Asian Info-communications Council

Document No. 38
22-26 November 2004
31st Conference (Shenzhen, China)

Working Group 1 (Technologies)

TITLE: A STUDY ON ADVANCED ROUTING AND DIFFSERV IN MULTIPROTOCOL LABEL SWITCHING NETWORKS

THEME: High Speed Network Technologies (ATM, FTTX, xDSL, PON, etc.)

SOURCE:

Tran Cong Hung (**P**osts & **T**elecommunications **I**nstitute of **T**echnology, Viet Nam)
E-mail: conghung@ptithcm.edu.vn

Nguyen Hoang Thanh (**P**osts & **T**elecommunications **I**nstitute of **T**echnology, Viet Nam)

E-mail: thanhnh@ptithcm.edu.vn

Summary:

Multiprotocol Label Switching_MPLS associates the strengths of both ATM and IP: fast switching in the core, efficient routing at the edge, ensuring the quality of service, provisioning and optimizing network resources. MPLS depends on IP routing. However, traditional IP routing does not solve completely the problem of traffic engineering and resource provisioning. With best effort, some parts in IP network are overloaded while the others are under-utilized. Consequently, we must extend the existing routing and signaling protocols for assuring quality of service and making the best use of network resources. We will go through main advanced routing issues in MPLS networks including constraint-based routing, explicit routing, fast reroute, traffic engineering and how to combine constraint-based routing and explicit routing to empower all MPLS strengths.

MPLS is hard to approach, but through network simulation, we can illustrate all the theory without a big invention to build a real MPLS network. Based on simulation in NS, we observe that advanced routing in MPLS can make the best use of network resources. Moreover, I also introduce Linux MPLS to unprofitable organizations like university environment in Vietnam.

First, we will make some MPLS basic definitions clear. Then, we will discuss the problem of traditional IP routing and how to expand the existing routing protocols to support traffic engineering. The next is MPLS DiffServ. And the last is MPLS advanced routing and DiffServ simulation. We also introduce an MPLS implementation in Linux.

Key words: explicit routing, constraint-based routing, fast reroute, traffic engineering, DiffServ, MNS (MPLS Network Simulation)

1. INTRODUCTION

Most of WANs in Vietnam are based on traditional IP routing which has a lot of weak points. Every node in IP network has to process two works: routing and forwarding. And the routing table is very big with hundreds of millions entries for the router to find the longest matching prefix address. That makes router work in slow performance when the traffic is overloaded. It leads to connection losing and traffic losing.

Internet is developing so far with a variety of value-added services to ensure security and quality of service. In recent years, telecommunication industry is looking up for a new switching technique that can associate the strong points of IP (routing) and ATM (switching). A model that has been considered is IP over ATM. This overlay approach enable IP and ATM to cooperate without changing its protocol. However, this approach is not scalable and flexible due to the fact that both end systems must maintain ATM and IP addressing. MPLS is an optimal solution.

2. ROUTING AND TRAFFIC ENGINEERING IN MPLS

MPLS can associate the strong points and overcome the weak points. Because of using shim header, it has the capability of fast switching in the core and efficient routing at the edge. MPLS also makes the best of network resources, support multicast and traffic engineering. In addition, the cost to deploy an MPLS network from existing networks is inexpensive.

A label identifies the path a packet should traverse. A label is carried or encapsulated in a Layer-2 header or between Layer-2 and Layer-3 header. The receiving router examines the packet for its label content to determine the next hop. Once a packet has been labeled, the rest of the journey of the packet through the backbone is based on label switching [26].

LIB (Label Information Base) is the database of all labels that an LSR creates and receives from MPLS neighbors for a certain prefix address. LFIB (Label Forwarding Information Base) is only a subset of LIB. It contains necessary information to forward a packet to the next hop. Edge router analyzes the IP header to decide which LSP to use. It adds a corresponding Label Switched Path Identifier and then forwards the packet to the next hop. Core routers just swap label based on its LFIB established before [1], [3], [11], [16].

Because the label distribution operation associates closely with routing operation, it is necessary to grasp the knowledge of IP routing protocols and Internet routing architecture ([6], [7], [8], [9], [10], [15]). However, traditional IP routing did not solve completely the problem of traffic engineering: some network parts are underutilized while the others are overloaded.

