Chapter 5

Transition Probability Prediction in Markov Reliability Modeling

5.1 Introduction

In the previous chapter, we described a technique to predict software system reliability based on UML modeling, by extending UML specification to support reliability analysis and design. The result of our approach is a Markov reliability model, from which we gave an estimate of the reliability of the system, using Cheung's approach [18]. Many researchers have proposed approaches based on Markov chain [14, 17-21]. We tried to apply these approaches to predict the software reliability of safety critical systems of NPP and came out to a conclusion that these approaches are not practical enough to be used by the practitioners in the industries. The reason is that the software reliability is based on the transition probabilities in between the states of Markov chain and all the authors have either assumed them or taken a coarse figure using analytical approach. Some authors computed them using operational profile but that can be possible after the deployment of the software system and hence it is not an early prediction. The detail of author's perception regarding transition probabilities is given in section 5.2. The contribution of this chapter is to propose a framework to address the problem of finding accurate transition probabilities of the MC for accurate reliability early prediction. This helps to

addresses some of the modeling and analysis limitations that are given in section 3.3.

5.2 Related work

Reliability prediction approach for component based software architecture has been proposed by Reussner [74]. But this approach is like a black box approach and based on Markov chain and UML and can predict software reliability. Also, the transition probabilities between the states of the Markov chain have been assumed. Gokahle and Trivedi [17] also propose methodology for software reliability prediction based on Markov Chains by assuming the transition probabilities in between the states of the Markov chain. Another approach for reliability prediction has been proposed by Cheung [18], which is based on Hidden Markov Chain. This approach uses five sources from system experts. This paper also lacks the method to compute the transition probabilities between the states of the Markov chain. They state that the transition probabilities can be obtained by assembling and deploying the components and executing the expected usage profile against them. However, for this software practitioners need to set up the whole system during architecture design, which is often neither desired nor possible. Recent approaches by Sharma et al. [75] and Wang et al. [76] extend Cheung's work to support different architectural styles and combined performance and reliability analysis. However, they rely on testing data or the software architecture's intuition to determine the transition probabilities.

Sato and Trivedi [77] combine a system model and resource availability model but assume fixed transition probabilities among services.

F. Brosch et al. [78] devised an approach based on Palladio Component Model which automatically gets transformed into a formal MC. They state that they compute P(Success|sj), the probability of success on condition that the system is in state sj, without giving any computational evidence. Moreover their basis to estimate the transition probabilities is MTTF and MTTR, which cannot be determined during architectural or design phase.

Gokhale et al. [79] again tried to address this issue up to some extent using Bayesian

approach but they define a posterior distribution of the random variable to find the transition probabilities based on any prior knowledge, which is also an analytical approach.

Goseva-Popstojanova et al. [80] proposed the method of moments to calculate the sensitivity of a system's reliability to component reliabilities and transition probabilities analytically. Indika et al. [81] proposed a method to evaluate reliability based on the architecture. They try to compute the transition probabilities in the MC from the expected number of visits to a communication link. But the uncertainty that is associated with this approach is to quantize the parameters which are required to compute the estimate of the number of visits.

Some approaches [82] also quote that usage profile should be used to find the transition probabilities of the system but this is not a generic solution as a same software may have different usage profile in it is installed at different locations.

All of the above methods have taken an analytical approach to quantify the sensitivity, where the applicability is limited to analytically solvable models. However, these analytical sensitivity analysis methods are hard to generalize.

5.3 The proposed method for transition probability prediction

To overcome the issues in the existing approaches, described in section 5.2, we propose a framework to estimate the transition probability for its suitability to give more accurate reliability prediction. We use it for the various systems of NPP. We choose stochastic process because of the many abstractions like internal architecture of the operating system, hardware, etc on which the software performance depends. Our framework contains five phases as shown in figure 5.1.

Each phase is described as follows.

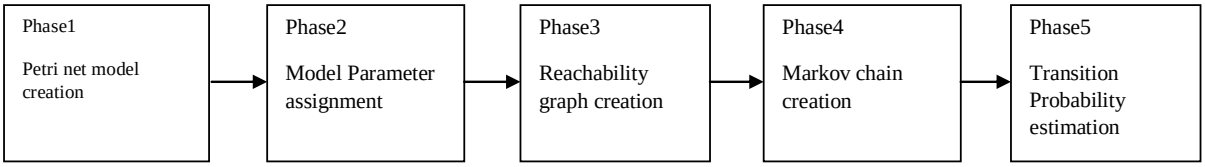

Figure 5.1: Transition Probability prediction framework.

5.3.1 Phase1: Petri net model creation

In this phase, the system is modeled using Petri net. The use of Petri net avoids two of the basic drawbacks of MC analysis. First, the model does not grow in size as the number of components in the model increase. Because Petri Nets model local states instead of global ones, they do not grow out of control as the model increases in complexity. Second, Markov Chain analysis is typically limited to modeling the probability of changes in the system with the exponential distribution [83]. However, processes pertaining to reliability, availability, maintainability, and safety do not necessarily conform to an exponential distribution. The method of modeling a system through Petri net is available in existing literatures [49, 51, 84-90]. The system designer or architect should identify the places and transitions carefully to include all the success and failure conditions of the system.

5.3.2 Phase2: Model Parameter assignment

In this phase, Time NET [30-31] is used to create SPN. SPN is a form of Petri net where transitions fire after a probabilistic delay determined by a random variable.

We need to assign delay of the transitions in SPN. The long-time behavior of this SPN can be studied by so-called stationary or steady-state evaluation. Time NET can be used for this purpose, which gives the throughput of all the transitions of SPN.

5.3.3 Phase3: Reachability graph creation

We create reachability graph from SPN in this phase. The method to create reachability graph from SPN can be found in several papers [48].Reachability problem is defined as: Given a computational (potentially infinite state) system with a set of allowed rules or

transformations, decide whether a certain state of a system is reachable from a given initial state of the system. In fact, this problem was shown to be EXPSPACE-hard [91] years before it was shown to be decidable at all (May, 1981). Papers continue to be published on how to do it efficiently. While reachability seems to be a good tool to find erroneous states, for practical problems the constructed graph usually has far too many states to calculate. To alleviate this problem, linear temporal logic is usually used in conjunction with the tableau method to prove that such states cannot be reached. LTL uses the semi-decision technique to find if indeed a state can be reached, by finding a set of necessary conditions for the state to be reached then proving that those conditions cannot be satisfied.

