CHAPTER 4

REPUTATION-BASED TRUST-INDEX COMPUTATIONS

4.1 INTRODUCTION

The reputation-based trust-index computations of various entities such as brokers, consumers and RPs in Multi-Broker Architecture and Hierarchical Broker Architecture are discussed in this chapter. The genuine feedbacks are considered for the computation of trust-indices using pre-evaluation-set to safeguard the trust-indices of RP from the malicious entities’ feedbacks. The simulation of Multi-Broker Architecture, Multi-Broker Architecture without delegation of consumer requests and Hierarchical Broker Architecture are discussed in this chapter. Their improved results are given. A very important issue in software security, namely that of assuring the integrity of computation that occurs on a remote, un-trusted host in grid environment is considered using quiz-based result verification scheme. The reputation-based trustworthiness of RP is computed based on the quiz-based result verification scheme in this thesis.

4.2 TRUST-INDEX COMPUTATION IN MULTI-BROKER ARCHITECTURE

The computation of satisfaction-index for the resource of RP, trust-indices of the RP, the brokers and the consumers in Multi-Broker Architecture are briefly described in this section. When a consumer sends its request to broker, the broker chooses the best trustworthy RP among the RP nominated
from a set, which comprised its (own) nomination as well as nominations from other brokers in the same domain as well as other brokers from other domains, as explained earlier. The notification is sent to both the concerned consumer and RP by the associated broker. After the completion of transaction, satisfaction (satisfaction-index or feedback) about the resource provided by the RP is sent by the consumer to the broker. The computation of the trust-indices of RP is done based on the satisfaction-index of consumers and brokers for the resource of RP. The computation of satisfaction-index is discussed in the following section.

4.2.1 Computation of Satisfaction-Index of RP

The satisfaction-index of an RP is computed by two different entities based on whether it is for a resource provided to a consumer or it is a resource monitored by the broker.

4.2.1.1 Satisfaction-Index of RP from consumer feedback

After utilizing the resource of the selected RP, the consumer \( U_c \) sends feedbacks on the resource (satisfaction-index) to the broker \( B \) who recommended the RP. The satisfaction-index \( S(U_c, \text{RP}) \) is a weighted sum of the percentage of completion of assigned work \( (S_1) \), whether the job completed within budget \( (S_2) \), the level of commitment of the RP to minimize the schedule-variance \( (S_3) \) and the level of intrusion, if any, that is detected by the audit data at the consumer’s site \( (S_4) \). The satisfaction-index is computed by Equation (4.1), as

\[
S_i(U_c, \text{RP}) = \sum_{j=1}^{j=4} \gamma_j S_j
\]  

(4.1)
Here $S_i(U_c, RP)$ is the satisfaction-index provided by the consumer ($U_c$) to the RP for $i^{th}$ transaction, $U_c$ is the ID of consumer, $\gamma_j$ is the weight associated with the satisfaction-index parameters $S_j$. Equal weights ($\gamma_j$) are assumed in the simulation for all $S_j$ parameters. The value of $S_2$ is assigned between -1 to +1 based on Budget Variance (BV) as given below.

$$BV = \frac{\text{proposed budget} - \text{actual budget}}{\text{proposed budget}}$$

Therefore, the value of $S_2$ is given as

$$S_2 = BV \text{ if } BV < 0 \text{ and } BV \geq -1$$
$$S_2 = 1 \text{ if } BV \geq 0$$
$$S_2 = -1 \text{ if } BV < -1$$

Similarly, $S_3$ is computed based on Schedule Variance (SV) as follows.

$$SV = \frac{\text{proposed schedule} - \text{actual schedule}}{\text{proposed schedule}}$$

Therefore, the value of $S_3$ is given as

$$S_3 = SV \text{ if } SV < 0 \text{ and } SV \geq -1$$
$$S_3 = 1 \text{ if } SV \geq 0$$
$$S_3 = -1 \text{ if } SV < -1$$

So, the values of $S_1$ ranges from 0 to 1 as continuous quantity, $S_2$ and $S_3$ varies from -1 to 1 as continuous value and $S_4$ as 0 or 1 as a discrete value based on whether there is an intrusion or not.
4.2.1.2 Satisfaction-Index of RP from broker feedback

The satisfaction-index of each RP associated with a broker is computed and maintained at that broker’s site based on the personal experience of the broker with each of these RPs. This is performed on the same lines as satisfaction-index computation on consumer’s feedback with \( B_c \) being the consumer in this case. In addition to the four parameters \( (S_1, S_2, S_3, S_4) \) considered above, the broker also uses the presence-index to fine tune the computation of satisfaction-index. Therefore, the satisfaction-index of the broker \( B_{RP} \) for the RP’s resource is given by Equation (4.2) as

\[
S_i(B_{RP}, RP) = \sum_{j=1}^{j=S} (\gamma_j * S_j)
\]  

(4.2)

Here \( B_{RP} \) is the ID of broker associated with the RP, \( S_5 \) is the presence-index (PI) of the RP, \( \gamma_j \) is the weight associated with the satisfaction-index parameters \( S_j \). PI is an average of the values of I-index and P-index. I-index is based on the bandwidth, system-capability, failover process, business continuity and disaster recovery process and adherence to standards. P-index is based on the affiliation of the entity to international standards of information security such as True Secure, ISO 270001 and BITS, and the policies of the RP on warranty, payment and the delivery. All these indices are either 1 or 0, based on whether the RP provides the agreed-upon facility or not. The satisfaction-index provided by the consumer \( U_c \) and the associated broker \( B_{RP} \) are consolidated, as explained below, into a single satisfaction-index for the particular transaction.

If the absolute difference between the feedback provided by the broker and the consumer, \( | S_i(U_c, RP) - S_i(B_{RP}, RP) | \) is greater than a set threshold value (T), then the feedback provided by the consumer is considered false (since the broker is a centralized agent and is more trust-worthy) and
only the satisfaction-index provided by the broker is considered as the satisfaction-index of the present transaction. Otherwise, an average of \( S_i(U_c, \text{RP}) \) and \( S_i(B_{\text{RP}}, \text{RP}) \) is used as the satisfaction-index \( (S_i(U_c, B_{\text{RP}}, \text{RP}) \). Thus

\[
\text{If } |(S_i(U_c, \text{RP}) - S_i(B_{\text{RP}}, \text{RP}))| \leq T \quad (S_i(U_c, B_{\text{RP}}, \text{RP}) = \frac{1}{2}(S_i(U_c, \text{RP}) + S_i(B_{\text{RP}}, \text{RP}))
\]

\[
\text{Else} \quad (S_i(U_c, B_{\text{RP}}, \text{RP}) = S_i(B_{\text{RP}}, \text{RP})
\]

### 4.2.2 Trust-Index Computation of RP

The broker associated with the consumer forwards the above feedbacks (satisfaction-indices) to all the brokers associated with that RP. These feedbacks are redefined as transaction information and are stored at the site of all the brokers associated with this RP. This consists of Consumer ID \( (U_c) \), RP, type of resources, satisfaction-index \( (S_i(U_c, B_{\text{RP}}, \text{RP})) \), cost \( (C) \) and the date of transaction (time stamp). A time stamp is included with each feedback to differentiate past transactions from the more recent ones. The broker decides the time stamp called breakpoint date based on the application and the consumer’s requirements. The frequency of requests made for an RP may decide this breakpoint date. This breakpoint date decides a transaction as being recent or past. The transactions graded as recent are assigned a higher weight \( (\alpha) \), past transactions, a lower weight \( (\beta) \), in the computation of trust-index of the RP, with \( (\alpha + \beta) = 1 \). If an application warrants the computation of trust-index based on only recent transactions, \( \beta \) is assigned a value of 0. If there is no need for any distinction between past and recent transactions, \( \alpha \) and \( \beta \) are assigned equal weights \( (\alpha = \beta = 0.5) \). The trust-index computation is given by Equation (4.3), as follows:
Here $S_i(U_c, B_{RP}, RP)$ is the satisfaction-index given by the consumer ($U_c$) and/or the broker ($B_{RP}$) of the RP for the $i^{th}$ transaction ranges between -1 to +1 as a continuous quantity, $t_1$ is the number of recent transactions after the breakpoint date and $t_2$ is the number of past transactions before the breakpoint date.

