ISSN:0975-9646

Jayant Biswas et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 2875-2881

Mathematical Interpretation of the Maturity Levels in the Ticket Based Information Technology Service Management Model

Jayant Biswas
Technical Consultant – ERP
IBM, Noida, India

Snigdha Srivastava
Software Asset Manager
TMF, Noida, India

Gaurav Prakash
Technical Consultant - ERP
IBM, Noida, India

***Abstract*— this document introduces a mathematical formula that can be used by the prevalent ticket based service desk models to measure the turn around time of a ticket. It also aims to form a tool to measure the effectiveness of multiple models over each other in a given infrastructural setup. The document defines the three maturity levels for a ticket based basic service desk model. This formula and the maturity levels may be generalized to fit all types of Service Desk models.**

***Keywords*— Turn around Time (TAT), Communication Distance, Requester, Resolver, Information Distance, Communication Bounce, Information Bounce, Information Technology Service Management (ITSM), Necessary Bounce, Information Technology Information Library (ITIL).**

1. Introduction

In a ticket based Service Desk [1] model two defining factors are - turnaround time of resolution and quality of resolution. When a requester requests for a ticket, the service desk team is responsible to raise a quality ticket and assign it to the correct resolver group. Then the ticket may travel among multiple resolver groups or may go back to the requester to get more information. In most of the models majority of the communication happens via the service desk which facilitates in the communication by removing the gaps like language barrier and knowledge barrier. The Service Desk constitutes of people who are considered as "Jack of All Trades"; who are required to have some knowledge about all things in Information Technology [1]. In some models multiple service desks are involved with each focusing on a specific set of tickets or specific geographical locations. Among the two defining factors mentioned above, this paper concentrates on the turn around time and the factors that directly influence it. This paper derives a mathematical formula and a new unit of measure to calculate the Communication Distance which is described in the subsequent sections.

II. The Formula

Based on the classic formula of Time and distance, the formula for Turn around time (TAT) can be stated as:

$$\text{TAT} = \frac{\text{Distance between the Requester and the Resolver}}{\text{Speed of actions leading to Resolution}}$$

The distance between the requester and the resolver can be explained as the length of the path followed by a request from the requester to the resolver till the request is fulfilled. This length is not the geographical length between the parties involved, so conventional units of length and distance such as meter, kilometre etc. would not be appropriate here. As the communication happens over the internet, geographical distances do not matter but what matters is the path/direction that is followed and the number of times a path is revisited in order to move ahead. Shortest distance between the Requester and the Resolver would be a straight line with two nodes.

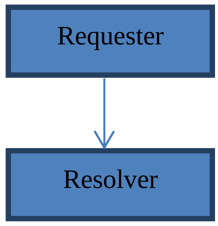


Fig 1.0 – Shortest Distance without Service Desk

In practice, there is at least one service desk in between the Requester and the Resolver making 3 nodes. The Service Desk shown in the Fig 2.0 could be located on premise or on the cloud [2], but both have same effect on the total distance travelled by a ticket from the requester to the resolver.

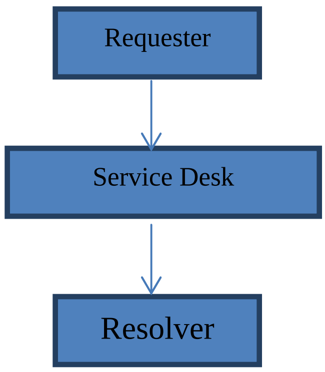


Fig 2.0 – Shortest Distance with Service Desk

So, we can see from the above figures that the distance in Fig 1.0 is 1 unit and the distance in Fig 2.0 is 2 units. In any further reference to the unit of distance, we would call it Communication distance or CD. So, the distance in Fig 2.0 is 2 CD. Now, let us check the factors that influence the distance between the Initial Requester and the Final Resolver.

Let us start with the simplest scenario where only one requester and only one resolver is involved and all the communication happens via the service desk. Referring the Fig. 2.0 the minimum communication distance is 2 CD. If the resolver needs more information from the requester then he would reach the requester via service desk. The requester would then respond via the service desk. So the same path is travelled again, back and forth.

