國立交通大學

電機學院 電信學程

碩士論文

使用虛擬網路概念之
高可靠度乙太接取網路架構
Virtual-Network-Based Highly Available Ethernet Access Network

研究生：易智超

指導教授：廖維國 博士

中華民國九十八年六月

使用虛擬網路概念之

高可靠度乙太接取網路架構

Virtual-Network-Based Highly Available

Ethernet Access Network

研究生：	易智超	Student:	Chih-Tsao Yi
指導教授：	廖維國 博士	Advisor:	Dr. Wei-Kuo Liao

國立交通大學
電機學院 電信學程
碩士論文

A Thesis
Submitted to College of Electrical and Computer Engineering
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master of Science
In
Comminication Engineering
June 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年六月

使用虛擬網路概念之
高可靠度乙太接取網路架構

學 生： 易智超　　　　　　　　　　指導教授： 廖維國 博士

國立交通大學　電機學院　電信學程碩士班

中文摘要

乙太網路被視為佈建與接取服務網路的一項關鍵技術。它具有，高擴充性、高頻寬、低價位與便於部署的優點。不過，乙太網路的主要缺點是它缺乏電信設備服務等級的效能。論文中，我們提出了可以符合電信設備等級的高容錯乙太服務接取網路架構，藉由我們提出的建議案亦可大大降低乙太接取網路的佈建成本。

因此，我們提出了以虛擬網路概念為基礎的高容錯乙太服務接取網路架構。雖然虛擬網路並非新的概念，但就我們所知、我們是第一個提出並將之應用於支援電信設備服務等級的乙太服務接取網路。為了達到服務不中斷的電信服務等級目的，我們由*一加一備援*架構來建構我們提出的高容錯乙太服務接取網路架構。值得一提的是，我們所提出的建議案是非常簡易且僅用軟體就可完全實踐。

另外、我們藉要求使用支援最基本之*虛擬區域網路*功能的第二層交換器來建構高容錯乙太服務接取網路。這些第二層交換器在一般市售價格裡，大約七千新台幣。在我們提出的架構裡，我們能服務多達 512 用戶於乙太服務接取網路裡。。若應用於專屬網路、則可兼具延展性的需求。

為了評斷我們提出的架構是屬於電信設備服務等級的架構，我們使用了”*連續時間*之馬可夫鏈 ”來做分析與驗證。透過分析、我們可以發現在連續時間之馬可夫鏈判定下，我們所提出的虛擬網路概念之高容錯乙太服務接取網路架構是可以符合電信設備服務等級。

Virtual-Network-Based Highly Available Ethernet Access Network

Student: Chih-Tsao Yi Advisor: Dr. Wei-Kuo Liao

Degree Program of Electrical and Computer Engineering

National Chiao Tung University

ABSTRACT

Ethernet is envisioned as a key technology to recent deployments of service network. Ethernet features in high scalability, very high bandwidth provisioning, inexpensive, and easy deployment. However, the major drawback of Ethernet is its lack of supporting carrier grade services. In this thesis, we consider to support carrier grade service for Ethernet access network. In addition, we aim at further reducing the switch cost by requiring less functionality in switches.

To do so, we propose a *virtual-network-based access architecture*. The concept of virtual network is not new. Nevertheless, as far as we know, we are the first to extend its usage to study the fault-tolerant-related issues in Ethernet access networks.

In this thesis, we elaborate on determining the configuration of virtual networks so that the service can be recovered when a fault occurs. Our goals for the virtual network configuration are basically requiring least VLAN functionalities in switches, enabling fast service recovery process upon an occurrence of faulty link/switch, and sustaining the scalability of access network. Our proposal basically employs the one-plus-one backup principle to enable fault tolerance.

Our proposed protocol is very simple and can be carried out entirely by software approach. Besides, our proposed architecture only needs minimum VLAN function for switches. Traditional application in our proposal, we can service up to 512 customer networks in the access network. In military network and the proprietary network, the scalability would not be the issue.

To evaluate the degree of carrier grade service of our proposal, fault-tolerant-related behavior is modeled by a continuous-time Markov chain (CTMC). Through the analysis, we also aim at discovering the rule of thumb to keep high degree of carrier grade.

誌 謝

當碩士論文寫到此頁，代表我的碩士生涯將要正式落幕了，兩年的研究所時光看似漫長，實則如過眼雲煙轉瞬即逝。雖然二年時間匆匆而逝，但已然在我的人生之中留下許多不可磨滅的奪目光采。

首先最要感謝的是廖維國老師，從入學到畢業，整整二年寒暑從未間斷的辛勤教導，吾以為學如逆水行舟，而老師則不斷帶著我一路勇往直前奮力不倦，不僅於課業上給予指導，更教會我做人做事的態度。如今輕舟悄然已過萬重山，老師對我的照顧及付出，點滴在心，無限感激。 另外，還要感謝口試委員李程輝教授、首特科技副總范國寶博士所提供的許多寶貴意見。學生獲益匪淺，藉由你們的蒞臨與指導下，使本篇畢業論文能更加完善。

終於完成一篇屬於自己的論文、我內心的雀躍真是筆墨無法形容的。回顧這兩年的研究所生涯，不論是期末考的挑燈苦讀、還是跟實驗室的大家一起歡樂等等，都是我的美好回憶。此時此刻，我的內心真是百感交集，一方面正期待完成碩士學業的未來生活，另一方面也對即將離開待了二年的交通大學感到不捨。尤其是充滿歡樂的 812b 實驗室裡，梓洋、永裕、宇哲、孟哲，振翔都是我研究所同甘共苦的好夥伴，一起討論課業，一起打屁嬉鬧。大家給予我這溫暖的友情，是我一輩子也不會忘記的。

最後、要真心的感謝我的妻子靖翎，若不是她一路陪伴我、無怨無悔地默默在背後支持我、鼓勵我。遇到難題適時給予中肯的建議，學業有所成就或延遲時，也沒我一起開心與擔心。若沒有了她的相伴與體諒，我不可能會有今天這樣的小小成果。

謝謝大家。

2009 年 6 月

Contents

List of Tables

List of Figures

Chapter 1 – Introduction

Ethernet is envisioned as a key technology to recent deployments of service network. Ethernet features in high scalability, very high bandwidth provisioning, inexpensive, and easy deployment [1][2][3][4]. However, the major drawback of Ethernet is its lack of supporting carrier grade services.

In this thesis, we consider to support carrier grade service for Ethernet access network. Similar to earlier work on proposing an Ethernet access architecture for supporting continuous IPTV services [5], we target at software approach to enable the service continuity, but for general classes of services. In addition, we aim at further reducing the switch cost by requiring less functionality in switches.

To do so, we propose a *virtual-network-based access architecture*. The concept of virtual network is not new. It has been used to enable load balance, deadlock avoidance, and fault tolerance in networks such as inter-connected networks and local area networks. Nevertheless, as far as we know, we are the first to extend its usage to study the fault-tolerant-related issues in Ethernet access networks.

In this thesis, we elaborate on determining the configuration of virtual networks so that the service can be recovered when a fault occurs. Our goals for the virtual network configuration are basically requiring least VLAN functionalities in switches, enabling fast service recovery process (< 50ms) upon an occurrence of faulty link/switch, and sustaining the scalability of access network. Our proposal basically employs the one-plus-one backup principle to enable fault tolerance and includes the way how the virtual networks are formed, a protocol to let the switches choose the virtual network to convey the services in a cooperative way, and VLAN configuration to form the virtual networks.

It is important to stress that our proposed protocol is very simple and can be carried out entirely by software approach. Besides, our proposed architecture only needs minimum VLAN function for switches, i.e., port base VLAN and IEEE802.1Q VLAN tagging. Such switches regard as very cheap switch, as we know the price of them around 7,000 NTD. Traditional application in our proposal, we can service up to

512 customer networks in the access network. In this example, there could be 168 access switches; the price to build up entire network is 1,176,000 NTD. Other case, 128 customer networks in the Access Network, it only require 294,000 NTD to build up the entire network. For both cases, the price per customer network is just 2,297 NTD. In our proposal, we also advise use the fault tolerant architecture in the military network and the proprietary network. We can support more customer networks by extending the VLAN filed to attach more information. In such network, the protocol format can be modified by the operator.

Regarding the concern of scalability, we note that a VLAN identifier in 802.1q is limited to distinguish 4096 networks (i.e. a single VLAN identifier is 12-bit wide) [7]. In access network, a customer network needs a VLAN id to run the district service and thus an access network at most supports up to 4096 customer networks. However, it is a kind of challenge for us to reach such high scalability. As it becomes clearly in later chapter, we need to encode all the information of virtual network identifier, switch identifier, customer network identifier in this 12-bit VLAN identifier field. With such a challenge, our proposal can still support up to 512 customer networks.

To evaluate the degree of carrier grade service of our proposal, we adopt the most common stochastic approach. In such approach, the network fault-tolerant-related behavior is modeled by a continuous-time Markov chain (CTMC). CTMC is a State-base model with transition rates between states. The continuous-time Markov chain (CTMC) was used to determine the steady-state availability [6] of the proposed virtual-network-based fault tolerant architecture. Through the analysis, we also aim at discovering the rule of thumb to keep high degree of carrier grade.

The rest of the thesis is organized as follows. In chapter 2 we introduce the Virtual Network concept. The VLAN Application is presented in chapter 3. And the VLAN Application Architecture Discussion is described in chapter 4, followed by conclusion in chapter 5.

Chapter 2 –Virtual Network Concept

In this chapter we will introduce the concept of Virtual Network. We generalize and categorize as three generic definitions of Virtual Network. They are Path-type Virtual Network, Broadcast-type Virtual Network and Fork-type Virtual Network (See the Figure 1). Follow we discuss three types virtual network, we can get the concept for developing the virtual-network-based fault tolerant architecture.

2.1 Virtual Network

A virtual network is defined by a set of virtual network nodes and a set of virtual network paths connecting the nodes. The virtual network path defines a path that consisting of one or more physical links between two virtual network nodes. Several virtual networks can co-exist in a physical network.

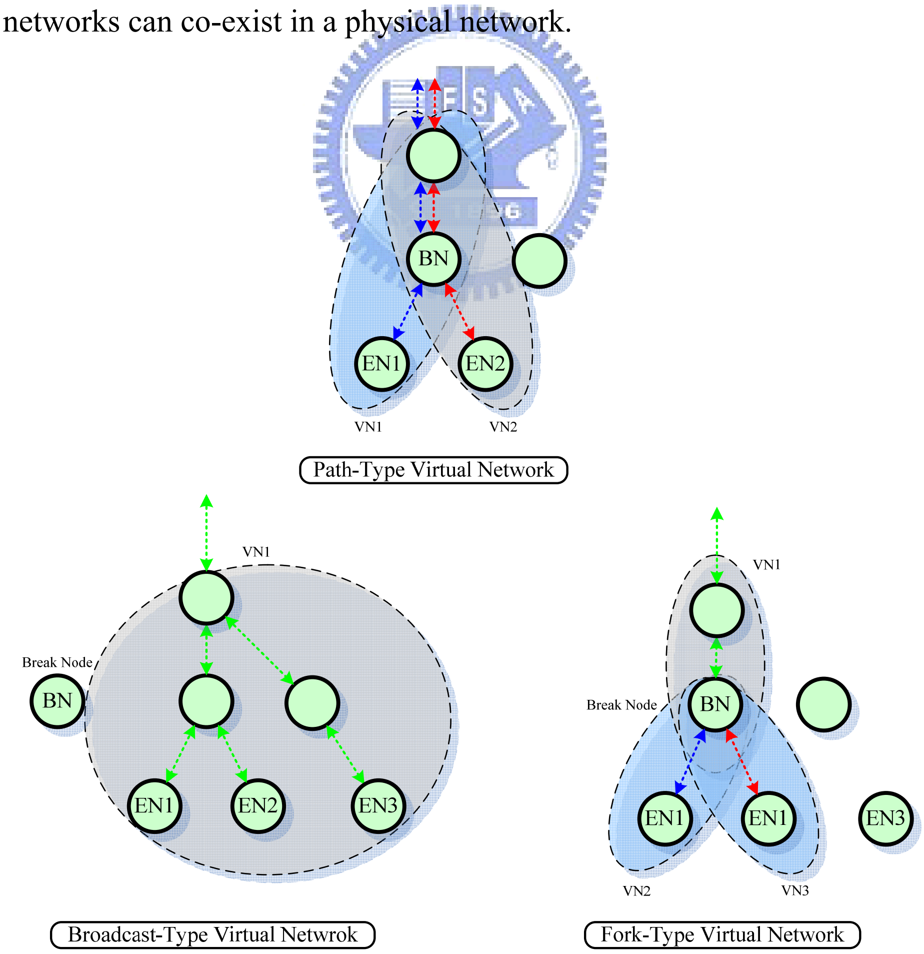
BN
EN1
EN2
VN1
VN2
Path-Type Virtual Network
VN1
Break Node
BN
EN1
EN2
EN3
Broadcast-Type Virtual Netwrok
VN1
Break Node
BN
EN1
EN1
EN3
VN2
VN3
Fork-Type Virtual Network

Figure 1 Three Types Virtual Network

2.2 Path-Type Virtual Network

Path-type virtual network indicate unique path from source to end node. We can see it at the Figure 1, while source deliver traffic to EN1 and EN2, it will only toward via VN1 and VN2. Active the path-type VN, source have to indicate the unique path information for the virtual network. The information has to include the BN (break node) and EN (end node).