This method of computing the trust-index of the RP is not robust. One RP may perform thousand transactions and cheat on tens out of every hundred transactions, while another RP may perform only ten transactions and is trustworthy in all the transactions performed. If the above procedure of simple aggregation is considered for computing the trust-index of the RP, both will have more or less the same trust-index, though the second RP is obviously more trustworthy compared to the first one. Therefore, trust-index of the RP is averaged over the aggregated satisfaction-indices of all past and recent transactions. Hence, the trust-index computation is modified by Equation (4.4), as

$$TI(RP) = \alpha \sum_{i=1}^{t_2} S_i(U_c, B_{RP}, RP) + \beta \sum_{i=1}^{t_1} S_i(U_c, B_{RP}, RP)$$

This improves the robustness of the model. Another factor known as the criticality weight, $CR_i$, of each transaction $i$ may also be considered to further fine-tune the computation of trust-index. This distinguishes between more critical and less critical transactions. This value of $CR_i$ is also between 0 and 1 as a continuous quantity. The cost ($C_i$) of each transaction is used to further refine the computation of trust-index of the RP. Higher weight is
assigned to transactions of higher cost than the cheaper transactions. Thus weight is proportional to the cost of the transaction. Normalized value of the cost of transaction is used in the computation of trust-index of RP. After incorporating this refinement, the trust-index is computed by Equation (4.5), as

$$TI(RP) = \left[ \frac{\sum_{i=1}^{n_1} (S_i(U_c, B_{RP}, RP) \ast C_{ni} \ast CR_i)}{t_1} - \frac{\sum_{i=2}^{n_2} (S_i(U_c, B_{RP}, RP) \ast C_{ni} \ast CR_i)}{t_2} \right]$$  \hspace{1cm} (4.5)

Here $C_{ni}$ is the normalized value of the cost of the $i^{th}$ transaction. Trust-index of the current $i^{th}$ transaction ($TI_{ct}(RP)$) alone is computed using Equation (4.6) as follows

$$TI_{ct}(RP) = S_i(U_c, B_{RP}, RP) \ast C_{ni} \ast CR_i$$  \hspace{1cm} (4.6)

While computing the trust-index of an RP, higher weight is assigned to direct experience than others’ recommendations. Thus the trust-index computation is modified by Equation (4.7), to

$$TI(RP) = a + b$$  \hspace{1cm} (4.7)

Here

$$a = \delta_1 \left[ \frac{\sum_{i=1}^{n_1} (S_i(U_c, B_{RP}, RP) \ast C_{ni} \ast CR_i)}{t_1} - \frac{\sum_{i=2}^{n_2} (S_i(U_c, B_{RP}, RP) \ast C_{ni} \ast CR_i)}{t_2} \right]$$

$$b = \delta_2 \left[ \frac{\sum_{i=1}^{n_1} (S_i(U_c, B_{RP}, RP) \ast C_{ni} \ast CR_i)}{t_1} - \frac{\sum_{i=2}^{n_2} (S_i(U_c, B_{RP}, RP) \ast C_{ni} \ast CR_i)}{t_2} \right]$$
Here, $\delta_1$ and $\delta_2$ are the weights assigned for direct experience and other’s recommendations, respectively. The trust-index of the RP, built at the broker’s site using the consumer’s and broker’s feedbacks, is consistently made available to all the brokers associated with the RP, for updating their databases. This ensures the correctness of trust-index of RP. Better trust-index is synonymous with better resource service.

4.2.3 Trust-Index Computation of Brokers

After the completion of each transaction, the trust-index of the broker associated with the selected RP ($TI(B_{RP})$) is computed and updated at the associated standards forum’s site. This process is similar to trust-index computation of the RP at the associated broker’s site. The trust-index of the broker ($TI(B_{RP})$) is a function of the trust-index ($TI_{ct}(RP)$) obtained from the current transaction and trust-indices of the past transactions ($TI_p(B_{RP})$) except current one ($TI_{ct}(RP)$), which are maintained at the standard forum’s site. The trust-index computation of the broker ($TI(B_{RP})$) associated with the selected RP is performed at the standard forum’s and given by Equation (4.8) as

$$TI(B_{RP}) = \left[ \frac{TI_p(B_{RP}) \times T + TI_{ct}(RP)}{(T+1)} \right]$$

(4.8)

Here ($TI_p(B_{RP})$) is the past trust-index of the broker, $B_{RP}$ for T past transactions.

4.2.4 Trust-Index Computation of Consumers

The trust-index of the consumer ($TI(U_c)$) is also computed and updated at the associated brokers’ site after the completion of each transaction. It is the weighted average of whether or not the consumer
provided feedback on the utilization of the resource \((x_1 = +1/-1)\) and whether it honored its commitment on payment \((x_2 = +1/-1)\). Cost of a transaction and commitment for payment is considered significantly important in the grid environment for selecting the best RP for a requested resource or computing the trust-index of a consumer. This is because of the payment-based economy system. The trust-index of the consumer for the \(i^{th}\) transaction \((TI_i(U_c))\) is computed using Equation (4.9) as

\[
TI_i(U_c) = w_1 \cdot x_1 + w_2 \cdot x_2
\]

Here, \(w_1\) and \(w_2\) are the equal weights assigned to \(x_1\) and \(x_2\), respectively. The trust-index of the consumer \(U_c\) after \(t\) transactions \((TI(U_c))\), is computed by Equation (4.10), as

\[
TI(U_c) = \frac{\sum_{i=1}^{t} TI_i(U_c)}{t}
\]

The associated brokers expect the consumer’s trust-index to be greater than a minimum set-value to process the consumer’s request. This set-value is selected by the brokers based on the criticality of the resource and cost involved in the transaction. The trust-index of a new consumer will be assigned with the same minimum set-value by the associated broker. Later, trust-index of the consumer is updated based on its deal in the current transaction (whether the payment is made and the feedback is provided). Trust-index of a trust-worthy consumer would increase with every transaction, whereas it would decrease for an untrustworthy consumer. If the trust-index of the consumer is less than a minimum set-value, the consumer is penalized by not processing its next three requests, and then the consumer is reassigned with the same initial trust-index, allowing for further transactions. However, the trust-index of a new consumer is greater than the trust-index of untrustworthy (old) consumer due to its past untrustworthy transactions.
4.2.5 Simulation and Results

This architecture is simulated with 10 domains, 40 to 80 brokers, 400 consumers and 400 RPs. Currently ten different resource types with fixed criticality rates are considered for the purpose of simplicity; however, this can be easily extended without losing generality.

Given the total number of requests to be generated and the mean of the distribution, the generation of number of requests to be issued during the $i^{th}$ time instant is according to a Poisson distribution. Here, 1000 requests with the mean value of 30 are used. The probability mass function for the Poisson process is given by Equation (4.11)

$$P(X = r / T = t) = (e^{\lambda t} * (\lambda t)^r) / r!$$

(4.11)

Equation (4.9) gives the probability that $r$ events will occur during duration of $t$. The maximum probable number of requests to be generated at the time instance $i$ ($X$) may be found by computing the total number of probable requests for a time duration of $i$ ($R_i$) and subtracting that from the accumulated sum of number of requests generated till the last instance ($\sum R_j$, $j=1$ to $i-1$). Thus, $X = R_i - \sum R_j$ where $j=1$ to $i-1$, $R_r=r$ for which $P(X = r, T = i)$ is maximum. The concise flow of simulation is given below.

4.2.5.1 The Consumer-Simulator module

This module simulates the consumer requests that are sent to the broker. This logic is handled with multiple threads so that multiple consumers’ requests may be spawned off simultaneously. The simulator algorithm is given below.
While (true) {

1. A Request object is generated on behalf of the consumer $U_c$.

2. Brokers maintain multiple queues, one for each resource type. Of the brokers associated with the consumer who has generated the request, the one with the minimum queue-length of the requested resource type is identified. The trust-index of the consumer is verified and the request is sent via a socket connection to the chosen broker. The ‘Broker-Simulator’ module handles the request from this point onwards.

3. The consumer waits to receive the RP ID from the Broker.

4. The ‘Broker-Simulator’ module ‘selects’ an appropriate RP in accordance with the logic explained in the subsequent section. If no RP is available, the ‘Broker-Simulator’ returns a null, ‘Simulator’ exits; else a valid RP ID is returned.

5. After utilizing the resource of the RP, the consumer sends a feedback to the broker who recommended the RP. The broker, who is associated with the selected RP, also generates the feedback. If the consumer’s satisfaction-index deviates from the broker’s satisfaction-index by more than a set value, the consumer’s feedback is ignored as false and the broker’s feedback is considered for updating the trust-index of RP. Otherwise, both indices are consolidated for the computation of satisfaction-index of the present transaction.

6. Trust-indices of the broker who chooses the selected RP and the consumer involved in the current transaction are also updated based on the trust-worthiness of the current transaction (as given in section 4.2). }
4.2.5.2 The Broker-Simulator module

This module implements the behavior of the brokers. Multiple threads are used to simulate the existence of multiple independent brokers. The Broker simulator algorithm is given below.

a) All the brokers are continually waiting to accept requests.

b) When a broker receives a request, the queue length of the corresponding requested resource type of that broker is incremented.

c) The broker delegates the request to T top brokers chosen on the basis of their average trust-indices. The broker also checks if an appropriate RP is associated with itself.

d) Each of the T brokers checks if appropriate RP is associated with it and nominates the best RP that satisfies the consumer’s requirements.

e) The best RP is chosen from all these nominations based on the least number of pending requests. If there are more RPs with the same pending requests, the RP with the highest trust-index is chosen. In case of a tie again, the one with the lowest cost is selected. Thus, this order of filtering, first based on the consumer’s constraint, queue length at the RP, next on trust-index and finally based on the cost ensures priority for the selection of suitable RP.