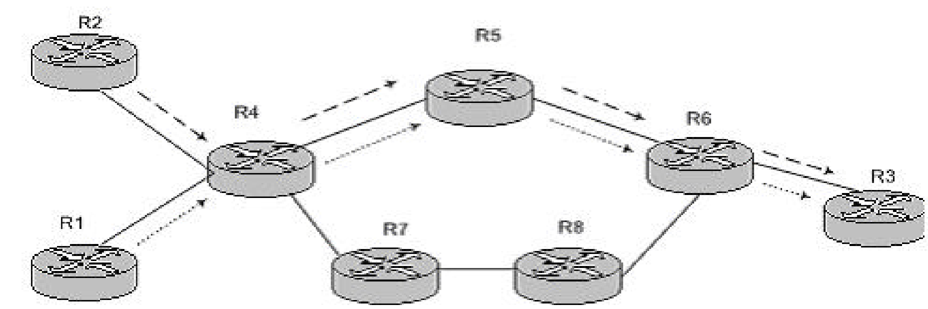


Figure 1: IP routing based on destination and lowest cost

Path from R2 to R3

Path from R1 to R3

Bandwidth of link (R4,R5), (R5,R6) is 100M, bandwidth of link (R4,R7), (R7,R8), (R8,R6) is 150M. Supposing that routing is based on hop count, traffic from R1 to R3 will follow the path (R4, R5, R6). Links along this path will be congested because bandwidth of each link is 100M while total bandwidth of R1 and R2 is 100 + 100 = 200M. At that time, all links along the path (R4, R7, R8, R6) is underutilized. Routing based on bandwidth also leads to the same fact that resource is underutilized along the path (R4, R5, R6) while load in path (R4, R7, R8, R6) is heavy because bandwidth of each link is 150M.

We observe that to support traffic engineering, besides the capability of optimal path finding based on a scalar metric, we must take account of current bandwidth of all links. IP routing just meets the first condition.

3. REROUTING

Fast reroute is a useful tool to support traffic engineering. MPLS operates like connection-oriented. Hence, it takes much time to establish new tunnel, especially when the current one is corrupt. MPLS has an efficient way to fast switch the corrupt path to a new one by switching to a pre-route LSP established before. MPLS uses label stack when a node or a link is down. [1]

We should modify the CSPF algorithm to find out K shortest paths first. In this way, we have prepared K entire paths. But if this path fails, MPLS will switch to an entirely new path.

But in some cases, we don't have to rebuild the entire ones, we just rebuild one part. When link from B to D is down, B pushes label 67 to its stack before sending packets to C, C swaps label 67 to 13, E removes label 13 and forwards packets to D. So when D receives packets labeled 67, D thinks that those packets were sent by B normally. D does not have to rebuild its LFIB.

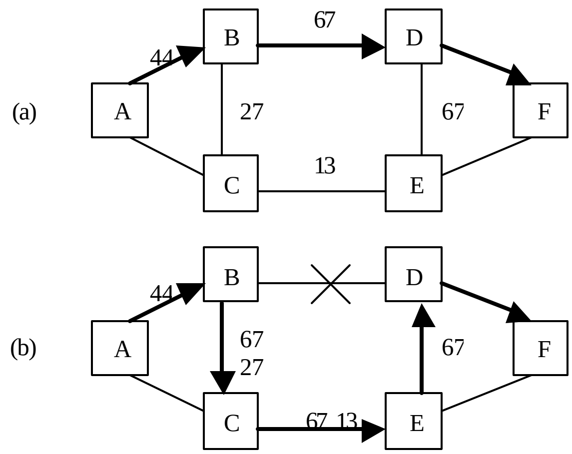


Figure 2: Label stack and switching protection

Another routing issue is dynamic routing. It ensures the bandwidth requirement of tunnels in the future. The main idea of dynamic routing algorithms is that new tunnel should not infer too much to paths that will have a big burden in the future.

TRAFFIC ENGINEERING

Advance routing associates strongly with and expand to traffic engineering.

First, we will go through the definition of Traffic Engineering (TE). TE is to optimize the network performance; it includes traffic oriented and resource oriented. Traffic oriented enhances the QoS of traffic streams (minimization of packet loss). Resource oriented pertains to the optimization of resource utilization (efficient management of bandwidth, minimizing congestion)([40],[17],[20], [21]).