For the application, each requirement has a unique virtual path for it. Every requirement is pre-defined a path. We obtain the less effort for source to distinguish virtual path while the requirement be required.

Counter the amount for path-type virtual network in entire network. It can be relate to the elements in entire network. For an example, while the network can support 128 customer networks, 128 end nodes and 42 virtual network nodes. And only one virtual network node exist in the path that between source to the end node. In such case, the amount of virtual network is 5376. This method can not obtain the advantage of scalability. We could use path-type virtual network in small network architecture.

2.3 Broadcast-Type Virtual Network

Broadcast-type virtual network indicate break nodes (BNs) that source will not deliver traffic pass though them. Source follows the broadcast-type virtual network to deliver traffic to end nodes just like the broadcast function to deliver the traffic from source to end node (EN). The end node could be indicated at the broadcast-type virtual network or not. The function for broadcast-type virtual network to indicate the end node information is belonging optional. We can see it at the Figure 1, while source deliver traffic to end nodes, it will only depend on the break node. To active the broadcast-type virtual network, source have to indicate the information for the break nodes; to indicate the information of end nodes belong optional that depend on the requirement of operator.

For the applications, we can use the nature for source delivers traffic to all end nodes without break node. We can also use the broadcast nature for some specific purposes, like multicast service, IPTV [5][8].

Counter the amount for broadcast-type virtual network in entire network. It can be count with amount of elements in entire network. For example, while the network can support 128 customer networks, it includes 128 end nodes and 42 virtual network nodes. The amount of virtual network is 42.

2.4 Fork-Type Virtual Network

Fork-type virtual network indicate both break nodes (BNs) and end node (ENs). Source distinguishes which virtual network nodes be as break nodes and decide virtual network with the information of break nodes for source to follow the unique virtual path in the fork-type virtual network and deliver traffic unto last break nodes. The path will depend on the information of end node. While the traffic delivers unto last break node then fork-type virtual network will depend on the forwarding information of end node to deliver the traffic from last break node to end node. We can see it at the Figure 1, while source deliver traffic unto last break node, it depend on the forwarding information of end node. After this, the streams are delivering from last break node to end node. To active the fork-type virtual network, source has to indicate the information for the break nodes and the information of end nodes, it depend on the requirement of end nodes. Via the fork-type virtual network capability, until the traffic is delivering to the end node, no other virtual network node that without relationship will receiver these streams.

For the applications, we can use the nature that source delivers traffic to all end nodes without break node. We can use it as the fault tolerance virtual network and to backup the fault nodes. We also can allocate particular break nodes in the path of each end node. By this way, attach more than one break nodes, to improve the bandwidth efficiently before traffic deliver unto last break node. Bandwidth utilization rate of virtual network node can be control by the source.

Counter the amount for fork-type virtual network in entire network. It can be count with amount of elements in entire network. For example, while the network can support 128 customer networks, 128 end nodes and 42 virtual network nodes. The amount of virtual network is 128.

2.5 Virtual-Network-Based Fault Tolerant Architecture

In this thesis, we consider applying fault tolerance system architecture via virtual network concept, see the Figure 2. We use fork-type virtual network concept to make the fault tolerance system as a predicted backup path for a fault node. Make break node (BN) is same as the backup switch and end node (EN) is same as the customer network in the fault tolerance system. There would be several customer networks (ENs) and one backup switch (BN) for a fault tolerance requirement. We plan the traffic flow will follow the path in the fork-type virtual network to skip the fault switch then deliver traffic by break node and forward to customer networks.

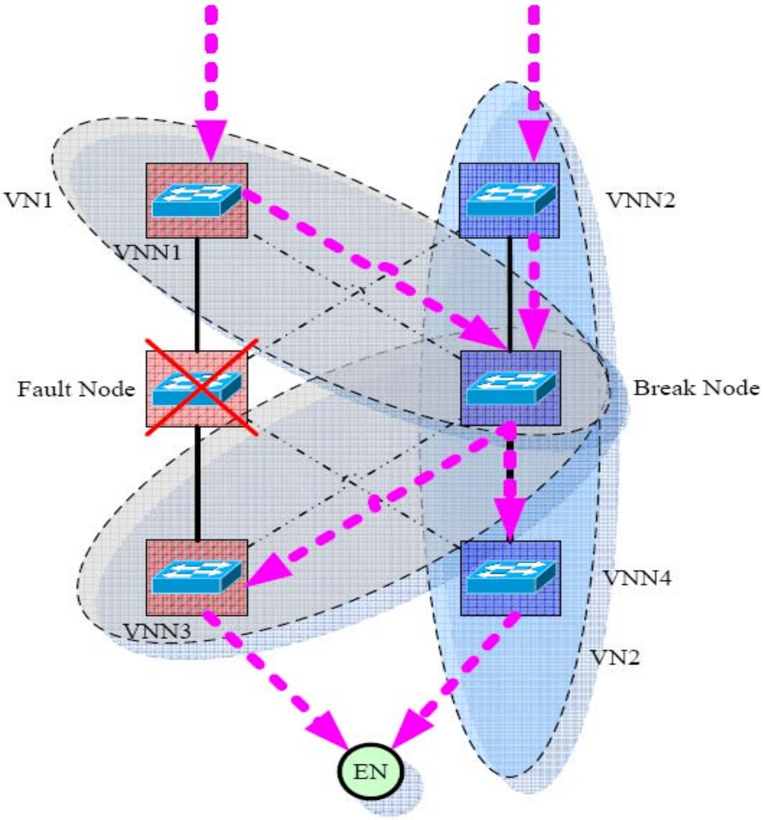

Figure 2 Planed Virtual-Network-Based Fault Tolerant Architecture

From comparison at above sections, we can conclude that [Fork-type Virtual Network] is the best method for developing virtual-network-based fault tolerant architecture. The fork-type virtual network has two advantages. 1) *Best bandwidth efficient* and *No effort* for other nodes. Until the traffic is delivering to the customer networks, no other virtual network node that without relationship will receiver these streams. 2) *Bandwidth utilization rate* of virtual network nodes *can be control* by source. Source attaches more than one break node and forwarding information for each EN in the traffic. By the way, the bandwidth utilization rate of each virtual network nodes can be control by the source.

While there is a fault switch (fault node), the source will distinguish and decide one fork-type virtual network to failover the traffic that duty of the fault switch. The traffic will be delivered on the path that unique path for source to the Break Node (BN). After it, the break node depend on the forwarding information that be attached in the stream to deliver to customer networks (ENs). (See the Figure 3)

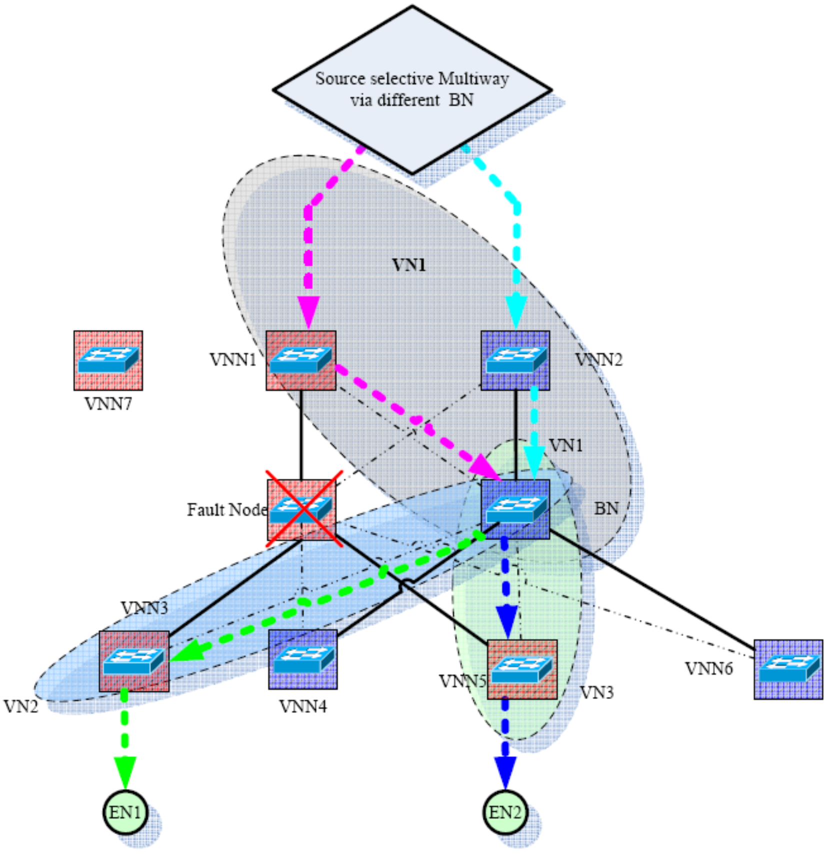

Figure 3 Fork-Type Virtual-Network-Based Fault Tolerant Architecture

Chapter 3 – VLAN Application

Base on the virtues of fork-type virtual network, we develop fork-type virtual-network-based fault tolerant architecture (See the Figure 3) and implement the broadcast-type virtual network as the advance requirement, such as IPTV.

Common VLAN implementation allows for any-to-any communication. Each host on the VLAN can communicate with any other host on that segment. VLAN are conventionally used to simplify network administration, improve security and limited broadcast domain of each VLAN. Increase the apparent bandwidth for network users. No chances for people gain access to the information that they are not authorized. VLAN does not pass broadcast traffic to nodes that are not part of the same VLAN, it is automatically reduces broadcasts in the network. VLAN give us a way of organizing and forming traffic the way we want to, etc.

We make our proposal be compatible with VLAN application and deviate from conventional VLAN application. We use VLAN to practice fault tolerance system. Base on the VLAN identifier filed have 12 bits. Raise an example for an access network that can support up 512 customer networks. Our proposal also can be used in such as the military network and the proprietary network while the protocol format can be modify by the operator. We can support more customer networks by extending the VLAN filed to attach more information.

3.1 Previous Work

In this thesis, we consider to support carrier grade service for Ethernet access network. Similar to earlier work on proposing an Ethernet access architecture for supporting continuous IPTV services [5]. The previous work is related to interconnect switches, a software design pattern and core protocol to coordinate the IGMP proxies in the redundant pair to maintain a consistent view of multicast membership information. Update any information in the table between and peer with the other process. Upon a status change of interconnect switches to reconfigure. The reconfigure will do the operation, such as modifying the forwarding database in the switch or VLAN configuration, and induce the specified channels from the upstream through IGMP messages.

3.2 VLAN Application Protocol

We target at software approach to enable the service continuity, but for general classes of services. In addition, we aim at further reducing the switch cost by requiring less functionality in switches.

We use the network architecture, one-plus-one fault tolerance architecture, in the earlier work [5] to practice our proposal. (Figure 4)

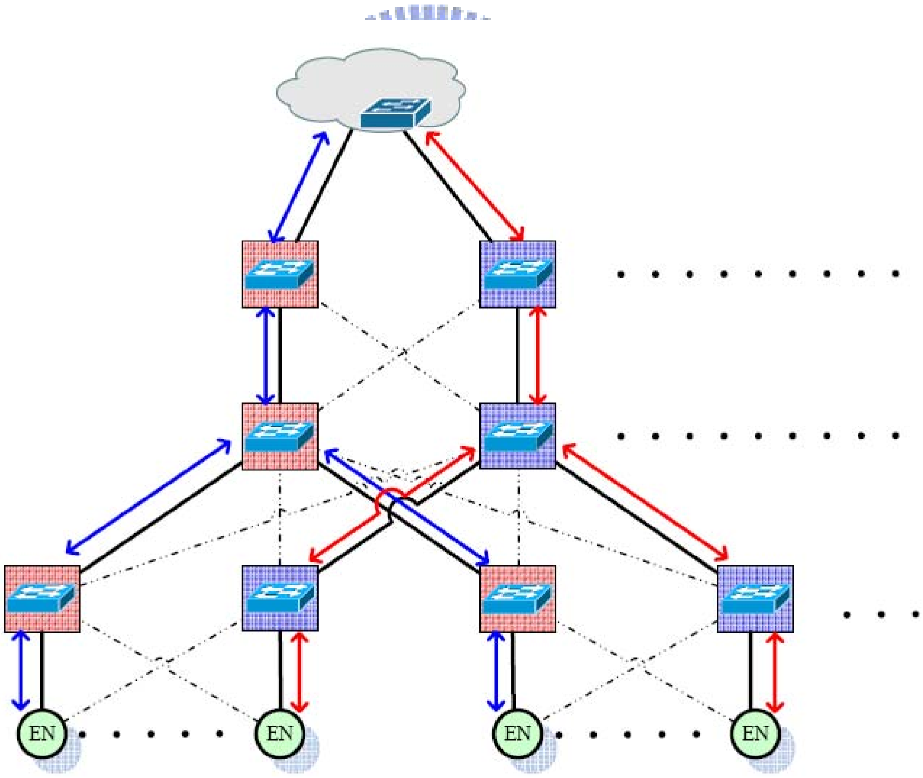

Figure 4 One-Plus-One Fault Tolerance Architecture

We trace the customer network as the End Node (EN), access switches as the virtual network nodes and Break Switch as the Break Node (BN) in the fork-type virtual network. We also attach identifier of each customer network that is same as the forwarding information of each customer network. We assign *hierarchical identifier* (See the Figure 5) to customer networks and virtual network nodes in fault tolerance system. Follow the relationship that between access switches and customer network, the identifiers for access switches in each level can carry by the identifier of customer network.