After the utilization of the resource, the broker obtains the feedbacks and computes the trust-indices of the RP and the consumer. These
trust-indices are updated at the sites of all brokers associated with the RP and the consumer and broker’s trust-index is updated at standard forums.

The choice of the number of brokers is an important factor in the performance of the architecture. The graphs in this section clearly indicate that this should be a compromising function between the number of idle brokers and maximum queue lengths. The \((M/G/\infty)/(k/FIFO)\) queuing model is chosen. The average number of waiting consumers is computed using Equation (4.12)

Average number of waiting consumers

\[
\frac{n}{k} \cdot \lambda \cdot E(T) + \frac{(n \cdot \lambda)^2}{2} (V(T) + E^2(T)) \left(1 - \frac{n}{k} \cdot \lambda \cdot E(T)\right)
\]

Here n is number of consumers, k is number of brokers, \(\lambda\) is the mean number of requests generated by a consumer, T is the discrete random variable denoting the resource service time, E(T) and V(T) are the mean and standard deviation of the resource service time. This formula has been derived from the Pollaczek-Khintchine relation. A similar formula can be derived for computing the number of idle brokers as a function of the number of brokers. The optimum number of brokers can be computed, as a compromising function of these two values giving desired weights to each of them.

4.2.5.3 Results of the simulation

Cost-loss to consumers is plotted against the number of transactions, for the cases of selection of RP based on trust, trust with different threshold values and the case without trust. Cost-loss is defined as the fraction of cost incurred by the consumer for unsatisfactory service. If m
is the amount paid by the consumer for the resource of an RP and 0.3 is the satisfaction-index given by the consumer after the transaction, cost-loss to the consumer is given by \((1-0.3) \times m\). Cost-loss for the case of selection of RP without using trust-index is more than the cases where the trust-index forms the basis of selection is shown in Figure 4.1. As expected, cost-loss decreases as the threshold value for the trust-index is increased. Higher the threshold value, stricter the selection process, more emphasis on trust-index for the selection of suitable RP, less cost-loss for the consumer. This indicates the robustness of the use of trust-index in the selection process.

Use of broker’s feedback, in addition to the consumer’s feedback for the computation of trust-index of RP, has a positive impact on choosing a ‘trust-worthy’ RP. This results in reduced cost-loss (Figure 4.2), as compared to the case where consumer’s feedback alone is considered for the computation of trust-index.

In Figure 4.3, the maximum queue length of any broker at different time instances is plotted against time, varying the number of brokers. It can be seen that the waiting time or the maximum queue length decreases as the number of brokers increases. This suggests that the higher the number of brokers the better. The number of idle brokers at different time instances is plotted against time for different number of brokers as shown in Figure 4.4. This graph suggests to decrease the number of brokers since number of idle brokers is proportional to the number of brokers. Such competing requirement suggests that a compromising function may be used to arrive at an optimal value for the selection of number of brokers. The cost-loss graph (Figure 4.1), graphs for cost-loss with feedback from the brokers and the consumer (Figure 4.2), analysis of maximum-queue length (Figure 4.3) and idle-broker (Figure 4.4) are plotted with 10% malicious RP.
Figure 4.1 Cost-Loss Performance Graph

In the case of failure of any broker, all the resources supported by that broker are also supported by at least one more broker. This is because each entity (consumer or RP) is associated with more than one broker. Hence all the information available with the failed broker is indeed available with at least one other broker in a consistent manner. There will not be any loss of precious data. However, as expected, there may be performance degradation because of longer queues at other broker sites in case of failure of any other broker.
Figure 4.2 Analyses of Broker and Consumer Feedbacks for Cost-Loss

Figure 4.3 Waiting Time Performance Graph
The trust-indices of RP, consumers and brokers in multi-broker architecture without delegation are discussed in this section. The procedures used to compute satisfaction-indices of consumers about RP’s resources filter the dishonest feedback of consumers and prediction of genuine feedback in case of false feedback is also discussed here.

4.3.1 Computation of Satisfaction-Indices

The broker examines its database to find out the best ‘suitable’ trustworthy policy-matched RP available and notifies both the entities. The RP then supplies the resource to the consumer and the transaction takes place. At the end of the transaction, feedbacks (between -1 and 1) in the form of a satisfaction-indices are provided by the consumer ($S_i(U_c, RP)$) and the
associate broker \( S_i(B_{RP}, \text{RP}) \). The RP also provides a feedback about the consumer for the committed transaction to the associated broker.

The satisfaction-indices provided by the consumer \( U_c \) and the associated broker \( B_{RP} \) are consolidated into a single satisfaction-index for the particular transaction \((i^{th}\) transaction) as given below:

- Let the absolute difference between the feedback provided by the broker and the consumer \( D \) be \(| S_i(U_c, \text{RP}) - S_i(B_{RP}, \text{RP}) |\) and the allowable minimum set threshold difference value be \( T \) (0.05 is used as minimum threshold value).

- Case 1: If \( D > T \), then the feedback provided by the consumer is considered as false for the current transaction (since the broker is assumed to be a more trust-worthy entity). But, if the nature of consumer’s feedback is identified as genuine using the pre-evaluating-set, (as explained in section 4.3.2.1), only for this \( i^{th} \) transaction, a false feedback is provided by the consumer either wanted or by mistake. So, an appropriate true consumer’s feedback for this transaction is predicted using the broker’s feedback as explained in section 4.3.2.2. Then, an average of this predicted satisfaction-index \( S_i(U_c, \text{RP}) \) of consumer and satisfaction-index of broker \( S_i(B_{RP}, \text{RP}) \) is used as the consolidated satisfaction-index \( S_i(U_c, B_{RP}, \text{RP}) \) of \( i^{th} \) transaction.

- Case 2: If \( D > T \), then the feedback provided by the consumer is considered as false for the current transaction. If the nature of consumer’s feedback is also identified as dishonest using the pre-evaluating-set, then the satisfaction-index provided by the broker alone is considered as the satisfaction-index of the present transaction \( S_i(U_c, B_{RP}, \text{RP}) \), by ignoring the false feedback of consumer.
- Case 3: If (D < T), then the feedback provided by the consumer is considered as genuine one. If the nature of consumer’s feedback is also identified as genuine using the pre-evaluating-set, then an average of satisfaction-index \( (S_i(U_c, \text{RP})) \) of consumer and satisfaction-index of broker \( (S_i(B_{RP}, \text{RP})) \) is used as the satisfaction-index \( (S_i(U_c, B_{RP}, \text{RP})) \).

- Case 4: If (D < T), then the feedback provided by the consumer is considered as genuine one. If, however, the nature of consumer’s feedback is identified as dishonest using the pre-evaluating-set, then an average of satisfaction-index \( (S_i(U_c, \text{RP})) \) of consumer and satisfaction-index of broker \( (S_i(B_{RP}, \text{RP})) \) is used as the satisfaction-index \( (S_i(U_c, B_{RP}, \text{RP})) \). The consumer has given a genuine feedback for this transaction, though it is identified as dishonest.

This satisfaction-index \( (S_i(U_c, B_{RP}, \text{RP})) \) is used to update the trust-index of the RP maintained at the brokers’ site. This ensures that the behavior of the RP is updated regularly and properly at the brokers’ site.

The RP’s feedback on the consumer for the \( i^{th} \) transaction includes the percentage of utilization of assigned resource \( (S_1) \) and whether the payment commitment was honored or not \( (S_2) \) and whether the feedback about the resource of RP is provided or not \( (S_3) \). Then, the broker finds a satisfaction-index about the consumer as the weighted sum of \( S_1, S_2 \) and \( S_3 \). It is given by Equation (4.13) as

\[
S_i(B_{RP}, U_c) = \sum_{j=1}^{3}(\gamma_j^*S_j)
\]

(4.13)

Here, the value of \( S_1 \) as 1 if percentage of usage is less than equal to the assigned one and as -1 if percentage of usage is more than the assigned
one, $S_2$ as 1 if the payment had been made as committed, -1 if not, as a discrete quantity and $S_3$ as 1 if feedback is provided and -1 if not, as another discrete quantity. Equal weights ($\gamma_j$) are considered for the parameters of $S_j$. However, the problem yet to be considered is false feedback being submitted by the consumers. The following section deals with the strategies to resolve this problem.

### 4.3.2 Monitoring Consumer Feedback

Filtering of dishonest feedback and predicting an appropriate true feedback in case of a transaction with dishonest consumer’s feedback are discussed in the following paragraphs.

