

Building Scalable Fault Tolerant Network

Raheel Ahmed Memon¹, Minhoo Shin^{1*}, Yeonseung Ryu¹, Jong-myung Rhee² and Dongho Lee³

Department of Computer Engineering¹

Myongji University, Yongin, Gyeonggi-do, Korea

Department of Information and Communication Engineering²

Myongji University, Yongin, Gyeonggi-do, Korea

Agency of Defense Development, Korea³

{raheel, shinminho, ysryu, jmr77}@mj.ac.kr

Abstract— Providing fault-tolerance in Ethernet has been a challenging issue. Though it is now possible to create a fault tolerant Ethernet in small-scale networks by using heartbeat mechanism, but for large-scale networks fault tolerance is still a challenge. This work is a contribution to such type of networks which are large and requires fault tolerance. Recursive Scalable Autonomous Fault-tolerant Ethernet (RSAFE) is a novel idea for providing Fault-tolerance in large-scale networks by using its recursive structure. Besides fault tolerance, RSAFE has many desirable features such as RSAFE uses TCP/IP and IP multicast standard communication protocols, in RSAFE growth of network will not effect to the existing network structure, also RSAFE uses Commercial Off-The-shelf products and there is no need of development or modification in existing products, so RSAFE is easier scheme to implement in existing networks. Further, comparison shows that our proposed scheme RSAFE is a cost-effective solution than previously proposed schemes.

Keywords— Fault-tolerant, Mission critical systems, Large-scale networks, Ethernet, Scalability, Recursive architecture

I. INTRODUCTION

In the network based mission critical systems such as unmanned vehicles, military weapon and aviation equipment control systems, Ethernet connectivity must provide fault tolerance to all the constituents for the smooth operations of such systems. *Fault Tolerant Ethernet* (FTE) developed by Honeywell is a response to such needs [7]. FTE provides the multiple paths to all its nodes for smooth flow of data without any interruption. FTE is designed with the following four goals. First, no single point of failure should cause the loss of communication between nodes. Second, failures must be detected and recovered within a specified time. Third, communication protocol standards such as TCP/IP and IP multicast must be supported. Fourth, the existing Commercial Off-The-Shelf (COTS) products can be used without modification [2][3].

There are two approaches widely used to prevent network failures [1]. The first approach offers *hardware solutions* which uses multiport Network Interface Cards (NIC) that provide standby ports to a network which are automatically activated in the case of the failure of an active port. Second approach offers *software based solutions*, which detect failures in a network and executes recovery algorithms diverting the data through alternative paths to destinations. Software based approaches use a heartbeat mechanism which

performs quite fine in a small network but as the size of network grows the heartbeat starts consuming large bandwidth. This scalability problem of *heartbeat* mechanism limits the application of software-based approaches in large networks [3].

In our previous work [1][3][8], we proposed a software-based approach called Scalable Autonomous Fault tolerant Network (SAFE). It resides in layer 2 and 3 of OSI model with multiple subnet architecture where all nodes in subnet send and receive Ethernet based heartbeats to each other to detect and recover the faults in network paths. But in SAFE there is a central subnet and all other subnets connect to that central subnet.

In this paper, we propose a new approach called *Recursive Scalable Autonomous Fault tolerant Ethernet* (RSAFE). This is a solution for large scale and mission critical systems to provide multiple paths if any failure is detected on any of the paths, then an alternative path can reroute packets to the destination without loss of information. In RSAFE, a large network is divided into various subnets (a subnet contains limited number of nodes that FTL can support), further limited number of subnets are grouped together to form different groups, then groups are recursively combined to form levels, until only one group remains in the highest level.

RSAFE uses a heartbeat mechanism for fault detection. As we described earlier, a heartbeat mechanism may cause large bandwidth consumption as the size of network grows. But, in our proposed approach by dividing the large network into small subnets and limiting the size of subnet, heartbeat mechanism can be implemented efficiently.

