

# Implementation of FPGA based Encoding schemes for NoC

Neelappa

Government Engineering College Engineering Kushalnagar, Karnataka, India

## ABSTRACT

As technology shrinks, the power dissipated by the links of a network-on-chip (NoC) starts to compete with the power dissipated by the other elements of the communication subsystem, namely, the routers and the network interfaces (NIs). In this paper, we present a set of data encoding schemes aimed at reducing the utilization of the area and delay of the links of an NoC. The proposed different data encoding schemes are simulated and implemented on FPGA board.

**Keywords:** ECC, HDL, FPGA

## I. INTRODUCTION

Shifting from a silicon technology node to the next one results in faster and more power efficient gates but slower and more power hungry wires. Global interconnect length does not scale with smaller transistors and local wires. Chip size remains relatively constant because the chip function continues to increase and RC delay increases exponentially. At 32/28 nm, for instance, the RC delay in a 1-mm global wire at the minimum pitch is 25× higher than the intrinsic delay of a two input NAND fan out of long global- and semi-global-tier interconnect networks, especially in high performance designs. Power dissipation on these busses mainly occurs during signal transitions and reducing them will reduce total power dissipation. Therefore, various techniques have been proposed in literature to encode data on a bus to reduce the average and peak number of transition [1].

The basic elements which forms a NoC-based interconnect are network interfaces (NIs), routers, and links. As technology shrinks, the power dissipated by the links is as relevant as (or more relevant than) that dissipated by routers and NIs[3][4]. An ever more significant fraction of the overall power dissipation of a network-on-chip (NoC) based system-on-chip (SoC) is due to the interconnection system. In fact, as technology shrinks, the power contribute of NoC links starts to compete with that of NoC routers [5].

In this paper we focus on reducing the utilization of the area and delay of the network links. Links dissipate power due to the switching activity (both self and coupling) induced by subsequent data patterns traversing the link [6].

The basic idea is to opportunely encode the data before their injection in the network in such a way as to reduce the switching activity of the links. Differently from the previous approaches on data encoding in NoCs. The switching activity of combinational circuits depends on the logic and structure of the circuit and the switching at the output of the input latch[7].

A coding technique that reduces the coupling switching activity by taking the advantage of end-to-end encoding for wormhole switching has been presented in M. Palesi, G. Ascia, F. Fazzino, and V. Catania.

The basic idea of the proposed approach is encoding the flits before they are injected into the network with the goal of minimizing the self-switching activity and the coupling switching activity in the links traversed by the flits. In fact self-switching activity and coupling switching activity are responsible for link power dissipation. Many works have been proposed dealing with the power dissipation in on-chip communication on different components of the interconnection networks (like routers, links and NIs). The main aim is to reduce the power dissipation by the links. Works have been done in the field of area of link power

reduction. In this techniques like shielding increasing line to line spacing, repeater insertion etc. are used. These techniques increase the chip area. The encoding scheme is another method which concentrates on reducing link power dissipation. The two categories of encoding techniques can be observed. The first techniques concentrate on reducing the power due to self switching activity of individual bus lines while ignoring the coupling switching activity. In this bus invert, NC-XOR graycode, T0-XOR has been proposed. In the above techniques encoding schemes are not suitable as coupling capacitances contributes majorpart of total interconnect capacitance. Due to this large part of power consumption is due to coupling switching activity. The works in the secondary category concentrates on coupling switching activity. Many works are proposed in that one concentrates on providing the control lines for reducing switching activity, other provides less control lines but the decoding logic is complex.

## II. DESIGN AND IMPLEMENTATION

In this section, we present the encoding scheme whose goal is to reduce power dissipation by minimizing the coupling transition activities on the links of the interconnection network. Let us first describe the power model that contains different components of power dissipation of a link. The dynamic power dissipated by the interconnects and drivers is

$$P = [T_{0 \rightarrow 1}(C_s + C_l) + T_c C_c] V_{dd}^2 F_{ck} \dots \dots \dots (1)$$

Where  $T_{0 \rightarrow 1}$  is the number of  $0 \rightarrow 1$  transitions in the bus in two consecutive transmissions,  $T_c$  is the number of correlated switching between physically adjacent lines,  $C_s$  is the line to substrate capacitance, is the load capacitance,  $C_c$  is the coupling capacitance,  $V_{dd}$  is the supply voltage, and  $F_{ck}$  is the clock frequency.

