Revisit Consensus in a Dual Fallible Clustered-MANET

K.Q. Yan **S.C. Wang** **Y.F. Tsou**
kqyan@cyut.edu.tw **scwang@cyut.edu.tw** **s9414613@cyut.edu.tw**
Chaoyang University of Technology

摘要

在隨意式無線網路的環境中，經常會因為傳輸媒介的損毀或雜訊的干擾，以致影響相關應用的執行。為了要提昇隨意式無線網路的可靠度，本研究在一個階層式的隨意式無線網路上探討合議問題。期望藉由合議值的達成，使系統的容錯能力得以提昇。本研究所提出的協定，可以使用最少次數的資訊交換，即能獲致最高之容錯能力。
關鍵詞：合議，容錯，分散式系統，階層式隨意式無線網路

Abstract

A Mobile Ad-hoc Network (MANET) may suffer from various types of transmission medium (TM) failure. In order to enhance the fault-tolerance and reliability of the MANETs, the consensus problem in the MANET model based on hierarchical clustering structure (clustered-MANET) is revisited in this paper. The proposed protocol is called Dual Consensus Protocol (DCP), which can make each correct mobile node reach a common value to cope with the faulty component in the clustered-MANET.
Keywords: Consensus, fault-tolerance, distributed system, hierarchical clustering structure, MANET

1. Introduction

The MANETs have attracted significant attentions recently due to its features of infrastructure less, quick deployment and automatic adaptation to changes in topology. Therefore, MANET suits for military communication, emergency disaster rescue operation, and law enforcement [2].

The reliability of the environment is one of the most important aspects in MANET. In order to provide a reliable environment in a MANET, we need a mechanism to allow a set of mobile nodes (MNs) to agree on a common value [8]. The Byzantine Agreement (BA) problem [2,3,8] is one of the most fundamental problems to reach a common value in a distributed system.

The BA problem was first introduced by Lamport [5] in 1980. With the agreement, many applications [8] can be achieved. A closely related sub-problem, consensus problem, has been studied extensively in the literature [4]. Lamport argued that the consensus problem under the assumption of synchronous behavior, showing that $(3f+1)$ nodes are required to allow f failures [7]. The previous research [3] had solved the consensus problem in an unreliable communication system, but it treated all faulty TMs are malicious. Actually, the symptom of a faulty TM can be classified into dormant and malicious fault [8]. The dormant fault of a communication always can be identified by the receiver if the transmitted message or information were encode appropriately (such as by NRZ-code and Manchester code [7]) before communication, it means that the dormant faulty TM can be detected. On the other hand, the malicious faulty TM is unpredictable.

In this paper, we concern the solution of consensus problem. The definition of the problem is to make the correct nodes in an n nodes distributed system to reach consensus. Each node chooses an initial value to start with, and communication to each other by exchanging messages. A group of multiple nodes is referred to make a consensus if it satisfies the following conditions [6].

(Agreement): All correct nodes agree on the same value.

(Validity): If the initial value of all nodes is v_i, then all correct nodes shall agree on v_i.

In a consensus problem, many results are based on the assumption of node failure in a fail-safe network [2,4,6]. Based on this assumption, a TM fault is unfairly treated as a node fault [5], regardless the correctness of an innocent node; hence an innocent node does not involve consensus. This is a contradiction with the definition of consensus problem, which requires all correct nodes to reach consensus.

In this paper, we consider a distributed system whose nodes are reliable during the consensus execution in clustered-MANET, while the TM may be disturbed by some faults. An efficient and reliable protocol to achieve consensus in an unreliable communication environment of tradition network topology has been proposed [5]. The proposed protocol can tolerate $\lceil c/2 \rceil - 1$ faulty TMs where c is the connectivity of network [7].

The rest of this paper is organized as follows. Section 2 discusses the MANET. Section 3 illustrates the concept of DCP by an example. The fault tolerant capability and correctness of the proposed protocol is shown in Section 4. Finally, the conclusion is given in the last section.

2. Mobile Ad-hoc Network (MANET)

MANET is composed of many mobile nodes [1].

In MANET, each node can dynamically form a network without any infrastructure such as base station. Each node connects to each other by multi-hop wireless TM. Besides moving randomly, each MN acts as a router to help other MN in the network transmit data packets. In the MANET, there are few cluster managers (CM) to take charge of message transmission of all MNs. Therefore, the CMs could be crashed due to a large number of work burdens and then effect the performance of message transmission of whole network. Fortunately, in the MANET, nodes often together and further form clusters due to common characteristics.

Fig. 1 shows the network topology, which is a hierarchical clustering network framework with several layers. Each cluster has its own CM. When the child cluster network is established, its CM will automatically establish hierarchical relation with its parent CM by exchanging data and obtaining its up and down layer manager.

Using this method to establish hierarchical manager architecture to manage and provide whole network environment to transmit data. When the message sender and receiver nodes are in the same cluster, it will exchange message directly by the two nodes to decrease the burden of the CM.

If the message sender and receiver nodes are located in different clusters, the sender node will transmit the message to its CM and then the group manager will transmit the message to its up-layer group manager according to the hierarchical framework. By using the relay between group managers, the message package can finally be transmitted to the receiver node.

