*International Journal of*
Intelligent Engineering & Systems

http://www.inass.org/

# MPLS-based Network Fault Recovery Research

**Yimin Qiu**[1,2*], **Jinguang Gu**[2], **Hongbing Zhu**[3], **Yi Zhou**[1,4]

[1] *Computer School of Wuhan University Wuhan, China*
[2] *College of Computer Science and Technology Wuhan University of Science and Technology Wuhan, China*
[3] *Faculty of Information Design, Hiroshima Kokusai Gakuin University, Hiroshima, Japan*
[4] *WISDRI Engineering and Research Incorporation Limited Wuhan, China*

[*] *Corresponding author's Email: ghw1978@sohu.com*

**Abstract:** In the past years there has been an enormous growth in the use of Internet, and new real-time connection-oriented services like streaming technologies and mission-critical transaction-oriented services are in use and new ones are currently emerging. The research on more reliable network becomes an inevitable trend presently. MPLS is a next generation backbone architecture, which can speed up packet forwarding to destination by label switching. However, if there is not a backup LSP when the primary LSP fails, MPLS frames cannot be forwarded to destination. Therefore, fault recovery has become an important research area in MPLS Traffic Engineering. At present, Protection Switching can be approached by two famous methods, Makam and Haskin based on which other methods basically come into being. When the place of failure is away from ingress node, a problem will come out. These two famous methods both have their disadvantage. In order to minimize or eliminate their drawback, this thesis tries to do some exploration on the MPLS-based recovery model. The model in the thesis using the Reverse Backup Path to solve the loop of data back to the path is too long, and the simulation experiment shows that new method of MPLS-based recovery has less packet disorder and much lower delay and packet losses.

**Keywords:** MPLS; Fault Recovery; Protection Switching; Reverse Backup Path; NS-2

## 1. Introduction

With the rapid development of a large number of new businesses, many researchers are very interested in complex networks, such as Internet[1], social networks[2], metabolic networks[3], food webs[4], citation networks[5] and so on. In recent years, how to improve the network quality of service has become urgent. Multi-Protocol Label Switching (MPLS) [6] uses label to execute fast packet switching in network, and integrates switching technique and IP routing technique for the purpose of constructing a new kind of network structure with higher stability and higher agility. MPLS, a next generation backbone architecture, can speed up packet forwarding to destination by label switching. However, if there is not a pre-established backup path when the network fails, MPLS frames cannot be forwarded to the destination. Therefore, fault recovery becomes an important research area currently.

Research on fault recovery based on MPLS is not only one of the important research fields but also the base for the development of next generation network. In essence, fault recovery of MPLS research mostly focuses on two aspects, Protection Switching and Re-routing. This thesis focuses mainly on the Protection Switching. At present, two famous methods are applicable to Protection Switching - Makam and Haskin, on base of which other methods basically come into being. When the place of failure is away from ingress node, a problem will come out. For Makam, more packets will be lost. For Haskin, path of Packet retrace is longer and packet delay time is longer, and packet disorder is prone to emerging.

In addition, path protection, recovery concept [7, 8] and discussing algorithm [9], which are important aspects of MPLS network's research, have drawn concerns from many researchers and scholars. They are the basic functions under the situation of random failures.

This paper utilizes Network Simulator Version 2 (NS-2) based on C++ language exploitation, with TCL language program, to present a MPLS-based network model with local world. Based on the research in these methods, the thesis aims to do some exploration on the MPLS-based recovery model.

The paper is organized as follow: we present the characters of MPLS and its fault-tolerant recovery theory in section 2 and Section 3. Section 4 investigates a new recovery model under random failures, uses NS-2 to implement simulation for the new method, and compares it with the present methods. Finally, conclusions and further research directions are presented.

## 2. The characters of MPLS

Multi-Protocol Label Switching (MPLS) is a next generation Internet Broad Band technique. It makes use of core networks exchange and marginal IP router technology synthetically to advance the whole network capability. MPLS promotes the implementation of advanced features such as Quality-of-Service and traffic engineering in an effective manner [10, 11].