Second, we talk about traffic and resource control. The traffic engineer acts as the controller in an adaptive feedback control system which includes a set of interconnected network elements (a network performance monitoring and measurement system, network configuration management tools). The traffic engineer observes the network state, characterizes the traffic and applies the control actions to the controller. A famous TE scheme is MATE (Multiple Adaptive Traffic Engineering) [40]. MATE avoids network congestion by balancing the load among multiple paths based on measurement and analysis of path congestion. The goal of ingress node is to distribute the traffic across LSPs so that loads are balanced and congestion is minimized. Ingress node includes these following components: Filtering and distribution, TE (monitoring phase and load balancing phase), Measurement and analysis (ingress node transmit probe packets periodically to egress node). The most important component of Traffic Engineering is **TE database**. [4]

4. DIFFSERV

We know that IntServ and DiffServ help solve the IP quality problem. Each service has particular strong points [1].

Table 1: IntServ vs. DiffServ

	IntServ	DiffServ
QoS ensuring	Per traffic stream	Per BA traffic (Class of service)
Configuration require	Per end-to-end session	In DS domains
How to classify traffic	Multi fields in header	DS Field
Ensuring period	Short time	Long time
Signaling	RSVP	No signaling protocol

Strong points	Ensure end-to-end quality and dynamic resource allocation	Simple to do, well done with IP architecture
Weak points	Hard to implement when the number of flow increases	Just ensure quality at each hop

In this paper, we just focus on DiffServ (DS). DS is a QoS architecture which marks packets based on user requirement and then applies different processing services to these packets. It uses the IP TOS (Type of Service) field (renames it the DS byte) to carry information about IP packet service requirements. It operates at Layer 3 only.

The main ideas of DiffServ are (a) to classify traffic at the boundaries of network and (b) to regulate (condition) this traffic at the boundaries. Packet will be assigned a value call DS code point (DSCP), containing specific value used to select a PHB. Nodes in the core just forward these packets based on this DSCP. A classifier selects packets based only on the contents of the DS field and service contract between customers and service provider specifying forwarding services a customer should receive. [23], [25]

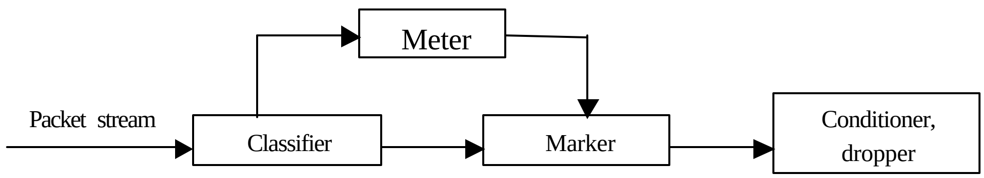


Figure 3: DS node model

Queue is a place where packets are stored and deleted. Scheduling is a process to decide which packets are served and which are dropped. We have a lot of objects queues: Drop-tail (a simple FIFO), FQ (a fair queue), SFQ (Stochastic Fair queuing), DRR (deficit round robin scheduling), RED (random early-detection gateways), CBQ (class-based queuing), CBQ/WRR (weighted round-robin). This is an example of WFQ [1].