If the receiver node is located out of the whole transmission range, the packet will uniformly be transmitted to the highest CM by which exchanges data with the highest CM in the other range to reach the goal of data delivering.

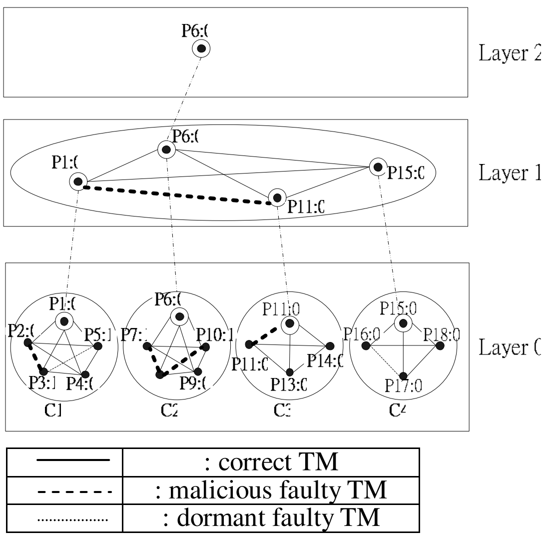

Fig. 1 MANET based on Hierarchical cluster structure

3. The Dual Consensus Protocol (DCP)

The proposed protocol DCP is used to solve the consensus problem due to faulty TMs, which may send wrong messages to influence the system to achieve consensus in a clustered-MANET. DCP protocol consists two phases and needs two rounds of message exchange to solve the consensus problem. In the first round of the message exchange, each node multicasts its initial value through TM and then receives the initial value of other nodes as well. In the second round, each node acts as the sender, sending the vector received in the first round to each other, and constructs a matrix, called the MAT_i, $1 \le i \le n$. Finally, the decision making phase will reach consensus among the all nodes. The proposed protocol DCP is presented in Fig. 2. Moreover, the procedure for setting MAT_i is shown in Fig. 3.

DCP protocol (for node P_i with initial value v_i)

Message Exchange Phase:

Round 1: Multicast (v_i), then receives the initial value from the other nodes, and construct vector V_i.

Round 2: Multicast (V_i), and then receive column vectors broadcasted by other nodes, and construct MAT_i.

Decision Making Phase:

Step 1-1: Each **λ** value is eliminated and does not join to majority. Take the local majority on messages received from each cluster of each row k of MAT_i to new MAT_i. If (local MAJ_k = ?), then set the local majority value = ϕ.

Step 1-2: Take the normal majority value of each row k of new MAT_i to MAJ_k.

Step 2: Search for any MAJ_k. If ($\exists MAJ_k = \neg v_i$), then $DEC_i := \phi$; else if ($\exists MAJ_k = \#$) AND ($v_{ki} = v_i$), then $DEC_i := \phi$; else $DEC_i := v_i$, and terminate.

Fig. 2. The DCP protocol to reach consensus

Procedure MATRIX (for node P_i with initial value v_i)

1. Receive the initial value v_i from node P_j, for $1 \le j \le n$ and $j \ne i$.
2. Construct the vector $V_i = [v_1, v_2, \ldots, v_j, \ldots, v_n]$, $1 \le j \le n$ and $j \ne i$. If a dormant TM, called TM_{ik}, was found, then $v_k = \lambda$
3. Multicast V_i to all nodes, and receive column vector V_j from node P_j, $1 \le j \le n$
4. Construct a MAT_i (Setting the vector v_j in column j, for $1 \le j \le n$). If a dormant TM, say TM_{ik}, was found, then $V_k = [\lambda, \lambda, \ldots, \lambda]$

Fig.3. Procedure for setting MAT_i on each node P_i

Subsequently, an example of executing the DCP in the clustered-MANET is shown in Fig. 1. There are eighteen nodes (denoted by P1, P2, …, P18) in the clustered-MANET. All nodes in the network are joined to the lowest layer (Layer 0). Four of the clusters of Layer 0 are shown in the Fig. 1. Nodes P1, P6, P11 and P15 are the CM of these clusters. The initial value of nodes P_i is 0 (for i = 1, 2, 4, 6, 9, 10, 11, 13 to 18); the initial value of other nodes is 1.

In the first round of message exchange, each node P_i multicasts its initial value v_i through TM to all other nodes, where $1 \le i \le n$, and receives the initial value of other nodes as well. Then each node uses the received message to construct vector V_i as shown in Fig. 4(a). In the second round of message exchange, each node multicasts its vector V_i and receives the vectors from other nodes to construct the matrix MAT_i as shown in Fig. 4(b).

The message exchange phase has completed after two rounds by DCP. In order to reduce the incorrect values of the TM were interfered within dormant or malicious. Each node takes majority in Step 1 of decision making phase. By the first of Step 1, each node takes the local majority on the value received from a cluster and constructs an 18*4 matrix. Then each node takes the normal majority value from each row of 18*4 matrix in the end of Step 2. Then, all nodes agree on the same value ϕ, and consensus is reached.

4. Fault tolerance capability analysis

The following lemmas and theorems are used to prove the correctness and complexity of DCP.

Lemma 1: If there is a majority value = $\neg v_i$ in MAT_i, then there is at least one node with an initial value which disagrees with v_i in the network.

