ENHANCING FAULT TOLERANCE AND REROUTING STRATEGIES IN MPLS NETWORKS

[1]Rajni Arya, [2]Komal Garg (Guide)

[1]M.Tech. Scholar, [2]Assistant Professor, C.S.E Department,

R.P.S. Group of Institutions, Mohindergarh, Haryana, (India)

ABSTRACT

MPLS is a packet-forwarding technology which uses labels to make data forwarding decisions. When fault occur in active LSP, the recovery scheme must re-direct the affected traffic to the recovery path which bypass the fault. In order to provide a reliable network, fault tolerance is an important issue in the network design. In multiprotocol label switching (MPLS) network; the main focus is on, how to protect the traffic against node and link failures. For this IETF proposed, two recovery mechanisms protection switching and rerouting. In order to improve the fault-tolerant performance of these recovery mechanisms, the proposed approach chooses the backup LSP (to recover the affected traffic) on the basis of three criteria namely: reducing request blocking probability, minimizing cost of network (i.e. minimum hop) and load balancing. An effort is made in order to show alone none of the criteria is sufficient. The simulation is done, to find the influence of these criteria's and drawbacks is shown if we choose single criteria.

Keywords: Multiprotocol Label Switching, Protection Switching, Rerouting.

I INTRODUCTION

1.1 Background

MPLS stands for Multi Protocol Label Switching. MPLS was proposed by Internet Engineering Task Force (IETF) as a technology, which plays an important role in providing the facilities for traffic engineering and quality of service in the network. MPLS network provide the ISPs with the required tractability to handle the traffic through ER-LSPs.

In networking, communication is carried out in the form of frames which travel from source to destination covering a precept of hop by hop transport in a store and forward manner. When the frames comes at each node it find the next hop in order to make sure that the frame manage its way towards its destination by performing a route table lookup. MPLS provide connection oriented service for variable length frame and rising as a standard for the next generation internet, MPLS is mechanism where labels are assigned to data packets and forwarding is done based on the contents of those labels without checking the originals packets itself, permitting tractability in using protocols and to forward packet via any type of transport medium. In networking, MPLS is a rising technology i.e. replacing the existing technology and it is highly in demand now a days. MPLS provide better solution and tractability to divert and forward packet when failure occurs. MPLS was designed by keeping in mind the strength and weakness of ATM because ATM (asynchronous transfer mode) and frame relay are ancestors of MPLS. MPLS is replacing technology as it requires less over head.

As the Internet grows tremendously in the last few years a lack of availability, reliability and scalability was found for mission critical networking environment. In conventional routing, packets are forwarded on the bases of destination address and a single metric like minimum hop-count or delay.

So conventional routing suffers from drawback, as this approach induces traffic to converge into the same link; therefore congestion increases in a significant manner in the network and because of that network comes in the state of an unbalanced network resource utilization condition. Traffic Engineering (TE) comes as the solution of this problem, which provides bandwidth guarantee, explicit route and an effective resource utilization of the network. Now a day's main demand in the network is background speed, so the current research focuses on traffic engineering with LSPs for better control over the traffic distribution in the network.

1.2 The Bigger Picture of Internet

In the past few years Internet has achieved a great success and become the backbone of our lives. The size of Internet increases in significant manner as the number of users is grown exponentially. Now a day's million of users worldwide communicate each other through the

Internet and the number is still growing continuously at a tremendous rate. Internet is formed by connecting many Local Area Networks (LANs), Metropolitan Area Networks (MANs) and Wide Area Networks (WANS), therefore Lan, Man and Wan combined forming the Internet through a backbone. The backbone provides a trunk connection to the Internet [1] through which clients can get access by sharing a set of lines or frequencies instead of providing them individually. Figure 1.1 show the structure of internet.

