frac{\lambda}{\mu} \)

From Little’s Law

\[ W = \frac{L}{\lambda} \]  \hspace{1cm} (3.7)

\[ W = \frac{\rho}{\lambda \cdot (1-\rho)} \]  \hspace{1cm} (3.8)

By simplifying

\[ W = \frac{1}{\mu - \lambda} \]  \hspace{1cm} (3.9)
Little’s law plays a major role in checking the consistency of the data and if two parameters are known in the system, remaining parameters can be calculated easily and also accurately.

3.4 HIGH LEVEL ARCHITECTURE OF SESSION MANAGEMENT IN IPTV

The foremost functionality of an IPTV is to deliver the scheduled multimedia contents to its paid subscribers. It employs secure IP multicast to disseminate the contents.

In a managed network, it is important to ensure that:

- A subscriber is allowed to join a multicast group only if there is enough bandwidth with the right service priority to handle the requested stream within the access network.

- The reserved subscriber resources are released when the channel zapping is activated or the system is powered off.

- During channel zapping, interaction or handshake between network entities related to bandwidth, service priority or admission control are to be optimized. This saves precious time and contributes to a faster channel zapping speed.

Based on the points mentioned above the session set up procedure in IPTV is shown in Figure 3.6.
3.4.1 Multicast Session Set up Procedure

The service provider of the IPTV network establishes the multicast session in order to distribute the contents to the paid subscribers. The step by step group establishment session is discussed below.

1. The service provider initiates the session setup request to the Authentication and Session Management (ASM) entity including a group multicast IP address and media offer for the scheduled content service.

2. The ASM reserves transport resources according to the type of multicast routing adopted.

3. The response for the reservation request is returned once the routing strategy is finalized and bandwidth is reserved.

4. The ASM forwards the request to the IPTV Control, which verifies that the subscriber is paid to the service or not.

5. The IPTV control replies to the ASM with the bandwidth required for the specific scheduled content channels and the newly computed session key to the service provider in order to encrypt the multimedia contents and to ensure only the paid subscribers are receiving the program.

6. In case the media offer has changed or new parameters are received, the ASM requests admission control for the confirmation phase.

7. If step 6 is used, the response for the admission control request is returned.
8. Finally, the service session setup response is forwarded to the service provider.

9. The service provider encrypts the contents via the commonly generated session key and sends the request to the Transport Processing Functions to view the channel.

10. An interaction between the Transport Processing Functions and Resource Admission Control entities occurs in order to guarantee the necessary bandwidth for the channel.

11. The needed bandwidth is reserved.

12. The encrypted multimedia contents are broadcasted to the subscribers.

13. Whenever subscriber zaps between channels the service provider sends a request to the Transport Processing Functions to change the channel.

14. The operation is identical to that of step 10.

15. Step 15 repeats the functionality of step 11.

16. Following that the encrypted multimedia media contents are forwarded to the service provider.
3.5 CALCULATION OF ORI

The subscribers of the channel randomly join and leave which initiates the rekeying request by the resource and admission control server to the rekey server. When a subscriber leaves from the channel, session key is to be updated to prevent the future access by the same subscriber. When
numerous leaves or join event occur for program $x$ whose value is $x$. Due to join or leave the subscriber’s count for program $x$ changes from $k$ to $j$ $(k < j)$ at time $T_0$ to $T_1$. The steps involved in establishing a multicast session is portrayed in Figure 3.6. Due to delay a loss occurs during the rekeying interval. The expected loss is denoted as $£$.

The expected loss at the time $T_{next}$ is:

$$£ = \sum x_i (T_{next} - T_a) \quad (3.10)$$

Where $a$ ranges from $k$ to $j$.

The main aim of finding an optimal rekeying interval for IPTV broadcasting is to improve the performance by minimizing the loss from the service provider end.

The assumptions made in the construction of rekeying model are:

- Rekeying request to the system is assumed to be Poisson process with rate $\lambda$. $k$ is the maximum number of the subscribers in the batch.

- After buffering the rekeying requests to the queue, service begins by the time $t$ with the density function $d(t) = \alpha e^{-\alpha t}$, $t \geq 0$, $\alpha > 0$ where $\alpha$ is the rate of time $T$.