The effective switched capacitance varies from type to type, and hence, the coupling transition activity,  $T_c$  is a weighted sum of different types of coupling transition contributions.

$$T_c = K_1 T_1 + K_2 T_2 + K_3 T_3 + K_4 T_4 \dots \dots \dots (2)$$

Where  $T_i$  the average number of Type I transition and  $K_i$  is its corresponding weight. We use  $K_1 = 1$ ,  $K_2 = 2$ , and  $K_3 = K_4 = 0$ . number of Type I transition may lead to

a considerable power reduction. Using (2), one may express (1) as

$$P = [T_{0 \rightarrow 1}(C_s + C_l) + (T_1 + 2T_2)C_c] V_{dd}^2 F_{ck} \dots \dots \dots (3)$$

$C_l$  Can be neglected

$$P \propto T_{0 \rightarrow 1} C_s + (T_1 + 2T_2) C_c \dots \dots \dots (4)$$

Here, we calculate the occurrence probability for different types of transitions. Consider that flit ( $t - 1$ ) and flit ( $t$ ) refer to the previous flit which was transferred via the link and the flit which is about to pass through the link respectively.

### 2.1 Scheme I

In scheme I, we focus on reducing the numbers of Type I transitions and Type II transitions. The scheme compares the current data with the previous one to decide whether odd inversion or no inversion of the current data can lead to the link power reduction.

#### 2.1.1 Power Model

If the flit is odd inverted before being transmitted, the dynamic power on the link is

$$P' \propto T'_{0 \rightarrow 1} + (K_1 T'_1 + K_2 T'_2 + K_3 T'_3 + K_4 T'_4) C_c \dots \dots (5)$$

Where  $T'_{0 \rightarrow 1}$ ,  $T'_1$ ,  $T'_2$ ,  $T'_3$  and are the self-transition activity, and the coupling transition activity of Types I, II, III, and IV, respectively. Now, defining

$$T_x = T_3 + T_4 + T_1^{***} \text{ and } T_y = T_2 + T_1 - T_1^{***}$$

one can rewrite (8) as

$$T_y > T_x \dots \dots \dots (6)$$

Assuming the link width of  $w$  bits, the total transition between adjacent lines is  $w - 1$ , and hence

$$T_y + T_x = w \dots \dots \dots (7)$$

Thus, we can write (10) as

$$T_y > \frac{(w-1)}{2} \dots \dots \dots (8)$$

This presents the condition used to determine whether the odd inversion has to be performed or not.