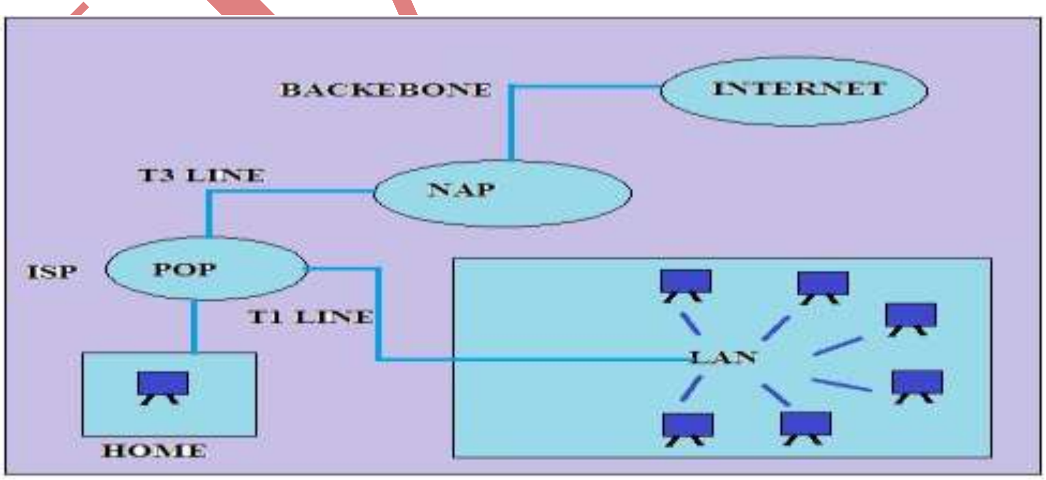

Figure 1.1: Showing the numbers of users are connected to each other via Internet.

A point-of-presence (POP) is an access point from one place to the rest of the internet. Each Internet service provider (ISP) has a point-of-presence on the Internet or we can say an average ISP can have more than 50 POPs those are interconnected having a ring topology to guarantee reliability. In POP Border Router (BR) are connected to other ISPs, Access Router (AS) are connected to remote customers, hosting Routers are connected to the web server and the Core Routers are the one that are connected to other POPs [2].

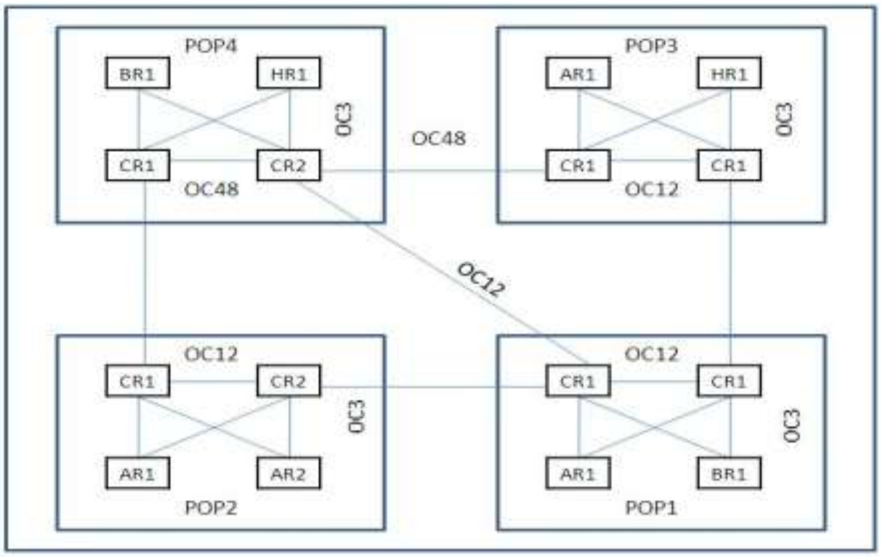

Figure 1.2: Simple ISP [2]

1.3 Research Motivation and Problem

As the Internet grows rapidly, it is becomes more and more difficult to satisfy the needs of customers and service provider because people want more advanced services from the Internet rather than the existing best effort services. Quality of service (QoS), Virtual Private Network (VPN) and Traffic Engineering (TE) enables ISPs to provide improved services to particular traffic flow. But IP, which was designed to provide best effort services, does not able to provide these new features. Multiprotocol Label Switching (MPLS) is a networking technology which is able to handle these features in an efficient manner.