MPLS is a label switching technology between the second layer and the third layer of the network. Because it is designed for IP specially, it can be combined with the high switching capability of secondly layer with the sensitive identity of the third layer, and give IP network such characters as High Switching, Flow Control, QoS and etc. Assorts with oriented-connect structure over connectionless IP network, providing effective resource storage and predefine router. Therefore, MPLS has been applied more and more to be a core network switching technique.

Node equipments of a whole network can be divided by MPLS based into two kinds: Label Edge Router (LER) and Label Switching Router (LSR). LERs form the juncture part of MPLS, and LSRs form the hardcore. The LSRs use the Label Distribution Protocol [12] to establish the LSPs. LERs initiate or halt Label Switching Path to link and fulfill the transmittable function of the traditional IP data packages and signs. Ingress LER accomplishes classification, searching path, transmitting table of IP package, production and the Forwarding Equivalence Class (FEC) mapped to label of LSP table. Egress LER halts LSP, and transmits residual packages according to the pop-up label.

MPLS unites IP and ATM's merits, and then it has powerful router and it satisfies diversified new applications' requirements [11].

## 3. MPLS based fault-tolerant recovery

With all kinds of new web services appearing and developing, the path resilience becomes an important concern. Any fault of network of each section could cause services paralysis, and make users unacceptable, thus leading to huge economical loss in ISP internet. Therefore, preventing communications halt and reducing package loss are a major concern. Path-oriented MPLS can provide quicker and more divinable protection and recovery than conventional hop-to-hop router model can.

When MPLS network's links or nodes fail, it is necessary to respond more frequently to the fault. Otherwise, even if the fault is very short duration, it may also lead to large packet loss, resulting in poor quality of service issues and network performance decline. The choice of fault recovery scenarios depend on the meshwork's spare capacity, recovery cost, calculative efficiency, recovery efficiency, expand factors and other conditions [11, 13]. The recovery mechanism provided by MPLS can not only ensure the resuming of data transmission in any given time when the fault occurs, but also keep the necessary QoS after the resuming of the data transmission.

Network traffic, combined with MPLS, as the broadest choice, offers and manages core meshwork to accelerate the concept of MPLS based recovery models.

The essentiality of MPLS based recovery also emphasizes the correlative recovery time with all-purpose router algorithm. All-purpose router algorithm has the merit to cause the network haleness and presence. But it also has faults. It will engross a lot of recovery time when it needs to recover from the fault. Overpassing MPLS based recovery models to augment all-purpose router algorithm can provide changeable network recovery. When MPLS uses pre-established LSPs to transmit data flow, that pre-established LSPs may allow MPLS pre-establishes recovery framework or backup LSPs.

In order to ensure the smooth run of the network and traffic transfer at full steam, when a link or node fails, the domestic and overseas experts attach much more importance to the research of MPLS based recovery models. The MPLS based recovery mechanisms can be classified as local protection or global protection [14]. The main difference between the global and local recovery is determined by the way they handle the recovery of the MPLS network.

The local protection is performed in a distributed manner, whose aim is to protect against a neighboring link or node failure. The local protection mechanisms involve significant backup path computations and management tasks in order to protect the entire MPLS path. The traffic on the working path is switched over to the backup path upon a failure on the working path in MPLS based local protection mechanisms. The recovery path selection or switching is done by the nearest point of failure upstream Label Switch Router or LSR when it uses local recovery [15].

The global protection is performed in a centralized manner and the aim is to protect against any link or node failure on the entire path or a segment of the path. In the case of global protection, the Path Switch LSR (PSL) switches the traffic from the failed working path to the backup path. However, the PSL is not usually adjacent to the point of failure. The global protection mechanisms require the fault notification be propagated to the PSL so that the PSL can perform the switch over to the backup path. And when the failure on the working path is modified, the traffic may be switched over back to the working path [15].