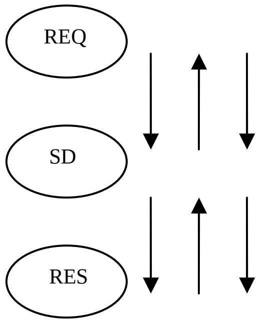


Fig 3.0 – Increase in Distance when more Information is required

The total distance becomes 6 CD if the path is traversed once due to need for more information. If the path is traversed twice, the total distance becomes 10 CD, till the request is resolved. We can see that in the above model 2 CD is necessary to fulfil the request but, 4 CD more is required in need of information, so, although this 4 CD adds to the total communication distance, this can be avoided if there is no lack of information at the first place.

In this paper we will widely use few terms. Readers are requested to go through the following paragraph carefully to understand the paper better.

1. *Communication Distance* – The distance travelled by the ticket from the initial requester to the final resolver where action is expected from each resolver and the ticket reaches every resolver in good quality and with complete information.
2. *Communication Bounce* – The event "Move the ticket ahead" is referred to as communication bounce which is a necessary bounce so that further action on the ticket can be taken. A Communication Bounce cause communication distance.
3. *Information Distance* – A distance travelled by the ticket when it is pushed back in need for more information is called the information distance. This can happen when the ticket contain incomplete information or is of bad quality.
4. *Information Bounce* – The event "More information needed" or "Can't perform the task without more information" is referred to as information bounce. A Information Bounce cause Information Distance.

In fig 3.0 only one resolver is involved so there is no distance between resolvers. There is an information distance of minimum 4 CD in between the requester and the resolver. The series becomes 2, 6, 10, 14, 18, 22, and so on, where the first value 2 is the minimum communication distance between the initial requester and the final resolver and subsequent values are added by a common difference of 4 CD with every demand for more information (Information Bounce). Putting this into the formula for arithmetic progression, i.e. S = a + n * d

We get, a = 2,

n = number of times more information is solicited,

d = common difference = 4.

So, if the requester had to provide information to the resolver team 3 times (excluding the first time when the request was created) Distance = 2 + 3 * 4 = 14 CD. This increase in distance would obviously increase the turnaround time of the request as at each node the action owner would spend some minimum time to act upon the request and move it towards resolution. Please note that when we are talking about distance, we assume that each action owner is acting on the request without any delay and spends a fixed amount of time in processing the request. Thus in our study the turnaround time would be directly proportional to the distance with a factor of 1. i.e. 1 CD α 1 TAT, 2CD α 2 TAT

In practical scenarios often there are multiple resolver teams involved and each has to perform its duty to resolve the ticket. Let's say that there are two resolvers involved and a requester originates a ticket via service desk. This ticket travels to the first resolver who does its work and then sends the ticket back to the service desk with the information that the ticket should be assigned to the second resolver. The second resolver receives the ticket from the service desk and performs the activity that resolves the ticket. The model would look like as below:

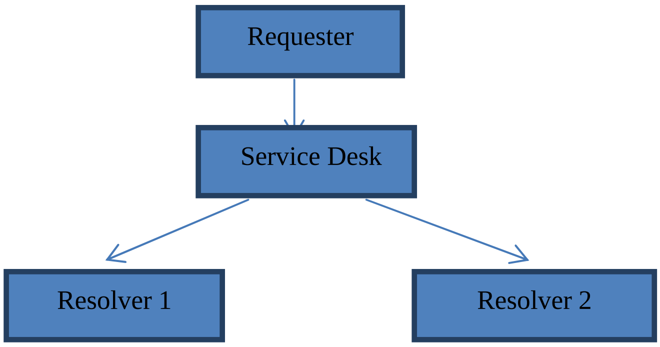


Fig 4.0 – Model with Two resolver Teams

If no additional information is required, the ticket has to travel 2 more necessary communication distances before getting resolved. So, the minimum distance becomes 4. This model gives rise to few cases:

Case 1: All communication happens via service desk: As shown above the minimum distance is 4 CD. If the first resolver finds that the information to fulfil the request is insufficient, the request travels back to the requester via the service desk. It comes back to the first resolver via the service desk again. If the information is now sufficient, the resolver 1 does its activity and then requests the Service desk to assign the ticket to the Resolver 2 who performs its activity and subsequently resolves the ticket.