MPLS got a lot attention by the researchers, not only because it provides new features like QoS, VPN and TE, but it provide a way to control particular traffic in an efficient manner and it also improve the network performance and efficiency. A crucial component of Quality of service (QoS) is fault tolerance or recovery from network failure. Real time applications and other critical task need instant response to a network failure; otherwise the required QoS cannot be met.

In conventional IP network, error handling is done by detecting a link or node failure in the network at the local router, spreading this information to the whole network area, so that network topology changes accordingly. One major flaw of this recovery mechanism is its slow response. The reasons of this problem are as follows:

- Failure can be detected at the lower layer very quickly, but they don't have any knowledge of where failure should be reported to, as they have no information for routing.
- As layer 3 is a connection less protocol, so it does not report it either.

However, MPLS deals with failures in an efficient manner, because MPLS is connection oriented and it lies between layer 2 and layer 3, this means it can detect failure more quickly. Unlike OSPF, MPLS knows the information of current sessions and connections in network. This provide great aid to the node of MPLS network, in order to notify the network failure to the edge node and take the immediate restoration steps for those paths even before the network topology finally converges.

The main idea of this work is based on three criteria: reducing request blocking probability, minimizing network cost and load balancing. For the reduction of request blocking probability, Minimum Interference Routing Algorithm (MIRA) [3] is the best algorithm for reducing request blocking probability, but MIRA suffers from computational complexity problem, as this algorithm frequently computes maximum flow. While minimum network cost algorithms suffers from bad performance in terms of request rejection ratio in heavily loaded network, and load balancing gives undesirable results in lightly loaded network. So we can say that alone none of this approach is sufficient.

Therefore in this project, we propose an effective approach for enhancing the fault-tolerant performance based on all these criteria. When there is a failure in LSP, due link or node failure, then the proposed approach is initiated to perform the recovery of the carried traffic in failed LSP and the backup is chosen on the basis of above three criteria.

1.4 Benefits of MPLS

MPLS benefits include better performance, lower total cost of ownership, greater flexibility to accommodate new technologies better security and survivability. MPLS provide better performance by using classes of service and priority queuing, by this network know which traffic s important and ensures that it takes priority over other traffic. MPLS allow devices to handle IP traffic by enabling IP capacity on that devices and forward packets using pre-calculated routes that is not used in regular routings along explicit paths. MPLS interfaces to existing routing protocols such as RSVP and OSPF. In addition its supports IP, ATM, frame relay, L2 protocols. An MPLS-enabled network simplifies the overall network infrastructure with the convergence of multiple technologies. Enterprises can eliminate multiple, complex overlay networks and are able to transport a variety of new applications over the network using voice, video and data. Simplification of the network greatly reduces capital and operating costs.

MPLS promise a foundation that allows ISPs to deliver new services that are not supported by conventional IP routing techniques. In order to meet the growing demand of resources ISPs face changeless not only providing superior baseline service but also providing latest high quality service according to the modem need. Packet forwarding has been made easy and effective since router simply forward packet base on the fixed labels and support the delivery of services with quality of service (QOS) along with appropriate level of security to make IP secure while reducing overheads like encrypting data that is required the secure information on public IP networks.

II PROBLEM DISCUSSION AND PROPOSED SOLUTION

2.1 Problem Statement

When fault occurs in LSP, due to failure of link or node in the network, the carried traffic in failed LSP has to be transmitted through the backup LSP and the selection of backup LSP is based on the following criteria:

- Reducing the request blocking probability
- Minimizing cost of network
- Load balancing

2.1.1 Reducing request blocking probability

The major task of traffic engineering is to reduce the request blocking probability, to make sure that maximum numbers of requests are accepted in the network, in order to improve operator revenues and increase client satisfaction. Minimum Interference Routing Algorithm (MIRA) [3] is one of the best algorithms for constraint based routing which reduces the request blocking probability. The basic concept of MIRA is based on the relationship between the maximum flow [4] value between two nodes and the bandwidth (that can be routed between nodes). In MIRA critical links are the links, which cause a decrease in maximum flow values between pair of nodes. Therefore, weights are allocated to the links according to their criticality. In the end a shortestpath- like algorithm is used to evaluate the path with minimum critical links. But MIRA suffers from computational complexity problem, as this algorithm frequently computes maximum flow.