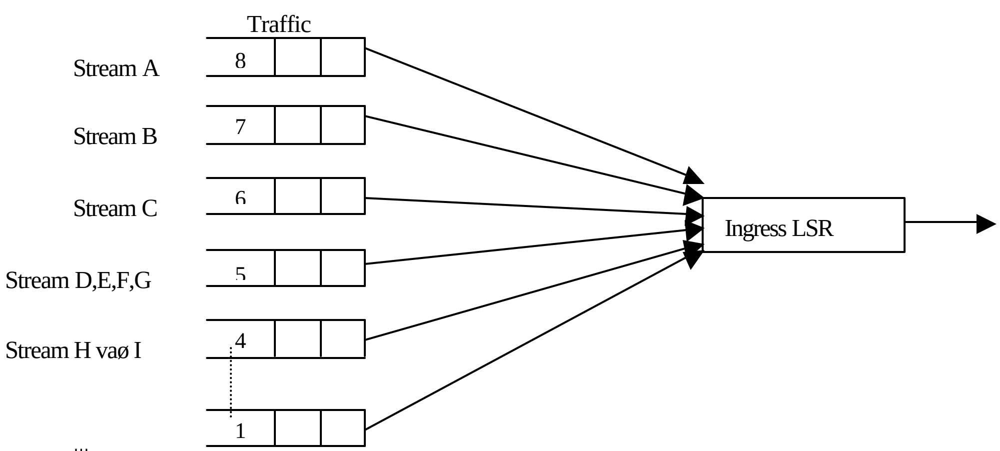


Figure 4: multi-stream MPLS for each class of

Four streams have level 5th priority; two streams have level 4, one stream for all the other levels. Weighted summary 8+7+6+5(4)+4(2)+3+2+1 = 55. Each stream in group D-E-F-G with level 5th priority take 5/55 total bandwidth (32,109 cells for each stream per second), each stream in group H-I with level 4th priority take 4/55 total bandwidth (25,687 cells for each stream per second)…

Per-Hop-Behavior (PHB) is the externally observable forwarding behavior applied at a DS-compliant node to a DS behavior aggregate. There are three kinds of PHB: BE, EF and AF. BE does nothing. EF has lowest delay and packet lost. AF provides many levels for packet ensuring. In case of congestion, DS node will destroy packets that have low priority. [1], [23], [24]

MPLS ensures the quality of packets strongly. It supports both Integrated Service and Differentiated Service. Next, we study how MPLS supports DiffServ. The job of MPLS network is to select and map between DS and MPLS LSPs.

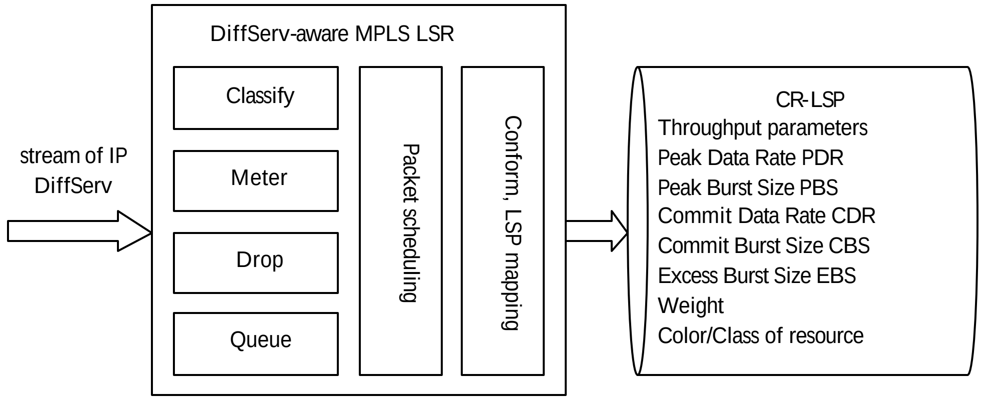


Figure 5: Model of LSR supporting DiffServ

DSCP in IP header is used only by ingress and egress LSP. Ingress router uses DSCP to decide how to encode MPLS label. DS node delays packets within a traffic stream to cause it to conform to some defined traffic profile which containing descriptions of the temporal properties of a traffic stream such as rate and burst size. Hence label selection decides how this traffic is treated.

To associate DSCP with MPLS, we must map the information in DSCP field into EXP filed of shim header because there are 6 bits in DSCP field and 3 bits in EXP field [1]. There are two kinds of LSP: E-LSP and L-LSP. In E-LSP, PHB is inferred from EXP, no signaling more but need EXP-PHB mapping configuration. L-LSP requires shim header and supports max 8 PHBs for each LSP. On the other hand, in L-LSP, PHB is inferred from label or label and EXP/CPL, Label-PHB mapping must be signaled at LSP establishment (by RSVP or LDP), one PHB for each LSP and it can be applied in shim header or header of data link layer. So which LSP should we choose? E-LSP is useful when resource is limited. Moreover, E-LSP model is resembled to DiffServ standard model. On the other hand, L-LSP is used in networks without shim header, supports many PHBs. Choosing E-LSP or L-LSP depends on specific network.