Proof: The majority value in the k-th row = $\neg v_i$ means that there are at least $\lceil (n\text{-}d+1)/2 \rceil$ $\neg v_i$'s in the k-th row where d is the number of dormant faults. Since the number of malicious faulty TMs is at most $\lfloor (n\text{-}d\text{-}3)/2 \rfloor$-1, and $(\lfloor (n\text{-}d+1)/2 \rfloor+1)\text{-}(\lfloor (n\text{-}d\text{-}3)/2 \rfloor\text{-}1) = 2$. Therefore, there exists at least one value $\neg v_i$ received from a correct TM. In other words, a node has a different initial value $\neg v_i$.

Lemma 2: Let the initial value of node P_i be v_i and TM_{ij} is correct or dormant, then the majority value at the i-th row in MAT_j should be v_i.

Proof:

Case 1: Since TM_{ij} is correct, the node P_j will receive v_i from node P_i in the first round and $v_{ij} = v_i$ in MAT_j. Meanwhile, the value v_i of node P_i will be broadcasted to the other nodes. There are at most $\lceil (n\text{-}d\text{-}3)/2 \rceil$-1 malicious faulty TMs in the network. In the second round, node P_j receives at least $(n\text{-}d\text{-}1)\text{-} \lceil (n\text{-}d\text{-}3)/2 \rceil = \lceil (n\text{-}d+1)/2 \rceil$ v_i's in the i-th row of MAT_j, where d is the number of λ which will be eliminated during the voting of majority. Hence, there are at least $\lceil (n\text{-}d+1)/2 \rceil$ v_i's in the i-th row, and the majority value in the i-th row should be equal to v_i.

Case 2-1: TM_{ij} is dormant and n is an old number, the node j will receive λ from node P_i in the first round and $v_{ij} = \lambda$ in MAT_j. Meanwhile, the value v_i will be broadcasted to other nodes. There are at most $\lceil (n\text{-}d\text{-}3)/2 \rceil$-1 maliciously faulty TMs and d dormant TMs in the network. After the second round, node P_j receives at least $(d+1)$ λ's and at least $n\text{-}(d+1)\text{-}\lceil (n\text{-}d\text{-}3)/2 \rceil\text{-}1 = \lfloor (n\text{-}d+1)/2 \rfloor+1$ v_i's in the i-th row of MAT_j, where d is the number of λ which will eliminated during the voting of majority. Hence, there are $n = (d+1)$ non-λ's and at least $\lfloor (n\text{-}d+1)/2 \rfloor$ (greater than $\lceil (n\text{-}(d+1)+1/2 \rceil = \lceil (n\text{-}d)/2 \rceil$ the majority required when n is in odd) v_i's in the i-th row, so, the majority value in the i-th should be equal to v_i.

Case 2-2: TM_{ij} is dormant and n is an odd number, the node P_j will receive λ from node P_i in the first round and $v_{ij} = \lambda$ in MAT_j. Meanwhile, the value v_i of node P_i will be broadcasted to the other nodes. There are at most $\lfloor (n\text{-}d+3)/2 \rfloor$ malicious faulty TMs and d dormant TMs in the system. After the second round, node P_j receives at least $(d+1)$ λ's and at least $n\text{-}(d+1)\text{-}(\lfloor (n\text{-}d\text{-}3)/2 \rfloor\text{-}1) = \lfloor (n\text{-}d+1)\ /2 \rfloor$ v_i's in the i-th row of MAT_j, where d is the number of λ which will be eliminated during the voting of majority. Hence, there are $n\text{-}(d+1)$ non-λ's and at least $\lfloor (n\text{-}d+1)/2 \rfloor+1$ (greater than $\lceil [n\text{-}(d+1)+1]/2 \rceil\text{-}1 = \lfloor (n\text{-}d)\ /2 \rfloor$) v_i's in the i-th row, so, the majority value in the i-th row should be equal to v_i. ■

Lemma 3: If the initial value of node P_i is v_i, whether the TM_{ij} is correct or dormant, the majority value at the i-th row of MAT_j, $1 \le j \le n$, should be either be v_i or not be able to be determined with $v_{ij} = \neg\ v_i$.

Proof: By Lemma 2, when TM_{ij} is correct or dormant, the majority value of the i-th row in node P_j is v_i, for $1 \le j \le n$. When TM_{ij} is under the influence of malicious fault, we consider the following two cases after running the first round.

Case 1: $v_{ij} = v_i$. Since there are at most $\lceil (n\text{-}d\text{-}3)/2 \rceil$ malicious faulty TM connected with node P_j, at most $\lceil (n\text{-}d\text{-}3)/2 \rceil$ values that may be $\neg v_i$'s in the second round. The number of v_i's is $[(n\text{-}d)\text{-}\lceil (n\text{-}d\text{-}3)/2 \rceil] = \lceil (n\text{-}d+3)/2 \rceil$ in the i-th row where d is the number of λ which will be eliminated during the voting of majority; therefore, the majority of the i-th row in MAT_i is v_i.