To provide inter-subnet connectivity, we select two master nodes in each subnet, one of which plays Primary Master Node (PMN) as an active node and the other plays Secondary Master Node (SMN) as a backup node that replaces PMN in case of PMN's failure. The master node is responsible for providing status updates of corresponding subnet to other subnets in the network. Suppose x is the current path of Node₄ in Subnet₁. Now if path x fails to send two consecutive heartbeat messages to the active master node, then the master node will assume that the current path x is failed and notify all the subnets that path x to Node₄ in Subnet₁ is failed and the new path to approach Node₄ in Subnet₁ is now y which is an alternative path.

In section 2, we review the background of fault-tolerant network architecture. In section 3 and section 4, we present

our RSAFE scheme in detail. We then show comparison study in Section 5. Finally, we summarize the work in Section 6.

II. RELATED WORKS

There have been various solutions for fault tolerance in mission critical systems. According to Song et.al [4] these solutions can be categorized in to three distinct types of network architectures namely: (1) Single network with redundant cables; (2) Redundant networks; and (3) Single network and single cable with reliable multi path. Each of these general types of architecture provides various; hardware, software and protocol solutions.

Single Network with redundant cables: One Network interface controller is connected to two or more redundant cables. From redundant cables one cable functions as a primary communication path while other cable(s) remain on standby and are activated in the case if any fault occurs in the primary cable.

Redundant Network: Two or more independent networks are maintained. One of them functions as primary network for communication while other network(s) remain on standby and are activated in the case if any fault occurs in the primary network.

Single network and single cable with reliable multi paths: A ring structure is the typical topology for this type of fault-tolerant network. The network itself provides a self-healing mechanism such that when open wires occur at one point of the network, the traffic is rerouted through an alternative path to its destination node.

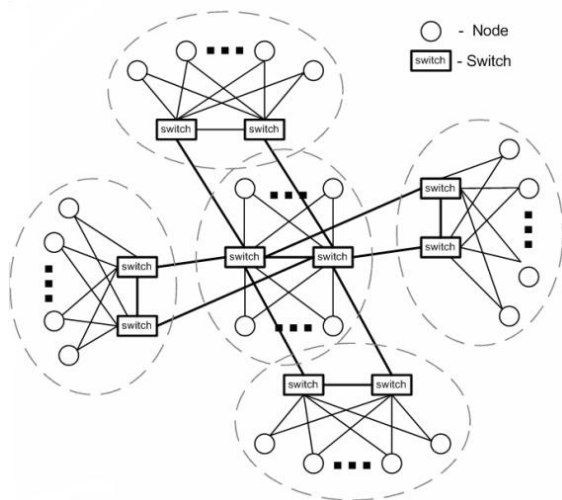


Figure 1. SAFE Architecture

The first approach Single Network with Redundant Cables has been used in Scalable Autonomous Fault-tolerant Ethernet (SAFE) [1]. As shown in Figure 1, SAFE divides large network into several subnets and each of subnet is having a limiting number of nodes. Each node in a subnet has two network interface cards corresponding to PORT A and PORT B, and each subnet having two switches. Due to such architecture this model is sometimes called a dual-Fault Tolerant Ethernet (DFTE) Network Model.

Since each node is having two networks interface cards so it provides four possible paths for communication within a subnet. The limitation of this scheme is that it uses central connectivity of subnets where each subnet is connected with a central subnet, so if both switches of central subnets will fail, then whole the communication will be interrupted.