Fault-tolerant recovery technology is intended to help in the case of failures after the establishment of a new connection to send data. Since the actual network is complex and changeable, in order to be able to do the development of networks on the basis of forward-looking and study how to use and integrate existing network resources to enable the network to achieve maximum performance, network researchers employ a network simulation software to simulate the real network [16].

The model in the thesis combines with the advantages of Makam, Haskin and Hundessa, using the Reverse Backup Path to solve the problem when the wrapping of data back to the path is too long. In the later part of the thesis, we use the network simulation tool NS-2 to implement simulation, and the simulation experiment result shows that improved method of MPLS-based recovery has less packet disorder and much lower delay and packet losses.

## 4. NS-2 introduction

Network Simulator (Version 2), widely known as NS2, is simply an event-driven simulation tool that has proved useful in studying the dynamic nature of communication networks. Simulation of wired as well as wireless network functions and protocols (e.g., routing algorithms, TCP, UDP) can be done by using NS2. In general, NS2 provides users with a way of specifying such network protocols and simulating their corresponding behaviors.

Due to its flexibility and modular nature, NS2 has gained constant popularity in the networking research community since its birth in 1989. Ever since, several revolutions and revisions have marked the growing maturity of the tool, thanks to the substantial contributions from the players in the field. Among these are the University of California and Cornell University who developed the REAL network simulator, on which NS is based. To investigate network performance, researchers can simply use an easy-to-use scripting language to configure a network, and observe results generated by NS2. Undoubtedly, NS2 has become the most widely used open source network simulator, and one of the most widely used network simulators.

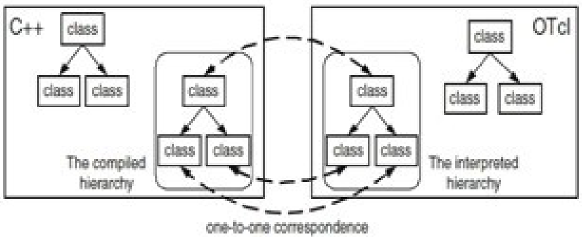

Figure 1 Two language structures of NS-2

NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl). While OTcl acts as the front-end(i.e., user interface), C++ acts as the backend running the actual simulation. As can be seen from Figure.1, class hierarchies of both languages can be either standalone or linked together using an OTcl/C++ interface called TclCL [17]. There are two types of classes in each domain. The first type includes classes which are linked between the C++ and OTcl domains. In the literature, these OTcl and C++ class hierarchies are referred to as the interpreted hierarchy and the compiled hierarchy, respectively. The second type includes OTcl and C++ classes which are not linked together. These classes are neither a part of the interpreted hierarchy nor a part of compiled hierarchy.

## 5. A new fault-tolerant recovery model

Table 1 Tag value and its significance

| Label value | Definition |
|---|---|
| 4 | Defined as the reversal of data packets, when the LSR receives this data packet, it is a faulty path in front of the work produced, and will send this packet to the backup path; if there is no alternate path, the packet will be sent to the upstream LSR. |
| 5 | Defined as Tag packets, LSR packet sent to this Tag, the data stream after the packets on the temporary storage in the Buffer, the longer the transmission path to the work; received Tag packet LSR, in this packet to the alternate path for transmission or play on the mark 6, an LSR knows that the former is its Tag packets. |
| 6 | An LSR packet prior to sending out Tag. |
| 7 | Using Ping protocol verification failure is restored; if the recovery path, data flow will switch back to the original working path; otherwise, to continue to packets sent to the alternate path. |

### 5.1 Experiment methods

In this section, we discuss several famous models, that is, tolerant abilities under failures. Here, failure means that links are broken. The most working methods of famous MPLS-based fault recovery mechanism need to transmit the message back to PSL or ingress node. For instance, Makam model needs to send FIS back to PSL, while Haskin model transmits information stream reverse to the ingress node over again [18, 19]. The packets of MPLS network have to transfer via LSP. However, LSP needs to pre-establish through tag, so commonly we should set up a backup LSP directly. When the information stream has to deliver back to the ingress node, we will label the backup LSP and send back. Pre-establishing an actual backup LSP will waste resources; therefore, this paper presents a new model which combines with the advantages of Makam [20], Haskin [21] and Hundessa [22], by using the Reverse Backup Path to solve the problem when the wrapping of data back to the path is too long, and evaluates models in the same conditions with the network simulation tool NS-2 to implement simulation. The purpose of models study is to find a recovery method with less packet disorder, much lower delay and fewer packet losses.