Case 2: $v_{ij} = \neg v_i$. There are at most $\lceil (n\text{-}d\text{-}3)/2 \rceil$ malicious faulty TMs. Therefore, in the second round, the total number of $\neg v_i$'s does not exceed $\lceil (n\text{-}d\text{-}3)/2 \rceil +1 = \lceil (n\text{-}d\text{-}1)/2 \rceil$ and the number of v_i's is at least $(n\text{-}d\text{-}1)\text{-}(\lfloor (n\text{-}d+1)/2 \rfloor) = \lfloor (n\text{-}d\text{-}1)/2 \rfloor$. If $n\text{-}d$ is an even number, then $\lceil (n\text{-}d\text{-}1)/2 \rceil = \lfloor (n\text{-}d\text{-}1)/2 \rfloor$, the majority of the i-th row in MAT_j cannot be determined. If $n\text{-}d$ is an odd number, then $\lceil (n\text{-}d\text{-}1)/2 \rceil > \lfloor (n\text{-}d\text{-}1)/2 \rfloor$. Hence, the majority of the i-th row in MAT_j is v_i.

Lemma 4: If $(\neg\exists MAJ_k = \neg v_i)$ AND $\{(\exists MAJ_k = ?)$ AND $(v_{ki} = v_i)\}$ in MAT_i, then $DEC_i = \phi$ is correct.

Proof: If there has a MAJ_k = ?, there are exactly $(n\text{-}d)/2$ v_i's and $(n\text{-}d)/2$ $\neg v_i$'s in the k-th row. If $v_{ki} = v_i$ in MAT_i, then all $(n\text{-}d)/2$ $\neg v_i$'s should be received in the second round. There are $\lceil (n\text{-}d\text{-}3)/2 \rceil$ malicious faulty TMs in the system. Therefore, in the second round, node P_i at least receives $(n\text{-}d)/2 - \lceil (n\text{-}d\text{-}3)/2 \rceil \ge 1$ value $\neg v_i$ from node P_k without disturbance. The initial value of node P_k should disagree with the initial value of node P_i; hence it is correct to choose $DEC_i = \phi$. If $v_{ki} = \neg v_i$, we claim that $\neg v_i$ ought to be passed through malicious TM from node P_i, and the initial value of node should be $\neg v_{ki} = v_i$. To prove, if TM_{ki} is correct, then the initial value of node P_k should be $\neg v_i$.

By Lemma 2, the majority value of the k-th row in MAT_i is $\neg v_i$. This is contradiction with the condition of ($\neg\exists MAJ_k = \neg v_i$). If the initial value of node P_k was $\neg v_i$, then by Lemma 3, MAJ_k should be either $\neg v_i$ or ? for $v_{ki} = v_i$. It is a contradiction.

Theorem 1: Protocol DCP is correct.

Proof: By Lemmas 1, 2, 3 and 4, the theorem is proved.

Theorem 2: Protocol DCP can reach a consensus.

Proof:

(1) Agreement:

Part 1: If a correct node agrees on ϕ, all correct nodes should agree on ϕ. If the correct node P_p with initial value v_i agrees on ϕ, by Theorem 1, there is at least a correct node P_k with initial value $\neg v_i$ in the network. By Lemma 4, the majority value in the k-th row of MAT_j, $1 \le j \le n$, should be either $\neg v_i$ or ? for $v_{kj} = v_i$. All correct nodes with initial value v_i agree on ϕ. Similarly, for the correct node P_p with initial value $\neg v_i$, the majority value of the p-th row in MAT_j, $1 \le j \le n$, either should be v_i or cannot be determined with $\neg v_{ij} = v_i$. All correct nodes with initial value $\neg v_i$ agree on ϕ, too.

Part 2: If a correct node agrees on v_i, all correct nodes should agree on v_i. If the correct node P_i with initial value v_i and $DEC_i = v_i$, but there exists some correct node P_i, $j \neq i$, has $DEC_j \neq v_i$, then that is impossible. To show this, if $DEC_j = \phi$, by Part 1, then $DEC_i = \phi$. This is a contradiction with the assumption as above. If $DEC_j = \neg v_i$, unless the initial value of node P_i is $\neg v_i$, otherwise it is impossible according to the definition consensus problem. However, if the initial value of node P_j is $\neg v_i$, by Lemma 4, MAJ_i is equal to $\neg v_i$ or ? with $v_{\underline{ij}} = v_i$ in $MAT_{\underline{i}}$; then, $DEC_{\underline{i}} = \phi$, which is a contradiction. Hence, all correct nodes should agree on the same value.

(2)Validity: The initial value of all nodes should be the same. If there is a value $\neg v_i$ in MAT_j, $1 \le j \le n$, then the value must be caused by malicious faulty TM. There are at most $\lceil (n-d-3)/2 \rceil$ malicious faulty TMs, hence there are at most $\lceil (n-d-3)/2 \rceil$ $\neg v_i$'s in each row. Since the value received in the first round may be $\neg v_i$, the majority of each row for all MAT_j, should be MAJ_j= ? (If the value received in the first round is $\neg v_j$, $1 \le j \le n$); or v_j. By Step 2 of the protocol DCP, all correct nodes should agree on v_i.

Theorem 3: The amount of information exchange by DCP is $O(n^2)$.

Proof: In the first round, each node sends out (n-1) copies of its initial value to other nodes. In the second round, an n-element vector is sent to the other n-1 nodes in the network; therefore, the total number of message exchange is $(n-1)+(n*(n-1))$. Therefore, the complexity of information exchange is $O(n^2)$

Theorem 4: One round of message exchange to achieve consensus is impossible.