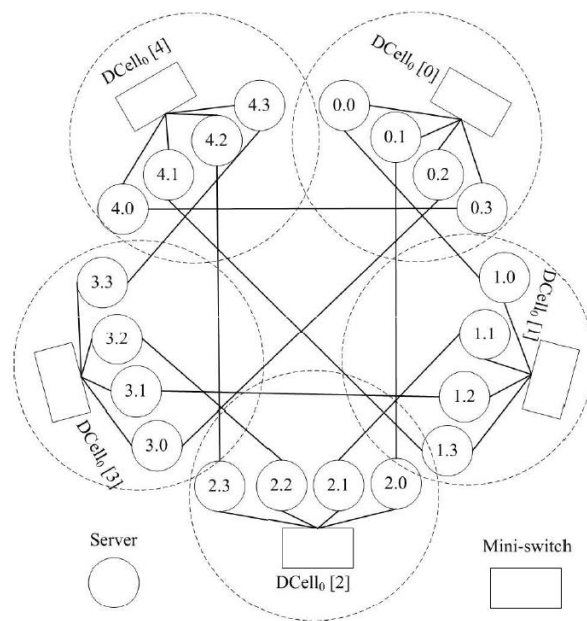


Figure 2. DCell Architecture

For inter-subnet connectivity a work has been done by Chaunxiong Guo, et al. [6] for interconnectivity in between exponentially increasing number of servers in data centers and called it DCell. Figure 2 shows the DCell Architecture, DCell has many good features for data centre networking and recursive structure is one among them. High-level DCell is constructed from many low-level DCells while DCells at a same level are fully connected to each other. Second feature of DCell is its Scalability which doubles exponentially as the node degree increases. Third feature is the distributed fault tolerant protocol that does not allow single point of failure and route the information through shortest-path even in the presence of server link or node failures. Fourth feature of DCell is that it has higher network capacity than the traditional tree-based structures. Since in DCell networks each server is connected to different levels of DCells via its multiple links it becomes a rather costly solution.

The fault tolerant architecture presented in this paper is a novel idea of inter-subnet connectivity for SAFE that provides multiple paths in inter-subnet connectivity and it is also it is covering the cons of SAFE and DCell. RSAFE doesn't have any central subnet to like SAFE, so any subnet failure will not cause the failure of whole network. Also RSAFE is not much expensive solution as like DCell structure, because in RSAFE each node is not directly connected to all its upper levels.

III. RSAFE

A. Overall Architecture

Our proposed scheme RSAFE is a low cost fault tolerant architecture for mission critical networks. RSAFE has six advantages. First, RSAFE has a recursive structure, which provides multiple paths to all nodes within a subnet and also it has multiple routing paths between all subnets. Second, RSAFE is a scalable solution for growing networks without degrading their performance. Third, RSAFE structure can tolerate single path failure events efficiently without causing any loss of communication. Fourth, RSAFE can detect and recover faults within a specified time. Fifth, RSAFE uses standard communication protocols such as TCP/IP and IP multicast. Finally, RSAFE solutions are based on Commercial Off-The-Shelf products, so there is no need of new development or modification in existing products.

The architecture of RSAFE is shown in Figure 3 RSAFE comprises of three basic building blocks; Subnet, Group and Level. Together these components form a Recursive structure.