5.3.4 Phase4: Markov Chain creation

MC contains behavioral and failure states. The reachability graph, created in the phase 3 can be converted into MC as follows. The MC state space is the reachability set $R(M_0)$, and the transition rate from state M_i to M_j is given by $q_{ij} = \lambda'_i$, the firing rate of transition t_i transforming M_i into $Mj(q_{ij} = \lambda'_{i1} + \lambda'_{i2} + \ldots$, if there are two or more transitions $t_{i1}, t_{i2}, \ldots$ transforming M_i into M_j); $q_{ij} = 0$ if no transitions transforming M_i into $M_j, i \neq j$;and q_{ij} is determined so as to satisfy $\sum_j q_{ij} = 0$. The square matrix $Q = [q_{ij}]$of order $s = |R(M_0)|$ is known as transition rate matrix [48]. As the transition rate matrix contains rates, the rate of departing from one state to arrive at another should be positive, and the rate that the system remains in a state should be negative. The rates for a given state should sum to zero, yielding the diagonal elements to be (equation 5.1),

$$q_{ii} = -\sum_{j \neq i} q_{ij} \tag{5.1}$$

With this notation, and assuming $p_t = Pr(X(t) = j)$, the evolution of a continuous-time Markov process is given by the first order differential equation (equation 5.2)

$$\frac{\partial p_t}{\partial t} = p_t Q \tag{5.2}$$

The probability that no transition happens in some time r is given by equation 5.3,

$$Pr(X(s) = i \ \ \forall s \in (t, t + r)|X(t) = i) = \mathrm{e}^{q_{ii}r} \tag{5.3}$$

That is, the probability distribution of the waiting time until the first transition is an exponential distribution with rate parameter $-q_{ii}$, and continuous Markov process are thus memory less processes. Also a test for the Markov property has been suggested by Anderson and Goodman (1957) as given in equation 5.4,

$$-2\log_n \lambda = 2\sum_{i,j}^{m} n_{ij} \log_e[\frac{p_{ij}}{p_j}] \tag{5.4}$$

where

$$p_j = \frac{\sum_i^m n_{ij}}{\sum_{i,j}^m n_{ij}}$$

where

p_{ij} = probability in cell i, j of the transition probability matrix.

p_j = marginal probabilities of jth column.

n_{ij} = transition frequency total in cell i, j of the original count of observed transitions.

m = total number of states.

5.3.5 Phase5: Transition Probability estimation

For state space ϵ, if all states $x, y \in \epsilon$ communicate, then the Markov chain has a stationary distribution $\pi = \pi_x, x \in \epsilon$ where each π_x is the proportion of time spent in state x after the Markov chain has run for infinite time, and this probability does not depend upon the initial state of the process. Such a Markov chain is called ergodic, for which it is possible to compute steady-state probability distribution π by solving the linear system, given in equation 5.5.

$$\pi Q = 0, \sum_{i=1}^{s} \pi_i = 1 \tag{5.5}$$

The transition probability p_{ij} of the Markov chain, created from SPN can be computed with the help of transition rate matrix Q. Since transition rate q_{ij} represents the transition of a state to another state per unit time and therefore if we take the ratio of transition rate q_{ij} (of going from state i to state j) and the sum of all transition rates except it transits to itself, we will get the transition probability from one state to other (p_{ij}). Clearly if it transits to itself infinitely, it will not be ergodic and in this case p_{ij} will be zero i.e.

$$p_{ij} = \begin{cases} \frac{q_{ij}}{\sum_{k \neq i} q_{ik}} & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

From this transition probability matrix P can be written as:

$$P = I - D_Q^{-1} Q \tag{5.7}$$

where,

$D_Q = diag\{Q\}$ is the diagonal matrix of Q

5.4 A Case Study

We extend the same case study, as we have taken in section 4.2 to illustrate our approach.

5.4.1 ECCS Design Requirements

There can be number of failures in a NPP, which are listed in table 5.1.

For all such events, the ECCS shall be capable of maintaining or re-establishing sufficient cooling of the fuel and fuel channels so as to limit the release of radioactive material from the fuel in the reactor and to maintain fuel channel integrity. For such events, the ECCS shall meet the following requirements [92]:

i The release of radioactive material from the fuel in the reactor shall be limited such

Table 5.1: Failures in NPP.

S. No.	Failures
1	Failure of any feeder pipe in the primary heat transport system
2	Failure of a calandria tube
3	Fuel channel flow blockage
4	Failure of a fuelling machine to replace a closure plug
5	Inadvertent opening of pressure relief or control valves on the primary heat transport system
6	Failure of steam generator tubes
7	Failure of any pipe or header in the primary heat transport system

that the reference dose limits are not exceeded.

ii For events listed in table 5.1, there shall be no failure of fuel in the reactor due to lack of adequate cooling.

iii Fuel and all the fuel channels in a reactor shall be kept in a configuration such that continued removal of the decay heat can be maintained by the ECCS.

iv After adequate cooling of the fuel is re-established by the ECCS, the system shall be capable to supply sufficient cooling flow for as long as it is required to prevent further damage to the fuel.

The ECCS shall be designed such that the fraction of time for which it is not available can be demonstrated to be less than 10^{-3} years per year. The availability of the safety support equipment that is necessary for correct operation of the ECCS shall be commensurate with the availability requirement of the ECCS. Availability calculations to demonstrate that this requirement can be met shall be included or referenced in the safety report of NPP. Such calculations shall be based on direct experience or reasonable extrapolations there from.

5.4.2 Test Facility

All the ECCS equipments shall be monitored or tested at a frequency of 1 month which is adequate to demonstrate compliance with the availability requirements specified in section 5.4.1. A CBS has been designed and developed for this purpose. The functions of the Test Facility are follows:

Data acquisition: It acquires the data from the sensor in the form of voltage or contacts. The data can be analog or digital. Analog data represents the process parameters like tank pressure, tank level, etc and digital data represents the status of valves, pumps, etc.