## 2.2 Encoding Architecture

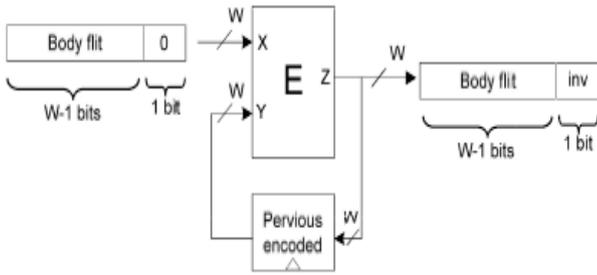


Figure 2.1.1(a) Circuit diagram

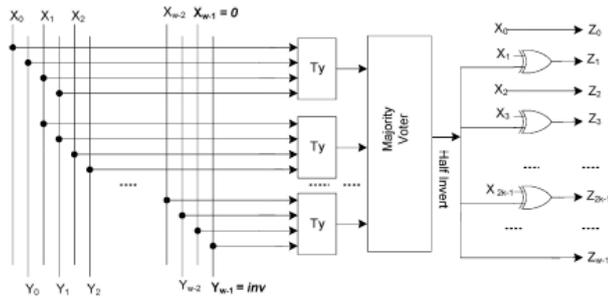


Figure 2.1.2(b) Internal View of the encoder block

The proposed encoding architecture, which is based on the odd invert condition defined, is shown in Fig.2.1.2. We consider a link width of  $w$  bits. If no encoding is used, the body flits are grouped in  $w$  bits by the NI and are transmitted via the link. In our approach, one bit of the link is used for the inversion bit, which indicates if the flit traversing the link has been inverted or not. More specifically, the NI packs the body flits in  $w - 1$  bits [Fig. 2.1.2(a)]. The encoding logic  $E$ , which is integrated into the NI, is responsible for deciding if the inversion should take place and performing the inversion if needed. The generic block diagram shown in Fig. 2.1.2(a) is the same for all three encoding schemes proposed in this paper and only the block  $E$  is different for the schemes. To make the decision, the previously encoded flit is compared with the current flit being transmitted. This latter, whose  $w$  bits are the concatenation of  $w - 1$  payload bits and a “0” bit, represents the first input of the encoder, while the previous encoded flit represents the second input of the encoder [Fig. 2.1.2(b)]. The  $w - 1$  bits of the incoming (previous encoded) body flit are indicated by  $X_i (Y_i)$ ,  $i = 0, 1, \dots, w - 2$ . The  $w^{th}$  bit of the previously encoded body flit is indicated by  $inv$  which shows if it was inverted ( $inv = 1$ ) or left as it was ( $inv = 0$ ). In the encoding logic, each  $T_y$  block takes the two adjacent bits of the input flits (e.g.,  $X_1X_2Y_1Y_2$ ,  $X_2X_3Y_2Y_3$ ,  $X_3X_4Y_3Y_4$ , etc.) and sets its output to “1” if any of the transition types of  $T_y$  is detected. This means that the odd inverting for this pair of bits leads to the reduction

of the link power dissipation. The  $T_y$  block may be implemented using a simple circuit. The second stage of the encoder, which is a majority voter block, determines if the condition given in (8) is satisfied (a higher number of 1s in the input of the block compared to 0s). If this condition is satisfied, in the last stage, the inversion is performed on odd bits. The decoder circuit simply inverts the received flit when the inversion bit is high.

## 2.3 Scheme II

In the proposed encoding scheme II, we make use of both odd and full inversion. The full inversion operation converts Type II transitions to Type IV transitions. The scheme compares the current data with the previous one to decide whether the odd, full, or no inversion of the current data can give rise to the link power reduction. The odd inversion condition is obtained as

$$T_y > (w - 1) / 2 \dots \dots \dots (9)$$

Similarly, the condition for the full inversion is obtained as

$$T_2 > T_4 \dots \dots \dots (10)$$

The full inversion condition is obtained as

$$T_2 > T_4^{**} \dots \dots \dots (11)$$

When none of (9) or (11) is satisfied, no inversion will be performed.