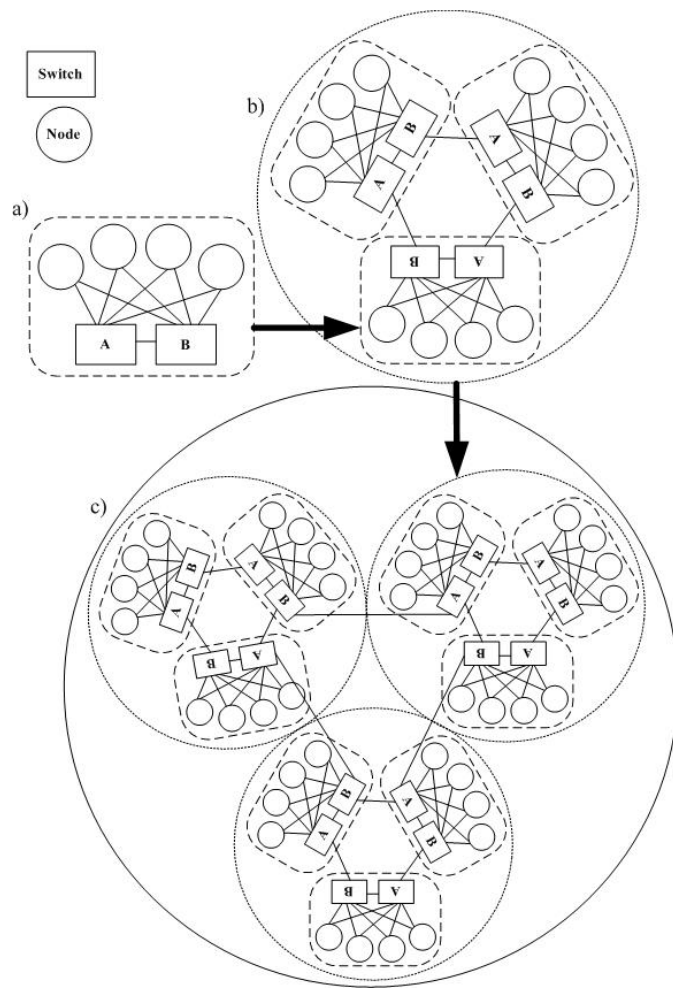


Figure 3 Three basic building blocks of Recursive Scalable Autonomous Fault-tolerant Ethernet: a) Subnet, b) Group and c) Level

1) Subnet: Figure 3 (a) shows that a Subnet is constructed from limited number of nodes and two switches. There are four possible paths for communication between two nodes. In the subnet one data path is defined as the primary

path for communication between two nodes of a subnet. Rest of the paths will remain on standby and will activate in the case if the primary data-path encounter a fault.

2) Group: Figure 3 (b) shows that a limited number of subnets together form a group where each subnet has a dual connectivity with other subnets of the group. The connectivity is formed as Second Switch of Subnet₀ is connected to first Switch of Subnet₁ and Second Switch of Subnet₁ is connected to first Switch of Subnet₂, for last subnet the Second Switch of Subnet_n is connected to first Switch of Subnet₀.

3) Level: Figure 3 (c) shows that a limited number of Groups together form a Level. Once the Subnets and Groups build the next step is to build levels, Level₀ is combination of limited number of groups, and Level₁ to onward is the combination of different Levels. In Figure 3. There is shown Level_{0,0} only, but suppose that there Level_{0,m} levels are available then the Level₁ will build by combining all the levels of Level₀. Similarly all the required levels can be created recursively; we can say that, in this structure all the higher levels are constructed recursively from lower levels. And for connectivity between the Levels or groups the subnets are selected randomly from each level/group and connected to each other.

In the given example number of nodes in a subnet is 4, and number of subnets in a group is 3, further number of groups in a level is 3.

B. Algorithm for Building RSAFE Architecture

In brief, we build RSAFE architecture in the following way. The algorithm starts with a set of nodes and switches. In the first step, we build a list of subnets each of which contains two switches and a fixed number of nodes, all chosen uniformly at random. Starting with the set of subnets, considering each subnet as a group, we build a level, which is a set of super-groups, i.e., sets of groups. We repeat this level-building procedure recursively as follows. In each phase, we are given a set of groups. We repeatedly create a super-group, a group of groups until every group belongs to a super-group. This new set of super-groups is called a level, and the next level is created from this group-set using the same algorithm. The recursive procedure ends when there is only one group left in the level.

Algorithm 1. Building RSAFE Network: RSAFEGen(·)

```

1: procedure RSAFEGen( NDList, SWList )
2:   SubnetList := SubnetListGen( NDList, SWList )
3:   Level := LevelGen( SubnetList )
4:   return Level
5: end procedure
-----
6: procedure SubnetListGen( NDList, SWList )
7:   SubnetList := ∅
8:   while NDList != ∅
9:     Nodes := PickRandom( NDList, SUBNET_SIZE )
10:    Switch1 := PickRandom( SWList, 1 )
11:    Switch2 := PickRandom( SWList, 1 )
12:    For each nd in Nodes
13:      Connect( nd, Switch1 )