Testing: Test Facility is used to test the healthiness of various mechanical equipments of the ECCS to meet its availability/reliability.

MIMIC display: It shows the dynamics of the ECCS in the pictorial format, as shown in figure 4.1.

Alarm Generation: It generates the alarm, if any important plant parameter crosses the alarm set point limits. It helps to alert the control room operators to take necessary actions.

Data display: It shows the analog and the digital data in various formats viz tabular trend, graphical trend, bar graph, etc. The provision of showing data in different formats is given for analysis purpose, after any accident or during normal operation of the plant.

Print records: The operator can take printouts of important data which may be useful for some analysis or to take necessary action in the future or just for information to the operators. User administration: It gives the access only to the registered control room operators to avoid unsafe operations on ECCS. It allows the super user to create and delete the account of the operators. The super user can also reset the password of the registered operator.

History: It stores the history of all the analog and digital signals of past 1 year for the purpose of analysis. Report generation: It generates the reports of ECCS equipment testing on demand. These reports provide the evidence to the controlling authority that

Table 5.2: Components of EU.

Module	**Numbers**
Analog Input Module	2
Digital Input Module	5
Relay Output Module	5
GBINC	2
HUB	1
RT-20	1

all the ECCS equipments are tested at the frequency of one month.

5.4.3 Test Facility Architecture

The architecture of the TFS is shown in figure 5.2. The embedded unit (EU) acquires data from the field sensors in the form of voltage or current or RTD (in case of temperature measurement) and communicates to the display unit (DU) for monitoring purpose. The field sensors are required to know the dynamics of the ECCS system like level of heavy water or light water in tanks, pressure in nitrogen gas tank (to ensure its capability to pressurize the water on demand), status of the valves, pumps, water flow rate, etc. Monitoring is required for some analysis purpose or to take some preventive action, by the operator. Apart from the monitoring, DU can also send commands to EU for alarm generation or to test the ECCS equipments.

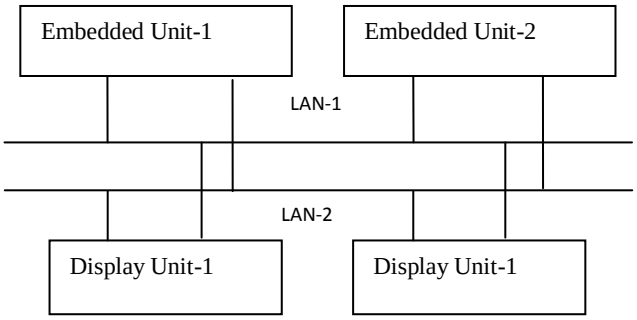

Figure 5.2: Architecture of Test Facility System.

The important components of EU are given in table 5.2. Apart from these module, EU also contains DC power supplies for these AIM, DIM, ROM cards and RT-20 Motorola

based 80280 16-bit processor of speed 1.4MHz, designed by Bhabha Atomic Research Centre, India.

During testing, EU sends the codes, mapped with testing messages to DU to know the ongoing dynamics of the ECCS. The dynamics of the ECCS gets displayed in the form of MIMIC, alarm messages, testing messages and also gets recorded for future use. Since the reliability requirements of TF is very high (10^{-3}year/year), redundant EU, DU and LAN has been used. EU is a real time; the software has been developed on real time operating system VxWorks, using C language and burned on EPROM - Motorola 80280, using RT-20. DU is a non real time system and is used for monitoring and sending request (to EU) is developed on Linux and C++ platform, using QT libraries.

5.4.4 Communication Module in TF

EU waits for an acknowledgement after a single transmission of a message. The packet is retransmitted up to maximum number of 5 transmissions if the timer expires or a negative acknowledgement comes after some acknowledgement time value of 2 seconds. For the retransmission mechanism, EU has a retry count which represents the number of transmissions for a specific packet send. In DU there is a state variable sR which stores sequence number of the packet to be received. This is used to detect duplicate packet to avoid duplicate status and alarm messages. After receive variable lifetime of 2 seconds timer expires, associated with receive lifetime value (T2), sR is destroyed or reset. EU has a variable sS to store the sequence number of packet to be transmitted or outstanding transmission. sS is used to relate a received acknowledgement to outstanding transmission and allow DU to detect duplicate frames. The moment transmit variable lifetime timer expires, sS is reset. The following algorithm has been developed:

1. Whenever EU sends a new packet, value of count is set to 1.

2. EU waits for acknowledgement for T1, after sending packet.

3. If EU receives acknowledgement before 2 seconds, packet send is successful and sS is set to 0/1(complement).

4. If $count < 5$ and acknowledgement is not being received, EU transmits packet and set $count = count + 1$, else EU terminates send process unsuccessfully, where sS will not change.

5. If 2 seconds elapsed after last data send, sS is destroyed.

6. At initial stage $sS = 0, p_{dsr}$ is source place, in which a token can enter at any time. p_{suc} and p_{usuc} are sink places in which token exits immediately.

5.5 Application of proposed framework for prediction of transition probabilities in Markov reliability model

We describe the communication module in TF in section 5.4.4. We apply our proposed framework to predict the transition probabilities in the Markov reliability model. We assume that requirements are clearly specified in the specification which is required to associate the time delay of the timed transitions of SPN.

5.5.1 Phase 1: Petri net model creation

Based on the communication protocol; we create a Petri net model of EU and DU in Time NET tool. For this we find the places and transitions of EU and DU. The details of places of EU are given in table 5.3(a) and transitions are given in table 5.3(b).

It is to be noted that in the tables, i' denotes the complement of i bit sequence number i.e. $0' = 1$ and $1' = 0$.

Now we create Petri net model after identifying the set of places $\{p_{dsr}, p_{srdyi}, p_{swi}, p_{toni}, p_{sis}, p_{rir}, p_{suc}, p_{usuc}\}$ and transitions $\{t_{si}, t_{swi}, t_{sik}, t_{siacksuc}, t_{siackusuc}, t_{comi}\}$ as shown in the figure 5.3. Conventionally timed transitions are shown by thick bar.

5.5.2 Phase2: Model Parameter assignment

As we describe in section 5.3.2, we use a tool Time NET for SPN creation. We keep a delay transitions as per the tolerant limit that is given in the specification of the system, as given in table 5.4.