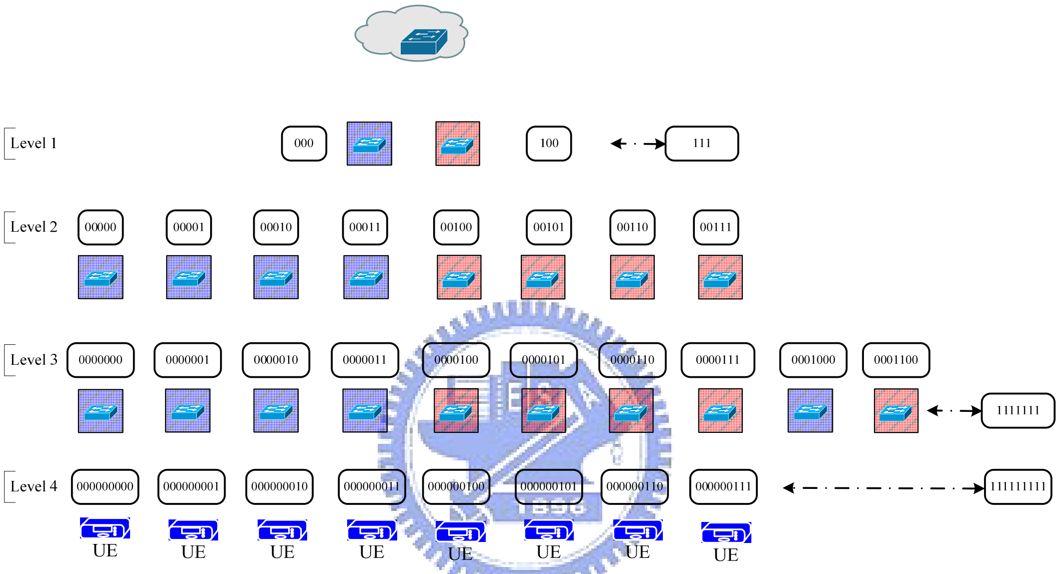

Figure 5 Hierarchical Identifier Number Assignment

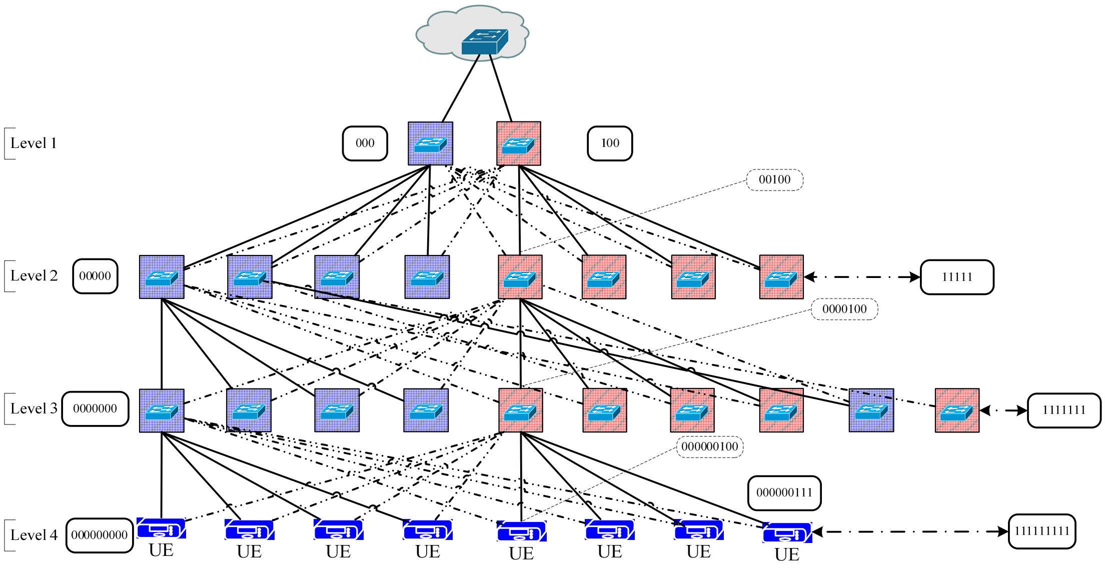

Figure 6 Physical Line Connections

The key for the IEEE 802.1Q is in its tags, VLAN identifier filed. We separate the VLAN identifier filed (Total 12 bits) as three parts, [Rbit+Lbit+Ibit], see the Figure 7.

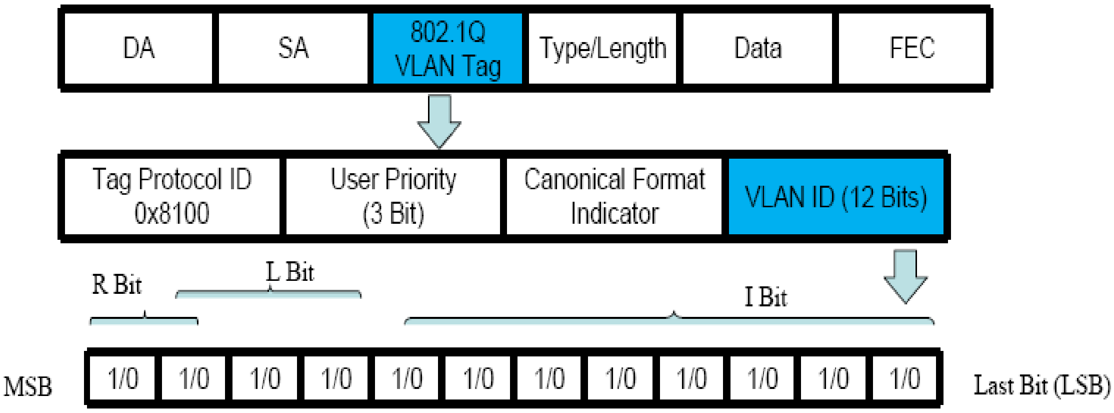

Figure 7 Protocol Format

R Bit (Requirement Type Bit): Indicate the stream belong [Backup requirement], [Advance requirement, Multicast service] or [Normally]. For normally streams, the R Bit is active as [00]. For backup requirement streams, the R Bit is active as “1”. For the advance requirement streams, the R Bit is active as “0 1”. While R Bit is active as “1” or “0 1” that mean the protocol be active, switch that receiver such stream will check L Bit first then compare I Bit to forward to customer networks.

L Bit (Levels Bit): Indicate levels of break nodes that execute the protocol requirement. There is unnecessary to allocate L Bit for first one level break switches (BNs) and customer networks (ENs). The reason for the first one is the source will distinguish the fault tolerance requirement and determine to deliver to the duty of the first one level break node. The reason for second is identifier of customer network is indicated in I Bit already.

I Bit (Information Bit): Indicate the forward information, identifier of customer network, in the VLAN stream. We assign *Hierarchical identifier* to customer network and virtual network node (access switches). By this way, the identifier of each level access switches that duty of customer network can get from Forward Information, identifier of customer network.

Base on the VLAN identifier filed have 12 bits. Rise an example for an access network that can support up to 168 access switches (Refer to Figure 5) which belong first to third levels and 512 customer networks that are belonging fourth level. In this example, first level has 4 pairs, second level has 16 pairs and the third level has 64 pairs. Finally, the fourth level, Customer networks, support to 512 customer networks.

The VLAN application protocol streams are generated by source. The streams replace the original streams by source and deliver to the customer networks while there are fault switches or to control the bandwidth utilization rate requirement to change the access switches in the fork-type virtual network.

3.3 VLAN Protocol Application

To give an example that support maximum customer networks, 512 customer networks, in our proposal. We introduce the VLAN traffic follow in our virtual-network-based fault tolerant architecture. Every switch have its own VLAN table to handle the VLAN stream flow, all the VLAN information in the VLAN table are pre-defined by the network operator. While there are access switches fail or the source detect there are over-utilization of capacity when other capacity is available in the access network [9] then source will depend on our protocol to change the VLAN identifier filed to meet the fail situation or for the performance optimization in terms of capacity utilization. The VLAN file is decomposed [Rbit+Lbit+Ibit]. The follow figure will introduce the below situations. First introduce the original VLAN traffic flow for source to customer networks (See the Figure 8). Follow is for a first level switch fails (See the Figure 9), a second level switch fails (See the Figure 10), a third level switch fails (See the Figure 11), first level and second level switches fail (See the Figure 12), first level and third level switches fail (See the Figure 13), second level and third level switches fail (See the Figure 14), and three levels switches fail (See the Figure 15). Finally introduce an example for requirement that over-utilization of capacity when other capacity is available to change the switches in the fork-type virtual network path. (See the Figure 16).

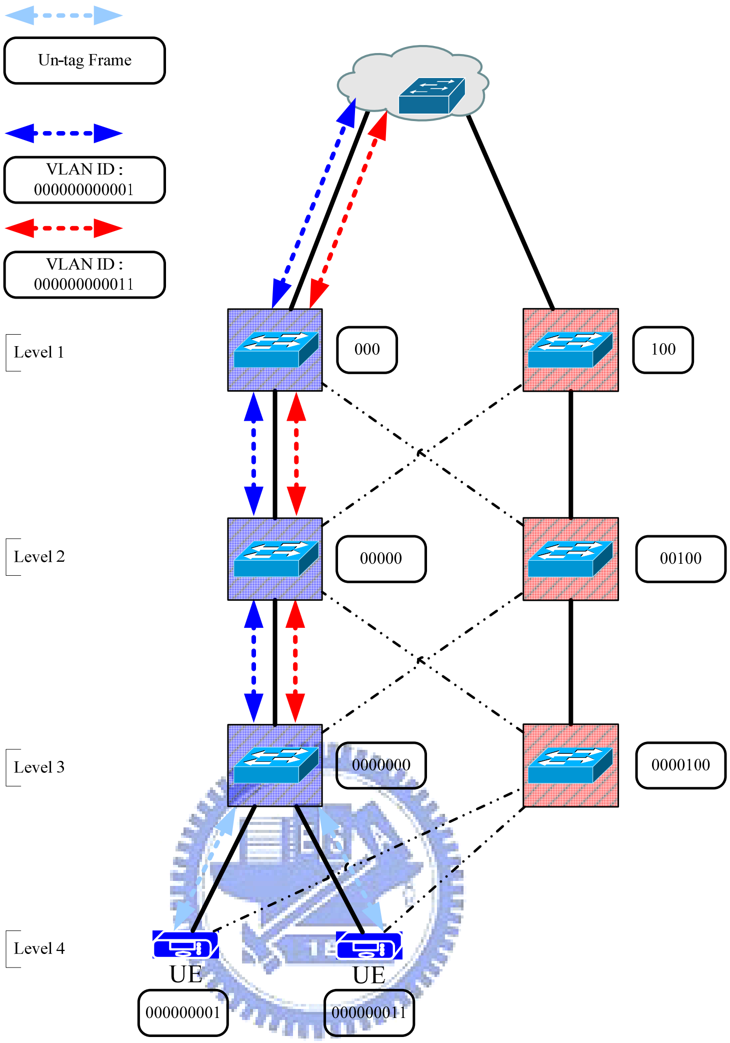

Figure 8 Original Traffic Flows

In the normal situation, the source will depend on the VLAN identifiers to deliver the traffic to the customer networks. The traffic flow will just flow between switches via the VLAN in the VLAN table of each switch.

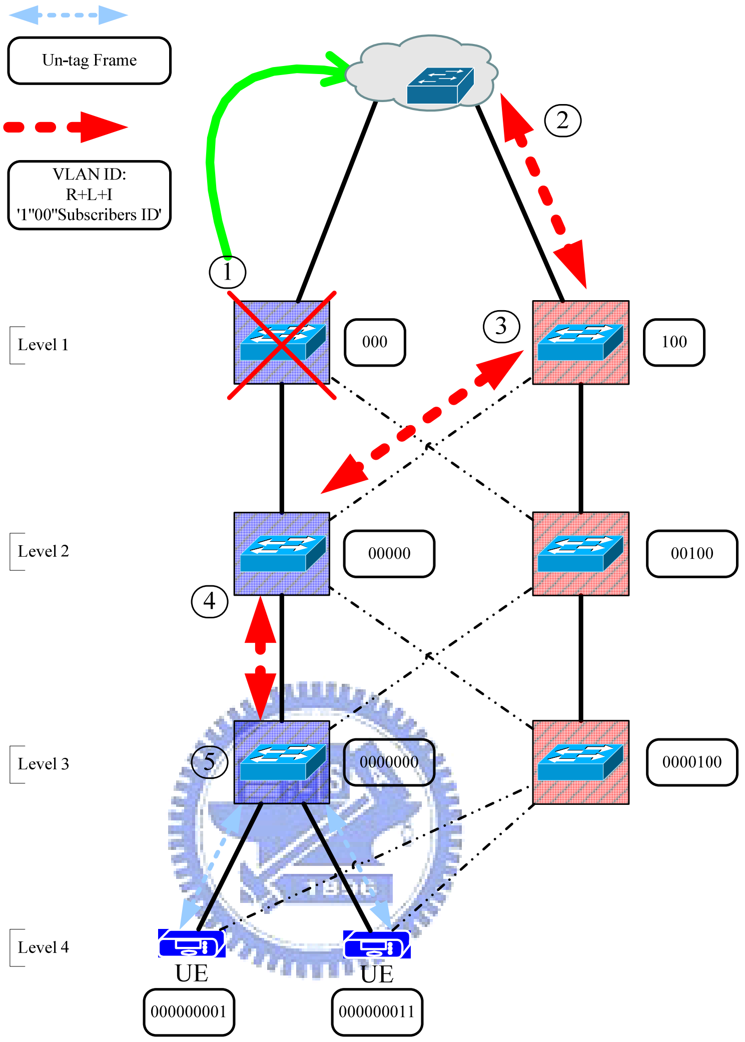

Figure 9 Traffic Flows of First Level Switch Fails

Step 1: First level switch fails, such event be noticed to Top Switch.