5. SIMULATION

With new technology, we should use simulator to illuminate behavior of algorithms and test the network performance in different environments and different kinds of topology, traffic model, different priority levels of traffic (best-effort, more than Best-effort, signaling traffic, real time traffic), network parameters (type of queue, length of queue, bandwidth, delay time). Moreover, we can monitor packet receiving, total received packets, number of out-of-order packets and graph, evaluate some statistics.

NS is open source simulation software on Linux, written in C++ and OTck. Information Science Institute in California has developed it. NS simulates IP networks including routing protocols and queue management. MNS has been implemented by extending NS. It is very useful for us to simulate various MPLS applications without constructing a real MPLS network. This is the conceptual model, architecture of MPLS node and the way MPLS nodes process traffic to ensure QoS request. [26], [27], [28].

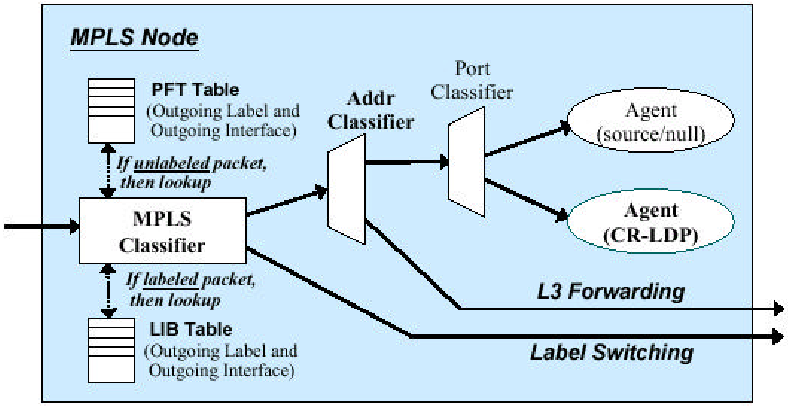


Figure 6: Architecture of MPLS node for label switching

To build a test-bed, we write a simulation program in Tcl and have to follow these steps. First, we have to create network topology including nodes, links, and queues at every node. Second, we create transport agent (TCP or UDP), bind it to corresponding node, and then connect them. Next, create application entity like FTP, CBR… Last, we write a script to schedule all the events [29], [32]. This is a simple script.

```
$ns at 1.2 "$LSRmpls4 setup-erlsp         8 4_5_6_8          3500"
$ns at 1.4 "$LSRmpls2 setup-erlsp         7 2_3_4_L3500_7    3600"
$ns at 1.6 "$LSRmpls2 bind-flow-erlsp     9 -1               3600"
$ns at 1.7 "$LSRmpls2 erb-dump"
$ns at 1.7 "$LSRmpls2 lib-dump"
$ns at 2.0 "$LSRmpls7 send-crldp-withdraw-msg  3600"
$ns at 2.0 "$LSRmpls8 send-crldp-withdraw-msg  3500"
```

We can monitor the LIB table at each node.

```
      --)At 1.7 seconds:
      --) ___ERB dump___ [node: 2] (--
----------------------------------------------------------------------
FEC   LSPid   LIBptr  SLIBptr  QoSid  aPATHptr iLabel iIface FailNext
7     3600      8       -1       -1      -1      -1      -1       *

      ___LIB dump___ [node: 2]
---------------------------------------------
     #      iIface     iLabel    oIface   oLabel    LIBptr   Linkerror?
     0:        -1         1         3        0         -1         -1
     1:        -1         2         5        0         -1         -1
     2:        -1         3         3        2         -1         -1
     3:        -1         4         5        3         -1         -1
     4:        -1         5         5        7         -1         -1
     5:        -1         6         3        6         -1         -1
     6:        -1         7         3        7         -1         -1
     7:        -1         8         5       10         -1         -1
     8:        -1        -1         3       11         -1         -1
```

We observe that with traditional IP routing based on hop counts, network resource is not utilized; some of them are overload while the others are underutilized. Network is congested, lead to packet loss in links along the path (1_3_5_7_9). The result has shown in figure 10.