$t_{si}, t_{sik}, t_{siacksuc}, t_{siackusuc}$ represent events that are supposed to occur within 1 millisecond, we set the value. As per the specification of the system the waiting time of the acknowledgement must not be more than 2 seconds, so we associate a delay of 2 seconds for t_{swi}.

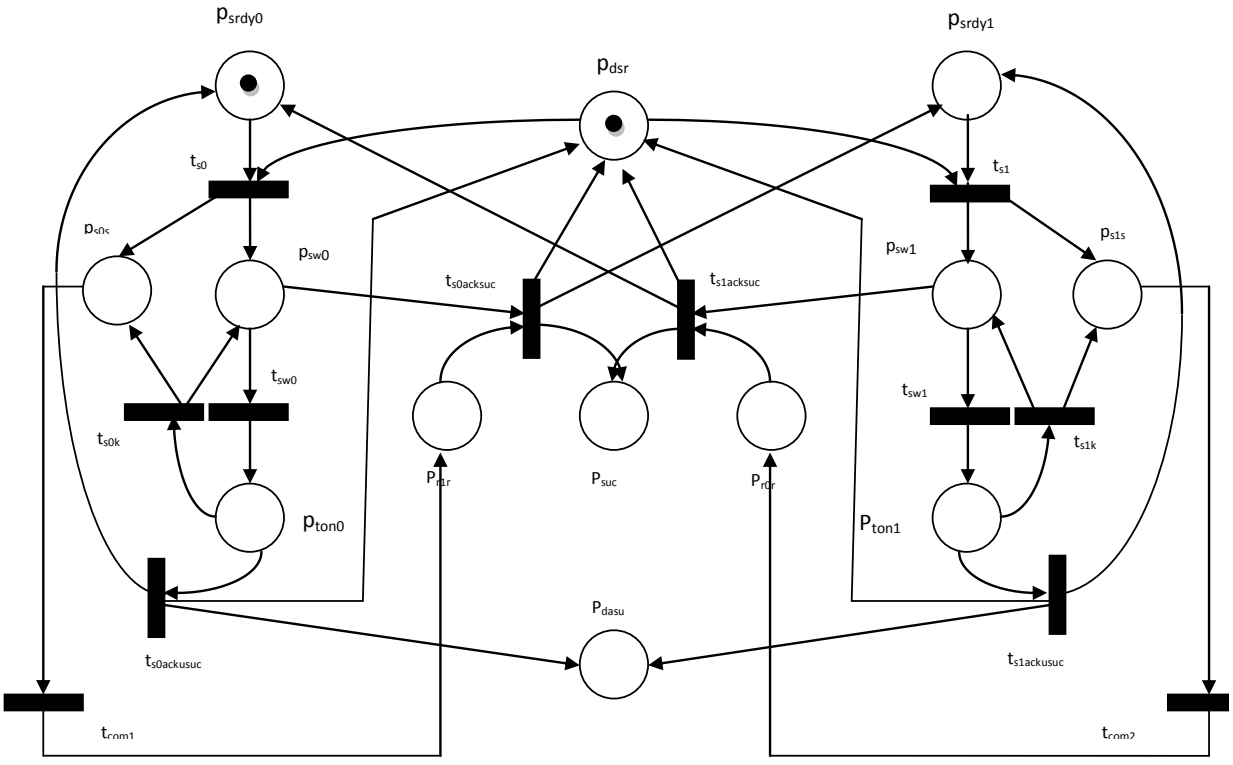

Figure 5.3: SPN of Embedded Unit.

Time NET gives the throughput after performing steady-state evaluation as shown in table 5.5.

5.5.3 Phase3: Reachability graph creation

Table 5.6 shows the marking of SPN of EU with their types. The full reachability graph is shown in figure 5.4. For the sake of convenience; we map the throughput in a sequence, as given in equation 5.8:

$$\{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7, \lambda_8, \lambda_9, \lambda_{10}, \lambda_{11}, \lambda_{12}\} = \{\lambda_{s0}, \lambda_{s1}, \lambda_{sw0}, \lambda_{sw1}, \lambda_{s0acksuc}, \lambda_{s1acksuc}, \lambda_{s0k}, \lambda_{s1k}, \lambda_{s0ackusuc}, \lambda_{s1ackusuc}, \lambda_{com1}, \lambda_{com2}\} \quad (5.8)$$

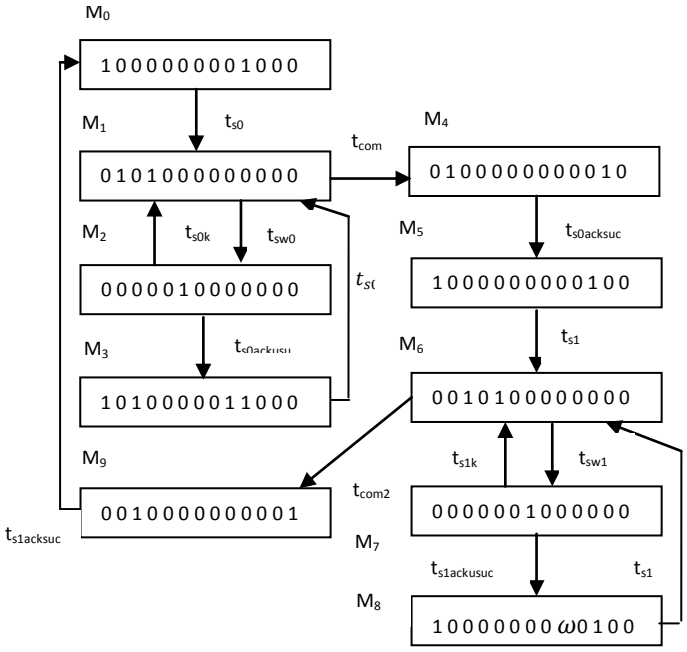

Figure 5.4: Reachability graph.