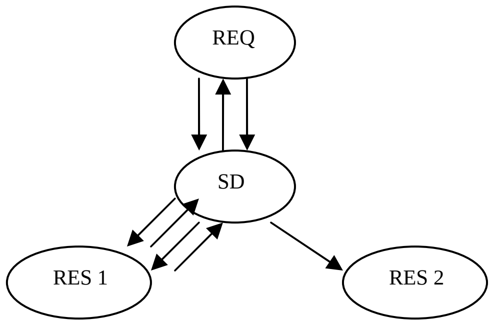


Fig 5.0 – Increase in Distance when more Information is required

If we count the number of arrows in the above figure, we get the total distance of 8 CD which involves 4 CD of Information distance and 4 CD of communication distance. So, comparing figure 3.0 with figure 5.0, there is an increase of 2 CD which is a necessary increase due to the addition of the resolver team. If any of the resolver teams need more information from the requester, the distance in need of information will increase by a common difference of 4 CD which is same as in the case of one resolver. The only difference here is the base value a_1 which is 2 in case of one resolver and 4 in case of two resolvers. As the number of resolvers increases, the base value increases by 2 CD resulting into another arithmetic progression series where a = 2 and d = 2. Putting this in the formula we get the following:
$S = a + n * d = 2 + (n * 2)$, where n represents the communication bounce between resolvers and it is equal to the number of resolver groups minus 1. So, if there are two resolver groups involved, n = 1 and if there is only one resolver group involved then n = 0.

Example 1: We now know that for case 1, with every increase in resolver group, the CD increases by a Common difference of 2 and with every request for information, the CD increases by a common difference of 4. If we need 5 resolver teams to resolve a ticket and 4 information bounces happen between any of the resolvers and the requester via the service desk, and 2 information bounces happen between the resolver groups then we can find the total distance using the formula given below:

$S = a_1 + n_1 * d_1 + n_2 * d_2 + n_3 * d_3$

Where,
a_1 = minimum distance between the requester and the first resolver
n_1 = number of information bounces between the requester and any resolver.
d_1 = distance travelled in each information bounce from resolver to requester.
$a_2 = a_1 + n_1 * d_1$
n_2 = number of necessary communication bounces between the resolver groups (it increases with the increase in number of resolvers) = n_2 = number of resolver groups - 1.
d_2 = distance travelled in each necessary communication bounce between resolver groups
n_3 = number of information bounces between any two resolvers.
d_3 = distance travelled on each information bounce between two resolvers.
$a_3 = a_1 + n_1 * d_1 + n_2 * d_2$

The formula consists of three independent series viz.

$S_1 = a_1 + n_1 * d_1$ = Necessary communication distance between requester and the first resolver + Information bounce between the requester and the resolver. Please note, this part does not cover the information bounce among the resolver groups as there is only one resolver group.

$S_2 = a_1 + n_2 * d_2$ = Necessary communication distance between requester and the first resolver + Necessary communication bounces to reach the final resolver. This part does not cover any type of information bounce.

$S_3 = a_1 + n_3 * d_3$ = Necessary communication distance between requester and the first resolver + Information bounce among the resolver groups. This part does not cover the information bounce between the requester and any of the resolver groups.

Combining the three series together we get:
$S = a_1 + n_1 * d_1 + n_2 * d_2 + n_3 * d_3$ where, a_1 is common to all the three parts.