```

```

14:         Connect( nd, Switch1 )
15:     end for
16:     SubnetList := SubnetList  $\cup$  { (Switch1, Switch2, Nodes) }
17: end while
18: return SubnetList
19: end procedure
-----
20: procedure PickRandom( List, Count )
21:     // remove at most Count many random elements in List
22:     // and return them
23: end procedure
-----
24: procedure Connect( entity1, entity2 )
25:     // Connect entity1 and entity2 with an Ethernet line
26: end procedure
-----
27: procedure LevelGen( GroupList )
28:     If |GroupList| = 1 then
29:         return GroupList
30:     end if
31:     Level :=  $\emptyset$ 
32:     while GroupList  $\neq \emptyset$ 
33:         SupGroup := PickRandom( GroupList, SUPGROUP_SIZE )
34:         For i := 0 to |SupGroup| - 2
35:             ConnectGrp( SupGroup[i], SupGroup[i+1])
36:         end for
37:         ConnectGrp(SupGroup[0], SupGroup[|SupGroup|-1])
38:         Level := Level  $\cup$  { SupGroup }
39:     end while
40:     return LevelGen( Level )
41: end procedure
-----
42: procedure ConnectGrp( Group1, Group2 )
43:     Connect( PickSwitch( Group1 ), PickSwitch( Group2 ) )
44: end procedure
-----
45: procedure PickSwitch( Group )
46:     // Among all the switches contained in the group,
47:     // return a randomly chosen switch
48: end procedure

```

Algorithm 1 describes the procedure in detail. The algorithm uses two constants as system parameters: SUBNET_SIZE for the size of each subnet, and SUPGROUP_SIZE for the size of each super-group (Table 1). For example, in Figure 3 SUBNET_SIZE is 3 and SUPGROUP_SIZE is 4.

Table 1. Parameters and Variables in Algorithm 1

Parameter/Variable	Description
SUBNET_SIZE	Maximum number of nodes in a subnet
SUPGROUP_SIZE	Maximum number of groups in a super-group
NDList	List of all nodes
SWList	List of all switches
SubnetList	List of subnets
GroupList	List of groups
Switch1/2	First/second switch of a subnet
Level	A level, i.e., a list of super-groups
SupGroup	A super-group, i.e., a group of groups

The main function RSAFEGen() is given a set of nodes NDList and a set of switches SWList. This function requires the following prerequisites:

- PR1. The number of switches is even (i.e., |SWList| is even)
- PR2. $|SWList| * (SUBNET_SIZE - 1) / 2 \leq |NDList|$ and $|NDList| \leq |SWList| * (SUBNET_SIZE) / 2$

These prerequisites are assumed so that no nodes and switches are left unused at the end of the algorithm. However, this assumption is not relevant to the correctness of the algorithm.

The algorithm starts when calling RSAFEGen() function (line 1-5) with two parameters: node-set NDList and switch-set SWList. The function first builds a set of subnets using given nodes and switches by calling SubnetListGen() function (line 2). Then, it calls the recursive functions LevelGen() function (line 3) to obtain the RSAFE architecture.

The SubnetListGen() function (line 6-19) repeatedly retrieves SUBNET_SIZE nodes and two switches chosen uniformly at random from NDList and SWList, respectively, (line 9-11) to construct subnets until no more nodes are left (line 8-17). For each subnet, we connect each node with each switch (line 12-15), and then we add the subnet, denoted by a list of two switches and a node-set, to the subnets-set SubnetList (line 16). In the process, we use two functions: PickRandom() function, which returns specified number of elements from the given list chosen at random and removes chosen elements from the list, and Connect() function, which connects two entities with an Ethernet wire.

The LevelGen() function (line 27-41) is the core function that builds the RSAFE architecture. The function is given a group-set GroupList as a parameter. At the first call, this group-set contains subnets that were created in SubnetListGen() function. In the subsequent calls, the parameter GroupList contains super-groups that were created from the previous call to the same function. In line 28-30, we first check whether we reached the highest level by checking the size of GroupList; if the size equals to one, we simply return the group-set, which is the highest level itself. Otherwise, we proceed to build a level out of GroupList. In line 32-39, we repeatedly extract SUPGROUP_SIZE (or less) groups out of GroupList to form a super-group, connect the groups in the formed super-group, and then add the super-group to the set Level. When constructing a super-group (line 34-37), we build a ring-topology of all the groups in the super-group, that is, given the ordered list of groups, we connect two consecutive groups in a circular manner. To connect two groups, we call ConnectGrp() function (line 42-44), which connects two switches each randomly chosen from each group (the selection implemented by PickSwitch() function). At the end of the function, variable Level contains all the super-groups that we constructed from GroupList. Then we make a recursive call to LevelGen() function passing Level as a parameter. The return value of LevelGen() is returned immediately.