5.5.4 Phase4: Markov Chain creation

The Markov chain (MC) of a SPN, shown in figure 5.3, is given in figure 5.5 and can be obtained as described in section 5.3.4. We find the transition rate matrix, shown in given in equation 5.9.

$$
\begin{array}{c}
 \\ M_0 \\ M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \\ M_6 \\ M_7 \\ M_8 \\ M_9
\end{array}
\begin{array}{c}
\begin{array}{cccccccccc} M_0 & M_1 & M_2 & M_3 & M_4 & M_5 & M_6 & M_7 & M_8 & M_9 \end{array} \\
\left[\begin{array}{cccccccccc}
-\lambda_1 & \lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\lambda_3-\lambda_{11} & \lambda_3 & 0 & \lambda_{11} & 0 & 0 & 0 & 0 & 0 \\
0 & \lambda_7 & -\lambda_9-\lambda_7 & \lambda_9 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \lambda_1 & 0 & -\lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -\lambda_5 & \lambda_5 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -\lambda_2 & \lambda_2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -\lambda_4-\lambda_{12} & \lambda_4 & 0 & \lambda_{12} \\
0 & 0 & 0 & 0 & 0 & 0 & \lambda_8 & -\lambda_{10}-\lambda_8 & \lambda_{10} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \lambda_2 & 0 & -\lambda_2 & 0 \\
\lambda_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\lambda_6
\end{array}\right]
\end{array}
$$

$$
= \begin{array}{c}
 \\ M_0 \\ M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \\ M_6 \\ M_7 \\ M_8 \\ M_9
\end{array}
\begin{array}{c}
\begin{array}{cccccccccc} M_0 & M_1 & M_2 & M_3 & M_4 & M_5 & M_6 & M_7 & M_8 & M_9 \end{array} \\
\left[\begin{array}{cccccccccc}
-6.61 & 6.61 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -3.37 & 0.08 & 0 & 3.29 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.04 & -0.08 & 0.04 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 6.61 & 0 & -6.61 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1.64 & 1.64 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -6.58 & 6.58 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -3.35 & 0.08 & 0 & 3.27 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.04 & -0.08 & 0.04 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 6.58 & 0 & -6.58 & 0 \\
1.64 & 0 & 0 & 0.4 & 0.5 & 0.2 & 0 & 0 & 0 & -1.64
\end{array}\right]
\end{array}
\tag{5.9}
$$

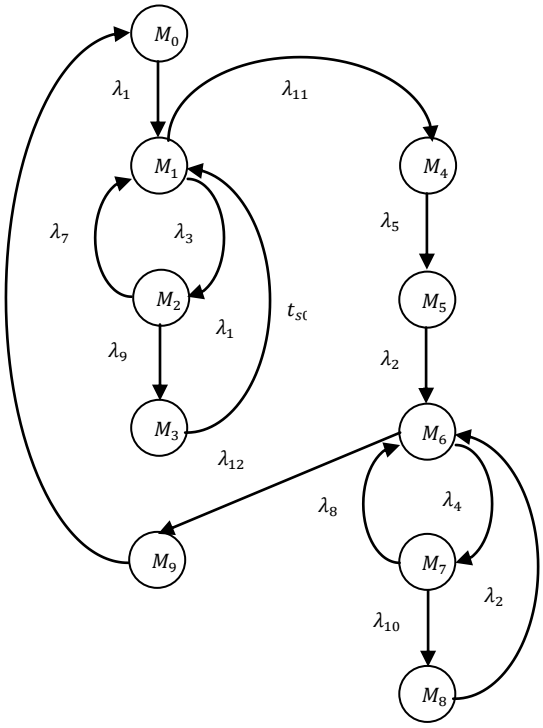

Figure 5.5: Markov chain creation.

5.5.5 Phase5: Transition Probability estimation

We estimate the transition probability matrix from equation 5.6. It is shown in equation 5.10.

$$
P = \begin{array}{c} \\ M_0 \\ M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \\ M_6 \\ M_7 \\ M_8 \\ M_9 \end{array}
\begin{array}{c}
\begin{array}{cccccccccc} M_0 & M_1 & M_2 & M_3 & M_4 & M_5 & M_6 & M_7 & M_8 & M_9 \end{array} \\
\left[\begin{array}{cccccccccc}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.024 & 0 & 0.976 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.5 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.025 & 0 & 0.975 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0.5 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0.2 & 0 & 0 & 0 & 0
\end{array}\right]
\end{array}
\tag{5.10}
$$

5.6 Validation of our approach

We validate the accuracy of our approach of predicting transition probabilities in Markov reliability model by computing the reliability of the communication module, based on our predicted transition probabilities and comparing it with its reliability, based on operational profile data of two years.

5.6.1 Reliability estimation, based on the predicted transition probabilities

From figure 5.3, table 5.3, figure 5.4 and figure 5.5, we know that there are only two failure states in the created Markov Chain M_3 and M_8. Rests of the states are Behavioral states. From equation 4.5, we get equation 5.11.

$$
\begin{aligned}
&[M_0 \quad M_1 \quad M_2 \quad M_3 \quad M_4 \quad M_5 \quad M_6 \quad M_7 \quad M_8 \quad M_9] \\
&\qquad = [M_0 \quad M_1 \quad M_2 \quad M_3 \quad M_4 \quad M_5 \quad M_6 \quad M_7 \quad M_8 \quad M_9] \times \\
&\qquad\qquad \begin{array}{c} \\ M_0 \\ M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \\ M_6 \\ M_7 \\ M_8 \\ M_9 \end{array}
\begin{array}{c}
\begin{array}{cccccccccc} M_0 & M_1 & M_2 & M_3 & M_4 & M_5 & M_6 & M_7 & M_8 & M_9 \end{array} \\
\left[\begin{array}{cccccccccc}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.024 & 0 & 0.976 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.5 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.025 & 0 & 0.975 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0.5 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]
\end{array}
\end{aligned}
\tag{5.11}
$$

Solving the equation 5.11, we get the following linear equations

$$M_0 = M_9 \quad \text{(i)}$$

$$M_1 = M_0 + 0.5M_2 + M_3 \quad \text{(ii)}$$

$$M_2 = 0.024M_1 \quad \text{(iii)}$$

$$M_3 = 0.5M_2 \quad \text{(iv)}$$

$$M_4 = 0.0976M_1 \quad \text{(v)}$$

$$M_5 = M_4 \quad \text{(vi)}$$

$$M_6 = M_5 + 0.5M_7 \quad \text{(vii)}$$

$$M_7 = 0.025M_6 \quad \text{(viii)}$$

$$M_8 = 0.5M_7 \quad \text{(ix)}$$

$$M_9 = 0.975M_6 \quad \text{(x)}$$

Using equation 4.4, we get

$$M_0 + M_1 + M_2 + M_3 + M_4 + M_5 + M_6 + M_7 + M_8 + M_9 = 1 \quad \text{(xi)}$$

Solving above 11 equations (i to xi), we get

$$M_0 = M_9 = 0.1602; M_1 = 0.16414; M_2 = 0.0039408; M_3 = 0.00197; M_4 = M_5 = 0.1602;$$
$$M_6 = 0.1603; M_7 = 0.004; M_8 = 0.002$$

We can notice that matrix P is like a sparse matrix and we found this sparse nature in all the design verification models of various other systems, which we developed for NPP. Hence it can be improved to take very less storage and computation time.