Proof:

Part1: Message exchange is necessary.

Without message exchange, a node cannot know whether or not a disagreeable value exists in other nodes; hence, consensus achievement is impossible.

Part2:One round message exchange is not enough to achieve consensus.

If node P_i is connected with node P_j by faulty TM_{ij}. Node P_i may not know the initial value in node P_j by using only one round of message exchange. Therefore, it is impossible to achieve consensus by using only one round of message exchange. ■

Theorem 5: If the total number of the faulty TMs t > $m+d$, where $m \le \lfloor (n-d+3)/2 \rfloor$, achieving consensus is impossible.

Proof: When t>m+d, n is an even number and each node has c TMs, c is odd number, in the system. It is possible that a node has more malicious faulty TM than correct TM even if the influence of d dormant faults was eliminated. Regardless of the number of rounds of message exchange, this node will always be confused by the messages transferred through those malicious faulty TMs. The decision making by the node may conflict with other nodes. In this case, consensus achievement is impossible. ■

Theorem 6: Using the minimum number of rounds, DCP can tolerate the maximum number of faulty TMs.

Proof: From Theorems 2, 4 and 5, the theorem is proved.

5. Conclusion

In the past, complex networks had studied in a branch of mathematics known as graph theory. The network topology developed in recent years [1] shows a mobile feature such that the previous protocols such as [5] cannot adapt to it. In this paper, the consensus problem on dual failure modes in clustered-MANET has revisited, the proposed protocol DCP makes all correct nodes reach consensus. DCP derives its bound of allowable faulty TMs with two rounds of message exchange.

Moreover, previous works about consensus problem had based on assumption that nodes are the only fallible components in the network, we plan to extend our protocol to consider the status (such as mobility) of nodes in clustered-MANET in future work.

References

[1] Banerjee, S. and Khuller, S., "A Clustering Scheme for Hierarchical Control in Multi-hop Wireless Network," Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, pp.1028-1037, 2001.

[2] T.C. Chiang, H.M. Tsai and Y.M. Huang, "A Partition Network Model for Ad Hoc Networks," IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, Vol. 3, pp.467-472, 2005.

[3] M. Fischer, "The Consensus Problem in Unreliable Distributed Systems (A Brief Survey)," Technical report, Department of

Computer Science, Yale University, 1983.

[4] M.J. Fischer, N. L. and M. S. Paterson, “Impossibility of Distributed Consensus with One Faulty Process,” Journal of the ACM, 32(2), pp. 374-382, 1985.

[5] L. Lamport, R. Shostak and M. Pease, “The Byzantine General Problem,” ACM Transactions on Programming Language and Systems, Vol. 4, No. 3, pp. 382-401, 1982.

[6] F.J. Meyer, and D.K. Pradhan, “Consensus with Dual Failure Modes,” IEEE Transactions on Parallel and Distributed Systems, vol. 2, no. 2, April 1991.

[7] M. Pease, R. Shostak and L. Lamport, “Reaching Agreement in Presence of Faults,” Journal of ACM, Vol. 27, No. 2, pp. 228-234, 1980.

[8] S.C., Wang, K.Q., Yan, and C.F., Cheng, “Byzantine Agreement under Unreliable Multicasting Network,” Pakistan Journal of Information and Technology, Vol. 2, No. 2, April-June, 2003, pp. 104-115.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
0	0	λ	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
1	λ	1	1	0	1	1	1	1	1	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
1	1	0	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	λ	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	λ	1	1	λ	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	λ	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	λ	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	λ	1	1	1	1	1	1	1
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig.4 (a) The vector received in the first round

MATi = (P_i, for i=1,4,6,9,17,18)

0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
0	0	λ	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
1	λ	1	1	0	1	1	1	1	1	**0**	**0**	**0**	**0**	1	1	1	1
0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
1	1	0	1	1	1	1	1	1	1	**0**	**0**	**0**	**0**	1	1	1	1
0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
1	1	1	1	1	1	1	λ	1	1	**0**	**0**	**0**	**0**	1	1	1	1
1	1	1	1	1	1	λ	1	1	λ	**0**	**0**	**0**	**0**	1	1	1	1
0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
0	0	0	0	0	0	0	λ	0	0	**1**	**1**	**1**	**1**	0	0	0	0
1	1	1	1	1	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
0	0	0	0	0	1	1	1	1	1	**0**	**0**	**0**	**0**	1	1	1	1
1	1	1	1	1	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
1	1	1	1	1	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	1	0	0
0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	1	0	0	0
0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0

MATi = (P_i, for i=2)

0	0	λ	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
0	0	λ	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
1	λ	λ	1	0	1	1	1	1	1	**0**	**0**	**0**	**0**	1	1	1	1
0	0	λ	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
1	1	λ	1	1	1	1	1	1	1	**0**	**0**	**0**	**0**	1	1	1	1
0	0	λ	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
1	1	λ	1	1	1	1	λ	1	1	**0**	**0**	**0**	**0**	1	1	1	1
1	1	λ	1	1	1	λ	1	1	λ	**0**	**0**	**0**	**0**	1	1	1	1
0	0	λ	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
0	0	λ	0	0	0	0	λ	0	0	**1**	**1**	**1**	**1**	0	0	0	0
1	1	λ	1	1	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
0	0	λ	0	0	1	1	1	1	1	**0**	**0**	**0**	**0**	1	1	1	1
1	1	λ	1	1	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
1	1	λ	1	1	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
0	0	λ	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	1	0	0
0	0	λ	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	1	0	0	0
0	0	λ	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
0	0	λ	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0