#### 4.3.2.1 Filtering of dishonest feedback

In our model, it is assumed that the brokers are trustworthy. Hence, feedback from a consumer is checked by brokers for genuineness. The way to identify false feedback is by trying to find the inconsistency between a rater's actual ratings and its rating habit. Dishonest ratings refer to those ratings which intentionally exaggerate or badmouth the facts.

A pre-evaluating set is used here to resolve the dishonest feedback problem. A pre-evaluating set (reference set) is a set of scenarios describing an RP's behavior in a transaction, including trustworthy, untrustworthy and in between behaviors (with trust-index varying from -1 to +1 in steps of 0.1). There is a pre-evaluating set in each domain in our model. An algorithm to filter out dishonest feedback using the Normal Test is given below:
Algorithm

- Let the consumer whose feedback to be rated be $C_1$. Let the feedbacks by consumer $C_1$ to the pre-evaluating set be $x_1 = \{ CR_0, CR_1, \ldots CR_{19} \}$

- Let the feedbacks by the broker to the pre-evaluating set be $x_2 = \{ BP_0, BP_1, \ldots BP_{19} \}$

- Compute the expectation values $E(x_1)$ and $E(x_2)$ of the sets $x_1$ and $x_2$, where $E(x_1)$ is the average of the values in set $x_1$ and $E(x_2)$ is the average of the values in set $x_2$.

- Compute the standard deviations of the set $x_1$ and $x_2$ as $S_1$ and $S_2$ using the Equation (4.14) and Equation (4.15), respectively, as follows.

$$S_1 = \sqrt{E(x_1^2) - [E(x_1)]^2} \quad (4.14)$$

$$S_2 = \sqrt{E(x_2^2) - [E(x_2)]^2} \quad (4.15)$$

- Compute the Z value using Equation (4.16) that decides whether a feedback is honest or dishonest as

$$Z = \frac{E(x_1) - E(x_2)}{(\frac{1}{n_1} + \frac{1}{n_2})\sqrt{(n_1s_1^2 + n_2s_2^2) / (n_1 + n_2 - 2)}} \quad (4.16)$$

- Here $n_1$ and $n_2$ are the number of entities that exist in the sets $x_1$ and $x_2$, respectively. If this $|Z|$ value > 1.96, then the feedback from the
consumer is considered as dishonest one as it deviates from the broker’s feedback by more than 5%. The nature of consumer’s feedback and thus the consumer, in turn, is identified as dishonest and updated into the database of the associated broker’s site.

### 4.3.2.2 Prediction of honest feedbacks

The false feedback is not considered in the computation of the trust-index of the RP. Hence, transactions which yield false ratings can be rendered invalid and trust-index of the participating RP after such transactions can be left unmodified. However, this can be unfair to the RP who had acted honestly in the transaction committed. There is a high probability that its trust-index could have been incremented if the participating consumer delivered a true feedback.

So, in cases where consumer’s false feedback is detected, the following method is implemented to predict an appropriate true feedback for the committed transaction, according to the broker’s feedback.

- Using the same expectation values and standard deviation formula from the previous algorithm, compute the regression line using Equation (4.17) as

\[
\frac{x_1 - E(x_1)}{s_1} = \frac{r(x_2 - E(x_2))}{s_2}
\]  

(4.17)

Here \( r \) is the correlation coefficient where

\[
r = \frac{E(x_1, x_2) - E(x_1)E(x_2)}{s_1s_2}
\]
Hence, by substituting the broker’s feedback (a satisfaction-index \( S_i(B_{RP}, RP) \)) in \( x_2 \), an appropriate true feedback \( x_1 \) (a satisfaction-index \( S_i(U_c, RP) \)) of a consumer can be obtained.

### 4.3.3 Trust-index Computation of Consumers

The associated brokers expect the consumer’s trust-index to be greater than a minimum set-value to process the consumer’s request. The trust-index of a new customer will be assigned with the minimum set-value (0.5) by the associated broker. Later, trust-index of the consumer is updated based on its deal in the current transaction, whether the payment is made as agreed, the feedback provided is genuine or not, etc.

It is the responsibility of the broker to identify the bad-mouthers and the ballot stuffers. Bad-mouthers are the consumers who purposely give a bad feedback even when the transaction is satisfactory, i.e., negative discrimination. Ballot stuffers are consumers who purposely give a good feedback even when the transaction is not satisfactory, i.e., positive discrimination.

Initially, the number of genuine and dishonest feedbacks of consumers compared to the broker’s feedbacks for the pre-evaluating set is obtained. If the difference between the consumer’s feedback and the broker’s feedback is more than a minimum set value, the consumer’s feedback is identified as dishonest. Otherwise, it is considered as genuine feedback. If the consumer’s feedbacks are identified to be false, the consumer will be punished. Its trust-index will be reduced by a value, \( R \), a reduction factor which is computed by Equation (4.18) as follows:

\[
R = \sum_{i=1}^{N} \left| \frac{S_i(U_c, RP) - S_i(B_{RP}, RP)}{T} \right| \quad (4.18)
\]
Here, $N$ and $T$ are the number of consumer’s false feedbacks and the total number of transactions committed with the consumer $U_c$ in the pre-evaluating set. $|(S_i(U_c, RP) - S_i(B_{RP}, RP))|$ is the absolute difference between the consumer’s feedback $(S_i(U_c, RP))$ and broker’s feedback $(S_i(B_{RP}, RP))$ about the RP’s resource for the $i^{th}$ RP in the pre-evaluation set.

And the consumer’s trust-index will be incremented as an incentive if its feedback is found to be genuine. We propose this increment factor, $I$, which is computed by Equation (4.19) as follows:

$$I = \frac{M}{T}$$

(4.19)

Here $M$ and $T$ are the number of consumer’s honest feedbacks and the total number of transactions committed with the consumer $U_c$ in the pre-evaluating set. The initial trust-index of the consumer $(TI_{\text{initial}}(U_c))$ is computed by Equation (4.20) as follows, based on the initial old trust-index of the consumer $TI_{\text{old}}(U_c)$ (=0.5), increment factor $(I)$ and decrement factor $(R)$.

$$TI_{\text{initial}}(U_c) = TI_{\text{old}}(U_c) + I - R$$

(4.20)

Trust-index of a trust-worthy consumer would increase with every transaction, whereas for an untrustworthy consumer, the trust-index would reduce as they transact with RP. If the trust-index of the consumer is less than a minimum set-value, the consumer is penalized by not processing the next three requests for further transactions. However, the trust-index of a new consumer is greater than the trust-index of untrustworthy (old) consumer due to its past untrustworthy transactions.
After every transaction, the consumer’s feedback is checked for its genuineness by the associated broker. For the honest feedback, the further increment factor is computed as same as in Equation (4.17) just by incrementing both M and T by 1.

For the dishonest feedback, the further reduction factor is computed using Equation (4.21) as follows

\[
R = \frac{R \times T + |S_c(U_c, RP) - S_c(B_{RP}, RP)|}{T + 1}
\]  

(4.21)

where \(|S_c(U_c, RP) - S_c(B_{RP}, RP)|\) is the absolute difference between the consumer’s feedback \((S_c(U_c, RP))\) and the broker’s feedback \((S_c(B_{RP}, RP))\) about the RP’s resource for the current transaction.

The trust-index of the consumer \(U_c\) for the current \(i\)th transaction is computed purely based on the associated broker’s satisfaction-index \((S_i(B_{RP}, U_c))\), is given using Equation (4.22) as follows:

\[
TI_i(U_c) = S_i(B_{RP}, U_c) 
\]  

(4.22)

Here \(S_i(B_{RP}, U_c)\) is the satisfaction-index given by the broker about the consumer for the \(i\)th transaction as already explained.

The trust-index of the consumer is updated after every transaction based on the old trust-index of the consumer \((TI_{old}(U_c))\), trust-index of consumer for the current transaction \((TI_c(U_c))\) and updating factor \((UF)\). It is computed using Equation (4.23) as follows,

\[
TI(U_c) = \frac{(TI_{old}(U_c) \times T + TI_c(U_c))}{T + 1} + UF
\]  

(4.23)
Here UF= +1 if honest feedback is given for the current transaction and UF= -1 if dishonest feedback is given for the current transaction.

### 4.3.4 Trust-index Computation of RP

The trust-index of an RP is computed mainly based on the honest feedbacks of the consumers along with the feedbacks of the brokers \((S_i (U_c, B_{RP}, RP))\), as given in section 4.3.2. A time stamp is included with each feedback for trust-index computation. The trust-index of the RP is the aggregation of the satisfaction-index of all past and recent transactions. This improves the robustness of the model. Other factors known as the criticality weight, \(CR_i\) and normalized cost \(C_{ni}\) of each transaction \(i\) may also be considered to further fine tune the computation of trust-indices as explained earlier. Thus, the trust-index of RP \((TI(RP))\) is given by Equation (4.24) as

\[
TI(RP) = \left[ \frac{\sum_{i=1}^{t_1} (S_i (U_c, B_{RP}, RP) \ast C_{ni} \ast CR_i)}{t_1} + \frac{\sum_{i=1}^{t_2} (S_i (U_c, B_{RP}, RP) \ast C_{ni} \ast CR_i)}{t_2} \right]
\]

(4.24)

Here, \(t_1\) is the total number of recent transactions after break point date; \(t_2\) is the total number of past transactions before break point date.