So, using equation 4.3, we get

$$\overrightarrow{p} = [0.1602, 0.16414, 0.0039408, 0.00197, 0.1602, 0.1602, 0.1634, 0.004, 0.002, 0.1602]$$

Hence the reliability of the communication module, using equation 11 is given by:

$$\begin{aligned} R_{com}^{est} &= 1 - p(M_3) - p(M_8) \\ &= 1 - 0.00197 - 0.002 \\ &= 0.99603 \end{aligned}$$

Rewriting the reliability,

$$\boxed{R_{com}^{est} = 0.99603} \tag{5.12}$$

5.6.2 Reliability estimation, based on the operational profile data

To estimate the reliability, based on the operational profile, we use Ramamoorthy and Bastani model [32] because of its suitability for real time systems. We present the method by proposing a framework, containing four phases, given in figure 5.6. These phases are illustrated as under:

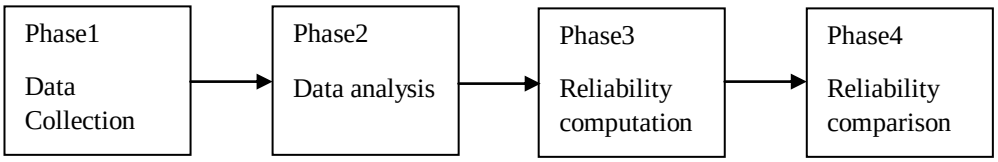

Figure 5.6: Reliability Computation framework.

5.6.2.1 Phase 1: Data Collection

DU maintains the record of every analog and digital data up to 3 years. This TF is running in 7 units, namely (i) Tarapur Atomic Power Station (TAPS)-3 (ii) TAPS-4 (iii) Rajasthan Atomic Power Station (RAPS)-3 (iv) RAPS-5 (v) RAPS-6 (vi) Kaiga Generating Station (KGS)-3 (vii) KGS-4. We could able to collect all the records of 720 days from the permission of shift charge engineer of the respective stations, from all the 7 units. The operational profile data of TAPS-3 is given in table 5.8.

5.6.2.2 Phase 2: Data Analysis

Every command initiated by the operator and its response from the EU is recorded with the time stamping to know the successful or unsuccessful operation. This is also very important for performing analysis in case of any fault, failure or event. The time stamping is done by the EU. There are message codes for each message, which is sent by EUand the actual message is generated by DU. For acknowledgement purpose a different message code is being sent by DU to EU.Every data is time stamped by EUin the format of $''dd/mm/yyyyhr : min : sec : msec''$. For example, in the table 5.7, there are five fields, explained under:

1. Message Code: every message is mapped with some code for message creation. For example, $''3032''$ means command to open motor valve-1.

2. Message sent by DU:This is the command sent by the DU to EU to perform the required operation. For example $''01/03/201223 : 15 : 28 : 203032''$ means DU has sent the message code 3032 to EU along with date and time information.

3. Message interpreted by EU:EU interprets the message code. For example "Motor valve-1 to be opened" means EU has interpreted that a command has come to open the motor valve-1.

4. Acknowledgement message sent by EU to DU:EU sends the acknowledgement message after performing the intended function along with the time stamp.

5. Message displayed by DU: The acknowledgement message sent by the EU gets decoded by DU. For example: $''01/03/201223 : 15 : 28 : 22$ Motor valve-1 is opened" means at the given time, motor valve-1 is opened at the given time.

Let number of runs in 1 day = 1.

5.6.2.3 Phase 3: Reliability Computation

We use Ramamoorthy and Bastani model [32] for reliability computation, according to which the reliability is given by equation 5.13. We collected operational profile date of 2

years for this purpose that is shown in table 5.8.

$$R_j(t) = E_{\lambda_j}[\mathrm{e}^{-\lambda_j \int_0^t f(T_j(s))ds}] \tag{5.13}$$

where

λ_j = failure rate after j_{th} failure; $0 \leq \lambda_j \leq \infty$

$T_j(s)$ = testing process at time s after j_{th} failure

$f(T_j(s))$ = severity of testing process relative to operational distribution; $0 \leq f(T_j(s)) \leq \infty$

For operational data, we can assume $f(T_j(s)) = 1$. Hence reliability equation becomes

$$R_j(t) = E_{\lambda_j}[\mathrm{e}^{-\lambda_j \int_0^t ds}] \tag{5.14}$$

Hence, the reliability of the Communication module of TF based on operational profile is given by

$$R_{act}^{com} = \frac{119\mathrm{e}^{0\times1} + \mathrm{e}^{-1\times1} + 119\mathrm{e}^{0\times1} + \mathrm{e}^{-1\times1} + 179\mathrm{e}^{0\times1} + \mathrm{e}^{-1\times1} + 179\mathrm{e}^{0\times1} + 120}{720}$$

$$= 0.9964$$

$$\boxed{R_{com}^{act} = 0.9964} \tag{5.15}$$

5.6.2.4 Phase 4: Reliability Comparison

In this phase, we compare its estimated value (equation 5.12) and actual value (equation 5.15). Correct validation of reliability also validates the predicted transition probability that was an issue in the existing approaches as discussed in section 5.2.

$$\begin{aligned} R_{com}^{act} &> R_{com}^{est} \\ R_{com}^{diff} &= R_{com}^{act} - R_{com}^{est} \\ &= 0.99603 - 0.9964 \\ &= 0.00037 \end{aligned} \tag{5.16}$$

We get the magnificent results as the difference is negligible. We investigated the results and came to the following noticeable facts to justify this small difference.