### 5.2 Algorithm description

According to MPLS based recovery mechanisms, this paper uses a college graduate students' laboratory web framework to establish a small network simulation based on NS-2.

In order to solve the network fault cause MPLS packet disorder, we use Buffer and Tag of Hundessa mechanisms as well as Reverse Backup Path to reduce the packet loss rate and reduce the delay time and so on, and add the Ping protocol, verify that the network is a path to understand the operation of the network. This can save recovery time and reduce delays.

RFC3032 [23] defines the MPLS Label Stack Encoding, of which 4 to 15 are reserved labels are not used, so we use 4 to 15 to define label design needs in the information packet, which is shown in Table 1.

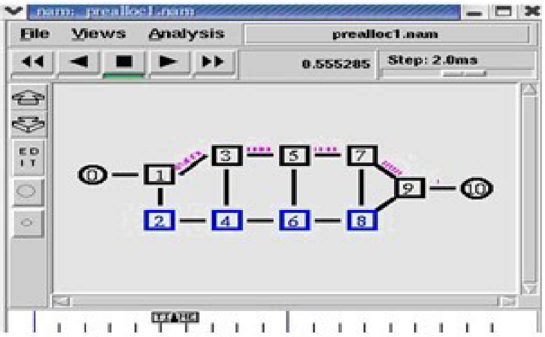

Figure 2 Network topology in NS-2

The network topology and settings used in the following simulations are the same for all simulated cases [19]. Figure2 shows the network topology and how traffic flows on the working path in normal operation [18]. The NS-2 allinone package includes a network viewer called nam (short for Network Animation) that can visualize the events in the network from trace files created by the NS-2 simulation. All network pictures in this chapter are screen shots from our simulations using the nam viewer. The simulation parameters are set in Table 2.

The network consists of nine MPLS enabled nodes (1-9) and two nodes (0, 10). Each MPLS node has an

Table 2 Simulation parameter setting

| Simulation parameter | value |
|---|---|
| Link Bandwidth | 10Mbit/s |
| Simulator | MNS2.0 |
| Duration | 2 sec |
| Flow start-up time | 0.5 sec |
| Traffic | 1.8Mbit/s |
| Failed link | LSR5-LSR7 |
| Link Down | 0.8 sec |
| Protection path | LSR1-LSR3-LSR5-LSR7-LSR9 |
| Global backup path | LSR1-LSR2-LSR4-LSR6-LSR8-LSR9 |

RSVP-TE agent attached, which is used for label distribution. The nodes 0 and 10 are of the standard node class in NS-2 and have no MPLS module or RSVP-TE agent attached. Node 0 has a CBR (constant bit rate) UDP agent attached and node 10 is set up to be traffic sink. At time 0.5s the CBR/UDP agent starts to send an UDP stream of packets that has destination address set to node 10. The size of the packets is set to 200 bytes and the send rate is set to 5000kbps. At time 1.8s the agent at node 0 stops the UDP stream towards node 10.

The working path is set to the shortest path between node 0 and node 10 (1, 3, 5, 7, 9). This LSP is set up with RSVP-TE signaling before the agent at node 0 starts to send traffic, and the lspid for the working path is set to 1000.

The propagation delay between two nodes is set to 1ms and the link bandwidth is 10 MB.