### 4.3.5 Trust-index Computation of Brokers

The broker’s trust-index is updated after every registration of an RP, deregistration of RP, or every time the trust-index of an RP, associated with this broker is updated. Trust-index is computed as the average of the
trust-indices of the RP that belong to the broker and it is given by Equation (4.25) as

\[
TI(B_{RP}) = \frac{\sum_{i=1}^{N} TI(RP_i)}{N}
\]  

(4.25)

Here, \(N\) is the number of RP, associated with the broker \(B_{RP}\), \(TI(RP_i)\) is the trust-index of the \(i^{th}\) RP, associated with the broker \(B_{RP}\).

### 4.3.6 Simulation and Results

This architecture is simulated with 10 domains, 50 brokers and 500 consumers and 500 RPs. For the purpose of simplicity, ten different resource types with fixed criticality rates are considered; however, this may be easily extended without losing generality. Given the total number of requests to be generated and the mean of the distribution, the generation of number of requests to be issued during the \(i^{th}\) time instant is according to a Poisson distribution. Here, we used 15000 requests with the mean value of 30.

Cost-loss to consumers is plotted against the number of transactions in the following three scenarios - selection of RP based on trust, based on trust and policy and the selection without trust. Cost-loss is defined as the fraction of cost incurred by the consumer for unsatisfactory service. Cost-loss where the selection of RP is made without using trust-index is more than the cases where the trust-index forms the basis of selection and cost-loss is more for selection of RP based on reputation alone compared with cases where selection is based on reputation and policy matched trust is shown in Figure 4.5. This shows the importance of reputation and policy matched trust-index for the selection of suitable RP and minimizing the cost-loss for the consumer community. This indicates the robustness of the use of trust-index in the selection process.
The number of exchanged messages required for selection of trustworthy RP is plotted against the number of transactions in the following scenarios - selection of RP in Multi-Broker Architecture with delegation of requests using \( r_f = 2 \) and \( r_f = 4 \) and the selection of RP without delegation of requests. It has been found that the number of messages for with delegation scenario is far higher than the delegation-less scenario (Figure 4.6). In the Multi-Broker Architecture without delegation, the number of messages drastically reduced.

![Cost loss graph](image)

**Figure 4.5 Comparison of cost-loss performance graph**
Figure 4.6 Comparison of number of messages exchanged graph

The broker’s trust-index in the presence of vicious RP has been plotted against the number of transactions in Figure 4.7. It is seen that with the forced deregistration concept the broker’s trust-index is almost maintained since the untrustworthy RPs are deregistered. The dynamically varying RP’s trust-index is plotted against the number of transactions in Figure 4.8. Here we have shown the differences in the RP’s trust-index with three cases: i) RP’s trust-index if false feedbacks are not detected and eliminated, ii) RP’s trust-index if false feedbacks are eliminated since the resulting transaction has rendered invalid feedback and iii) RP’s trust-index if an appropriate feedback of consumer is predicted even with consumer’s false feedback (highly close to the true feedback, had there been one). And the observation made is that the proposed prediction method is efficient to safeguard the RP’s trust-index and award the RP with its most deserving trust-index even when the consumer acts maliciously. Slight variation with predicted feedbacks is smoothened when the number of transactions increased.
Figure 4.7 Effect of forced deregistration of RPs

Figure 4.8 Effect of predicted feedbacks on an RP’s trust-index

4.4 COMPUTATION OF TRUST-INDICES IN HIERARCHICAL BROKER ARCHITECTURE
Since the unbiased, trustworthy RRAs are involved in the resource selection in the proposed Hierarchical Broker Architecture, the trust-indices of RP, consumer and resource brokers alone are calculated. The genuineness of the feedbacks (satisfaction-indices) is checked by the associated RRA using pre-evaluation set (reference set), the trust-indices of the concerned entities are updated accordingly. All these procedures are discussed in this section.

4.4.1 Computation of Satisfaction-Indices

The consumer utilizes the resources of the selected RP and at the end of the transaction provides a feedback to the RRA. This feedback includes the consumer’s ID, RP’s ID, the resource type, the satisfaction-index, transaction cost and the date of completion of the transaction. The satisfaction-index indicates the level of satisfaction achieved by the consumer on the resource provided by the RP. A value between -1 and 1 is used, with 1.0 indicating 100% satisfaction and -1 indicating the lowest satisfaction. The RRA collects the satisfaction-indices from both the consumer and the broker, who is associated with selected RP and rated these satisfaction-indices after checking its genuineness.

The satisfaction-index $S_i(U_c, RP)$ of the consumer ($U_c$) and the satisfaction-index $S_i(B_{RP}, RP)$ of the broker $B_{RP}$ about the RP’s resource are provided in the range from -1 to 1 for the $i^{th}$ transaction are computed as in section 4.2.

The satisfaction-indices provided by the consumer $U_c$ and the associated broker $B_{RP}$ are consolidated into a single satisfaction-index for the particular transaction ($i^{th}$ transaction) as given below:
Case 1: If the nature of consumer’s and broker’s rated feedbacks are identified as genuine using the reference set, then an average of rated satisfaction-index \( S_i (U_c, RP) \) of consumer and rated satisfaction-index of broker \( S_i (B_{RP}, RP) \) is used as the satisfaction-index \( S_i (U_c, B_{RP}, RP) \) about RP’s resource in the \( i^{th} \) transaction.

Case 2: If the nature of broker’s feedback is identified as dishonest and consumer’s feedback is a genuine one using the reference set, then the rated satisfaction-index \( S_i (U_c, RP) \) of consumer alone is assigned as the satisfaction-index \( S_i (U_c, B_{RP}, RP) \). The trust-index of broker is reduced because of its biasness.

Case 3: If the nature of consumer’s feedback is identified as dishonest and broker’s feedback as genuine using the reference set, then the RRA itself will generate the feedback about the RP’s resource for the \( i^{th} \) transaction and assigned as the satisfaction-index \( S_i (U_c, B_{RP}, RP) \). The broker may bias in the current transaction, though it is identified as genuine one.

Case 4: If the nature of consumer’s and broker’s feedbacks are identified as dishonest using the reference set, then the RRA itself will generate the feedback about the RP’s resource for the \( i^{th} \) transaction and assigned as the satisfaction-index \( S_i (U_c, B_{RP}, RP) \).

For the cases 3 and 4, the RRA will generate the feedback about the transaction since the concerned entities are not genuine. For the cases 1 and 2, the RRA will not generate the feedback about the transaction since any or both of the concerned entities are genuine and also it reduces its (own) load. The satisfaction-index \( S_i (U_c, B_{RP}, RP) \) formed at the RRA’s site, is used to
update the trust-index of the RP maintained at the brokers’ site. This ensures that the behavior of the RP is updated regularly and properly at the brokers’ site. Till the next reference set formation period, the genuineness of the feedbacks of malicious entities (identified as dishonest consumer and / or dishonest biased broker) are compared with the actual feedback considered by RRA. Based on this, the RRA, a party with no vested interest in the transactions, updates the similar and dissimilar feedbacks as its own feedback reference in its database and updates the trust-indices of the biased entities. During each dishonest feedback, notification is sent by the RRA for the malicious entities to improve its performance.

This feedback (either the consolidated feedback from the consumer and the broker or the consumer’s feedback alone in case of biased broker, or RRA’s feedback in case of both consumer’s and broker’s dishonest feedbacks) is propagated to the broker, who is associated with the selected RP. This is required to update the trust-index of that RP at the broker’s site. A periodic audit by the respective RRA’s ensures that the broker uses the feedback recommended by the RRA to compute the trust-index of the RP. This feedback is informed to the corresponding RP to ensure proving better resources in future in order to raise its trustworthiness. This feedback along with the cost of that transaction forms the transaction information for that transaction, and is maintained at the associated brokers’ site. The cost of the transaction is computed as explained earlier in Multi-Broker Architecture. The details of computing and updating the trust-indices are explained later in this section. However, the problem yet to be considered is false feedbacks being submitted by the consumers and brokers.

4.4.2 Monitoring Feedbacks
Filtering of dishonest feedback and predicting an appropriate true feedback in case of a transaction with dishonest entities’ feedbacks are discussed below.