### 2.3.1 Encoding Architecture

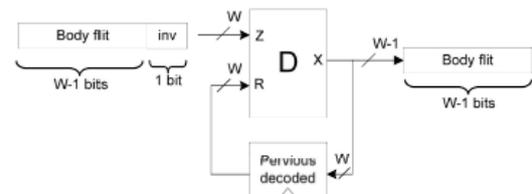


Figure 2.3.1(a) Circuit diagram

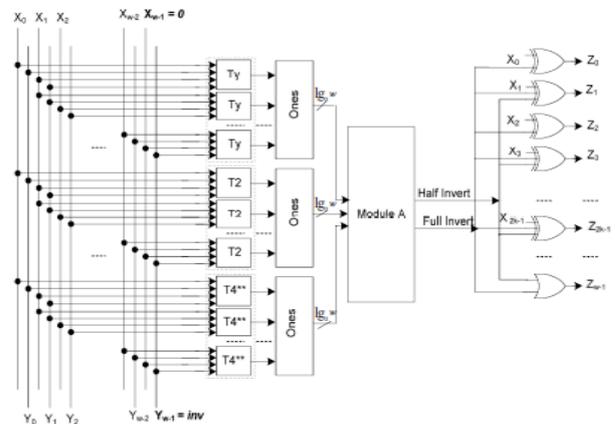


Figure 2.3.2(b) Encoder Architecture

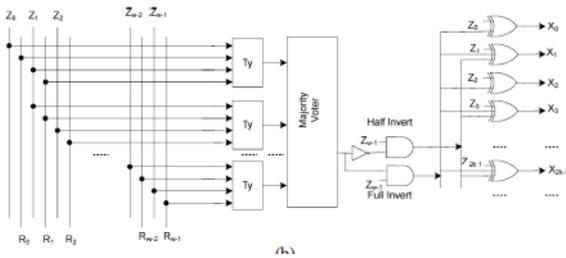


Figure 2.3.2(c) Decoder Architecture

**Encoding Architecture:** The operating principles of this encoder are similar to those of the encoder implementing Scheme I. The proposed encoding architecture, which is based on the odd invert condition of (9) and the full invert condition of (11), is shown in Fig.2.3.1(a). Here again, the  $w^{th}$  bit of the previously and the full invert condition of (11) is shown in Fig. 2.2.1(b). Here again, the  $w^{th}$  bit of the previously encoded body flit is indicated with  $inv$  which defines if it was odd or full inverted ( $inv= 1$ ) or left as it was ( $inv= 0$ ). In this encoder, in addition to the  $T_y$  block in the Scheme I encoder, we have the  $T_2$  and  $T_4^{**}$  blocks which determine if the inversion based on the transition types  $T_2$  and  $T_4^{**}$  should be taken place for the link power reduction. The second stage is formed by a set of 1s blocks which count the number of 1s in their inputs. The output of these blocks has the width of  $\log_2 w$ . The output of the top 1s block determines the number of transitions that odd inverting of pair bits leads to the link power reduction. The middle 1s block identifies the number of transitions whose full inverting of pair bits leads to the link power reduction. Finally, the bottom 1s block specifies the number of transitions whose full inverting of pair bits leads to the increased link power. Based on the number of 1s for each transition type, Module A decides if an odd invert or full invert action should be performed for the power reduction. For this module, if (9) or (11) is satisfied, the corresponding output signal will become “1.” In case no invert action should be taken place, none of the output is set to “1.” Module A can be implemented using full-adder and comparator blocks. The circuit diagram of the decoder is shown in Fig. 3. The  $w$  bits of the incoming (previous) body flit are indicated by  $Z_i(R_i)$ ,  $i = 0, 1, \dots, w - 1$ . The  $w^{th}$  bit of the body flit is indicated by  $inv$  which shows if it was inverted ( $inv = 1$ ) or left as it was ( $inv= 0$ ). For the decoder, we only need to have the  $T_y$  block to determine which action has been taken place in the encoder. Based on the outputs of these blocks, the majority voter block checks the validity of the inequality given by (8). If the output is “0” (“1”) and the

$inv= 1$ , it means that half (full) inversion of the bits has been performed. Using this output and the logical gates, the inversion action is determined. If two inversion bits were used, the overhead of the decoder hardware could be substantially reduced.

### 2.4 Scheme III

In the proposed encoding Scheme III, we add even inversion to Scheme II. The reason is that odd inversion converts some of Type I transitions to Type II transitions. Therefore, the even inversion may reduce the link power dissipation as well. The scheme compares the current data with the previous one to decide whether odd, even, full, or no inversion of the current data can give rise to the link power reduction.

**2.4.1 Power Model:** Let us indicate with  $P'$ ,  $P''$  and  $P'''$  the power dissipated by the link when the flit is transmitted with no inversion, odd inversion, full inversion, and even inversion, respectively. Similar to the analysis given for Scheme I, we can approximate the condition  $P''' < P$

The even inversion leads to power reduction when  $P''' < P$ ,  $P''' < P'$  and  $P''' < P''$  we obtain

$$T_e > (w - 1)/2, \dots, T_e > T_y, (12)$$

The full inversion leads to power reduction when  $P''' < P$ ,  $P'' < P'$  and  $P''' < P''$ , the full inversion condition is obtained as

$$2(T_2 - T_4^{**}) > 2T_y - w + 1, \quad (T_2 > T_4^{**})$$

$$2(T_2 - T_4^{**}) > 2T_e - w + 1 \quad \dots \dots \dots (13)$$

Similarly, the condition for the odd inversion is

obtained from  $P' < P$ ,  $P' < P''$  and

$P' < P'''$ , the odd inversion condition is satisfied when

$$T_y > (w - 1) / 2, \dots, T_e > T \quad \dots \dots \dots (14)$$

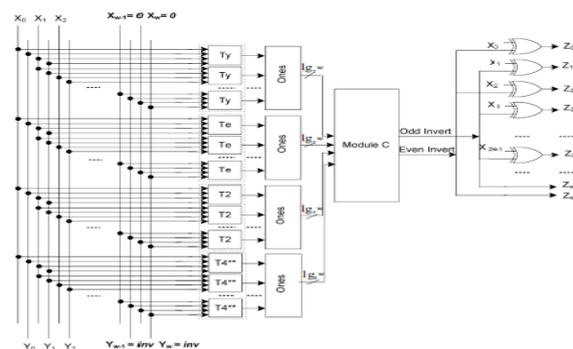


Figure 2.4.1. Encoding Architecture

**Encoding Architecture:** The operating principles of this encoder are similar to those of the encoders implementing Schemes I and II. The proposed encoding architecture, which is based on the even invert condition of (12), the full invert condition of (13), and the odd