Step 2: Top Switch change the VLAN as R bit as '1' L Bit as '00' with I bit as the Customer networks ID. Top Switch delivers the streams to first level Switch, 100.

Step 3: First level Switch, 100, depends on the 'R+L' forward to second level Switch, 00000.

Step 4: Second level Switch, 00000, depend on the 'R+L' forward to third level Switch, 0000000.

Step 5: Third level Switch, 0000000, depends on I bit to forward the stream to Customer networks.

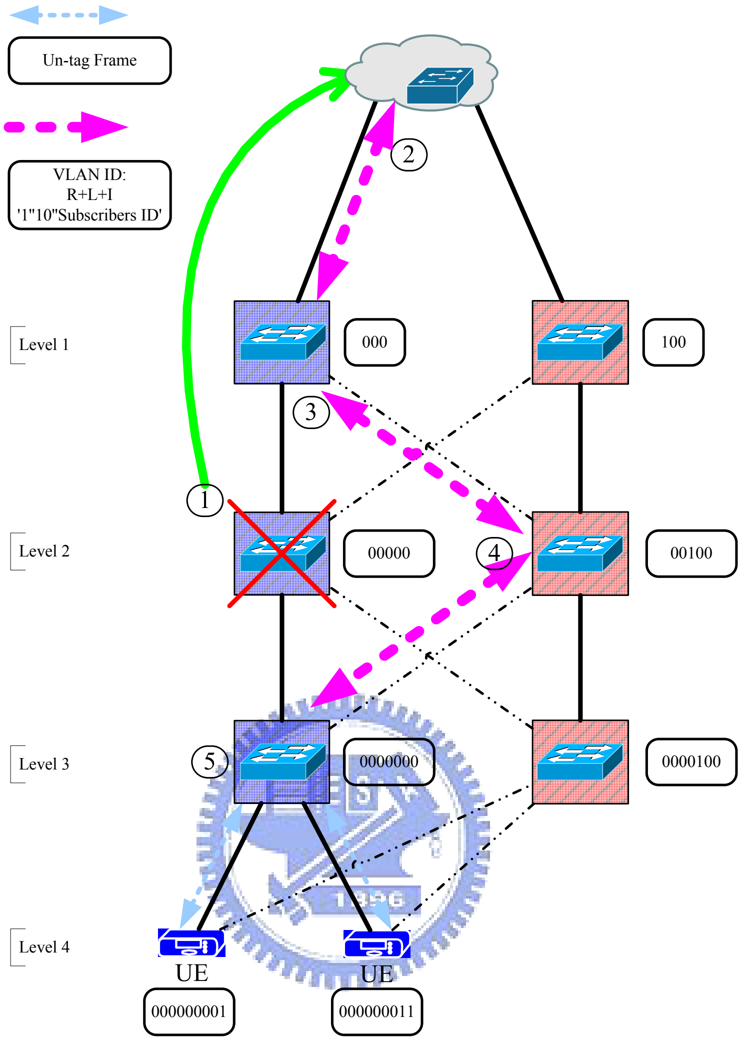

Figure 10 Traffic Flows of Second Level Switch Fails

Step 1: Second level Switch fails, such event be noticed to Top Switch.

Step 2: Top Switch change the VLAN as R bit as '1' L Bit as '10' with I bit as the Customer networks ID. Top Switch delivers the streams to first level Switch, 000.

Step 3: First level Switch, 000, depends on the 'R+L' forward to second level Switch, 00100.

Step 4: Second level Switch, 00100, depend on the 'R+L' forward to third level Switch, 0000000.

Step 5: Third level Switch, 0000000, depends on I bit to forward the stream to Customer networks.

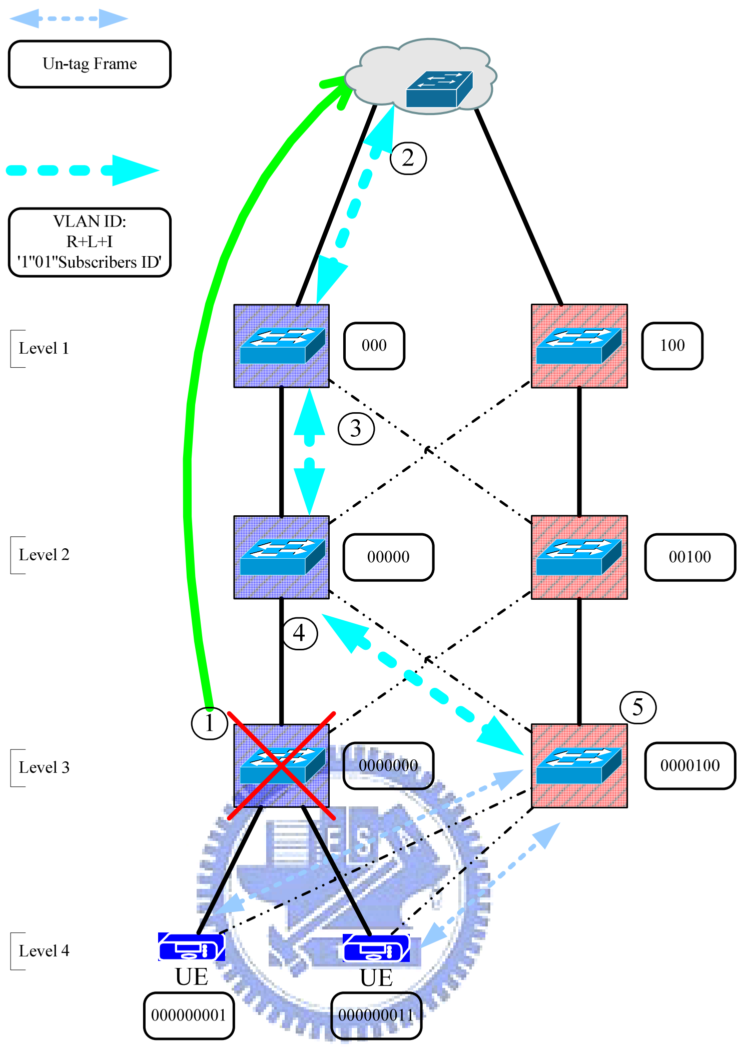

Figure 11 Traffic Flows of Third Level Switch Fails

Step 1: Third level Switch fails, such event be noticed to Top Switch.

Step 2: Top Switch change the VLAN as R bit as '1' L Bit as '01' with I bit as the Customer networks ID. Top Switch delivers the streams to first level Switch, 000.

Step 3: First level Switch, 000, depends on the 'R+L' forward to second level Switch, 00000.

Step 4: Second level Switch, 00000, depend on the 'R+L' forward to third level Switch, 0000100.

Step 5: Third level Switch, 0000100, depends on I bit to forward the stream to Customer networks.

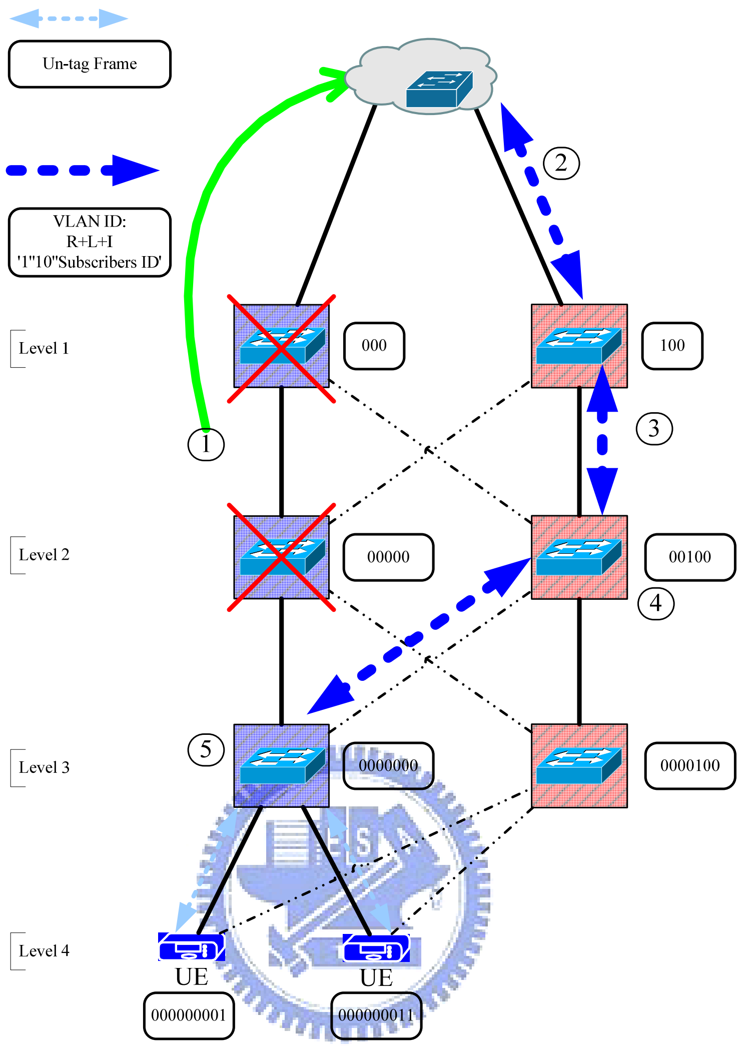

Figure 12 Traffic Flows of First Level & Second Level Switches Fail

Step 1: First level and second level Switches fail, such events be noticed to Top Switch.

Step 2: Top Switch change the VLAN as R bit as '1' L Bit as '10' with I bit as the Customer networks ID. Top Switch delivers the streams to first level Switch, 100.

Step 3: First level Switch, 100, depends on the 'R+L' forward to second level Switch, 00100.

Step 4: Second level Switch, 00100, depend on the 'R+L' forward to third level Switch, 0000000.

Step 5: Third level Switch, 0000000, depends on I bit to forward the stream to Customer networks.

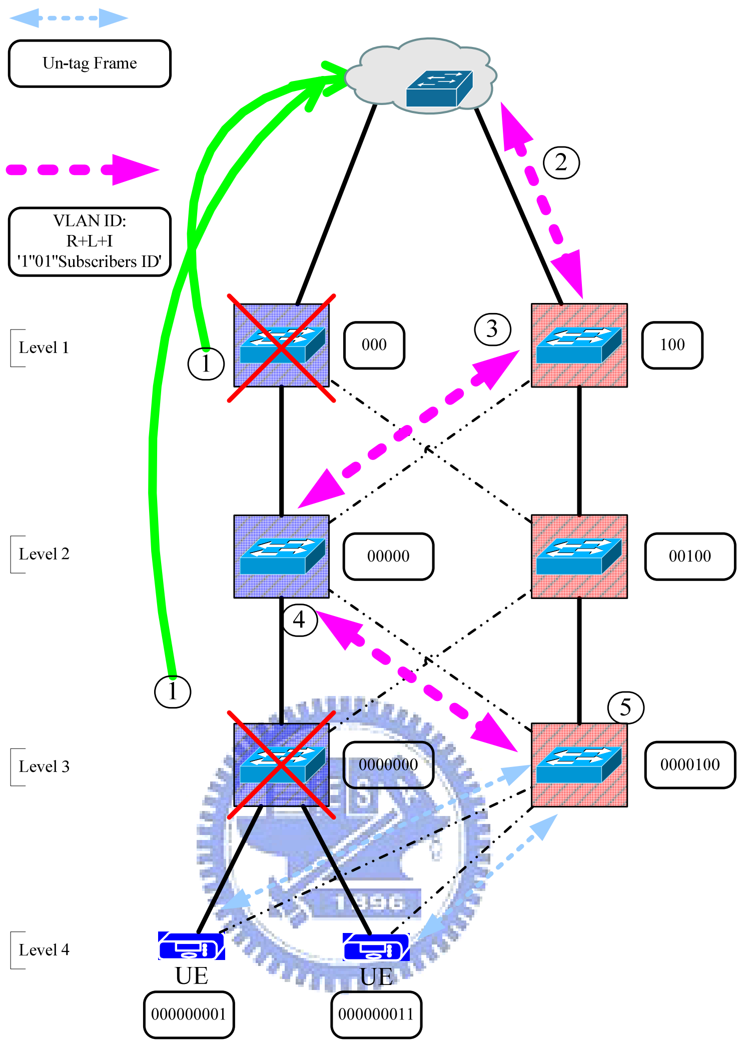

Figure 13 Traffic Flows of First Level & Third Level Switches Fail

Step 1: First level and third level Switches fail, such events be noticed to Top Switch.