4.4.2.1 Filtering of dishonest feedback

In this model, the RRAs are assumed to be trustworthy like DNS. Hence, the feedbacks from consumer or broker are checked by RRA for genuineness. The way to identify false feedback is by trying to find the inconsistency between a rater's actual ratings and its rating habit. By dishonest ratings, this model means those ratings which intentionally exaggerate or badmouth the facts.

A model of reference set is used here. A reference set is a set of RP describing an RP's behavior in a transaction, including trustworthy, untrustworthy and in between behaviors (with trust-index varying from -1 to 1 in steps of 0.1). There is a reference set in each domain in this model. An algorithm to filter out dishonest feedbacks is given below

Algorithm

- Let an entity (consumer or broker) whose feedback to be rated be E₁. Let the ratings (feedbacks) by the entity E₁ to the pre-evaluating set be \( x₁ = \{ER₀, ER₁, \ldots ER₁₀\} \)

- Let the ratings by the RRA to the reference set be \( x₂ = \{RP₀, RP₁, \ldots RP₁₀\} \)
Compute the expectation values $E(x_1)$ and $E(x_2)$ of the sets $x_1$ and $x_2$, where $E(x_1)$ is the average of the values in set $x_1$ and $E(x_2)$ is the average of the values in set $x_2$.

Compute the standard deviations of the set $x_1$ and $x_2$ as $S_1$ and $S_2$ using Equation (4.26) and Equation (4.27) as

$$S_1 = \sqrt{E(x_1^2) - [E(x_1)]^2}$$

(4.26)

$$S_2 = \sqrt{E(x_2^2) - [E(x_2)]^2}$$

(4.27)

Compute the $Z$ value using Equation (4.28), that decides whether a feedback is honest or dishonest as

$$Z = \frac{E(x_1) - E(x_2)}{\left(\frac{1}{n_1} + \frac{1}{n_2}\right)\sqrt{\left(n_1s_1^2 + n_2s_2^2\right)/\left(n_1 + n_2 - 2\right)}}$$

(4.28)

Here, $n_1$ and $n_2$ are the number of entities that exist in the sets $x_1$ and $x_2$, respectively. If this $Z$ value $> 1.96$, then the feedback from the entity is considered as dishonest as it deviates from the RRA’s feedback by more than 5%. The nature of the entity’s feedback is identified as dishonest and updated into the database of the associated RRA’s site. It should be explicitly stated that $Z$ follows normal distribution (using central limit theorem for large $n_1$ and $n_2$) and thus the threshold 1.96 can be obtained using normal distribution tables for a deviation 0.05.

### 4.4.2.2 Prediction of honest feedbacks

The false feedback is not considered in the computation of the trust-index of RP. Hence, transactions which yield false ratings can be rendered
invalid and trust-index of the participating RP after such transactions can be left unmodified. However, this can be unfair to the RP who had acted honestly in the transaction committed. There is a high probability that its trust-index could have been incremented if the participating, consumer delivered a true feedback.

So, in cases where false feedback providers are identified by comparing transaction feedback with reference feedback in reference set at the registration time or using newly formed reference set at later stage, the RRA itself will generate an actual true feedback for the committed transaction till the next reference set period, called checking period, as proposed in this model.

4.4.3 Trust-index Computation of Consumers and Brokers

Feedbacks about each of the transactions are collected from the entities to compare with RRA’s (own) feedback and to check for the genuineness during the checking period. Let the checking period be T. The details like number of Correct Feedbacks (CF, taking its feedback as reference), number of False Feedbacks (FF) and Total number of Feedbacks (TF) performed during that T period (including the reference set transactions since anyway RRA also generates feedbacks for reference set entities) are stored at the RRA’s site to update the trust-index of involved entities. Even though the nature of entities’ feedbacks (consumers and brokers) are identified as genuine, the RRA will randomly generate its own feedback about some transactions till the next reference set formation (but considered all transactions in the reference set), as said above for the dishonest feedback cases. And, then stores similar details at its (the RRA) site to update the trust-index of concerned entities. This is done to avoid malicious feedback providing behavior once the entity is identified as genuine feedback provider.
using reference set during the registration time or later. Algorithm is briefed below.

\[
\text{TF} = \text{CF} = \text{FF} = 0
\]

While T period do

If entity’s feedback deviates from the RRA’s feedback by more than 5%

\[
\text{FF} = \text{FF} + 1
\]

Else

\[
\text{CF} = \text{CF} + 1;
\]

\[
\text{TF} = \text{TF} + 1
\]

End while

After T period (i.e., as reference set formation period), the trust-indices of entities (TI(E), TI(E_c) if an entity is consumer, TI( E_b) in case of broker) are updated using Equation (4.29) as

\[
\text{TI}(E) = \frac{(\text{TI}_{\text{old}}(E) + \frac{\text{CF}}{\text{TF}})}{2}
\]  (4.29)

Here, TI_{old}(E) is old trust-index of the entity. Some sets of scenarios are tabulated below in Table 4.1, considering TF as 100 during T period.

**Table 4.1 Trust values of entities**

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>Old trust-index</th>
<th>CF</th>
<th>FF</th>
<th>New Updated trust-index</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>1</td>
<td>100</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2.</td>
<td>1</td>
<td>80</td>
<td>20</td>
<td>0.9</td>
</tr>
<tr>
<td>3.</td>
<td>1</td>
<td>50</td>
<td>50</td>
<td>0.75</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>4.</td>
<td>1</td>
<td>30</td>
<td>70</td>
<td>0.65</td>
</tr>
<tr>
<td>5.</td>
<td>1</td>
<td>10</td>
<td>90</td>
<td>0.55</td>
</tr>
<tr>
<td>6.</td>
<td>1</td>
<td>0</td>
<td>100</td>
<td>0.5</td>
</tr>
<tr>
<td>7.</td>
<td>0.5</td>
<td>100</td>
<td>0</td>
<td>0.75</td>
</tr>
<tr>
<td>8.</td>
<td>0.5</td>
<td>80</td>
<td>20</td>
<td>0.65</td>
</tr>
<tr>
<td>9.</td>
<td>0.5</td>
<td>50</td>
<td>50</td>
<td>0.5</td>
</tr>
<tr>
<td>10.</td>
<td>0.5</td>
<td>30</td>
<td>70</td>
<td>0.4</td>
</tr>
<tr>
<td>11.</td>
<td>0.5</td>
<td>10</td>
<td>90</td>
<td>0.3</td>
</tr>
<tr>
<td>12.</td>
<td>0.5</td>
<td>0</td>
<td>100</td>
<td>0.25</td>
</tr>
<tr>
<td>13.</td>
<td>0.1</td>
<td>80</td>
<td>20</td>
<td>0.45</td>
</tr>
<tr>
<td>14.</td>
<td>0.1</td>
<td>50</td>
<td>50</td>
<td>0.3</td>
</tr>
<tr>
<td>15.</td>
<td>0.1</td>
<td>30</td>
<td>70</td>
<td>0.2</td>
</tr>
<tr>
<td>16.</td>
<td>0.1</td>
<td>0</td>
<td>100</td>
<td>0.05</td>
</tr>
</tbody>
</table>

From the above scenarios, during the checking period T, if all feedbacks are identified as genuine using RRA’s feedback as reference, trust-index of entity is increased as incentive (except for the first scenario, as such there is no problem for this case). Similarly, in case of some false feedbacks, the trust-index of entities is reduced as punishment.

The trust-index computed is for the consumer entity. So the trust-index of the consumer entity (TI(E_c)) is given by Equation (4.30) as

\[
TI \ (E_c) = \frac{(TI_{old} \ (E_c) + CF)}{2} \tag{4.30}
\]

Here, TI_{old}(E_c) is the old trust-index of consumer entity. The broker’s trust-index is computed not only based on the mean of the ratio of Correct Feedbacks with the Total number of Feedbacks (i.e., \( \frac{CF}{TF} \)) with the old trust-index of the consumer but and also based on the average of trust-indices of associated RP. It reflects not only the broker’s nature of the feedback behavior but also the nature of the resource services (satisfactory
service or not) provided by the associated RP. So the trust-index of broker entity (TI (E_b)) is given by Equation (4.31) as

\[
TI (E_b) = \left[ TI_{old} (E_b) + \frac{\sum_{i=1}^{N} TI_i (RP) / N + CF / TF}{2} \right] / 2
\]  

(4.31)

Here, TI_{old}(E_b) is the old trust-index of broker entity, N is the number of associated RP, TI_i(RP) is the trust-indices of associated i^{th} RP. Computation of RP is discussed in the following paragraphs.