2.1.2 Minimizing costs of network

To accomplish a minimum cost of network, metrics like minimum hop count or link costs, have been conventionally included in routing algorithms. In order to minimize the cost of network many algorithms are proposed, for example Minimum hop algorithm [5]. Moreover, many other algorithms are proposed to make improvement in Minimum hop algorithm. Minimum hop algorithms are easy and computationally proficient. But in case of heavily loaded network, they give worse result [6] in terms of request refusal ratio. Link cost corresponds to the physical link length, so they are used in algorithms mainly for traffic engineering and they have no huge influence in networking architectures.

2.1.3 Load balancing

In network, load balancing plays an important role to decrease congestion. The basic concept of load balancing is to distribute load in such a way that improves the overall performance of network. But in lightly loaded network load balancing shows bad performance, for example routing packets on longer paths.

2.1.4 MIRA, Minimizing cost of network and Load Balancing

In this approach [6], three criteria (Load Balancing, MIRA and Minimizing cost of network) are used to calculate the path for the affected traffic. But this approach suffers from the problem computational overhead, because this approach computes all the three criteria throughout the process of packet forwarding.

2.2 Proposed Solution

Our proposed solution closely relates the integrated solution proposed by authors in [7]. In order to compute backup path by using above three criteria, the main challenge is to define the optimal weighting (W_1, W_2, W_3) for each element in the cost function given by (Eq. 1) [8]. Initially, the weight connected with MinHop should be increased, in order to show, its good performance under lightly loaded network. Therefore equation2 shows, weight (W_1) is inversely proportional to the total network load. So it can be that, weight (W_1) is predominant under lightly loaded network and it starts to decrease as the total network load increases to reach the total network capacity. Next, Minimum Interference Routing Algorithm (MIRA) comes into play, when links criticality is changing (links are getting rapidly loaded). So we see in equation 2 weight (W_2) is directly proportional to the network load. Finally, in equation 3 a new parameter for the load metric element that will control load balancing influence in the overall cost function by limiting its undesirable effects under light load. Moreover, constants a, b and c are used in order to scale the numeric values to a comparable range.

$$= \quad {}_1 + \quad {}_2 \qquad \times (\,) + {}_3 \times$$

()………………………………………………1

$$= \times \frac{\quad -}{\quad} \; ; \; = 16 \times \times \frac{-}{\quad} \; ; \; =$$

……………………………………………………………2

$$==$$

0 $\qquad < \hbar \;\; \hbar \;\; = (\,)/3$

$$\frac{\hbar(\,)}{(\,)} \qquad \hbar$$ ……………………….3.

In simulation, we can see that the performance of the method by authors is good in overall situation. The request blocking probability of the proposed scheme is comparable with MIRA results. The load standard deviation values are comparable with values for load balancing under light load. Under high load, the proposed scheme achieves performance bounded by load balancing (upper standard deviation) and MIRA (lower standard deviation) due to the equally combined effect of these algorithms. Finally, we see the influence of the MinHop element under light load; the integrating solution has good performance compared to MinHop. Therefore, these results justify our weighting approach. Even with a set of intuitive weights, we show the relevancy of the three objectives, and the benefit of their combination.

III SIMULATION AND ANALYSIS

3.1 Simulation Environment

In this project, before I decided which simulator environment, I would use, I looked at some different available network simulators like J-Sim, NS2 and AMPL. Then after studying all these simulators, I decided to go AMPL In this section I will describe why I choose AMPL as simulator environment in this project.