- The rekeying server works on First-Come, First Served (FCFS) discipline. The service is non-preemptive. The service times are assumed to be distributed according to an exponential distribution with density function $s(t) = \mu e^{-\mu t}$, $t \geq 0$, $\mu > 0$ where $\mu$ is the service rate.
If the rekeying request and service are independent of time or if the behavior of the group is independent of time, the group is said to be in steady state. Otherwise it is said to be in transient state. Let $P(n)$ be the probability that there are $n$ rekeying requests at time $t$ and one rekey server. The probability of the subscriber group having $n$ rekeying requests in $t+\Delta t$ time is from one of four mutual exclusive ways as follows:

1. Presence of $n$ rekeying requests at time $t$ and no rekeying request in $t+\Delta t$ time

2. Presence of $n-1$ rekeying requests at time $t$ and one rekeying request in $t+\Delta t$ time

3. Presence of $n+1$ rekeying requests at time $t$ and no rekeying request and one rekey servicing in $t+\Delta t$ time

4. Presence of $n$ rekeying requests at time $t$ and one rekeying request and one rekey servicing in $t+\Delta t$ time:

$$P_n(t+\Delta t) = P_n(t) (1 - \lambda \Delta t) (1 - \mu_n \Delta t) + P_{n-1}(t) \lambda_{n-1} \Delta t (1 - \mu_{n-1} \Delta t) +$$

$$P_{n+1}(t) (1 - \lambda_n \Delta t) \mu_{n+1} \Delta t + P_n(t) \lambda_n \Delta t \mu_n \Delta t$$

By applying the limits to Equation 3.11 with respect to the system state, the probability for different state are as follows

When no rekeying request $n = 0$

$$\lambda P_0 = \mu P_1$$

(3.12)

When only one rekey request $n = 1$

$$\lambda P_1 + \mu P_1 = \lambda P_0 + \mu P_2 \Leftrightarrow (\lambda + \mu) P_1 = \lambda P_0 + \mu P_1$$

(3.13)
When only two rekeying request $n = 2$

$$\lambda P_2 + \mu P_2 = \lambda P_1 + \mu P_3 \quad \Leftrightarrow (\lambda + \mu)P_2 = \lambda P_0 + \mu P_3$$  \hfill (3.14)

When three rekeying request $n = 3$

$$\lambda P_3 + \mu P_3 = \lambda P_2 + \mu P_4 \quad \Leftrightarrow (\lambda + \mu)P_3 = \lambda P_2 + \mu P_4$$  \hfill (3.16)

When $k$ rekeying request arrives $n = k$

$$\mu P_k = \lambda P_{k-1} \quad \Leftrightarrow \mu P_k = \lambda P_{k-1}$$  \hfill (3.17)

Solving these simultaneous equations the values of $P_0, P_1..P_k$ can be obtained:

$$P_1 = (\frac{\lambda}{\mu})P_0$$
$$P_2 = (\frac{\lambda}{\mu})^2 P_0$$  \hfill (3.18)

So the probability that the group has $n$ rekeying request is:

$$P_n = (\frac{\lambda}{\mu})^n P_0 \quad 0 \leq n \leq k - 1$$  \hfill (3.19)

The initial probability can be calculated as:

$$\sum_{n=0}^{k} P_n = \sum_{n=0}^{k} (\frac{\lambda}{\mu})^n$$
$$P_0 \sum (\frac{\lambda}{\mu})^n = 1$$

So $P_0 = 1 - (\frac{\lambda}{\mu}) - (\frac{\lambda}{\mu})^{k+1}$

$$P_0 = \frac{1}{\lambda + 1}$$

The probability that there exists $k$ rekeying request is:

$$P_n = (\frac{\lambda}{\mu})^n 1 - (\frac{\lambda}{\mu})^{k+1}$$
$$P_0 = \frac{1}{\lambda + 1}$$  \hfill (3.21)
The parameters which decides the optimal batch interval are:

- Expected number of rekeying request in a batch of size k. \( (N_q) \)
- Expected number of rekeying request in the group. \( (N_s) \)
- Expected waiting time for rekeying request to process in the group \( (W_s) \)
- Expected waiting time for rekeying request to process in the batch \( (W_q) \)

By applying little’s law the various control parameters can be calculated:

\[
N_s = \left( \frac{\lambda}{\lambda - \mu} \right) - (k+1) \left( \frac{\lambda}{\mu} \right)^{k+1} / 1 - \left( \frac{\lambda}{\mu} \right)^{k+1} \quad (3.22)
\]

\[
N_q = N_s - \frac{\lambda}{\mu} \text{ where } \lambda' = \mu(1-P_0) \quad (3.23)
\]

\[
W_s = 1 / \lambda' N_s \quad (3.24)
\]

\[
W_q = 1 / \lambda' N_q \quad (3.25)
\]