### 4.4.4 Trust-Index Computation of RP

The trust-index of an RP is computed mainly based on the honest feedbacks about the RP’s resource (S_i(U_c, B_{RP}, RP)). A time stamp is used to differentiate past transactions from the more recent ones. The broker decides the breakpoint date based on the application. This breakpoint date decides a transaction as being recent or past. The transactions graded as recent are assigned a higher weight (\(\alpha\)) and past transactions a lower weight (\(\beta\)), in the computation of trust-index of the RP, with \((\alpha + \beta) = 1\). The trust-index of the RP is the aggregation of the satisfaction-index of all past and recent transactions. This improves the robustness of the model. Other factors such as the normalized cost (C_{ni}) and criticality, and CR_i of each transaction are used to further refine the computation of trust-index of the RP. Higher weight is assigned to transactions of higher cost than the cheaper transactions. Higher weight is assigned to transactions of more critical resources than the less critical one. Thus, the trust-index of RP (TI (RP)) is given by Equation (4.32) as
Here \( t_1 \) is the total number of recent transactions and \( t_2 \) is the total number of past transactions.

### 4.4.5 Simulation and Results

This model is simulated with 10 RRAs, 50 brokers, 500 consumers and 500 RPs. Currently ten different resource types are considered for the purpose of simplicity; however this can be easily extended without losing generality. The total number of requests to be generated and the mean of the distribution, the generation of number of requests to be issued during the \( i^{th} \) time instant is according to a Poisson distribution. Here, we used 1000 requests with the mean value of 30. The various graph results obtained, namely the job success rate graph, the cost loss graph and the effect of predicted actual true feedback in RP’s trust-index computation are shown below.

Job success rate is plotted in Figure 4.9 against the percentage of the malicious brokers with different threshold values with trust. In the proposed Hierarchical Three-tier Broker Architecture, the job success rate is much higher and stable compared to the earlier Multi-Broker Architecture with increasing the percentage of malicious brokers. This is because the biased brokers who provide biased feedbacks are identified and only the genuine feedback (either the genuine consumer’s feedback or RRA’s feedback itself) is considered as the satisfaction-index of the present transaction. Hence, the job success rate remains unaffected despite the
increase in the percentage of malicious brokers. Also as the threshold value increases, job success rate increases. So higher the threshold, stricter is the selection process and higher is the job success rate.

The cost-loss to consumers is plotted in Figure 4.10 against the number of transactions for the cases of selection of RP based on trust, with different threshold values and the case without trust-index in the proposed Hierarchical Broker Architecture as well as with distributed Multi-Broker Architecture. Cost-loss is defined as the fraction of cost incurred by the consumer for the unsatisfactory resource service. The cost-loss for the case of selection of RP without using trust-index is more than the cases where the trust-index forms the basis of selection. In the distributed Multi-Broker model, the cost-loss is much lower than the case without trust but it is still much higher than the cost-loss in Hierarchical Broker Architecture. This is so because the effect of malicious brokers and biased brokers is handled in this model. Hence, the consumer is guaranteed a trust worthy and good quality service by the RRA. Also for various threshold values, the cost loss varies. Higher the threshold value, the stricter selection process, more emphasis on the trust-index for the selection of suitable RP and less the cost-loss for the consumer. This indicates the robustness of the use of trust-index in selection process.
Cost-loss to consumers is plotted against the number of transactions in the following three scenarios - selection of RP based on trust alone, based on trust with policy-match and the selection without trust. Among the selection of RP based on trust-index, cost-loss where the selection of RP is made without using trust-index is more than the cases where the trust-index forms the basis of selection. And cost-loss is more for selection of RP based on reputation alone compared with cases where selection is based on reputation and matched-policy, is shown in Figure 4.11. This shows the importance of reputation and matched-policy for the selection of suitable RP that minimizes the cost-loss for the consumer community.
Figure 4.10 Cost-loss performance graph

The dynamically varying RP’s trust-index is plotted against the number of transactions in Figure 4.12. Here we have shown the differences in the RP’s trust-index with three cases: i) RP’s trust-index if false feedbacks are not detected and eliminated, ii) RP’s trust-index if false feedbacks are eliminated since the resulting transaction has rendered invalid feedback and iii) RP’s trust-index if an appropriate true feedback of consumer is predicted even with consumer’s false feedback. The observation made is that the proposed prediction method is efficient to safeguard the RP’s trust-index and award the RP with its most deserving trust-index even when the consumer acts maliciously. Slight variation with predicted feedbacks is smoothened when the number of transactions increased.
**Figure 4.11** Comparison of cost-loss of consumer with trust alone; trust with policy match and without trust

**Figure 4.12** Effect of predicted RRA true feedback on RP’s trust-index
4.5 TRUST BASED ON VERIFICATION OF COMPUTATIONS

A very important issue in software security, namely integrity of computation on a remote, un-trusted host in grid environment, is considered. The existing scheme for result verification in a distributed grid system is replication. This scheme suffers from collusion and redundant execution of the jobs leading to high resource consumption. In order to combat these issues, a scheme called quizzing is proposed in the Hierarchical Broker Architecture. Some tasks of given jobs are considered as quiz queries instead of separate set of quiz queries in order to avoid the differentiation of jobs from quiz queries at the RP’s site. This leads to better resource utilization compared to the simple quiz-based result verification scheme. The reputation based trustworthiness of RP is computed based on the quiz-based result verification scheme in this thesis.

4.5.1 Replication Problem

Replication scheme is used for verification of completed jobs’ results. Here, the broker gets the job from the consumer’s request. For job’s result verification, the broker creates k replicas of the same job and sends the copies to k+1 RPs that are selected based on their previous trust values. When all the chosen RPs have finished the jobs and returned the results, the broker compares the results and if more than half the numbers of hosts agree on the same result, it accepts that result as the correct one.

This method suffers from two drawbacks. First, in the heterogeneous grid environment, there may be malicious RPs which discuss among themselves and purposely provide the same bad results, thus resulting in collusion. The second problem in the replica concept is that each replica of the job consumes resources resulting in high resource consumption.
4.5.2 Quizzing concept

To overcome these issues of replication scheme, quizzing scheme has been proposed whose basic idea is to attach queries to each of the job. A set of quiz queries for that resource along with their results is determined by the consumer and maintained in the associated broker’s database. Whenever a consumer requests for a job to the broker, the job is sent along with the selected set of queries to the selected RP. The RP, which gives correct result to the set of queries, is considered to be non-malicious. But, this method also has a drawback because of the queries, which are completely different from the required consumer’s job. The malicious RP identifies the queries easily and gives the correct results only for the queries and not to actual job. To eliminate the above problem, the quizzing scheme, which is enhanced with indistinguishable set of queries from the given job, is used in this thesis. Here, the queries are nothing but the part of the job (tasks) so that they are not identified by the malicious RP. The RP computes the result for both the job and the quiz queries (part of job, that is, tasks) and sends it to the broker. The broker accepts the job only if the quiz queries (job’s tasks) are answered correctly, else discards it.

4.5.3 Proposed Architecture

Biased brokers are the greatest problem in a dynamic, heterogeneous grid environment. They may introduce malicious RP with erroneous jobs’ results. In order to consider the property of unbiased nature and trustworthiness, Hierarchical Broker Architecture is proposed with a neutral entity called RRA.
4.5.3.1 Schemes of replication and quizzing

If the replication scheme is introduced in the Multi-Broker Architecture, the broker selects k RP based on the previous trust values and to the level the RP can bear the cost given by the consumer. It then creates k replicas of the task and sends the copies to k+1 RPs that are selected. When all the chosen RPs have finished the job and returned the results, the broker compares the results and if more than half the number of hosts agrees on the same result, it accepts that result as the correct one and increases the trust values of the corresponding RP. The algorithm for replication scheme is given below.

1. The broker replicates the consumer’s job and sends to k+1 RP where k is the
   Replication factor
2. If (number of majority results > (k + 1)/2)
   Then
   Accept the majority result;
   Else
   Reject the whole task, reschedule it later;
3. If (result of job is accepted)
   Then
   If (RP gave most common result)
   Then
   Increase trust values for those RP;
   Else
   Decrease trust values for those RP

The problem with this approach is that for giving the majority result, the different RPs may collude among them and give the same result,
which may be taken to be correct. The second problem is that we replicate a single job to more RPs and increase the resource consumption of RP.

So, a quizzing scheme has been proposed. Here, when a consumer requests for a resource with the broker, a set of quiz queries are identified and computed by the consumers and stored in the associated broker's database along with their results. When a resource is requested for any consumer's job, the broker selects an RP that satisfies the consumer’s request and also has the higher trust-index. The broker then sends the consumer’s job along with the appropriate quiz queries to the selected trustworthy RP. After the completion of transaction, based on the percentage of quiz correctness, the trust value of the RP is updated at the associated broker’s site. Here, only a single RP is selected, so there is no possibility of collusion problem and unnecessary resource consumption at multiple RPs’ sites. The algorithm of quizzing scheme for result verification is given below.

While (job queue is not empty)

1. Selection
   a. Choose an RP who satisfies the consumer’s request based on its resource functionality, trust value and queue length.
   b. Select and insert x quiz queries along with the consumer’s job and send to the selected trustworthy RP.