3.1.1 J-Sim

J-Sim [9] (formally known as Java Sim) is an open source component based network simulator written in Java, developed by Hung-ying Tyan and some other people at the Ohio State University. It provides MPLS support through a third-party extension, but it does not include the RSVP-TE signaling protocol. The documentation is available from the simulator's website, and it does include good descriptions of native code implementation, the philosophy behind the simulator and some tutorials and guides for new implementations.

Installation of the simulator, as it seems usual with Java applications, requires setting environment variables and compiling the source codes with third-party tools more common in Linux/Unix platforms, and then applying patches needed by the extensions such as MPLS. J-Sim is a dual language environment, where the user manipulates classes written in Java using Tcl scripts, much resembling NS-2. This poses the same problems related to NS-2, i.e., the need to know both Java and Tcl in order to use the simulator and implement non-existent characteristics.

3.1.2 NS2

NS-2 [10] is a discrete event simulator targeted at network research. It is open source, developed mainly by VINT project, Xerox PARC, UCB, USC/ISI, and contributions by several other researchers and users.

NS-2 is coded in C++ in a modular fashion. The user interfaces with the simulator using the object-oriented script language OTcl. It was conceived natively to run under Unix systems (including Linux), although it is possible to install it under Microsoft Windows.

MPLS and RSVP-TE are not available as standard libraries in NS- 2. They were implemented through contributions from other researchers. The MNS (MPLS for Network Simulator) module was developed by GaeilAhn [11][12], its original location no longer being available in the internet. This module contains MPLS and CR-LDP, but not RSVP-TE. The MNS module was further extended by [13] and [14][15] to include RSVP-TE functionalities. These modules cannot be obtained directly from their authors' websites, but only through request by email or from users who already own the modules.

NS-2 learning curve is significantly steep. One has to know the script OTcl language and learn how to build scripts that interface with the simulation objects coded in C++. The available documentation is not written in a didactic style, making it difficult for the beginner to build initial simulations without investing a considerate amount of time in trial and error. The documentation is especially poor for the MPLS and RSVP-TE modules, requiring the user to read the source code in order to learn how to interface with it and detect the offered capabilities. It is open, but implementation of new functions or modifications demand studying large portions of the source code. Generation of results and statistics is not automatic. One has to build a trace file from the simulation and perform a post processing on the file, calculating the desired statistics, by means of a processing language such as awk. Simulations can easily produce very large trace files, demanding significant post processing times.

Due to those characteristics and to the fact that the main module needed for the simulations, MNS, is yet not fully supported, NS-2 also stimulated the devise of the new simulator.

3.1.3 AMPL

AMPL (for A Mathematical Programming Language) is a modeling language for mathematical programming. It is a commercial utility but a student version with fair limitation exists and is free to be downloaded from www.ampl.com. AMPL can be used further to pass the model to a solver, free or commercial, to be optimized, that is to be maximized or minimized with respecting a number of constraints. It is important to note that AMPL does not solve the problem by itself; it is simply used to model a system and to pass the system to a solver. AMPL can define system variables, objective, and constraints. Constraints may be linear or non-linear, equalities or inequalities.

3.1.3.1 AMPL uses three file types:

- .mod files: are used to store system model containing constraints, objective and variables. Equations are written in AMPL syntax and can contain a large variety of functions and operators. Groups of similar constraints and variables can be formed easily. A model file must start with a model; command.
- .dat files: define system data like parameter values, initial points, etc. A data file must start with a data; command.
- .ampl files: are AMPL script files. They are sort of batch files that are used to encapsulate different commands and execute them one after another. The command file can be passed to AMPL as an argument. For example the command AMPL myModel.ampl executes AMPL commands in the file myModel.ampl

one by one.