Let's put example 1 into the three different series.
4 information bounces between the requester and any resolver can be plotted as:
$a_1 + n_1 * d_1$, where $a_1 = 2$, $n_1 = 4$, $d_1 = 4$,
2, 2 + (1 * 4), 2 + (2 * 4), 2 + (3 * 4) , **2 + (4 * 4)**
2, 6, 10, 14, **18**

5 resolver groups are required to resolve the ticket. This can be plotted as:
$a_1 + n_2 * d_2$, where $a_1 = 2$, $n_2 = 5 – 1 = 4$ (n_2 will always be one less than the total number of resolvers), $d_1 = 2$,
2, 2 + (1 * 2), 2 + (2 * 2), 2 + (3* 2), **2 + (4* 2)**
2, 4, 6, 8, **10**

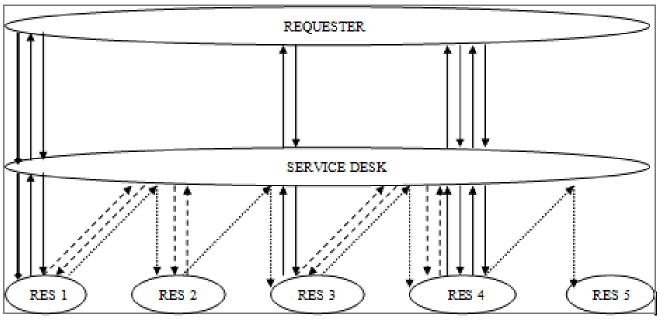
2 information bounces among the resolver groups can be plotted as:
$a_1 + n_3 * d_3$, where $a_1 = 2$, $n_3 = 2$, $d_3 = 4$,
2, 2 + (1 * 4), **2 + (2 * 4)**
2, 6, **10**

Taking the last values in the three series given above and combining them gives us the following:
$S = a_1 + n_1 * d_1 + n_2 * d_2 + n_3 * d_3$

Please note that in the invidual series a_1 is present in all the three series but, if the series are combined, a_1 will only be used once as it is common to all the individual series.

$S = 2 + (4 * 4 + 4 * 2 + 2 * 4) = 2 + 16 + 8 + 8 = 34$ CD

In the diagram below which depicts example 1 we can count the number of arrows that are required to fulfill the request. There are four types of arrows representing the various types of communication that happen among the parties involved.



Arrows	Represents
	Initial Transfer
	Information Distance Travelled between Requester and Resolver
	Communication Distance between the Resolvers.
	Information Distance Travelled between the Resolvers.

Fig 6.0 – Depiction of Example 1

The model described by case 1 has the following advantages and disadvantages.

Advantages:

1. Suitable for a newly setup ticket based Service desk model.
2. The service desk being the single point of contact for all the parties involved, it takes away the load of finding the correct assignment queues from the resolver teams. This responsibility is taken care by the service desk thus the resolver teams can focus more on the resolution process.
3. Uniformity is maintained in all the communications as all the communication pass through the service desk. This allows the service desk to put quality checks and keep a standard format in all communication.
4. The service desk is trained to handle users from different locations and languages so they are better equipped to communicate with the users provided they have the correct information from the resolver teams.

Disadvantages:

1. The model increases the communication distance and information distance thus affecting the Turn around time.
2. More resources required in Service Desk to manage greater volumes of inflow and outflow

Case 2: Resolvers can communicate directly and they talk to the requester via the Service desk: This model is more mature [3] than the model described in the case 1.