Step 2: Top Switch change the VLAN as R bit as '1' L Bit as '01' with I bit as the Customer networks ID. Top Switch delivers the streams to first level Switch, 100.

Step 3: First level Switch, 100, depends on the 'R+L' forward to second level Switch, 00000.

Step 4: Second level Switch, 00000, depend on the 'R+L' forward to third level Switch, 0000100.

Step 5: Third level Switch, 0000100, depends on I bit to forward the stream to Customer networks.

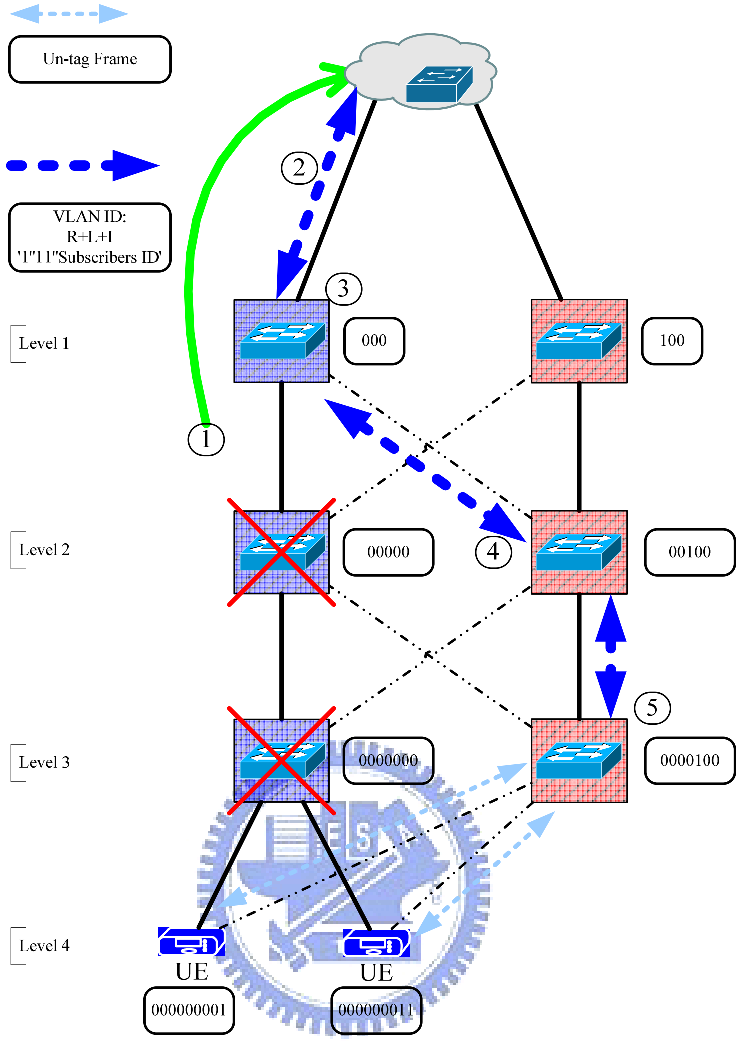

Figure 14 Traffic Flows of Second Level & Third Level Switches Fail

Step 1: Second level and third level Switches fail, such events be noticed to Top Switch.

Step 2: Top Switch change the VLAN as R bit as '1' L Bit as '11' with I bit as the Customer networks ID. Top Switch delivers the streams to first level Switch, 000.

Step 3: First level Switch, 000, depends on the 'R+L' forward to second level Switch, 00100.

Step 4: Second level Switch, 00100, depend on the 'R+L' forward to third level Switch, 0000100.

Step 5: Third level Switch, 0000100, depends on I bit to forward the stream to Customer networks.

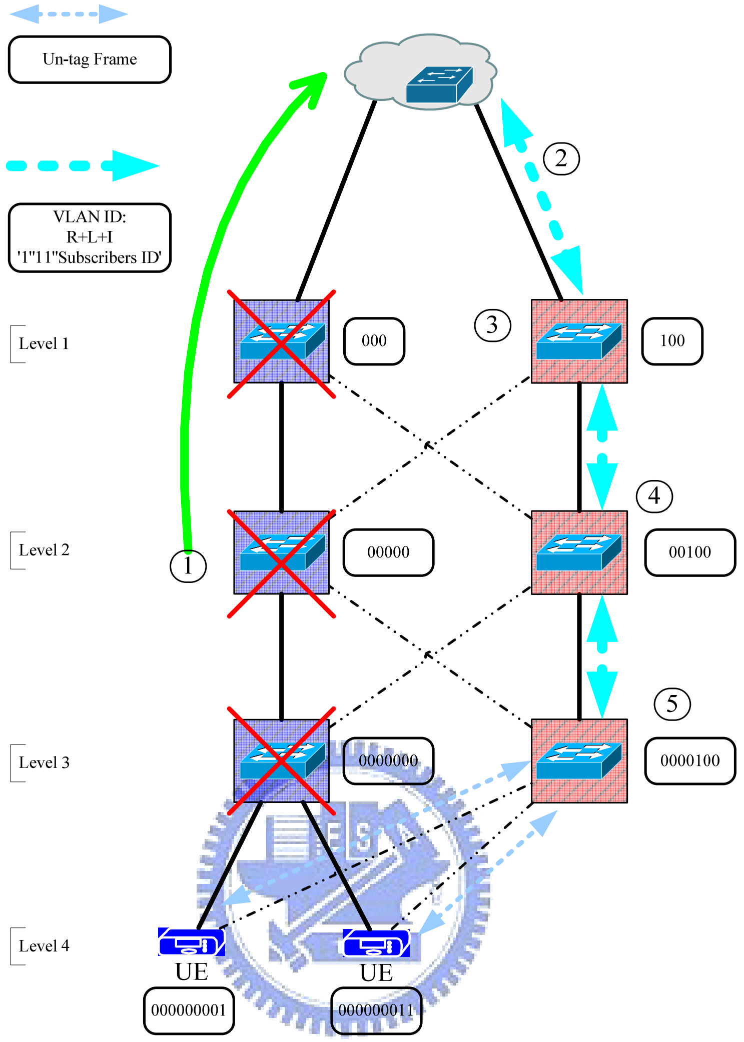

Figure 15 Traffic Flows of Three Levels Switches Fail

Step 1: First level, second level and third level Switches fail, such events be noticed to Top Switch.

Step 2: Top Switch change the VLAN as R bit as '1' L Bit as '11' with I bit as the Customer networks ID. Top Switch delivers the streams to first level Switch, 100.

Step 3: First level Switch, 100, depends on the 'R+L' forward to second level Switch, 00100.

Step 4: Second level Switch, 00100, depend on the 'R+L' forward to third level Switch, 0000100.

Step 5: Third level Switch, 0000100, depends on I bit to forward the stream to Customer networks.

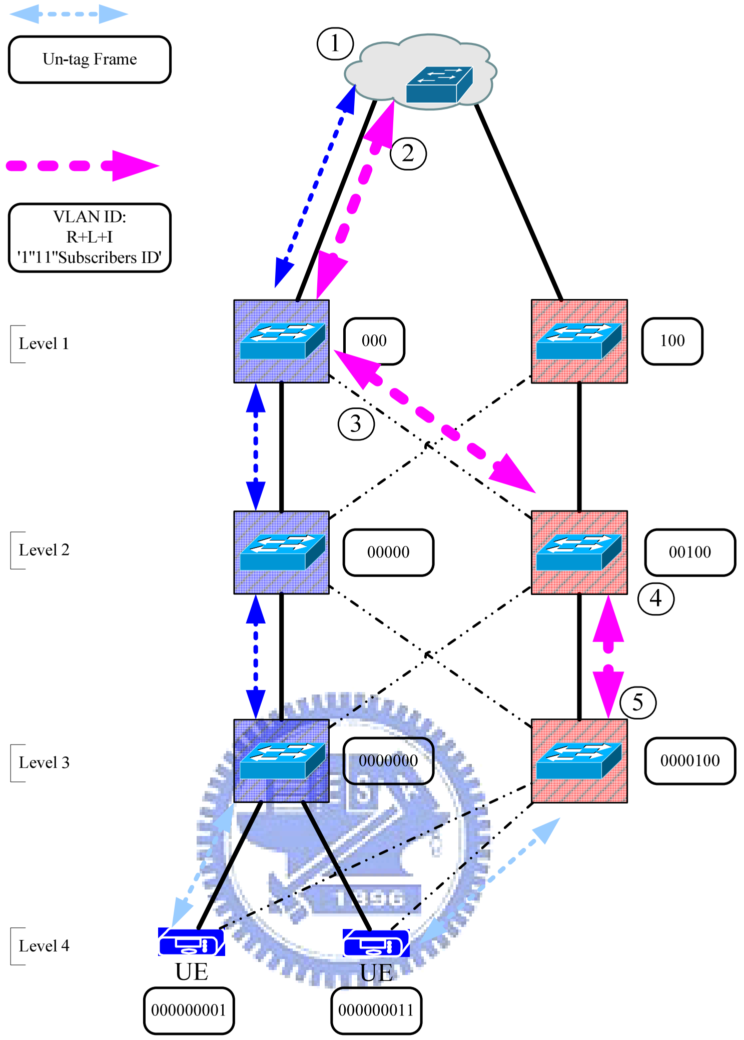

Figure 16 Over-Utilization Requirement

Step 1: Source detect second level and third level Switches traffic overloading.

Step 2: Top Switch change the VLAN as R bit as '1' L Bit as '11' with I bit as the Customer network, 000000011. Top Switch delivers the streams to first level Switch, 000.

Step 3: First level Switch, 000, depends on the 'R+L' forward to second level Switch, 00100.

Step 4: Second level Switch, 00100, depend on the 'R+L' forward to third level Switch, 0000100.

Step 5: Third level Switch, 0000100, depends on I bit to forward the stream to Customer networks.

3.4 Role in VLAN Application Protocol

There are three roles in the access network. Top Switch is related to the source in the fork-type virtual-network-based fault tolerant architecture. Access switches are related to virtual network nodes in the fork-type virtual-network-based fault tolerant architecture. Customer networks are related to end nodes in the fork-type virtual-network-based fault tolerant architecture.

Top Switch implements powerful capability. That due to the Top Switch has the duty to communicate between access network and Intranet. It implements all kinds VLAN function and powerful network process capability. It also has to handle fast failover process and deliver capability. Top Switch can send backup streams to replace duty of fault switches and depend on the bandwidth loading situation to control the bandwidth utilization rate [9]. Access switch suggest have basely requirements. We minimize required VLAN in switch to obtain the low cost advantage for developing entire access network. Normally, the switch that develop the access network will implement the VLAN function as below, Port base VLAN, IEEE802.1Q VLAN Tagging, Translated VLAN, Protocol-based VLAN, Double VLAN and Mac Base VLAN. The access switch also has the capability to notice fault event to Top Switch. For failure detection, our proposal relies on switch support for SNMP traps [10]. Each of the switches in the access network is configured to send SNMP traps to the Top Switch to synchronize its status with Top switch whenever any event of interest takes place. (See the tag 1 in the Figure 17) Typically, the events of interest are link failures, port failures, carrier loss, etc. Except this, the Top will sent the SNMP message flows thought upstream switch to the downstream switch. (See the tag 2&3 in the Figure 17) By this way, we can make sure all the events of interest can be detected.

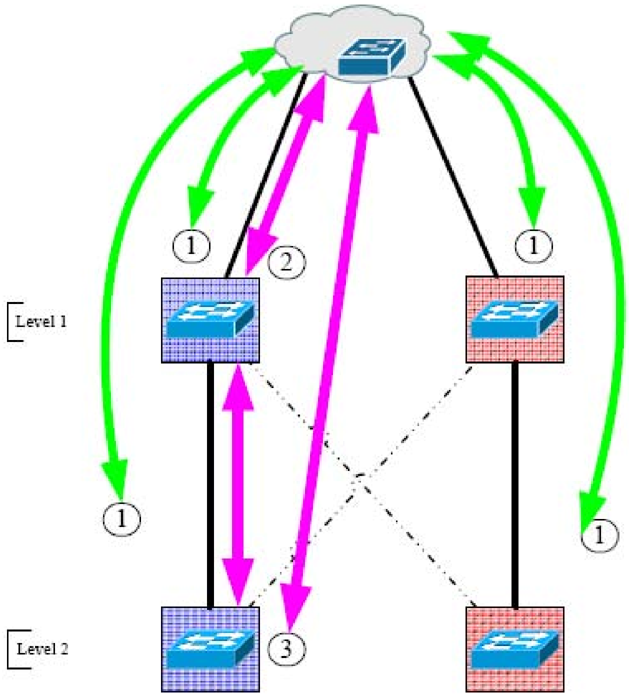

Figure 17 SNMP Message Flow between Switches

In our protocol, it only requires the minimum VLAN functions for access switch, Port base VLAN and IEEE802.1Q VLAN Tagging. These switches are regard as very cheap switch, as we know the price of them around 7,000 NTD.