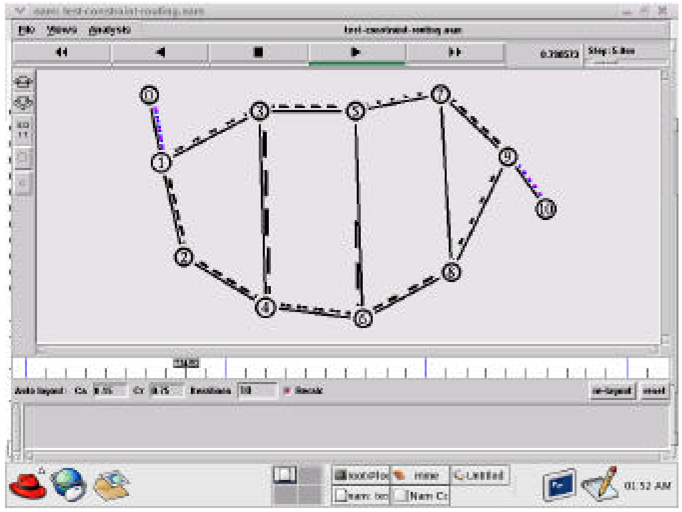


Figure 9: MPLS networks with constraint-based routing utilize network resource

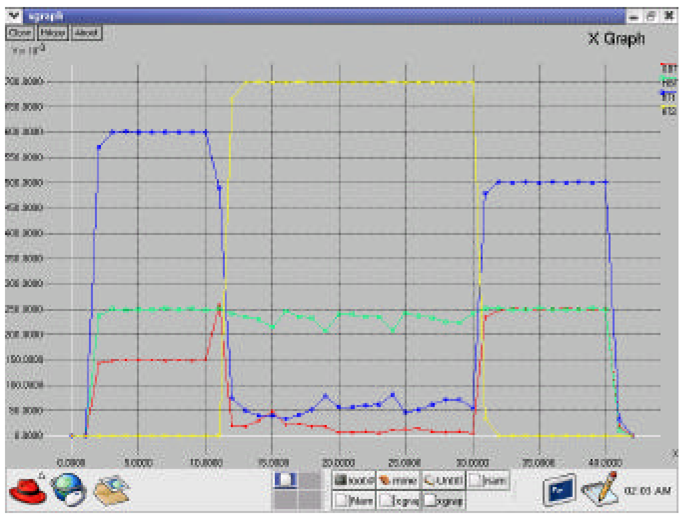


Figure 10: Network congestion lead to traffic lost in IP network

MPLS support constraint-based routing to find out the optimal path that meets the requirement and utilize networks resources. We use XGraph to illustrate bandwidth of each resource. Those graphs prove that MPLS advanced routing (constraint-based and explicit routing) can make the best use of network resources. The traffic load is distributed. Network performance is increased.

This is the initial parameter of our MPLS network which topology is in Figure 9. Bandwidth of links between nodes 0-3, 1-3, 9-10 is 3Mb, between 5-7, 4-6, 8-9 is 2Mb and the others is 1Mb.

```
$ns duplex-link $node0 $LSR1  3Mb  10ms DropTail
$ns duplex-link $LSR1  $LSR3  3Mb  10ms CBQ
$ns duplex-link $LSR3  $LSR5  1Mb  10ms CBQ
$ns duplex-link $LSR5  $LSR7  2Mb  10ms CBQ
$ns duplex-link $LSR7  $LSR9  1Mb  10ms CBQ
$ns duplex-link $LSR1  $LSR2  1Mb  10ms CBQ
$ns duplex-link $LSR2  $LSR4  1Mb  10ms CBQ
$ns duplex-link $LSR4  $LSR6  2Mb  10ms CBQ
$ns duplex-link $LSR6  $LSR8  1Mb  10ms CBQ
```

```
$ns duplex-link $LSR8  $LSR9  2Mb  10ms CBQ
$ns duplex-link $LSR3  $LSR4  1Mb  10ms CBQ
$ns duplex-link $LSR5  $LSR6  1Mb  10ms CBQ
$ns duplex-link $LSR7  $LSR8  1Mb  10ms CBQ
$ns duplex-link $LSR9  $node10  3Mb  10ms DropTail
```

There are four streams, the first stream request 1200kb, three streams requests 700kb. In this network, there is no path with bandwidth at least 1200kb in all links. So that it could not allocate for the first stream 1200kb. At 0.2, 0.4 and 0.6 second, three stream request 700kb. The first stream is allocated the LSP along the path 1-3-5-7-9-10, the second stream followed 1-2-4-6-8-9-10 and the last stream is distributed to path 1-3-4-6-5-7-8-9-10. In figure 11, the horizontal axis is time and the vertical axis is bandwidth of each stream. The first is Red, the three next are Green, Blue and Yellow. Each stream has bandwidth approximately 700kb. It means that we can distribute the network load and utilize the network resources.