### 5.3 New model simulation

This new recovery program is primarily to use Reverse Backup Path and the Global Repair Path to Recovery, a program to improve the use of this fault the upstream LSR Search Reverse Backup Path fast fault recovery mechanism that is used around the back to reduce packet loss while avoiding the problem of packet disorder. In this simulation experiment, LSR1-LSR2-LSR4-LSR6-LSR8-LSR9 is the Global Repair Path, and LSR5-LSR6-LSR8-LSR9 the Reverse Backup Path [18].

The program needs to use the Buffer and Labels. When a LSR receives the reverted packet, it can be determined that the corresponding working path has been broken. This LSR should send multi-packet marked with Tag, and stored to the Buffer with the work path's data flow. After the alternate path receives the reversed packet with Tag, the Tag is removed and the Buffer's packets are sent to the backup path continuously. So it can reduce the use of circuit bandwidth, and the packet will not upset the order. Each LSR has the same Buffer, marked on the packet or remove Tag feature; this would resolve the packets order confusion in Haskin model.

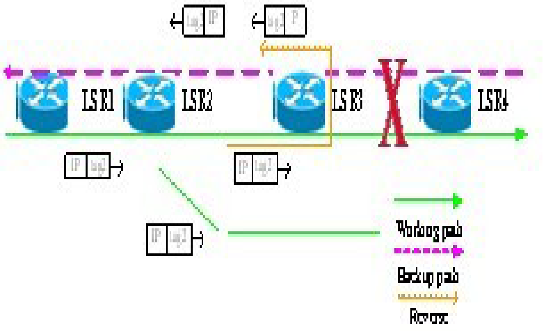

Figure 3 Exist the reverse backup path

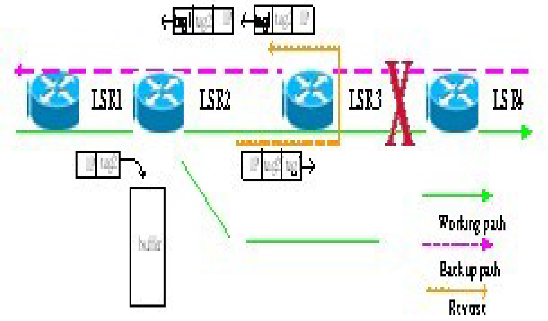

Figure 4 Cache data and receive reversed data packets

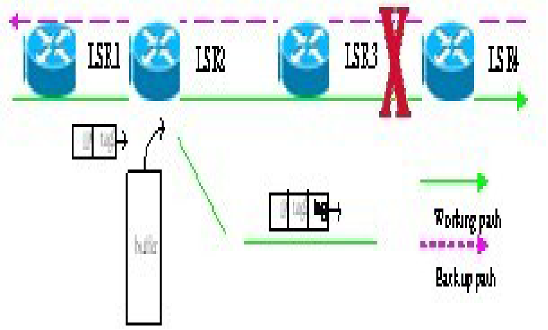

Figure 5 Forwarding data from the buffer

If the fault occurred between the LSR3 and LSR4, LSR3 first tests whether there is an replacement path.

If there is it should immediately change to the replacement path; otherwise it will send the data back to LSR2. Here tag1 denotes the external label which will receive the reverse marked packets around the back, and tag2 denotes internal packet label. When LSR2 receives reversed packets, it means that the path of a break in front of the work happens, it also should detectes whether exists the alternate path firstly. If there is, as shown in Figure.3, it must sent the retraced packets to the replacement path immediately. Meanwhile, when LSR2 receives reversed packets, it also receives the same LSP's data packet and marks it with a Tag, forwarding with the original path. In Figure.4, if LSR2 receives another same data packet, it should deposits in Buffer, until all data packets have been back.

When all the packets have been re-wound back after the reverse, LSR2 puts Buffer data packets within the forwarding from the Reverse Backup Path out in Figure.5. When all packets within the Buffer are absolute transmitted, the rest packets can be transmitted directly from the Reverse Backup Path, but they do not have to wait in the Buffer. At this time, Ping agreement is used to test whether the network failure has complete repaired; after the recovery, it can switch the data stream to the original working path so it can continue to go forward.