The optimal rekeying interval \( (\text{ORI}_{\text{opt}}) \) is defined as the difference between start of current batch rekeying process to the next batch rekeying process. This can be calculated by adding the expected waiting time and the rekeying time by the rekey server denoted as \( T \).

\[
\text{ORI}_{\text{opt}} = W_s + T \quad (3.26)
\]

\( \text{ORI}_{\text{opt}} \) can be calculated by varying the arrival rate and the service time taken by the rekey server. The rekeying time mainly depends on the kind of rekeying algorithm used by the rekey server.
3.6 PERFORMANCE ANALYSIS

In general, rekeying algorithms are evaluated based on three parameters:

- Communication cost
- Computation cost
- Storage cost

By considering these three parameters, the suitable rekeying algorithm can be applied. In batch rekeying algorithm the performance and security mainly depends on the batch size and rekeying interval.

In dynamic environment, members join or leave randomly, generates lot of rekeying messages. Number of rekeying messages reduces significantly for batch rekeying system with the compromise to strict forward and backward secrecy To optimize the rekeying interval, a queuing theory based model is used.

The proposed model is simulated using Java based simulator called Java Modeling Tools, which is mainly to simulate the various queuing models. The suitable rekeying model is identified by changing various system parameters like rekeying request rate, rekeying service rate and batch size.

3.6.1 Arrival Rate Analysis

The arrival rate of the subscriber is one among the key factor to determine the rekey interval. Since the arrival of a subscriber follows a Poisson distribution, the present arrival pattern can be predicted from the previous arrival rate. The Figure 3.7 shows the analysis of arrival when the past arrival rate is three subscribers / min. The Figure 3.8 shows the analysis of arrival when the past arrival rate is five subscribers / min. The horizontal
axis represents the number of future arrivals and vertical axis symbolizes the probability for the future arrivals.

**Figure 3.7** Expected Arrival rate for $\lambda=3$

**Figure 3.8** Expected Arrival rate $\lambda=5$
3.6.2 Service Rate Analysis

The next influential parameter in the model is service rate. The performance of rekeying server for two different service rate is illustrated in the Figure 3.9. If the expected arrival pattern is high, then the rekey server can select the algorithm, which yields the better service rate. The service rate and arrival rate are the two influential parameters in determining the rekey interval and the rekey algorithm. With the help of the probability theory, the expected values can be calculated easily.

![Figure 3.9 Expected service rate by the rekey server](image)

3.6.3 Rekeying Interval Analysis

With the help of arrival rate and service rate predictions, the rekeying interval can be fine-tuned.
Figure 3.10 shows the average waiting time for each batch of size 5 with the arrival rate of 105 sub/min and a service rate of 25 sub/min. The coefficient of variation of Interarrival time is assumed to be 1. To find an average value the model is simulated for a run length of 10.

**Figure 3.10 Batch wise waiting time analysis**

The group controller performs the rekeying operation with rekeying service rate ($\mu$) as 5 sec/req. The rekeying request follows Poisson distribution. Figure 3.11 shows the analysis of the rekeying interval for the varying batch size. Three different batch size are considered Optimal rekeying interval when $k = 4, 6, 8$. 
Figure 3.11 Optimal rekeying interval when $k = 4, 6, 8$

Figure 3.12 shows the relationship between expected rekeying request and the average waiting time when the value of batch size is five. The performance of the group mainly depends on the rekeying algorithm used. If the rekeying algorithm renders the service at the rate of $\mu = 2, 5, 9$, the average waiting time for the request to be served is shown in the graph.

Figure 3.12 Average waiting time when $\mu = 2, 5, 9$
From all these analyses it is clear that, the various parameters of the queuing based rekeying model can be modified to get an optimum performance. The rekeying interval becomes optimal depends on the program and subscriber’s choice. Certain programs like one day cricket match, the subscriber’s join or leave may occur during the beginning or end of the session and for a short span of time the group becomes static.

While optimizing the rekeying interval the batch size is assumed as constant for each run. The next chapter analyses the different influential parameters of batch size and identifies the suitable batch size.