In summary, Algorithm 1 leverages the recursive structure of RSAFE; we make recursive calls to build each level based

on the previous level. Our algorithm works for various configurations: different number of nodes, different subnet, sizes and different super-group sizes. The algorithm also leverages the random selection of switches when connecting two groups, naturally distributing routing loads among switches in the group. Note that, because of this randomness, groups could be connected differently in Figure 3; the figure just exemplifies an ideal uniformly random selection of switches when connecting groups.

IV. FAULT-TOLERANT MECHANISM

A. Handling Faults in each Subnet: RSAF basically detects network faults on the basis of subnet using heartbeat mechanism. A heartbeat message (HBM) is an Ethernet (layer 2) frame sent and received between nodes in each subnet. HBM can reach only the nodes in a subnet and cannot reach the outside nodes of the subnet because it is layer-2 data. Each node is responsible for detecting the faults of its belonging subnet and recovering from it. A node in RSAF network has three modules for handling faults occurred in its belonging subnet: HBM_Sender, HBM_Receiver and Fault_Recovery. HBM_Sender is responsible for sending HBM periodically to notify its information about network and HBM_Receiver is responsible for receiving a HBM from other nodes to update the topology according to the status of network. The HBM_Receiver determines a fault in a path if two consecutive HBMs not received on it. Once the network fault is detected, Fault_Recovery module adopts the alternative path to transmit the data.

Eq. 1 shows the failure over time T_{fot} in subnet, where $T_{hbrepeat}$ is heartbeat repetition interval and T_{lat} is delay latency (includes heartbeat transmission latency).

$$T_{fot} = 2 \times T_{hbrepeat} + T_{lat} \quad (1)$$

If we increase the number of nodes in a subnet, the heartbeat repetition interval should also be increased. This results in, however, increasing the fault recovery time because the node detects the faults when it has not received two consecutive heartbeats.

In proposed RSAF scheme, we can control the failure over time within a proper value by limiting the number of nodes in a subnet.

B. Inter-subnet Fault Detection: We adopt our previously proposed solution in order to exchange the information outside the subnets [1][3]. RSAF manages two master nodes in each subnet for inter-subnet fault recovery: Primary Master Node (PMN) and Secondary Master Node (SMN). PMN is an active master node and SMN is a standby master node, in case if PMN fails then SMN can recover the fault.

A node in RSAF network has two modules for inter-subnet fault handling: Master_Election and Notify_Fault.

Master_Election module of each node is responsible for managing master node in its belonging subnet. PMN is elected based on a voting and time-out mechanism. When subnet starts up or all master nodes of subnet fail, Master_Election begins master election procedure. First, it broadcasts a VOTE message to all nodes and starts a timer. When the timer expires, every node will collect the VOTE message. Second, the node with smallest IP will become the PMN, and node that have second smallest IP will become SMN.

Then the elected nodes (PMN and SMN) will broadcast a WIN message periodically to all other nodes in the subnet. The remaining candidate nodes will remain in WAIT status until they receive the WIN message commanding them to become NORMAL nodes. When the NORMAL node does not receive two consecutive WIN messages, it detects fault of master node.

Notify_Fault modules of PMNs communicate each other using IP packets to exchange the subnet status only when fault occurs. When a fault occurs in a subnet, master nodes can detect it by heartbeat mechanism described in section 4.A and notify to other master nodes to recover from the fault on the inter-subnet communication path. When master node receives notification from other master nodes, it in turn sends notification to nodes in its subnet to recover from the fault.

V. EVALUATION

In this section, we present comparison of RSAF with existing Fault Tolerant Schemes SAFE and DCell.