Customer network is required two ports that connect with switch in our protocol. In upstream, the un-tag frame stream will be delivered to access switch. The access switch will add VLAN identifier for them. After this, those VLAN identifier frames will be delivered unto Top Switch. In downstream, Top Switch will sent the traffic and those streams will follow the virtual network to deliver unto the edge access switch. After this, the edge access switch will un-tag the traffic and sent them to customer networks.

The key for VLAN application protocol is in its VLAN application protocol Streams. The VLAN application protocol Streams be generated by source. The streams replace the original streams and deliver to the customer networks while there are fault switches or for the bandwidth utilization rate requirement that over-utilization of capacity when other capacity is available in the access network to redirect the fork-type virtual network path by changing the break nodes in this path.

Chapter 4 – VLAN Application Architecture Analysis

In communication apparatus implement with ability, high reliability, support lasting service and offer fault-tolerant and high usability. Ethernet is envisioned as a key technology to recent deployments service Network. Ethernet have features, high scalability, and high bandwidth provisioning but very inexpensive and easy deployment. But Ethernet lacks carrier-grade feature of availability. Especially, we use the cheap switch to develop the fork-type virtual-network-based fault tolerant architecture.

The most common stochastic approach a network is modeled by a Continuous-Time Markov Chain (CTMC). Continuous-Time Markov Chain is a state-base model with transition rates between states. In this section, the continuous-time Markov chain (CTMC) was used to determine the steady-state availability of the proposed virtual-network-based fault tolerant architecture. We can get up to "99999" Carrier-grade Availability. The goal of analysis is to discover the rule of thumb to keep high carrier grade.

4.1 Markov Chain for Fault Tolerance System

Figure 18 is the state-transition diagram. The state-transition diagram of the continuous-time Markov chain is modeling failure, repair and failure detect and recover behaviors of our proposal, virtual-network-based fault tolerant architecture.

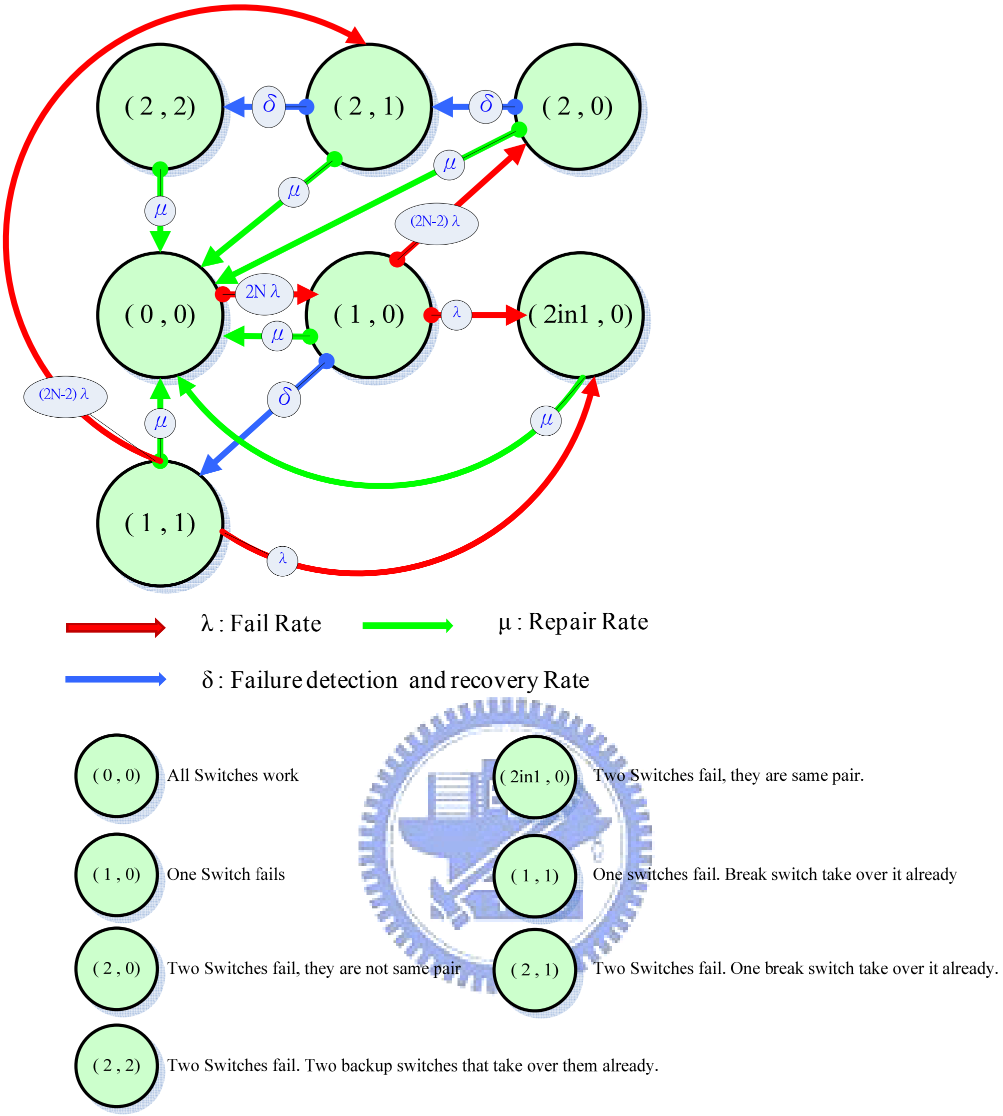
(2 , 2)
(2 , 1)
(2 , 0)
(0 , 0)
(1 , 0)
(2in1 , 0)
(1 , 1)
δ
μ
(2N-2) λ
2N λ
λ
λ : Fail Rate
μ : Repair Rate
δ : Failure detection and recovery Rate
All Switches work
One Switch fails
Two Switches fail, they are not same pair
Two Switches fail. Two backup switches that take over them already.
Two Switches fail, they are same pair.
One switches fail. Break switch take over it already
Two Switches fail. One break switch take over it already.

Figure 18 States Transitions Diagram

4.1.1 Markov State Definitions

In this thesis, the time to failure, time to repair and time to failure detect and recovery of switch are assumed to be exponentially distributed with mean 1/ λ , 1/ μ and 1/ δ , respectively. In Figure 18 when the state transfers from (0, 0) to (1, 0), (1, 0) to (2, 0) and (1, 0) to (2in1, 0) which indicates that switch fails, while the state transfers from (1, 0) to (1, 1), (2, 0) to (2, 1) and (2, 1) to (2, 2) which indicates a failure has been detected and recovered, and the break switch has taken over the role of the active switch. Finally, when every states except the state (0, 0), transfers to state

(0, 0) which indicates that event of the access network operator repairs. The associated failure detect and recover rate (δ) is the multiplicative inverse of the mean time that from the switch failed to the source detecting that the failure had occurred and break switch being recovered for it. The virtual-network-based fault tolerant architecture in state (1, 0) (2, 0) (2, 1) (2in1, 0) are assumed to be failed. The state transforms indicate the virtual-network-based fault tolerant architecture fails (i.e., some customer networks cannot receive packet via system). Note that in this thesis, there are N pair switches in the fault tolerance system, our proposal is one-plus-one fault tolerance system, total is 2N switches in the system.

As shown in Figure 18, state (i, j) represents the status of the switches that in the fault tolerance system, where i and j represent the status of Number of Fault switches and Number of Break Switches that Take over Fault Switch, respectively. While i equal to 「0」means all switches is working, i equal to 「1」mean one switch fails, i equal to 「2」mean two switches fail and they are not in the same pair and i equal to 「2in1」mean two switches fail and they are in the same pair. The element of j represents the Break Switch that takes over the Fault Switch. Finally, if i equal to j it means the system is working.

4.1.2 Markov State Assumptions

In this thesis, base on we focus the access network architecture, all switches in the access network should be back to operational mode while the event of operator repair. And the probability of more than three switches fail within a moment is small for ignoring. All failure events are assumed to be mutually independent. The time to failure, time to repair and time to failure detect and recovery of switch are assumed to be exponentially distributed with mean $1/\lambda$, $1/\mu$ and $1/\delta$, respectively. Let random variable X be the lifetime (i.e. time to failure [11]) is exponentially distributed with failure rate λ, Therefore, if a component obeys an exponential failure rate with parameter λ, then the MTTF (i.e., the expected lifetime [11]) can be determined as $1/\lambda$. We can get Mean Time to Failure, Mean Time to Repair and Mean Time to failure Detect and Recovery as be below equations (1)(2)(3).

Mean Time to Failure $\mathrm{MTTF}=E[X]=\int_0^\infty e^{-\lambda t}\,dt=\frac{1}{\lambda}$ (1)

Mean Time to Repair $\mathrm{MTTR}=E[Y]=\int_0^\infty e^{-\mu t}\,dt=\frac{1}{\mu}$ (2)

Mean Time to failure Detect and Recovery $\mathrm{MTTDR}=E[Z]=\int_0^\infty e^{-\delta t}\,dt=\frac{1}{\delta}$ (3)

4.2 Markov Chain Analysis

Let π (i, j) denotes the proportion of time that the system is in state (i, j). Note that in the steady state the rate at which transitions into state (i, j) must equal to the rate at which transitions out of state (i, j). Above concept is called the *global balance equation* [12] for the Markov chain in question. (i.e. for a Markov state, the sum of departure rates equal to the sum of arrival rates.) Thus, from Figure 18, we can obtain the global balances for each Markov state

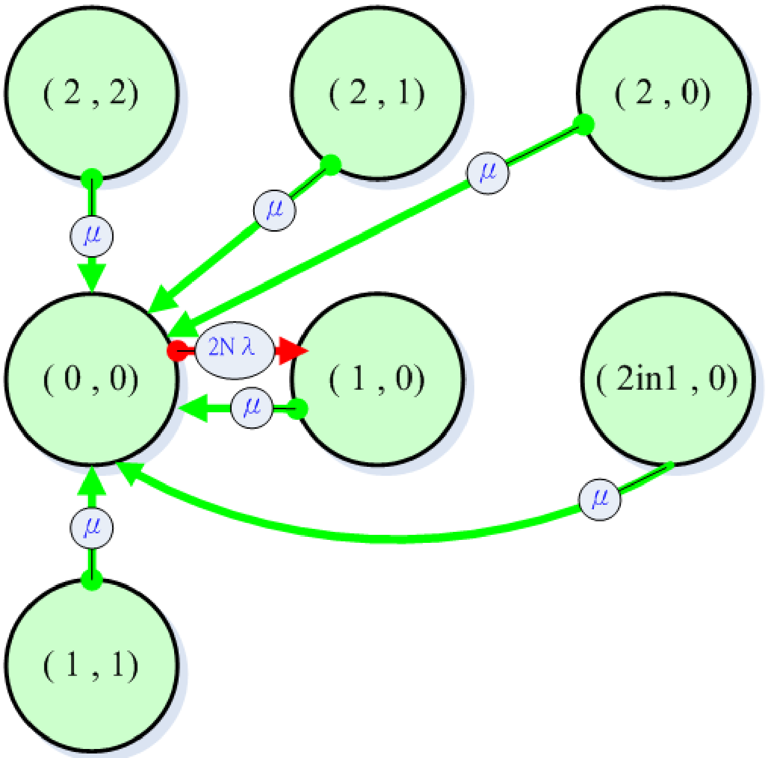

Figure 19 Global Balance for State (0, 0)

Form Figure 19, we obtain the below equation,

State (0,0): $2N\lambda\pi(0,0) = \mu(\pi(1,0) + \pi(1,1) + \pi(2in1,0) + \pi(2,0) + \pi(2,1) + \pi(2,2))$ (4)

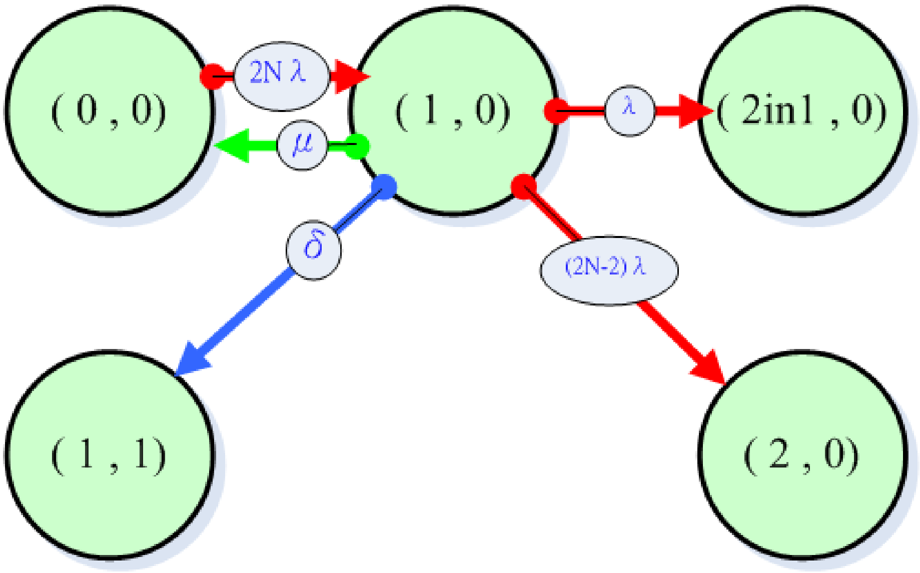

Figure 20 Global Balance for State (1, 0)