As seen as Figure 2, at time 0.8s a link on the working path breaks. When the upstream LSR on the working path detects the link break, the chosen recovery mechanism is started. If the failure occurred between the link LSR5 and LSR7, LSR5 will first search whether there has a backup path. When found, it will be transferred to the backup path; otherwise the data packets will reverse back to LSR3. When LSR3 receives the reversed data, it means that a failure has occurred; at the same time, it also searches for a backup path, and immediately switches the reversed data to the backup path. Simultaneously, while LSR3 receives the reversed data and the same LSP data, it will mark it a Tag and make it transmit along the original path. In the case where LSR3 receipts the same LSP data, it will be deposited in Buffer until the whole data package is retraced back.

When the whole data package is retraced back, LSR3 at once transmits the data in Buffer from Reverse Backup Path. All data package transferred, the latter data will be transmitted from Reverse Backup Path directly, and there is no need to wait in Buffer. Here, we can use Ping protocol to detect whether the network failure have recovered. Then, the data flow will be switched to the original working path after the network recovery.

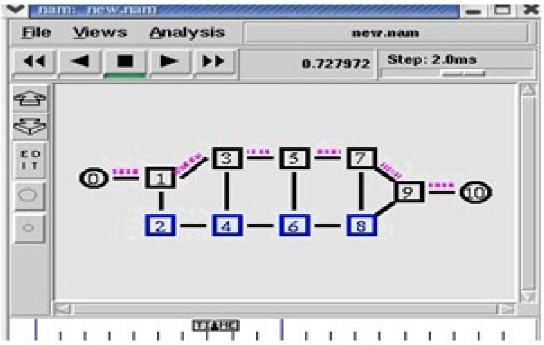

Figure 6 New model simulation in NS-2 (a)

To sum up, if the link failed, the node will first search for the Reverse Backup Path and the start Buffer mechanism, after the data packet enters the Buffer to wait. When the node detectes a Reverse Backup Path, data flow from the working path will switch to the Reverse Backup Path, and the packet around will forward out from this the back on the path. This node 0 from node 10 of the data sent along the LSR1- LSR3- LSR5- LSR6-LSR8-LSR9 Path Forwarding.

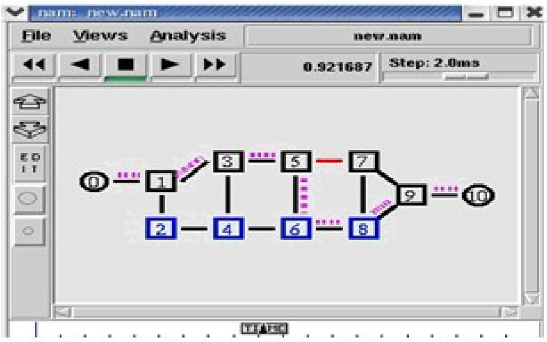

Figure 7 New model simulation in NS-2 (b)

Figure 6 is 0.727972 seconds simulation. The path is still working unimpeded. Figure 7 is 0.921687 seconds simulation. The data LSR5 have to switch to Reverse Backup Path (LSR5- LSR6- LSR8-LSR9). From the figure it can be seen through the Reverse Backup Path, one aspect is to reduce the data surrounding a return to the path of the program, which did not like to pass Haskin reverse recovery path as first, and then switch to the backup path; the other is to speed up the recovery speed, when the LSR5 received packets around the back, it directly switch to the recovery path, because of having Reverse Backup Path, while Haskin programs and Makam programs need to wait to go back to the LSR1 recovery path switch to avoid the packet disorder problems.