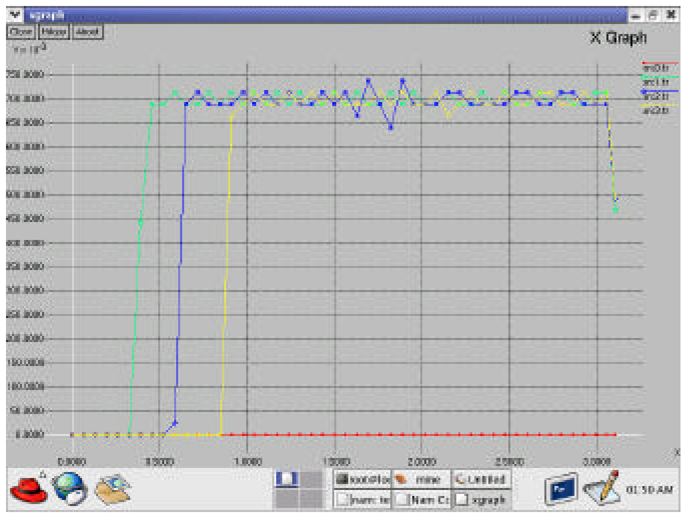


Figure 11: Flows with satisfied bandwidth in constraint-based routing in MPLS

MPLS supports DiffServ. In MNS, MPLS node includes a Service Classifier which forwards packets to different Packet Schedulers. ERB is Explicit Route information Table. There are different CBQ (Class Based Queuing) at each link.

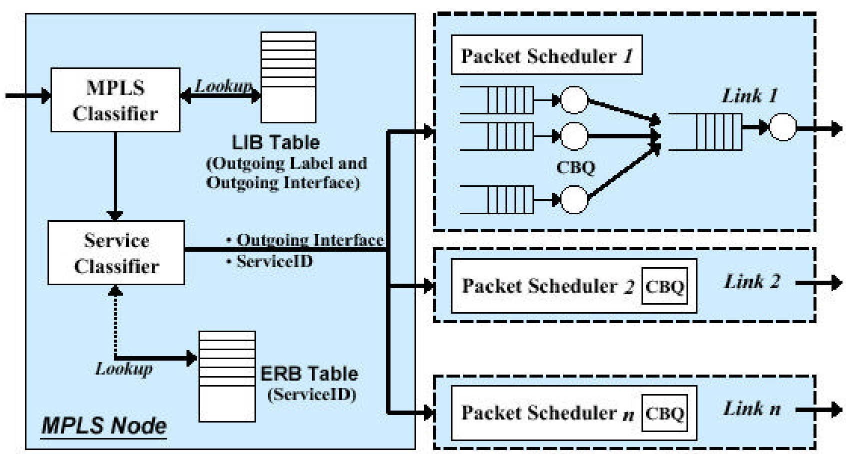
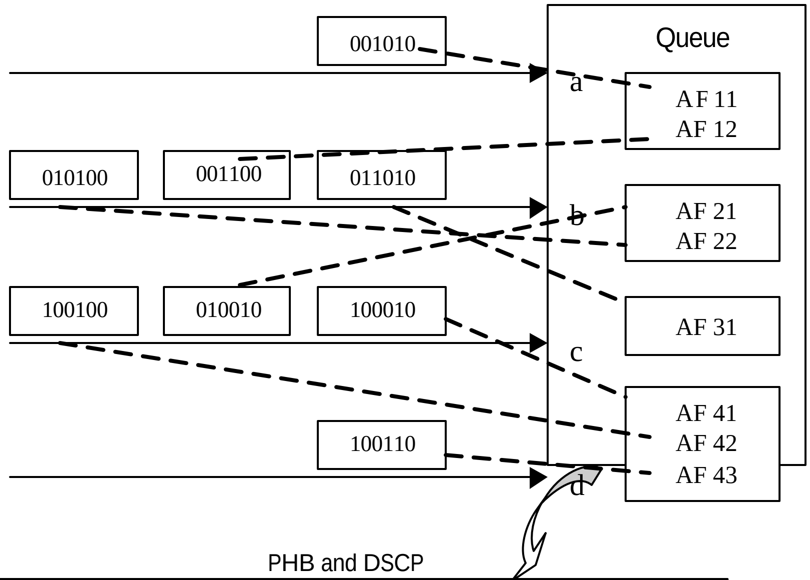