MATi = (P_i, for i=3)

0	**λ**	0	0	**1**	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
0	**λ**	λ	0	**1**	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
1	**λ**	1	1	**1**	1	1	1	1	1	**0**	**0**	**0**	**0**	1	1	1	1
0	**λ**	0	0	**1**	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
1	**λ**	0	1	**0**	1	1	1	1	1	**0**	**0**	**0**	**0**	1	1	1	1
0	**λ**	0	0	**1**	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
1	**λ**	1	1	**0**	1	1	λ	1	1	**0**	**0**	**0**	**0**	1	1	1	1
1	**λ**	1	1	**0**	1	λ	1	1	λ	**0**	**0**	**0**	**0**	1	1	1	1
0	**λ**	0	0	**1**	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
0	**λ**	0	0	**1**	0	0	λ	0	0	**1**	**1**	**1**	**1**	0	0	0	0
1	**λ**	1	1	**0**	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
0	**λ**	0	0	**1**	1	1	1	1	1	**0**	**0**	**0**	**0**	1	1	1	1
1	**λ**	1	1	**0**	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
1	**λ**	1	1	**0**	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
0	**λ**	0	0	**1**	0	0	0	0	0	**1**	**1**	**1**	**1**	0	1	0	0
0	**λ**	0	0	**1**	0	0	0	0	0	**1**	**1**	**1**	**1**	1	0	0	0
0	**λ**	0	0	**1**	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
0	**λ**	0	0	**1**	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0

MATi = (P_i, for i=5)

0	0	**0**	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
0	0	**1**	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
1	λ	**0**	1	0	1	1	1	1	1	**0**	**0**	**0**	**0**	1	1	1	1
0	0	**1**	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
1	1	**1**	1	1	1	1	1	1	1	**0**	**0**	**0**	**0**	1	1	1	1
0	0	**1**	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
1	1	**0**	1	1	1	1	λ	1	1	**0**	**0**	**0**	**0**	1	1	1	1
1	1	**0**	1	1	1	λ	1	1	λ	**0**	**0**	**0**	**0**	1	1	1	1
0	0	**1**	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
0	0	**1**	0	0	0	0	λ	0	0	**1**	**1**	**1**	**1**	0	0	0	0
1	1	**0**	1	1	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
0	0	**1**	0	0	1	1	1	1	1	**0**	**0**	**0**	**0**	1	1	1	1
1	1	**0**	1	1	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
1	1	**0**	1	1	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
0	0	**1**	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	1	0	0
0	0	**1**	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	1	0	0	0
0	0	**1**	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0
0	0	**0**	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0

MATi = (P_i, for i=7)

0	0	0	0	0	0	0	λ	0	0	1	1	1	1	0	0	0	0
0	0	λ	0	0	0	0	λ	0	0	1	1	1	1	0	0	0	0
1	λ	1	1	0	1	1	λ	1	1	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	λ	0	0	1	1	1	1	0	0	0	0
1	1	0	1	1	1	1	λ	1	1	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	λ	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	λ	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	λ	λ	1	λ	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	λ	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	λ	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	λ	0	0	0	λ	0	0	0	0	0	0
0	0	0	0	0	1	1	λ	1	1	λ	1	1	1	1	1	1	1
1	1	1	1	1	0	0	λ	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	λ	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	λ	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	λ	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	λ	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	λ	0	0	1	1	1	1	0	0	0	0

MATi = (P_i, for i=8)

0	0	0	0	0	0	λ	0	0	λ	1	1	1	1	0	0	0	0
0	0	λ	0	0	0	λ	0	0	λ	1	1	1	1	0	0	0	0
1	λ	1	1	0	1	λ	1	1	λ	0	0	0	0	1	1	1	1
0	0	0	0	0	0	λ	0	0	λ	1	1	1	1	0	0	0	0
1	1	0	1	1	1	λ	1	1	λ	0	0	0	0	1	1	1	1
0	0	0	0	0	0	λ	0	0	λ	0	0	0	0	0	0	0	0
1	1	1	1	1	1	λ	λ	1	λ	1	1	1	1	1	1	1	1
1	1	1	1	1	1	λ	1	1	λ	1	1	1	1	1	1	1	1
0	0	0	0	0	0	λ	0	0	λ	0	0	0	0	0	0	0	0
0	0	0	0	0	0	λ	λ	0	λ	0	0	0	0	0	0	0	0
1	1	1	1	1	0	λ	0	0	λ	0	λ	0	0	0	0	0	0
0	0	0	0	0	1	λ	1	1	λ	λ	1	1	1	1	1	1	1
1	1	1	1	1	0	λ	0	0	λ	0	0	0	0	0	0	0	0
1	1	1	1	1	0	λ	0	0	λ	0	0	0	0	0	0	0	0
0	0	0	0	0	0	λ	0	0	λ	0	0	0	0	0	1	0	0
0	0	0	0	0	0	λ	0	0	λ	0	0	0	0	1	0	0	0
0	0	0	0	0	0	λ	0	0	λ	0	0	0	0	0	0	0	0
0	0	0	0	0	0	λ	0	0	λ	1	1	1	1	0	0	0	0