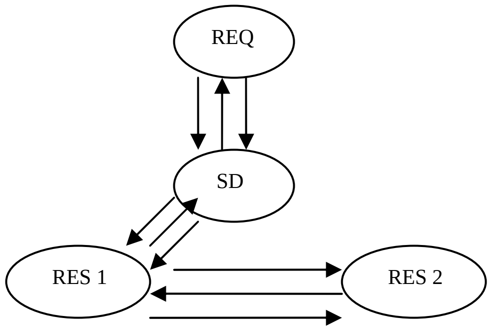


Fig 7.0 – Direct communication among the resolver teams

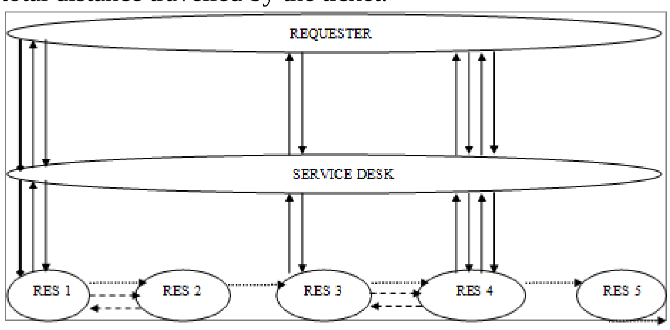
As depicted above, the communication among the resolver teams happen directly without the involvement of the service desk. This model reduces the necessary communication distance among the resolver groups by 1 CD. The information distance also reduces by 2 CD.

In the formula, $S = a_1 + n_1 * d_1 + n_2 * d_2 + n_3 * d_3$
The common difference d_2 becomes 1 CD and d_3 becomes 2 CD.

Example 2: If we need 5 resolver teams to resolve a ticket and 4 information bounces happen between any of the resolvers and the requester via the service desk, 2 information bounces happen between the resolver groups in need of more information then the communication distance becomes:

$S = 2 + 4 * 4 + 4 * 1 + 2 * 2 = 26$ CD

Please note that the Example 1 and example 2 have same criteria but, in example 2 the distance reduced by 8 CD. Due to the reduction in distance, this model can be considered more mature than the model depicted in case 1. We can count the arrows in the diagram below to find the total distance travelled by the ticket.



Arrows	Represents
	Initial Transfer
	Information Distance Travelled between Requester and Resolver
	Communication Distance between the Resolvers.
	Information Distance Travelled between the Resolvers.

Fig 8.0 – Depiction of Example 2

Advantages:

1. This model showcases better coordination among the resolver teams. When they communicate directly, they come to know about each other and mutually perform their tasks.
2. Better cooridination among the resolver teams reduces the turn around time. It also reduces the non productive time which spent in assigning the ticket from the first resolver to the service desk and then from the service desk to the second resolver and so on.
3. By improving service quality and responsiveness with a more efficient delivery system, IT can cope more effectively with cost reduction pressures and budget limitations.
4. As the ticket inflow/outflow at the service desk level decreases, the cost of managing the service desk reduces.

Disadvantages:

1. The turn around time is better than Case 1 but, there could be unnecessary bounces among the resolver teams due to factors like lack of ownership, incorrect assignment and bad quality of information.
2. The resolver teams gets added responsibility to check the availability of the resources in the teams where they want to assign the ticket or they want to ask for more information. This added responsibility reduces the focus on ticket resolution to some extent.

Case 3: Only the first communication between the requester and the resolver happens via the service desk. All the subsequent communication among all the participants happens directly: This is the most mature model. And can be depicted as below.

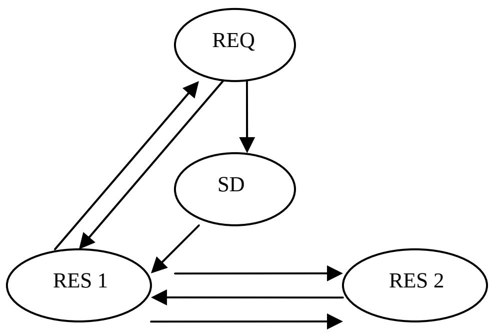


Fig 9.0 – Direct communication among all the teams after the first communication

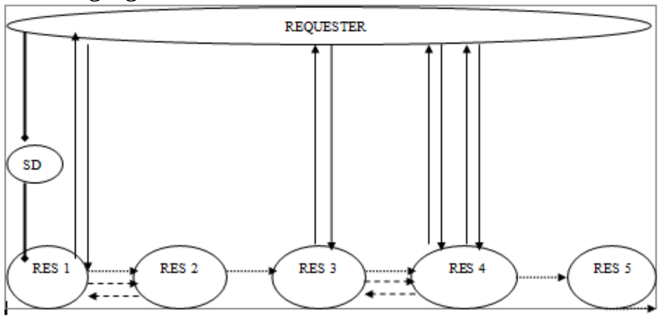
The service desk just facilitates the first transfer. All the subsequent transfers are done directly among the requester and all the resolvers. Taking the same example 1, we can see that d_1 decreases by 2 CD. All the others remain same as in example 2.