3.2 Simulation

In the following, Minimum hop and MinLength refer respectively to a minimal hop [4] and a minimal length algorithm minimizing respectively the number of hops and the physical length of the chosen path. Whereas, MIRA and load balancing refer respectively to the approaches described in previous chapter. These algorithms are evaluated in a simulation environment. Traffic demands are uniformly distributed between all ingress/egress pairs and the associated bandwidth request is uniformly distributed between [0, 10] Kbps. We use an integer linear programming (ILP) approach to calculate the LSP route according to each algorithm. The objective (Eq. 4) is to find the path with minimal cost where the cost function is giver by (Eq. 7). Note that for MinLength the cost is equal to the link length, while the MIRA cost is consistent with the definition introduced in [3]. A flow conservation constraint (Eq. 5) ensures that the algebraic sum of the flows at each node is null except (Eq. 7) for the source and destination nodes of the LSP. Moreover, a capacity limitation constraint (Eq. 6) ensures that the resulting bandwidth on each link does not violate the edge capacity.

IV CONCLUSION AND FUTURE WORK

In this project, we identify relevant objectives for fault tolerance of MPLS network. We set up clear common criteria for these algorithms, namely: request reducing blocking probability, minimizing cost of network, and load balancing. We categorize and evaluate the appropriate approaches for this problem. The study shows the drawbacks of partial considerations, and the need for a global solution. Finally, we propose a solution that covers the different criteria presented in the project. Our formulation helps in clarifying all the trade-offs involved in CBR, thus enables the design of more complete solutions. Our approach shows that combination of our set of objectives achieves better overall satisfying results. The simulations presented in this project could be extended to encompass a discrete-event approach taking into account limited life-time LSPs. Moreover, the objectives we fixed can be the basis for further studies of CBR with emphasis on techniques for on-line design of survivable networks with multi-priority traffic.

V REFERENCES

[1] Wikipedia, “Internet backbone”. *Free encyclopedia of information* [Online]. Available: http://en.wikipedia.org/wiki/Internet_backbone .

[2] Xipeng Xiao Hannan, A. Bailey, B. Ni, L.M. “Traffic engineering with MPLS in the Internet”. *IEEE Network Magazine. Pub*. Mar/Apr 2000. Vol. 14. Issue. 2. pp. 28-33.

[3] K. Kar, M. Kodialam, T.V. Lakshman, “Minimum Interference Routing of Bandwidth Guaranteed Tunnels with MPLS Traffic Engineering Applications”, *IEEE Journal on Selected Areas in Communications*, December 2000.

[4] Jenn-Wei Lin *, Huang-Yu Liu. "Redirection based recovery for MPLS network systems". *In the Journal of Systems and Software*, April, 2010.

[5] J. Moy, "OSPF: Anatomy of an Internet Routing Protocol", *New York: Addison-Wesley*, 1998

[6] R. Guerin, D. Williams, A. Orda, "QoS routing mechanisms and OSPF extensions". *In Proc. Globecom*, 1997.

[7] Q. Ma, P. Steenkiste, "On path selection for traffic with bandwidth guarantees", *In Proceedings of IEEE International Conference on Network Protocols*, Atlanta, GA, October 1997.

[8] SamerLahoud, G´eraldineTexier, and Laurent Toutain. "Classification and Evaluation of Constraint-Based Routing Algorithms for MPLS Traffic Engineering". *In2 rue de la Chˆataigneraie - CS 17607 - 35576 CessonS´evign´eCedex– France*, 2004.

[9] Hung-yingTyan. "J-Sim Simulator". http://www.j-sim.org.

[10] NS Network Simulator. http://www.isi.edu/nsnam.

[11] GaeilAhn. MNS http://flower.ce.cnu.ac.kr/~fog1/mns/.

[12] GaeilAhn; Woojik Chun. "Design and implementation of MPLS network simulator supporting LDP and CR-LDP". *IEEE International Conference on Networks (ICON 2000)*, p. 441-446.

[13] Boeringer,R.RSVP-TEpatchforMNS/ns-2.http://www.ideo-labs.com/index.php?structureID=44.

[14] Adami, D. et al. "Signaling protocols in diffserv-aware MPLS networks: design and implementation of RSVP-TE network simulator". *IEEE Global TelecommunicationsConference (GLOBECOM'05)*, vol. 2, p. 792-796.

[15]Callegari, C. Vitucci, F. "RSVP-TE patch for MNS/ns-2".http://netgroup-serv.iet.unipi.it/rsvp-te_ns/.