2. Verification of quiz results and updating the trust values of RP
   a. Upon receiving results from the selected RP, check the results for quiz queries of consumer’s job
      If (the result matches with the stored results in the database at the broker’s site)
Then
  Accept the results of that job
Else
  Reject the job in the queue, reschedule it later
b. Updating the trust value of the RP,
   If (job’s results are accepted)
   Then
       Increase the trust value of the RP.
   Else
       Decrease the trust value of the RP based on the percentage of quiz correctness

4.5.3.2 Working principle

Since the grid environment may contain malicious RP, the results provided by the RP should be verified under some process. In order to provide result verification for resources and considering the biased brokers problem in the grid environment, the quizzing concept has been introduced in the Hierarchical Broker Architecture so that the malicious RP can be figured out by their results. In the Hierarchical Broker Architecture discussed in section 3.5, the replication and quizzing schemes are plugged into the RRA’s site in this secured architecture for the performance comparison.

When a consumer submits a request of the job to the any of the interested RRA, the RRA selects an appropriate trustworthy RP by referring its database from the current nominated set of RP who satisfy completely for consumer job’s functionality with other constraints and having the higher trust value with lesser queue length, from various brokers. In case of quizzing scheme, a set of suitable indistinguishable quiz queries are selected from the database of the RRA. The quiz queries are the tasks of consumer’s job
submitted by the consumers in the consumer’s request which are stored in RRA’s site. The results of the quiz queries are computed by the consumer themselves and submitted to RRA along with their queries. The RRA attaches the quiz queries along with the consumer’s job and this quiz queries would be tasks of consumer’s job. The RP cannot identify the quiz queries since the attached quiz queries are the part of the job itself. It then sent as the job with quiz queries to the selected RP through the associated broker.

After the completion of job, the RP returns the results of the job along with the answers for the quiz queries. The results of quiz queries are verified at the RRA site. The RRA compares its results for the quiz queries from the database with the results computed by the RP. The job completion is accepted if the results of the quiz queries are correct else the job is discarded and the process is rescheduled later. It then based on verification of the quiz queries’ results; satisfaction about the RP’s resource is generated and sent to the broker of the selected RP to update the trust value of RP. Based on the updates of RP’s trust, the trust of the associated broker’s trust is also updated at the RRA’s site.

Since a part of job is provided as quiz queries instead of a separate set of queries, wastage of resource consumption is eliminated further. Also, the malicious RP cannot differentiate the quiz and the job to produce correct results only to quiz not to the entire job. The selected RP finishes the job with quiz and transmits results of job to the RRA through brokers which then hand it over to the consumer. It receives cost of resources from the consumer and pays the brokerage charges to the broker.

In case of replication scheme, a set of trustworthy RP (k RP) are chosen from the database at the RRA site instead of single RP and forward the consumer’s request to k RPs through their respective brokers. A thread in
RRA waits to receive the results of all k RPs through their respective brokers. As mentioned in the algorithm above, if \((k+1/2)\) received results are similar, the result is agreed as correct one and sent to the consumer. Else the same process is repeated starting from the selection of another k RP for the consumer’s request.

### 4.5.4 Trust Value Computations

After the completion of consumer’s job, the trust-index of RP is calculated based on the percentage of satisfactory service of the RP at the RRA site. In case of quiz-based scheme, the percentage of satisfactory service is based on the percentage of correctness of associated quiz queries from that RP. That recommended trust-index is sent to the associated broker for updating the new trust-index of RP and audited periodically also. The computation of trust values of brokers and consumers is similar to the procedure explained in section 4.4.

After the completion of transaction, the associated RRA will send an agent to update the trust-index of RP with the updating factor \((U)\) through the associated broker as an incentive, given by Equation (4.33) as

\[
U = +P_c(RP)
\]  

(4.33)

Here, \(P_c(RP)\) is the percentage of the correctness of the quiz queries. If some percentage of results of quiz queries is wrong, updating factor \((U)\) is calculated using (8.2) as punishment is given by Equation (4.34),

\[
U = -(1 - P_c(RP))
\]  

(4.34)
So, the trust-index of RP is now updated at the associated broker’s site by sending an agent from the RRA after the completion of transaction is given by Equation (4.35) as

\[
TI(RP) = \frac{TI_{old}(RP) + U}{2}
\]  

(4.35)

Here, \(TI_{old}(RP)\) is the old trust value of the RP. Some test case scenarios are tabulated below in Table 4.2. The scenarios show that the trust of RP increases if the quiz queries are hundred percentile correct. Otherwise, the trust value of RP got reduced as a function of lack of percentage of correctness of quiz query results. If trust value is greater than one, it is assigned as one since the trust value varies from -1 to +1.

**Table 4.2 Test case scenarios for trust values of RP**

<table>
<thead>
<tr>
<th>Old trust</th>
<th>Percentage of correctness</th>
<th>New trust</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>0.5</td>
<td>0</td>
<td>-0.5</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.5</td>
<td>-0.5</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>0</td>
<td>0.75</td>
<td>-0.25</td>
</tr>
<tr>
<td>0.5</td>
<td>0.75</td>
<td>0.25</td>
</tr>
<tr>
<td>1</td>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0.5</td>
<td>1</td>
<td>1.5</td>
</tr>
<tr>
<td>0.75</td>
<td>1</td>
<td>1.75</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1.</td>
</tr>
</tbody>
</table>
4.5.5 Simulation and Results

The Hierarchical Broker Architecture is simulated with 10 RRAs, 50 brokers, 100 consumers and 100 RP and 1000 consumer requests. The number of resources consumed is plotted against number of transactions for replication (with k=2) and proposed quizzing schemes as in Figure 4.13. The resource consumption is more in replica scheme compared to quiz-based verification scheme as shown from this simulation.

![Comparison of replication and quizzing](image)

**Figure 4.13 Performance of resource consumption**

4.6 CONCLUSION

Trust value computations of various entities such as RP, consumers and brokers under various architectures such as Multi-Broker Architecture, Multi-Broker Architecture without delegation and Hierarchical Broker Architecture are discussed in this section. Filtering the false feedbacks of consumer using variance from broker’s feedbacks and providing punishment or incentives based on its behavior in current transaction is implemented in
Multi-Broker Trust Model. Filtering of dishonest feedbacks and predicting an appropriate true consumer’s feedback according to broker’s feedback used in the trust-index computation are implemented and its improved effects are shown in this model. The same identification of dishonest feedbacks and prediction of appropriate true feedback in case of malicious consumers and biased brokers in Hierarchical Broker Architecture is also discussed to safeguard the trust-index of RP.

The simulation results in Multi-Broker Architecture show that higher the threshold value, the stricter the selection process for an RP and hence better the performance of the grid with less the cost-loss for the consumer community. Use of broker’s feedback in addition to the consumer’s feedback for the computation of trust-index of RP has a positive impact on choosing a ‘trust-worthy’ RP. It is also seen that the waiting time and the maximum queue length decrease as the number of brokers in each domain increases. Thus, this proposed architecture is robust in terms of selection of suitable RP for a resource-request, avoiding malicious RP, with quicker selection process, reduction in cost-loss and achieving reliability through redundancy.

The simulated results in Multi-Broker Architecture without request delegation and with reputation and policy-based resource selection indicate the reduction of the workload of the broker and the number of message exchanged compared with trust-based selection and conventional resource selection. It is also seen that with the deregistration concept, the broker’s trust-index is almost maintained since the untrustworthy RP’s are de-registered in this model. And the observation made is that the proposed filtering and prediction method of malicious feedback in this model is efficient to safeguard the RP’s trust-index and award the RP with its utmost deserving trust-index even when the consumer acts maliciously.
The simulation results in Hierarchical Broker Architecture reveal that higher the threshold value, the stricter the selection process for an RP and hence better the performance of the grid with more job success rate with less cost-loss for the consumer community, compared to the Multi-Broker Architecture. The result of the proposed model with integrated reputation and policy based resource selection methodology shows further improvement in reduction of cost-loss to the consumers. Filtering of dishonest feedbacks and predicting a true RRA’s feedback for the trust-index computation of RP, is implemented and its improved effects are shown. Filtering the false feedbacks of consumer and / or brokers using RRA’s feedback as reference and updating of trust-index of involved entities as punishment or incentives based on the genuineness of the current transaction’s feedbacks, are also demonstrated.

The Hierarchical Broker Architecture is enhanced as a secured trust model with plug-ins of replication scheme and the proposed quiz-based scheme for result verification process at the RP’s site. The quiz model avoids the collusion among the malicious RP, thus making the grid environment a feasible one for the distributed consumers, compared to replica scheme. Unlike replication, quiz-based scheme does not produce any replicas of jobs during the execution and hence higher resource consumption overhead is reduced much. Trust value of RP is computed based on the percentage of the correctness of the computations at the RP’s site to reflect the trustworthiness of computations.