$S = a_1 + n_1 * d_1 + n_2 * d_2 + n_3 * d_3$

$S = 2 + 4 * 2 + 4 * 1 + 2 * 2 = 18$ CD

So, it reduces by 8 CD as compared case 2 and 16 CD when compared to Case 1.

This example can be diagrammatically depicted in following fig. 10.



Arrows	Represents
——→	Initial Transfer
——→	Information Distance Travelled between Requester and Resolver
········→	Communication Distance between the Resolvers.
- - - - →	Information Distance Travelled between the Resolvers.

Fig 10.0 – Depiction of Example 2

Advantages:

1. This setup overcomes all the barriers and shows confidence in the professionalism and efficiency of all the stake holders.
2. If applied ideally, this model has the fastest Turn around time.
3. By improving service quality and responsiveness with a more efficient delivery system, IT can cope more effectively with cost reduction pressures and budget limitations.
4. The role of service desk in terms of ticket assignment reduces significantly thus allowing them the opportunity to learns new skills and take new roles.

Disadvantages:

1. This model leaves most of the powers to the Requester and the resolvers. The service desk plays a very limited role thus the quality or communication could get easily compromised.
2. There could be unnecessary bounces among all the stakeholders due to poor quality of communication, lack of ownership, lack of knowledge etc. which can affect customer satisfaction.
3. The pressure on the resolver teams as well as the requester increases as they have to make sure that the tickets get assigned to the right team. Managing ticket assignment by the resolver teams may further reduce the focus on resolution of more tickets in the queue.

III. Comparative Analysis

In the previous section we managed to derive a formula that fits in all the three cases. These cases have three variables viz. n_1, n_2 and n_3. If we keep any two variables constant and change the third one, we can get a view of the performance and maturity levels [3] of the service desk models defined by the three cases above.

Let's keep the variables $n_2 = 3$ (4 resolver team involved) and $n_3 = 1$ as constants and change the variable n_1 (Information Bounce between Requester and Resolver). The outcome would look like:

TABLE I

N1	Case1	Case2	Case3
1	16	11	9
2	20	15	11
3	24	19	13
4	28	23	15
5	32	27	17

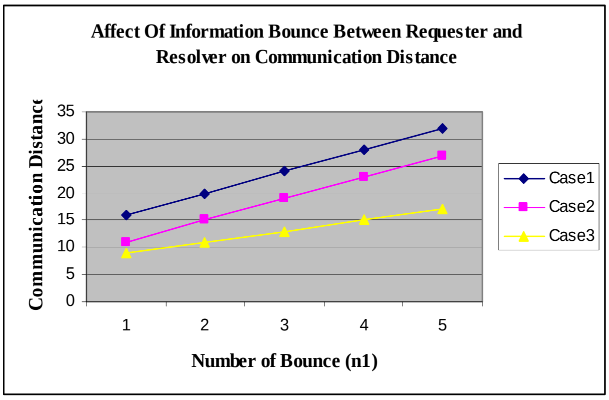


Fig 11.0 – Affect of Information Bounce between Requester and Resolver on the Communication Distance

The table and graph clearly shows the affect of increase in n_1 on the distance which in turn increases the Turn around Time.

The Case 3 fares much better than both Case 1 and Case 2. So, a mature model where the Requester and the Resolver Teams can talk directly significantly reduces the TAT. A service desk is required to initiate the dialog but, all subsequent communication happens directly.

Now, let's keep $n_1 = 1$ and $n_3 = 1$ as constants increase n_2 (Increase in number of Resolver Teams or Increase in necessary bounce among the resolvers).