Form Figure 20, we obtain the below equation,

State (1,0): $(\delta+\mu+(2N-1)\lambda)\pi(1,0) = 2N\lambda\pi(0,0)$ (5)

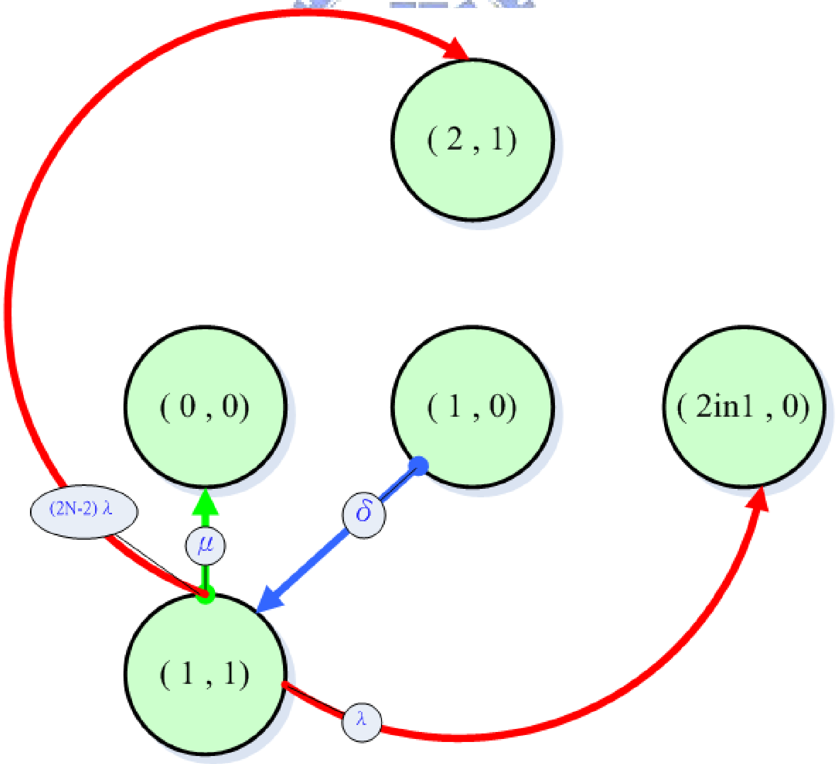

Figure 21 Global Balance for State (1, 1)

Form Figure 21, we obtain the below equation,

State (1,1): $(\mu+(2N-1)\lambda)\pi(1,1) = \delta\,\pi(1,0)$ (6)

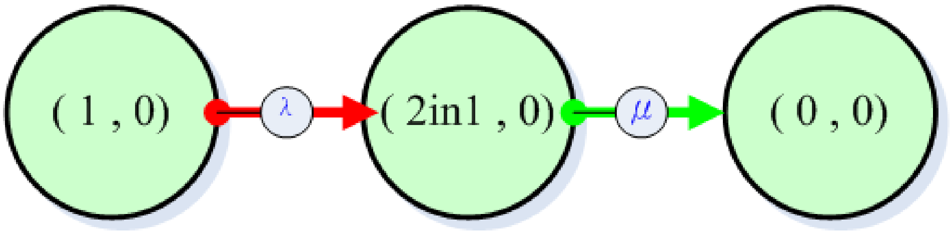

Figure 22 Global Balance for State (2in1, 0)

Form Figure 22, we obtain the below equation,

State (2in1,0): $\mu\pi(2in1,0) = \lambda\, \pi(1,0) + \lambda\pi(1,1)$ (7)

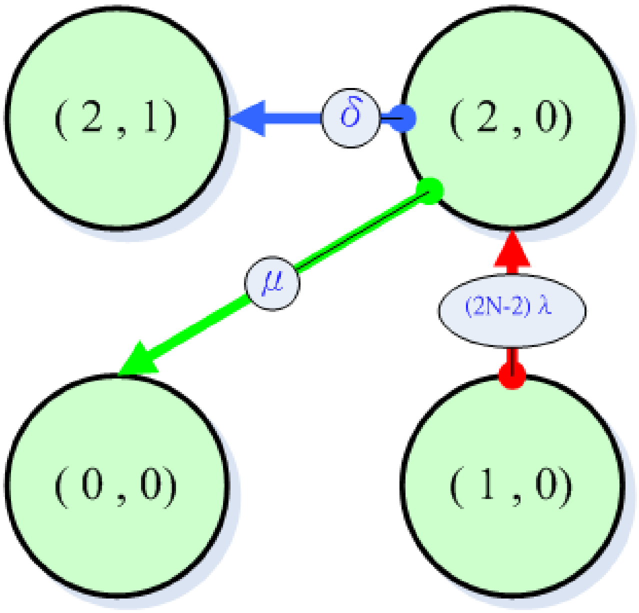

Figure 23 Global Balance for State (2, 0)

Form Figure 23, we obtain the below equation,

State (2,0): $(\delta+\mu)\ \pi(2,0) = (2N-2)\lambda\pi(1,0)$ (8)

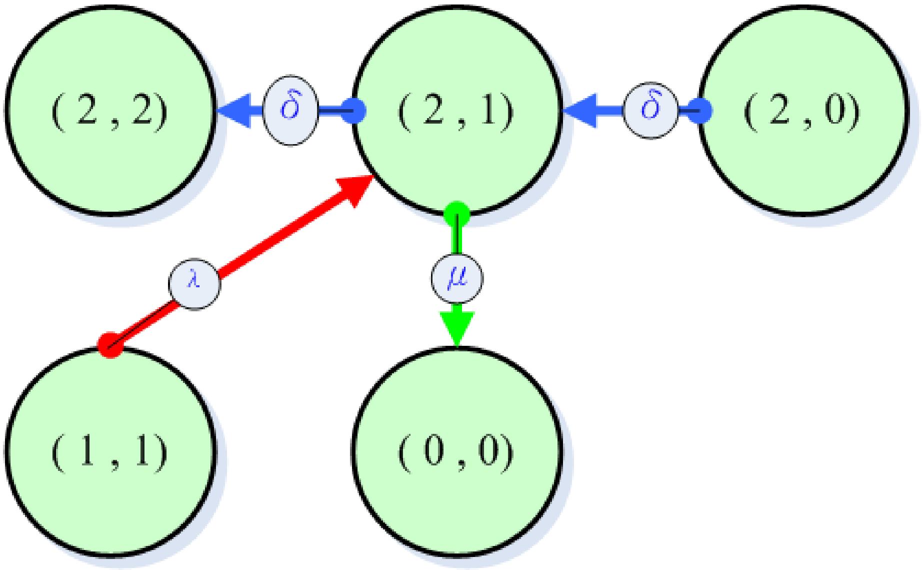

Figure 24 Global Balance for State (2, 1)

Form Figure 24, we obtain the below equation,

State (2,1): $(\delta+\mu)\pi(2,1) = \delta\pi(2,0)+ (2N-2)\lambda\pi(1,1)$ (9)

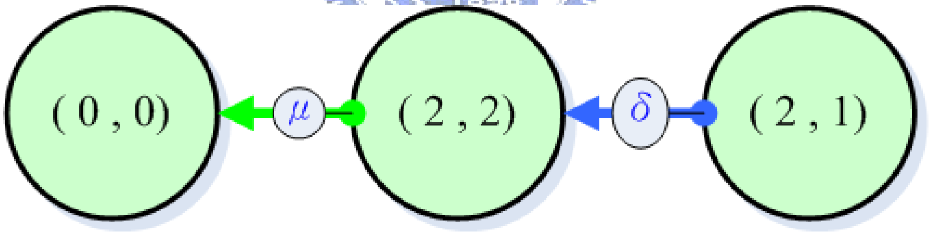

Figure 25 Global Balance for State (2, 2)

Form Figure 25, we obtain the below equation,

State (2,2): $\mu\pi(2,2) = \delta\ \pi(2,1)$ (10)

Consider the sum of all Steady-state Probability is equal to 1. By solving the preceding set of equations, along with this equation

$$\pi(0,0)+\pi(1,0)+\pi(1,1)+\pi(2in1,0)+\pi(2,0)+\pi(2,1)+\pi(2,2) = 1 \qquad (11)$$

By solving the preceding set of equations, we can get steady-state probabilities for each Markov states.

Thus, the availability (A_{HA}) of a Highly Available Ethernet Access Network can be determined based on π (0, 0) + π (1, 1) + π (2, 2). Then, the equivalent failure rate (λ_{HA}) and the equivalent repair rate (μ_{HA}) of a Highly Available Ethernet Access Network can be determined by applying the aggregation techniques described in [13]. (See the Figure 26)

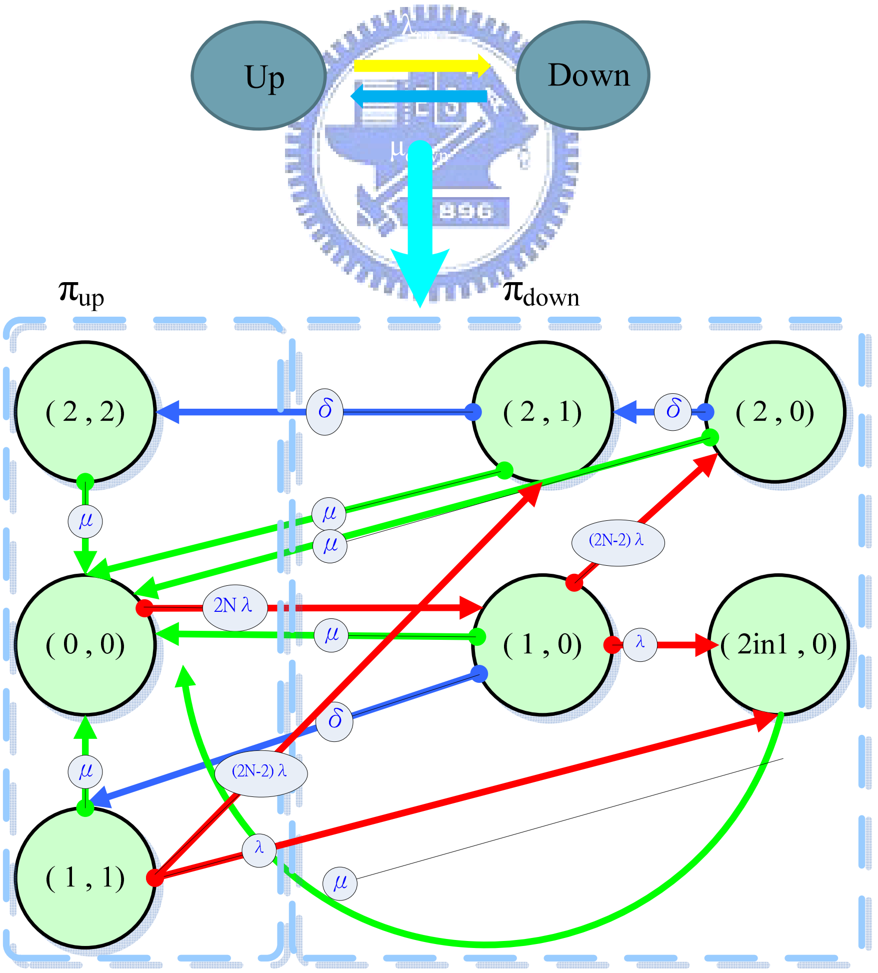

Figure 26 Aggregation in Availability Markov Model

Therefore, we obtain below equations (12) ~ (17).

$$\lambda_{up} = \frac{(2N)\lambda\cdot\pi(0,0)+(2N-1)\cdot\pi(1,1)+0\cdot\pi(2,2)}{\pi(0,0)+\pi(1,1)+\pi(2,2)} = \frac{2N\lambda\cdot\pi(0,0)+(2N-1)\cdot\pi(1,1)}{\pi(0,0)+\pi(1,1)+\pi(2,2)} \tag{12}$$

$$\mu_{down} = \frac{(\delta+\mu)\cdot\pi(1,0)+\mu\pi(2in1,0)+\mu\cdot\pi(2,0)+(\delta+\mu)\cdot\pi(2,1)}{\pi(1,0)+\pi(2in1,0)+\pi(2,0)+\pi(2,1)} \tag{13}$$

$$\mu_{down} = \frac{\delta\cdot\left(\pi(1,0)+\pi(2,1)\right)+\mu\cdot\left(\pi(1,0)+\pi(2in1,0)+\pi(2,0)+\pi(2,1)\right)}{\pi(1,0)+\pi(2in1,0)+\pi(2,0)+\pi(2,1)} \tag{14}$$

$$\mu_{down} = \frac{\delta\cdot\left(\pi(1,0)+\pi(2,1)\right)}{\pi(1,0)+\pi(2in1,0)+\pi(2,3)+\pi(2,1)}+\mu \tag{15}$$

$$\pi_{up} = \mu_{down} \;/\; (\lambda_{up} + \mu_{down}) = A_{HA} \tag{16}$$

$$\pi_{down} = \lambda_{up} \;/\; (\lambda_{up} + \mu_{down}) \tag{17}$$

4.2.1 Analysis for Fault Tolerance System

In real cases of switch in access network, the mean time to failure is belonging to year's grade. Mean time to repair is belonging to hour's grade. Mean time to failure detects and recovery is belonging to ms' grade [5].