MATi = (P_i, for i=10)

0	0	0	0	0	0	0	λ	0	0	1	1	1	1	0	0	0	0
0	0	λ	0	0	0	0	λ	0	0	1	1	1	1	0	0	0	0
1	λ	1	1	0	1	1	λ	1	1	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	λ	0	0	1	1	1	1	0	0	0	0
1	1	0	1	1	1	1	λ	1	1	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	λ	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	λ	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	λ	λ	1	λ	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	λ	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	λ	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	λ	0	0	0	λ	0	0	0	0	0	0
0	0	0	0	0	1	1	λ	1	1	λ	1	1	1	1	1	1	1
1	1	1	1	1	0	0	λ	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	λ	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	λ	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	λ	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	λ	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	λ	0	0	1	1	1	1	0	0	0	0

MATi = (P_i, for i=11)

1	**1**	**1**	**0**	**1**	0	0	0	0	0	1	λ	1	1	0	0	0	0
1	**1**	**0**	**0**	**1**	0	0	0	0	0	1	λ	1	1	0	0	0	0
0	**1**	**0**	**1**	**1**	1	1	1	1	1	0	λ	0	0	1	1	1	1
1	**1**	**1**	**0**	**1**	0	0	0	0	0	1	λ	1	1	0	0	0	0
0	**0**	**1**	**1**	**0**	1	1	1	1	1	0	λ	0	0	1	1	1	1
1	**1**	**1**	**0**	**1**	0	0	0	0	0	0	λ	0	0	0	0	0	0
0	**0**	**0**	**1**	**0**	1	1	λ	1	1	1	λ	1	1	1	1	1	1
0	**0**	**0**	**1**	**0**	1	λ	1	1	λ	1	λ	1	1	1	1	1	1
1	**1**	**1**	**0**	**1**	0	0	0	0	0	0	λ	0	0	0	0	0	0
1	**1**	**1**	**0**	**1**	0	0	λ	0	0	0	λ	0	0	0	0	0	0
0	**0**	**0**	**0**	**0**	0	0	0	0	0	0	λ	0	0	0	0	0	0
1	**1**	**1**	**1**	**1**	1	1	1	1	1	λ	λ	1	1	1	1	1	1
0	**0**	**0**	**0**	**0**	0	0	0	0	0	0	λ	0	0	0	0	0	0
0	**0**	**0**	**0**	**0**	0	0	0	0	0	0	λ	0	0	0	0	0	0
1	**1**	**1**	**1**	**1**	0	0	0	0	0	0	λ	0	0	0	1	0	0
1	**1**	**1**	**1**	**1**	0	0	0	0	0	0	λ	0	0	1	0	0	0
1	**1**	**1**	**1**	**1**	0	0	0	0	0	0	λ	0	0	0	0	0	0
1	**1**	**1**	**0**	**1**	0	0	0	0	0	1	λ	1	1	0	0	0	0

MATi = (P_i, for i=12)

1	**1**	**1**	**0**	**1**	0	0	0	0	0	0	1	1	1	0	0	0	0
1	**1**	**0**	**0**	**1**	0	0	0	0	0	0	1	1	1	0	0	0	0
0	**1**	**0**	**1**	**1**	1	1	1	1	1	1	0	0	0	1	1	1	1
1	**1**	**1**	**0**	**1**	0	0	0	0	0	0	1	1	1	0	0	0	0
0	**0**	**1**	**1**	**0**	1	1	1	1	1	1	0	0	0	1	1	1	1
1	**1**	**1**	**0**	**1**	0	0	0	0	0	1	0	0	0	0	0	0	0
0	**0**	**0**	**1**	**0**	1	1	λ	1	1	0	1	1	1	1	1	1	1
0	**0**	**0**	**1**	**0**	1	λ	1	1	λ	0	1	1	1	1	1	1	1
1	**1**	**1**	**0**	**1**	0	0	0	0	0	1	0	0	0	0	0	0	0
1	**1**	**1**	**0**	**1**	0	0	λ	0	0	1	0	0	0	0	0	0	0
0	**0**	**0**	**0**	**0**	0	0	0	0	0	1	λ	0	0	0	0	0	0
1	**1**	**1**	**1**	**1**	1	1	1	1	1	0	1	1	1	1	1	1	1
0	**0**	**0**	**0**	**0**	0	0	0	0	0	1	0	0	0	0	0	0	0
0	**0**	**0**	**0**	**0**	0	0	0	0	0	1	0	0	0	0	0	0	0
1	**1**	**1**	**1**	**1**	0	0	0	0	0	1	0	0	0	0	1	0	0
1	**1**	**1**	**1**	**1**	0	0	0	0	0	1	0	0	0	1	0	0	0
1	**1**	**1**	**1**	**1**	0	0	0	0	0	1	0	0	0	0	0	0	0
1	**1**	**1**	**0**	**1**	0	0	0	0	0	0	1	1	1	0	0	0	0