1. We have shown the reliability figure using operational profile of two years. The reason is that, we have installed a new version of this software in all units of NPP and collected data from the time at which it started in operational phase. As we can see, when time elapse is very long, reliability may get stabilized.

2. The predicted reliability depends on the accuracy of the assigned parameters in TimeNET tool. We show the sensitivity analysis of parameter assignment in section 5.7.

Interestingly, the unreliability figure is more of interest, especially if we deal with the safety critical or safety related systems of NPP.

The predicted unreliability figure, we got from our approach is given by

$$\begin{aligned} UR_{com}^{act} &= 1 - R_{com}^{act} \\ &= 1 - 0.99603 \\ &= 0.00397 \end{aligned} \tag{5.17}$$

The unreliability figure, based on the operational profile can be computed as

$$\begin{aligned} UR_{com}^{est} &= 1 - R_{com}^{est} \\ &= 1 - 0.9964 \\ &= 0.0036 \end{aligned} \tag{5.18}$$

Hence, the difference between the predicted and actual unreliability is given by:

$$\begin{aligned} UR_{com}^{diff} &= UR_{com}^{act} - UR_{com}^{est} \\ &= 0.000397 - 0.0036 \\ &= -0.00037 \end{aligned} \tag{5.19}$$

We get the accuracy of 89.73% in the predicted and actual (based on operational profile) unreliability figure which is quite rewarding.

Similarly we compute theR_{com}^{diff} and UR_{com}^{diff} or rest of the 6 atomic power stations and compared with the predicted reliability. The accuracy in the predicted and actual unreliability figure of Communication module of TF for all the 7 stations are given in table 5.9.

5.7 Sensitivity Analysis of Parameter assignment

We show the impact of the values of transition delay on transition probabilities. To illustrate this we change the delay of only one timed transition, say t_{s0} at the step of 10 milliseconds and noted down the changed throughput of all the timed transitions. We illustrate the change of throughput and transition probability from M_1 to M_4 for only t_{com1} in figure 5.7 and figure 5.8 respectively. We find that the transition probabilities do not deviate much if the assigned values of delay of timed transitions are not much deviated. Noticeably, if there is a little deviation in many transition probabilities, the cumulative effect will be substantial enough to give much deviated or incorrect reliability figure.

5.8 Performance estimation

We can also estimate the performance of a system, if we know the time spent in each state when control reaches to it. Although, it is not required for our case study but may be required for several real time systems. The time spent in any state is known as sojourn time. For each state i, the amount of time spent in that state in a given visit is an exponentially distributed random variable, with parameter w_i. In the case where the

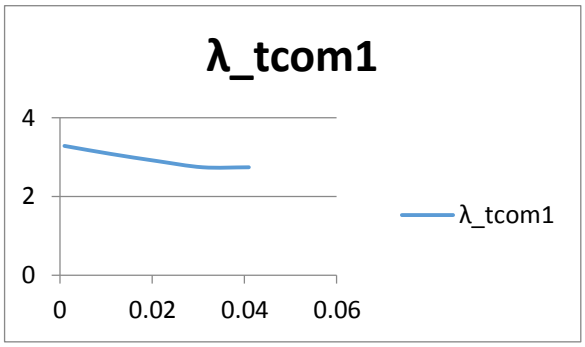

Figure 5.7: Throughput change for t_{com1}.

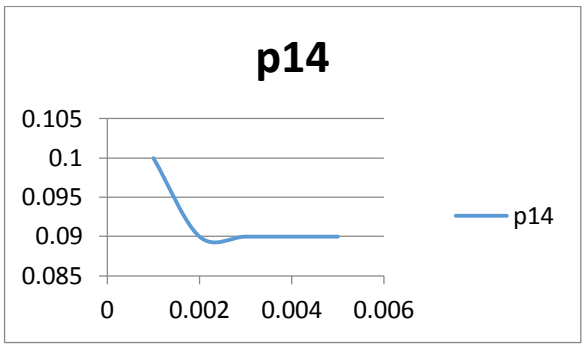

Figure 5.8: Transition probability change for t_{com1}.

state is absorbing, i.e., where state never transits, we define w_i to be equal to zero. In this section we derive the holding times from the transition rate matrix.

Let's define $Q = P'(0)$. Fix a state i. If $w_i = 0$, state never transits and hence for all times t, we have

$$\begin{aligned} P_{ij}(t) &= 0 \qquad \forall j \neq i \\ P_{ii}(t) &= 1 \end{aligned} \tag{5.20}$$

Thus we conclude that if $w_i = 0$, then $Q_{ij} = 0$ for all states j.

We now consider the more interesting case where $w_i \neq 0$. Let us make the approximation that in a small time δ, the chain will make only at most one jump; this approximation is not valid if δ is large, but is asymptotically valid as $\delta \to 0$. First consider a state $j \neq i$, we have:

$$\begin{aligned} Q_{ij} &= \lim_{\delta \to 0} \frac{P_{ij}(\delta) - P_{ij}(0)}{\delta} \\ &= \lim_{\delta \to 0} \frac{P_{ij}(\delta)}{\delta} \\ &\approx \lim_{\delta \to 0} \frac{1 - \mathrm{e}^{-w_i \delta} R_{ij}}{\delta} \\ &= -w_i R_{ij} \end{aligned} \tag{5.21}$$

The approximation we use is that if the chain goes from i to j in time δ, then the chain must make one jump in the interval$[0, \delta]$, and when it makes this jump it must go to state j.

Similarly, for Q_{ii}:

$$\begin{aligned} Q_{ii} &= \lim_{\delta \to 0} \frac{P_{ii}(\delta) - P_{ii}(0)}{\delta} \\ &= \lim_{\delta \to 0} \frac{P_{ii}(\delta) - 1}{\delta} \\ &\approx \lim_{\delta \to 0} \frac{\mathrm{e}^{-w_i \delta} - 1}{\delta} \\ &= -w_i \end{aligned} \tag{5.22}$$

Combining this with our analysis for the case where $w_i = 0$, we find:

$$Q_{ij} = \begin{cases} -w_i, & j = i \\ w_i P_{ij}, & j \neq i \end{cases} \tag{5.23}$$

From these equations and above, we perceive that given the transition rate matrix, we can compute the transition probability matrix and the holding times. Thus the transition rate matrix contains the same modeling information as the holding time chain specification.