invert condition of (14), is shown in Fig.2.4.1. The  $w^{th}$  bit of the previously encoded body flit is indicated by  $inv$  which shows if it was even, odd, or full inverted ( $inv = 1$ ) or left as it was ( $inv= 0$ ). The first stage of the encoder determines the transition types while the second stage is formed by a set of 1s blocks which count the number of ones in their inputs. In the first stage, we have added the  $T_e$  blocks which determine if any of the transition types of  $T_2$ ,  $T_1^{**}$  and  $T_1^{***}$  is detected for each pair bits of their inputs. For these transition types, the even invert action yields link power reduction. Again, we have four Ones blocks to determine the number of detected transitions for each  $T_y$ ,  $T_e$ ,  $T_2$ ,  $T_1^{**}$ , blocks. The output of the Ones blocks are inputs for Module C. This module determines if odd, even, full, or no invert action corresponding to the outputs “10,” “01,” “11,” or “00,” respectively, should be performed. The outputs “01,” “11,” and “10. Similar to the procedure used to design the decoder for scheme II, the decoder for scheme III may be designed.

### III. RESULTS AND DISCUSSION

#### 3.1.Simulation result for scheme 1

Input :16 bit input data(1100010010101011)  
 Encoded output :0110111000000001( 5 transitions are reduced)  
 Decoded output:16 bit original input data is obtained after decoding.  
 (1100010010101011)

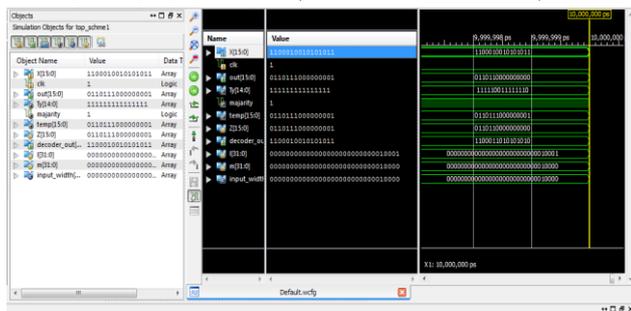


Figure 3.1. Scheme 1 simulation result

#### 3.2.Simulation result for Scheme 2

Input : 32 bit input data (10110101011001010101000110010101)  
 Encoded output:7 transitions are reduced(10101000011101000010001010000100)  
 Decoded output: 32 bit original input data is obtained after decoding data as  
 (10110101011001010101000110010101)

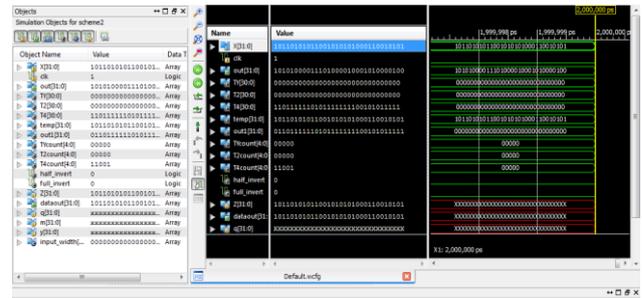


Figure 3.2. Scheme 2 Simulation result

#### 3.3.Simulation result for Scheme 3

Input : 32 bit input data (10110101011001010101000110010101)  
 Encoded output : 9 transition activities are reduced.  
 (10101000011101000100001110000100)  
 Decoded output : 32 bit original input data is obtained after decoding.  
 (10110101011001010101000110010101)

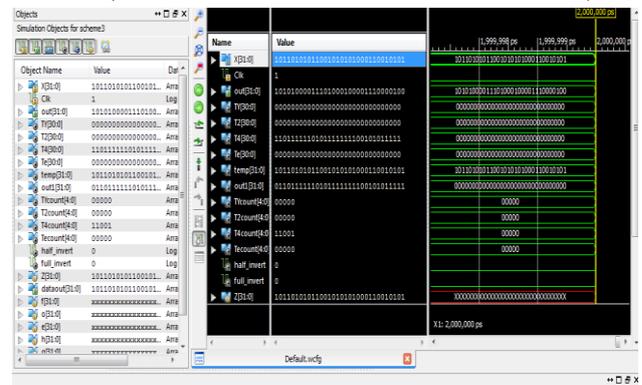


Figure 3.3. Scheme 3 simulation result

The comparison summary of the three different schemes are presented in the table 1.1