### 5.4 Simulation comparison and analysis

Figure 8 shows the number of dropped packets for different models and depending on which link that breaks. For Haskin model and this new recovery model, traffic is switched onto a pre-setup backup path by the LSR that detects the failure, so for both of those models the only packets that are dropped are the ones dropped during the failure detection time. For Makam model, the number of packet loss as the failure occurred with the end-point LSR distance from the entrance of LSR increased. Because the failure occurred, a failure notification must be sent to the up-stream packet (FIS) to the PSL. After the PSL receives the FIS, it will switch to the recovery path data to complete the path to recovery.

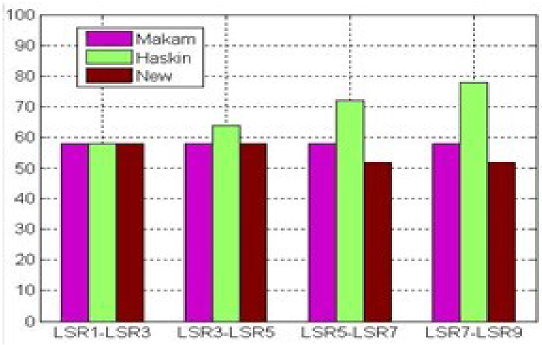

Figure 8 Packets dropped

As can be seen from the figure above, the data stream starting transmission, Makam model, Haskin model and this new recovery model in the link LSR1-LSR3 when the packet loss rate is the same, with the data transfer, Makam model data more and more packet loss, especially when the link after a failure. In contrast, Haskin model and this new recovery model data packet loss rate are much less than the Makam model. In the fault occurs, this new recovery model in the comparison of packet loss rate under the program to optimize much more off than those Haskin.

Through the simulation results of delay analysis in Figure.9, we can see that in the use of Haskin model, when the network fails, two conditions may present themselves: a longer delay of some of the data packet, about 50ms or soor shorter packet delay, about 1.45s after it returns to normal.

For Makam model, when the network fails, the fault LSR5 directly discards packets, and data packets do not need to wrap, so there are no packets to arrive.

For this new model, when the network fails, the fault LSR5 search to a Reverse Backup Path data will be forwarded to the Reverse Backup Path, so the packets do not need to wrap a long distance movements; and in each core LSR has a Buffer for temporary data, so the data packets need to wrap less than the Haskin model.

## 6. Conclusions

This paper analyzed the recovery algorithmic, simulated and demonstrated a MPLS backbone meshwork with NS-2 tool. Moreover, when a link or node fails in the working path, performs the switch over of the data traffic to the pre-established recovery path with a new recovery model. Therefore, it can guarantee the network transmission stability and reliability. This paper compiled with TCL programs by NS-2 validates the recovery model's feasibility and validity through simulation, as well as its one character of the less packet loss and the low delay. Based on the analysis, we can conclude that the new recovery model has lower delay than Haskin model and Makam model, and has reduced packet loss. In a word, the program is more optimized.

To select the experimental local network area in this paper is not suitable for real world and how to do it will be the next work. In addition, our model can not dynamically establish the Reverse Backup Path, and does not calculate the buffer size. However, these can boost the network fault recovery. Therefore, to build model with design target of ensuring network stability will be our focus.

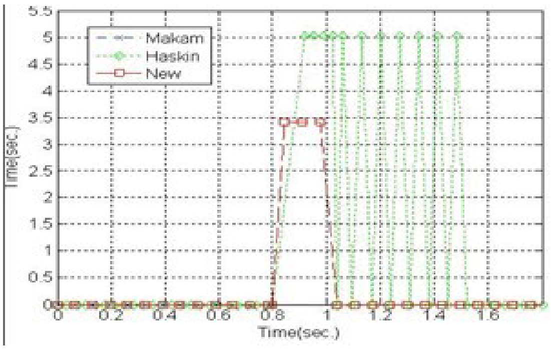

Figure 9 Delay

## 7. Acknowledgments

This work was partially supported by a grant from the NSF (Natural Science Foundation) of China under grant number 60803160, and was partially supported by a grant from NSF of Hubei Province of China under grant number 2009CDA136. This work was partially supported by China Postdoctoral Science Foundation under grant number 20060400275 and Jiangsu Postdoctoral Science Foundation under grant number 0601009B.