Table 2. Comparison of RSAF with SAFE and DCell

<i>RSAF</i>	<i>SAFE</i>	<i>DCell</i>
Node within a subnet having 4 routes to reach to destination	Node within a subnet having 4 routes to reach to destination	Node within a DCell having only 1 route
Inter-subnet connectivity is using a recursive structure so all the subnets connect to different subnets and different groups to provide maximum alternative paths.	Inter-subnet connectivity is done as; one of the subnet is selected as central subnet and all other subnets connected to that subnet.	DCell connectivity is dealt in the way that each node of a DCell is directly connected to Nodes of next DCells. So DCell provides large number of alternative paths
A cost-effective solution. In this cost increases linearly with respect to number of subnets.	Cost effective solution, but not providing much alternative paths	Higher Cost solution, as the structure grows the number of links increases quadratically. That's why wiring cost is high in DCell

As shown in Table 2, RSAF provides more connectivity than existing SAFE Scheme and less than DCell. But RSAF is a cost effective solution than DCell because the number of links in DCell is increasing in quadratic way. Hence, as the network grows, the cost also increases quadratically. But in RSAF the cost is linearly increasing with respect of the number of subnets in network.

Therefore, RSAF is a better solution for fault tolerance in large-scale networks because it provides maximum alternative

paths with limited number of links. Also, RSAFE is a cost-effective solution because the growth of network will not brings a big difference in the cost of implementation.

VI. CONCLUSION

In this paper, we presented a new scheme for Fault Tolerant Ethernet in large scale network called Recursive Scalable Autonomous Fault Tolerant Ethernet (RSAFE). In RSAFE, there are 3 basic building blocks; subnets, groups and levels. The recursive structure building is done by using Algorithm 1 where the subnets are constructed by combining multiple nodes and switches, groups are constructed by combining multiple subnets, and levels are constructed by combining multiple groups. Further building is done recursively, that each level construction is based on the previous level. RSAFE uses heartbeat mechanism to detect and recover the faults within a subnet. There two master nodes are elected; one as Primary Master Node (PMN) which acts as active master node and second as Secondary Master Node (SMN) acts as standby master node. Master node is responsible for exchanging the status updates of a subnet with other subnets. At the end comparison of previous schemes with RSAFE is done in Table 2, which shows that RSAFE is a cost effective solution for fault tolerance in large-scale networks.

VII. ACKNOWLEDGEMENT

This work was supported by Defense Acquisition Program Administration and Agency for Defense Development under the contract UD070019AD.

REFERENCES

- [1] K.Y Kim, Y.S Ryu, J.M Rhee, and D.H Lee, "*SAFE: Scalable Autonomous Fault-tolerant Ethernet*," Proc. of the 11th International Conference on Advanced Communication Technology (ICACT 2009), pp. 365-369, Feb. 2009.
- [2] H.A Pham, J.M Rhee, S.M Kim, and D.H Lee, "*A Novel Approach for Fault Tolerant Ethernet Implementation*," Proc. of the 4th International Conference on Networked Computing and Advanced Information Management (NCM 2008), Vol. 1, pp. 58-61, Sep 2008.
- [3] H.A Pham, J.M Rhee, Y.S Ryu, and D.H Lee, "Performance Analysis for a Fault-Tolerant Ethernet Implementation based on Heartbeat Mechanism," Proc. of 41th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, June, 2011.
- [4] S. Song, J. Huang, P. Kappler, R. Freimark, J. Gustin, and T. Kozlik, "*Fault-Tolerant Ethernet for IP-Based Process Control Networks*" Proc. of the 25th Annual IEEE International Conference on Local Computer Networks (LCN'00), pp. 116-125, Nov 2000.
- [5] H.A Pham, J.M Rhee, Y.S Ryu, and D.H Lee, "*An Adaptive and Reliable Data-Path Determination for Fault-Tolerant Ethernet Using Heartbeat Mechanism*". In Proc. of the 5th International Conference on Computer Sciences and Convergence Information Technology (ICCIT 2010), pp. 440-444, Dec 2010.
- [6] Chuanxiong Guo, Haito Wu, Kun Tan, Lei Shei, Yongguang Zhang, Songwu Lu. "*Dcell: a scalable and fault-tolerant network structure for data centers*" ACM SIGCOMM 2008.
- [7] Honeywell Automation and Control Solutions <http://hpsweb.honeywell.com/Cultures/en-US/Products/Systems/ExperionPKS/FaultTolerantEthernet/default.htm>
- [8] K.Y. Kim, S.M. Kim, J.M. Rhee, and Y.S. Ryu, "A Leader Management Scheme for Fault-Tolerant Multiple Subnets," Proc. of International