Table.1.1.Comparison summary

Logic utilization	Scheme 1	Scheme 2	Scheme 3
Input data	16-bit	32-bit	32-bit
Number of slice register	0% ( 31 out of 54576)	0% ( 65 out of 54576)	0% ( 65 out of 54576)
Number of slice LUTs	0%	1%	1%

<b>Number of fully used LUT-FF Pair</b>	<b>26% (19 out of 72)</b>	<b>21% (64 out of 291)</b>	<b>18% (63 out of 340)</b>
<b>Delay</b>	<b>7.880n sec</b>	<b>12.083n sec</b>	<b>11.373n sec</b>
<b>Frequency</b>	<b>126.904M Hz</b>	<b>82.761M Hz</b>	<b>87.928M Hz</b>
<b>Power</b>			

From the table it is clear that in terms of area the scheme III is better performance and in terms of delay the scheme I better performance compared to other schemes

#### IV. CONCLUSION

We have presented a set of data encoding schemes on FPGA aimed at reducing the logic utilization of the area and delay. The proposed encoding schemes are simulated and implemented on FPGA board. Overall, the proposed encoding scheme III has better performance compared to other two schemes even though it consumes more number of slices and delay.

#### V. REFERENCES

[1]. Buchegger S., Tissieres C., Le Boudec J. Y., "A Test-Bed for Misbehaviour Detection in Mobile Ad-Hoc Networks -How Much Can Watchdogs Really Do?", Mobile Computing Systems and Applications (WMCSA '04), pp. 102-111, 2004

[2]. Ning P., Sun K., "How to Misuse AODV: A Case Study of Insider Attacks against Mobile Ad-hoc Routing Protocols", In Proc. of the IEEE Workshop on Information Assurance, pp. 60-67, 2003

[3]. Stajano F., Anderson R., "The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks", In Proc. of Int. Workshop on Security Protocols, Springer, 1999 Yi P., Dai Z., Zhang S., Zhong Y., "A New Routing Attack in Mobile Ad Hoc Networks", Int.

[4]. Hattig .M, 2001, Ed., Zero-Conf IP Host Requirements, Draftietfzerofonreqts- 09.tct, IETF MANET Working Group, August 2001.

[5]. Perrig .A, Szewczyk .R, Wen .V, Culler .D, and Tygar J.D, 2001, "SPINS: security protocols for sensor networks": Proceedings of the 7th Annual International Conference in Mobile Computing and Networks (MobiCom 2001), Rome, Italy, pp. 189-199.

[6]. Basagni .S, Herrin .K, Rosti .E, and Bruschi .D, 2001, "Secure Pebble nets, in: Proceedings of 2nd MobiHoc", Long Beach CA, October 2001, pp.156-163.

[7]. Boukerche .A, El-Khatib .K, Xu .L, Korba .L, 2004, "Secure ad hoc routing protocol", Fourth International IEEE Workshop on Wireless Local Networks. Tampa, Florida, November 2004. NRC47394.

[8]. Pearlman .M. R, Haas .Z .J, Sholander .P, Tabrizi S. S, "On the impact of lternate path routing for load balancing in mobile ad hoc networks", Mobi HOC, 2000.

[9]. Yenumula B. Reddy, RastkoSelmic. 2011, "Agentbased Trust Calculation in Wireless Sensor Networks", SENSORCOMM 2011: The Fifth International Conference on Sensor Technologies and Applications, IARIA, pp 324-339.

[10]. Wenjia Li, Anupam Joshi And Tim Finin, 2011 "ATM: Automated Trust Management For Mobile Ad-Hoc Networks Using Support Vector Machine", In: 12th IEEE International Conference On Mobile Data Management (MDM), pp. 291-292

[11]. Govindan and P. Mohapatra, 2012 "Trust computations and trust dynamics in mobile adhoc networks: A survey, "IEEE Commun. Surveys & Tutorials, vol. 14, no. 2, pp. 279-298.

[12]. England, P., Shi, Q., Askwith, B., Bouhaf. 2012, " A Survey Of Trust Management In Mobile Ad-Hoc Networks "Proceedings of the 13th Annual Post Graduate Symposium on The Convergence of Telecommunications, Networking and Broadcasting.