## References

[1] A. Vzquez, R. Pastor-Satorras and A. Vespignani, *Large-scale topological and dynamical properties of the Internet*, Phys. Rev. E, June 2002, 65: 066130(12).

[2] S. Wasserman and K. Faust. *Social network analysis: methods and applications*, Cambridge: Cambridge University Press, 1994.

[3] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai and A.-L. Barabsi, *The large-scale organization of metabolic networks*, Nature, Oct. 2000, 407: 651-654.

[4] R. J. Williams and N. D. Martinez, *Simple rules yield complex food webs*, Nature, Mar. 2000, 404: 180-183.

[5] S. Redner, How popular is your paper? *An empirical study of the citation distribution*, Eur. Phys. J. B, Aug. 1998, 4: 131-134.

[6] Rosen E, Viswanathan A and Callon R, *Multiprotocol Label Switching Architecture*, RFC3031 IETF, 2001.

[7] B. Tian, LSP *protection and restoration*, Communications Technology, 2007, 5:49-51.

[8] W. Xinhong and W. Gang-xing, *Path protection and recovery mechanism for MPLS network*, Computer Science, 2002, 10:85-87.

[9] W. Zuxi, G. Jun, H. Hanping and S. Gang, *Research on packet loss and disorder of MPLS-based recovery*, Microcomputer Information, 2007, 21:84, 93-94.

[10] D.Awduche, J.Agogbua, M.O'Dell and J.McManus, *Requirements for traffic engineering over MPLS*, IETF RFC2702, September (1999).

[11] B. T. Doshi, S. Dravida etc, *Optical network design and restoration*, Bell Labs Technical Journal, 1999, IV (1): 58-84.

[12] L.Andersson, P.Doolan, N.Feldman, A.Fredette and B.Thomas, *LDP specification*, IETF RFC3036, January 2001.

[13] Y. Xiong and L. G. Mason, *Restoration strategies and spare capacity requirements in self-healing ATM networks*, In IEEE/ACM Transactions on Networking, 1999, VII (1): 98-110.

[14] V.Sharma and F.Hellstrand, *Framework for multi-protocol label switching (MPLS)-based recovery*, IETF RFC3469, February 2003.

[15] A. P. S. Virk and R. Boutaba, *Economical protection in MPLS networks*, Computer Communications, 2006, 3:402-408.

[16] Yimin Qiu, Jianxun Chen, Jinguang Gu and Xin Xu. "A Simulation for MPLS Global Recovery Model", In: *Proc. of First International Conf. On Intelligent Networks and Intelligent Systems*, 2008, 259-262.

[17] K. Fall and K. Varadhan, *The ns manual (formerly known as ns notes and documentation)*,Aug.2007.[Online].Available:http:www.isi.edu/nsnam/ns/ns-documentation.html.

[18] Yimin Qiu, *MPLS-based Network Fault Recovery Research*, China, Wuhan, Wuhan University of Science and Technology, 2009.

[19] Johan Martin Olof Petersson, *MPLS Based Recovery Mechanisms*, Norway, University Of OSLO, 2005.

[20] S.MakamV.SharmaK.Owens and C.Haung, *Protection/Restoration of MPLS networks*, Internet-Draft (draft-makam-mpls-protection-00.txt), 1999.

[21] D. Haskin and R. Krishnan, *A Method for Setting an Alternative Label Switched Paths to Handle Fast Reroute*, Internet-Draft(draft-haskin-mpls-fast-reroute-05.txt), November 2000.

[22] L .Hundessa and J.D.Pascual, "Fast Rerouting Mechanism for a Protected Label Switched Path", In: *Proc. of Tenth International Conf. On IEEE Computer Communications and Networks*, 2001, 527-530.

[23] Rosen E, Tappan D, Fedorkow G, Rekhter Y, Farinacci D, Li T and Conta A, MPLS Label Stack Encoding, RFC 3032, January 2001.