5.9 Conclusion

In this chapter we proposed an approach to predict the transition probabilities in a Markov reliability model. The reliability assessment in Markov model is based on transition probabilities in between the states of MC. In section 5.2, we infer that in the existing approaches authors have either assumed them on the basis of some coarse knowl-

edge or computed them using analytical methods which do not give accurate values. Some authors have computed them using operational profile but that is possible only after deployment of the system and hence it is not an early prediction. Our framework addressed the existing limitations and is described in section 5.2. We applied this framework on communication module of TF in section 5.5.We also illustrated the technique to compute the sojourn time of any state of Markov chain and shown that transition rate matrix contains the same modeling information as the sojourn time chain specification. Sojourn time helps in estimating the performance metrics. Sensitivity of parameter assignment has been shown to show its importance for correct estimation of design metrics. The validation of our approach is shown on the seven different sets of operational profile data of NPP in section 5.6, using Ramamoorthy and Bastani model. We also drew some noteworthy facts for getting small difference in the predicted and computed reliability figure.

Our evaluation results indicate that our framework provides meaningful estimation of reliability. The estimation of reliability for a small module will lead to estimation of reliability of the whole system to take early preventive action.

Table 5.3: EU places and transitions.

p_{dsr}	EU gets request from DU to send data
p_{srdyi}	EU is ready to send packet with $\vartheta(S_i) = i$
p_{swi}	EU is waiting for ack with seq. no. i'
p_{toni}	Ack timer timed out for data send with $\vartheta(S_i) = i$
p_{sis}	Data packet of i seq. no. sent
p_{rir}	Ack with seq. no. i has received
p_{suc}	Data send successful is reported to operator
p_{usuc}	Data send unsuccessful is reported to operator

(a) EU places.

t_{si}	EU sends the packet with seq. no. i. Set retry count $k = 1$ by the firing of this transition
t_{swi}	EU detects the timeout of the ack. Timer ($I(t_{sdackto}) = [T_1, T_1]$), where I assigns a closed firing interval to each transition
t_{sik}	EU sends the packet with seq. no. i. This transition is fired only when $k < N$ and subsequently k increases
$t_{siacksuc}$	Ack. With seq. no. i' is received and data send is completed successfully
$t_{siackusuc}$	Process of data send is terminated without success after N trials. This transition is fired when $K \geq N$
t_{comi}	DU sends the ack to EU with seq. no. i

(b) EU transitions.

Table 5.4: EU transitions with delay.

Transition	**Delay**
t_{si}	0.001
t_{sik}	0.001
t_{swi}	2.00
$t_{siacksuc}$	0.001
$t_{siackusuc}$	0.001
t_{comi}	0.1

Table 5.5: Throughput of the transitions.

Transition	**Delay**
λ_{s0}	6.609748407158103
λ_{s1}	6.583028510345760
λ_{s0k}	0.041071027462874
λ_{s1k}	0.040884540272115
λ_{sw0}	0.082128363780294
λ_{sw1}	0.081772575101630
$\lambda_{s0acksuc}$	1.642883346667905
$\lambda_{s1acksuc}$	1.635351620802226
$\lambda_{s0ackusuc}$	0.041071027462874
$\lambda_{s1ackusuc}$	0.040884540272115
λ_{com1}	3.285134551211776
λ_{com2}	3.270903004065200

Table 5.6: Markings of EU GSPN model.

M_0	$(1,0,0,0,0,0,0,0,0,1,0,0,0)$	tangible
M_1	$(0,1,0,1,0,0,0,0,0,0,0,0,0)$	tangible
M_2	$(0,0,0,0,0,1,0,0,0,0,0,0,0)$	tangible
M_3	$(1,0,0,0,0,1,0,0,1,1,0,0,0)$	tangible
M_4	$(0,1,0,0,0,0,0,0,0,0,0,1,0)$	tangible
M_5	$(1,0,0,0,0,0,0,0,0,0,1,0,0)$	tangible
M_6	$(0,0,1,0,1,0,0,0,0,0,0,0,0)$	tangible
M_7	$(0,0,0,0,0,0,1,0,0,0,0,0,0)$	tangible
M_8	$(1,0,0,0,0,0,0,0,\omega,0,1,0,0)$	tangible
M_9	$(0,0,1,0,0,0,0,0,0,0,1,0,1)$	tangible

Table 5.7: Command Messages and acknowledgement messages of Communication.

S. No.	**Message Code**	**Message sent by DU**	**Message interpreted by EU**	**Ack. Message sent by EU to DU**	**Message displayed by DU**
1	3032 (Motor valve-1 open) 3042 (Motor valve-1 opened ack)	01/03/2012 23:15:28:20 3032	Motor valve-1 to be opened	01/03/2012 23:15:28:22 3042	01/03/2012 23:15:28:22 Motor valve-1 is opened
2	4033 (Motor valve-1 close) 4043 (Motor valve-1 closed ack)	01/03/2012 23:15:30:20 4033	Motor valve-1 to be closed	01/03/2012 23:15:30:22 4043	01/03/2012 23:15:30:22 Motor valve-1 is closed

Table 5.8: Operational profile Data.

t	n	n_f
1	1	0
30	30	0
60	60	0
120	120	1
150	150	1
180	180	1
210	210	1
240	240	2
270	270	2
300	300	2
330	330	2
360	360	2
390	390	2
420	420	3
450	450	3
480	480	3
510	510	3
540	540	3
570	570	3
600	600	4
630	630	4
660	660	4
690	690	4
720	720	4

Table 5.9: Accuracy of predicted unreliability figure of communication module for given seven stations.

TAPS-3	**TAPS-4**	**RAPS-3**	**RAPS-5**	**RAPS-6**	**KGS-4**	**KGS-4**
89.73%	91.20%	90.80%	91.73%	90.48%	89.92%	91.18%