We use the equations at above section to counter the availability (A_{HA}) of two real cases in our proposal, virtual-network-based fault tolerant architecture. First one, total 42 access switches in the virtual-network-based fault tolerant architecture. The mean time to failure is Exponential distribution with λ (1 year). The mean time to repair is Exponential distribution with μ, (4/12/24 Hours). Finally, Mean time to failure detects and recovery is Exponential distribution with δ, (0.1/0.05/0.025 Second). The second, total 168 access switches in the virtual-network-based fault tolerant architecture. The mean time to failure is Exponential distribution with λ (1 year). The mean time to repair is Exponential distribution with μ, (4/12/24 Hours). Finally, Mean time to failure detects and recovery is Exponential distribution with δ, (0.1/0.05/0.025 Second).

By using the Matlab tool, we can obtain the availability and CTMC steady-state probability of these two cases in our proposals, virtual-network-based fault tolerant architecture. (See Table 1 to Table 4)

We use CTMC to analysis our virtual-network-based fault tolerant architecture, we can get up to "99999" Carrier-grade Availability with the conditions, λ: 1/1year, μ: 1/4Hours and total 42 Access switches in the fault tolerance system. The virtual-network-based fault tolerance system can support up to 128 customer networks. In the other case, λ: 1/1year, μ: 1/4Hours and 168 access switches in the virtual-network-based fault tolerance system that can support up to 512 customer networks. It also reaches to "9999" carrier-grade availability.

Failure rate, λ = 1/1year;   Total 42 Access Switches

Availability	1/μ : 4 Hours	1/μ : 12 Hours	1/μ : 24 Hours
1/δ:0.1 second	0.99999143262912	0.99992930918285	0.99974575485110
1/δ:0.05 second	0.99999149913628	0.99992937540831	0.99974582042313
1/δ:0.025 second	0.99999153239003	0.99992940852109	0.99974585320918

Table 1 Availability for case I

Failure rate, λ = 1/1year;   Total 168 Access Switches

Availability	1/μ : 4 Hours	1/μ : 12 Hours	1/μ : 24 Hours
1/δ:0.1 second	0.99996961422258	0.99979131820273	0.99940764798025
1/δ:0.05 second	0.99996956126430	0.99979126690047	0.99940760015909
1/δ:0.025 second	0.99996961422258	0.99979131820273	0.99940764798025

Table 2 Availability for case II

Failure rate, $\lambda = 1/1$year;   Total 42 Access Switches

	$1/\mu$: 4 Hours $1/\delta$:0.1 second	$1/\mu$: 12 Hours $1/\delta$:0.1 second	$1/\mu$: 24 Hours $1/\delta$:0.1 second
$\pi(0,0)$	0.98118279569892	0.94559585492228	0.89680589680590
$\pi(1,0)$	0.00000013067411	0.00000012593521	0.00000011943747
$\pi(1,1)$	0.01847126216290	0.05151093230958	0.09277290437004
$\pi(2in1,0)$	0.00000843442595	0.00007056309349	0.00025417266797
$\pi(2,0)$	0.00000000000002	0.00000000000002	0.00000000000002
$\pi(2,1)$	0.00000000234288	0.00000000653360	0.00000001176724
$\pi(2,2)$	0.00033737469522	0.00282251720582	0.01016689495138
Availability	0.99999143262912	0.99992930918285	0.99974575485110

	$1/\mu$: 4 Hours $1/\delta$:0.05 second	$1/\mu$: 12 Hours $1/\delta$:0.05 second	$1/\mu$: 24 Hours $1/\delta$:0.05 second
$\pi(0,0)$	0.98118279569892	0.94559585492228	0.89680589680590
$\pi(1,0)$	0.00000006533728	0.00000006296768	0.00000005971877
$\pi(1,1)$	0.01847132749972	0.05151099527711	0.09277296408874
$\pi(2in1,0)$	0.00000843442595	0.00007056309349	0.00025417266797
$\pi(2,0)$	0.00000000000000	0.00000000000000	0.00000000000000
$\pi(2,1)$	0.00000000117144	0.00000000326681	0.00000000588362
$\pi(2,2)$	0.00033737586667	0.00282252047263	0.01016690083500
Availability	0.99999149913628	0.99992937540831	0.99974582042313

	$1/\mu$: 4 Hours $1/\delta$:0.025 second	$1/\mu$: 12 Hours $1/\delta$:0.025 second	$1/\mu$: 24 Hours $1/\delta$:0.025 second
$\pi(0,0)$	0.98118279569892	0.94559585492228	0.89680589680590
$\pi(1,0)$	0.00000003266870	0.00000003148386	0.00000002985940
$\pi(1,1)$	0.01847136016830	0.05151102676093	0.09277299394811
$\pi(2in1,0)$	0.00000843442595	0.00007056309349	0.00025417266797
$\pi(2,0)$	0.00000000000000	0.00000000000000	0.00000000000000
$\pi(2,1)$	0.00000000058572	0.00000000163340	0.00000000294181
$\pi(2,2)$	0.00033737645239	0.00282252210604	0.01016690377682
Availability	0.99999153239003	0.99992940852109	0.99974585320918

Table 3 Steady-State Probability & Availability for case I

Failure rate, λ = 1/1year;   Total 168 Access Switches

	1/μ : 4 Hours 1/δ:0.05 second	1/μ : 12 Hours 1/δ:0.05 second	1/μ : 24 Hours 1/δ:0.05 second
π(0,0)	0.92875318066158	0.81291759465479	0.68480300187617
π(1,0)	0.00000014843054	0.00000012991823	0.00000010944335
π(1,1)	0.06619863576596	0.15225199483317	0.21625347009650
π(2in1,0)	0.00003022775534	0.00020856455445	0.00059247556038
π(2,0)	0.00000000000002	0.00000000000002	0.00000000000002
π(2,1)	0.00000001045374	0.00000002404284	0.00000003414962
π(2,2)	0.00501779693282	0.03462169199650	0.09835090887396
Availability	0.99996961422258	0.99979131820273	0.99940764798025

	1/μ : 4 Hours 1/δ:0.04 second	1/μ : 12 Hours 1/δ:0.04 second	1/μ : 24 Hours 1/δ:0.04 second
π(0,0)	0.92875318066158	0.81291759465479	0.68480300187617
π(1,0)	0.00000019790724	0.00000017322425	0.00000014592444
π(1,1)	0.06619858628926	0.15225195152715	0.21625343361541
π(2in1,0)	0.00003022775534	0.00020856455445	0.00059247556038
π(2,0)	0.00000000000004	0.00000000000004	0.00000000000003
π(2,1)	0.00000001393832	0.00000003205711	0.00000004553282
π(2,2)	0.00501779344823	0.03462168398221	0.09835089749075
Availability	0.99996956126430	0.99979126690047	0.99940760015909

	1/μ : 4 Hours 1/δ:0.03 second	1/μ : 12 Hours 1/δ:0.03 second	1/μ : 24 Hours 1/δ:0.03 second
π(0,0)	0.92875318066158	0.81291759465479	0.68480300187617
π(1,0)	0.00000014843054	0.00000012991823	0.00000010944335
π(1,1)	0.06619863576596	0.15225199483317	0.21625347009650
π(2in1,0)	0.00003022775534	0.00020856455445	0.00059247556038
π(2,0)	0.00000000000002	0.00000000000002	0.00000000000002
π(2,1)	0.00000001045374	0.00000002404284	0.00000003414962
π(2,2)	0.00501779693282	0.03462169199650	0.09835090887396
Availability	0.99996961422258	0.99979131820273	0.99940764798025

Table 4 Steady-State Probability & Availability for case II

4.3 Switch Architecture Profit

By our design, we develop entire virtual-network-based fault tolerant architecture only by Layer 2 device, Switch. Our protocol is a layer 2 protocol, the fast VLAN forwarding and checking capabilities is the fundamental capability for the switch. Base on our proposal will shorten the time to failure detects and recovery.

We can also earn the benefits of low cost and some advance by replacing Layer 3 device to develop the fault tolerance system to handle the fault tolerance process. We consider the Layer 3 Device, Router may be the top level of Entire Network. Control fault tolerance process within the router could increase the effort of it. All pattern of the fault tolerance process may have to travel entire network, it should take long time and impact the bandwidth of the service Network. Finally, such as the router are more expensive than the Layer 2 Device, Switch.

Chapter 5 – Conclusion

In this thesis, we illustrate the virtual-network-based fault tolerant architecture. Follow discuss the virtual network concept we propose to develop the fault tolerance system architecture with the VLAN application protocol. To do so, we propose a virtual-network-based access architecture. The concept of virtual network is not new. It has been used to enable load balance, deadlock avoidance, and fault tolerance in networks such as inter-connected networks and local area networks. Nevertheless, as far as we know, we are the first to extend its usage to study the fault-tolerant-related issues in Ethernet access networks. We elaborate on determining the configuration of virtual networks so that the service can be recovered when a fault occurs. We consider applying fault tolerance system architecture via virtual network concept. Use the fork-type virtual network to make the fault tolerance system as a predicted backup path for a fault device. The fork-type virtual network has two advantages. 1) Best bandwidth efficient and No effort for other nodes. Until the traffic is delivering to the customer networks, no other device that without relationship will receiver these streams. 2) Bandwidth utilization rate of switches can be control by source.

VLAN are conventionally used to simplify network administration, improve security and limited broadcast domain of each VLAN. We make our proposal be compatible with VLAN application and deviate from conventional VLAN application. We use VLAN to practice fault tolerance system. The fast VLAN forwarding and checking capabilities is the fundamental capability for the switch. We propose a layer 2 protocol, base on our proposal will shorten the time to failure detect and recover. The continuous-time Markov chain was used to determine the steady-state availability [6] of the proposed virtual-network-based fault tolerant architecture. Through the analysis, we also aim at discovering the rule of thumb to keep high degree of carrier grade.

We can obtain three key motivations, Carrier-grade Availability, Low Cost and High Scalability. Apart from these, our proposal provides reliable and simple Network topology architecture, redundant protection machine for all elements in the fault tolerance system, make fault tolerance system is not only dedicated to take over the role of the active switch if the active switch failed but also have its own duty for delivering user traffic that can be controlled by Top Switch via VLAN application protocol. Finally, we can use the Advance Requirement for the multicast service, IPTV.

Reference

[1] IEEE Std 802.1ah-2008. “IEEE Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks, Amendment 8: Management Information Base (MIB) Definitions for VLAN Bridges” Institute of Electrical and Electronics Engineers, 2008.

[2] IEEE Std 802.3ab-1999. “IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements", Institute of Electrical and Electronics Engineers, 1999.

[3] IEEE Std 802.3ah-2004. “IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements", Institute of Electrical and Electronics Engineers, 2004.

[4] IEEE Std 802.3an-2006. “IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements", Institute of Electrical and Electronics Engineers, 2006.

[5] Wei-Kuo Liao, Ping-Hai Hsu, Shu-Kang Tseng, and Kang-Chiao Ling “An Ethernet Access Architecture for Highly Available IPTV”, Global Telecommunications Conference, Dec. 2008 PP 1-5

[6] Chia-Tai Tsai, Rong-Hong Jan, Chien Chen, Chia-Yuan Huang, “Implementation of Highly Available OSPF Router on ATCA”, IEEE International Symposium on Pacific Rim Dependable Computing, December 2007, pp.147 - 154

[7] IEEE Std 802.1Q-1998. “IEEE Standards for Local and Metropolitan Area Network: Virtual Bridged Local Area Networks”, Institute of Electrical and Electronics Engineers, 1998.

[8] Sundar Vedantham, Seong-Hwan Kim, and Deepak Kataria, Agere Systems Inc. “Carrier-Grade Ethernet Challenges for IPTV Deployment”, IEEE Communications Magazine, July 2006 PP 24-31

[9] G. Ash, “Traffic Engineering & QoS Methods for IP-, ATM-, & TDMBased Multiservice Networks,” Internet Draft, Oct 2001.

[10] Sharma, S., Gopalan, K., Nanda, S., Chiueh, T., “Viking: A Multi-Spanning-Tree Ethernet Architecture for Metropolitan Area and Cluster”, INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies Volume 4, 7-11 March 2004, pp. 2283 - 2294 vol.4.

[11] K.S. Trivedi, “Probability and Statistics with Reliability, Queuing and Computer Science Applications”, 2nd Edition, John Wiley and Sons, Inc, New York, 2002, pp. 405－504.

[12] Ng Chee-Hock, Soong Boon-Hee, “Queuing Modeling Fundamentals with Applications in Communication Networks”, 2nd Edition, John Wiley and Sons, June 2008, pp. 106－107.

[13] M. Lanus, Y. Lin, and K.S. Trivedi, “Hierarchical Composition and Aggregation of State-Based Availability and Performability Models”, IEEE Transactions on Reliability, 52(1), 2003, pp. 44－52.