MATi = (P_i, for i=13)

1	**1**	**1**	**0**	**1**	0	0	0	0	0	1	1	1	1	0	0	0	0
1	**1**	**0**	**0**	**1**	0	0	0	0	0	1	1	1	1	0	0	0	0
0	**1**	**0**	**1**	**1**	1	1	1	1	1	0	0	0	0	1	1	1	1
1	**1**	**1**	**0**	**1**	0	0	0	0	0	1	1	1	1	0	0	0	0
0	**0**	**1**	**1**	**0**	1	1	1	1	1	0	0	0	0	1	1	1	1
1	**1**	**1**	**0**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0
0	**0**	**0**	**1**	**0**	1	1	λ	1	1	1	1	1	1	1	1	1	1
0	**0**	**0**	**1**	**0**	1	λ	1	1	λ	1	1	1	1	1	1	1	1
1	**1**	**1**	**0**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0
1	**1**	**1**	**0**	**1**	0	0	λ	0	0	0	0	0	0	0	0	0	0
0	**0**	**0**	**0**	**0**	0	0	0	0	0	0	λ	0	0	0	0	0	0
1	**1**	**1**	**1**	**1**	1	1	1	1	1	λ	1	1	1	1	1	1	1
0	**0**	**0**	**0**	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0
0	**0**	**0**	**0**	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0
1	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	1	0	0
1	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	1	0	0	0
1	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0
1	**1**	**1**	**0**	**1**	0	0	0	0	0	1	1	1	1	0	0	0	0

MATi = (P_i, for i=14)

1	**1**	**1**	**0**	**1**	0	0	0	0	0	1	1	1	1	0	0	0	0
1	**1**	**0**	**0**	**1**	0	0	0	0	0	1	1	1	1	0	0	0	0
0	**1**	**0**	**1**	**1**	1	1	1	1	1	0	0	0	0	1	1	1	1
1	**1**	**1**	**0**	**1**	0	0	0	0	0	1	1	1	1	0	0	0	0
0	**0**	**1**	**1**	**0**	1	1	1	1	1	0	0	0	0	1	1	1	1
1	**1**	**1**	**0**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0
0	**0**	**0**	**1**	**0**	1	1	λ	1	1	1	1	1	1	1	1	1	1
0	**0**	**0**	**1**	**0**	1	λ	1	1	λ	1	1	1	1	1	1	1	1
1	**1**	**1**	**0**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0
1	**1**	**1**	**0**	**1**	0	0	λ	0	0	0	0	0	0	0	0	0	0
0	**0**	**0**	**0**	**0**	0	0	0	0	0	0	λ	0	0	0	0	0	0
1	**1**	**1**	**1**	**1**	1	1	1	1	1	λ	1	1	1	1	1	1	1
0	**0**	**0**	**0**	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0
0	**0**	**0**	**0**	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0
1	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	1	0	0
1	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	1	0	0	0
1	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0
1	**1**	**1**	**0**	**1**	0	0	0	0	0	1	1	1	1	0	0	0	0

MATi = (P_i, for i=15)

0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	**0**	0	0
0	0	λ	0	0	0	0	0	0	0	1	1	1	1	0	**0**	0	0
1	λ	1	1	0	1	1	1	1	1	0	0	0	0	1	**1**	1	1
0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	**0**	0	0
1	1	0	1	1	1	1	1	1	1	0	0	0	0	1	**1**	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0
1	1	1	1	1	1	1	λ	1	1	1	1	1	1	1	**1**	1	1
1	1	1	1	1	1	λ	1	1	λ	1	1	1	1	1	**1**	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0
0	0	0	0	0	0	0	λ	0	0	0	0	0	0	0	**0**	0	0
1	1	1	1	1	0	0	0	0	0	0	λ	0	0	0	**0**	0	0
0	0	0	0	0	1	1	1	1	1	λ	1	1	1	1	**1**	1	1
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	**0**	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	**0**	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	**0**	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	**0**	0	0

MATi = (P_i, for i=16)

0	0	0	0	0	0	0	0	0	0	1	1	1	1	**0**	0	0	0
0	0	λ	0	0	0	0	0	0	0	1	1	1	1	**0**	0	0	0
1	λ	1	1	0	1	1	1	1	1	0	0	0	0	**1**	1	1	1
0	0	0	0	0	0	0	0	0	0	1	1	1	1	**0**	0	0	0
1	1	0	1	1	1	1	1	1	1	0	0	0	0	**1**	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0
1	1	1	1	1	1	1	λ	1	1	1	1	1	1	**1**	1	1	1
1	1	1	1	1	1	λ	1	1	λ	1	1	1	1	**1**	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0
0	0	0	0	0	0	0	λ	0	0	0	0	0	0	**0**	0	0	0
1	1	1	1	1	0	0	0	0	0	0	λ	0	0	**0**	0	0	0
0	0	0	0	0	1	1	1	1	1	λ	1	1	1	**1**	1	1	1
1	1	1	1	1	0	0	0	0	0	0	0	0	0	**0**	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	**0**	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1	**0**	0	0	0

Fig.4(b) MAT constructed at the end of 2nd rounds

Fig.4 An example of DCP execution