TABLE III

n2	Case1	Case2	Case3
1	12	9	7
2	14	10	8
3	16	11	9
4	18	12	10
5	20	13	11

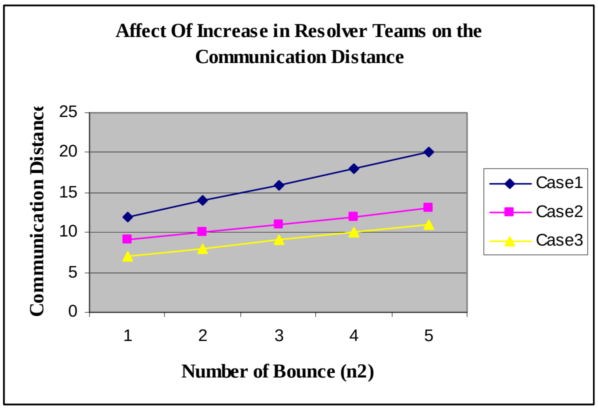


Fig 12.0 – Affect of Increase in Resolver Teams on the Communication Distance

From the figures above we can find that in case 1, the distance increase faster than the other two cases if the number of resolver teams increase. It should be noted that although there is a difference between Case 2 and Case 3 but both are parallel to each other so their rate of increase is same while the rate of increase in distance for case 1 is the highest.

Finally, let's keep $n_1 = 1$ and $n_2 = 3$ as constants increase n_3 (Increase in number of Information Bounce among the Resolver Teams).

TABLE III

n3	Case1	Case2	Case3
1	16	11	9
2	20	13	11
3	24	15	13
4	28	17	15
5	32	19	17

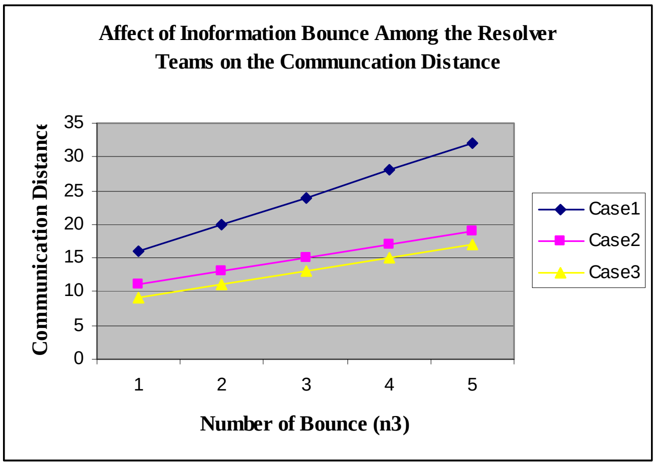


Fig 13.0 – Affect of Information Bounce among Resolver Teams on the Communication Distance

The figures above show that if the Information Bounce increases among the resolver groups, the distance would increase at a faster speed in Case 1. The distance also increases for Case 2 and Case 3 but their trends are same.

IV. Conclusions

This paper describes the three levels of maturity of a typical ticket based service desk model. This model can be generalized to fit the variations like increase in the number of the service desks between the requester and the resolver teams or between the resolver teams. Also, this paper depicts the effect of the increase in any type of bounce on the overall distance and the turn around time. Using the findings of this paper, an implementation manager would be able to choose the appropriate service desk model that fits the company's requirements. The manager can take into considerations the cultural, demographic and geographic features of the requester groups and the resolver teams. Once a model is in place, the model becomes mature and some bounces may look irrelevant over the time. Then, the manager may decide to go to the next maturity level [3] taking into consideration the advantages and disadvantages of each level.

ACKNOWLEDGMENT

We wish to acknowledge the valuable contribution of Mr. Nishank Bajaj, Incident Manager for a Global project and Mr. Virender Singh, Delivery Lead for ERP in a global company for reviewing this paper and providing very valuable inputs which helped us in completing the paper.

REFERENCES

[1] Todd Bridgeman, IPM – The ITSM Procedural Model for the New Service, Kinetic IT, 2012
[2] Harnish, Ed, Cloud-Based vs. On-Premises: How to Maximize Benefits of Each Infrastructure Model, eWeek.com, 27 August 2010
[3] IT Help Desk in Higher Education, ECAR Research Study 8, 2007
[4] ITIL – Introducing the service desk, UCISA