Figure 11: Traffic processing of MPLS node and link

Module DiffServ support four classes of stream (4 physical RED queues). Each class has three levels of packet dropping (3 virtual queues). Each combination of physical queue and virtual queue is associated with a scheduling plan and level of packet dropping. We can configure the parameters for these virtual queues.



AF11=001010 AF21=010010 AF31=011010 AF41=100010	AF12=001100 AF22=010100 AF32=011100 AF42=100100	AF13=001110 AF23=010110 AF33=011110 AF43=100110

Figure 12: PHB and DSCP mapping

NS DiffServ has three main components: Policy, Edge Router and Core Router. Policy includes classes of service defined by administrator for edge routers to mark incoming packets. There are five kinds of policy.

1. Time Sliding Window with 2 color markers (TSW2CMPolicier) uses CIR and two levels of packet drop precedence.
2. Time Sliding Window 3 color markers (TSW3CMPolicer): uses CIR, PIR and three levels of packet drop precedence. Medium level is used when packet rate is over CIR and lowest level is used when packet rate is over PIR.
3. Token Bucket (tokenBucketPolicer): uses CIR, CBS and two levels of packet drop precedence. Incoming packet is marked with low level only when it is bigger than the token bucket.
4. Single Rate Three Color Marker (srTCMPolicer): uses CIR, CBS, EBS and three levels of packet drop precedence.
5. Two Rate Three Color Marker (trTCMPolicer): uses CIR, CBS, PIR and PBS for three levels of packet drop precedence.

Class PolicyClassifier contains Policer table mapping a policy and initial code point to downgraded code point.

The second component is Edge Router, which marks packets by code point followed above defined policy. The last one is Core Router which checks the code point of pre-marked and forwards to according queues.

User defined policies can be added by modifying and recompiling NS source code. All policies are put in PolicyClassifier Class, including source node, destination node, type of meter, initial code point and other state information like CIR (committed information rate), PIR (peak information rate), bucket CBS, EBS, and PBS (byte), CBS: committed burst size, EBS: excess burst size, PBS: peak burst size, C bucket: current size of committed bucket, E bucket: current size of excess bucket, P bucket current size of peak bucket, incoming time of last packet, average sending rate, Length of Sliding Window

```
$ns simplex-link $edge $core 10Mb 5ms dsRED/edge
$ns simplex-link $core $edge 10Mb 5ms dsRED/core
```

```
set qEC [[$ns link $edge $core] queue]
# Initial parameters for queue from edge to core:
$qEC meanPktSize $packetSize
$qEC set numQueues_ 1
$qEC setNumPrec 2
$qEC addPolicyEntry [$s1 id] [$dest id] TokenBucket 10 $cir0 $cbs0
$qEC addPolicyEntry [$s2 id] [$dest id] TokenBucket 10 $cir1 $cbs1
$qEC addPolicerEntry TokenBucket 10 11
//Policer map from policy type, initial code point to downgrade code point
$qEC addPHBEntry 10 0 0
//mapping code point 10 to physical queue 0 and virtual queue 0
$qEC addPHBEntry 11 0 1
$qEC configQ 0 0 20 40 0.02
//queue, minimum threshold, maximum threshold, maximum packet dropping probability
$qEC configQ 0 1 10 20 0.10
set qCE [[$ns link $core $e1] queue]
# Set DS RED parameters from Core to Edge:
$qCE meanPktSize $packetSize
$qCE set numQueues_ 1
$qCE setNumPrec 2
$qCE addPHBEntry 10 0 0
$qCE addPHBEntry 11 0 1
$qCE configQ 0 0 20 40 0.02
$qCE configQ 0 1 10 20 0.10
$qE1C printPolicyTable
$qCE2 printCoreStats
```

We must establish and config parameters for DiffServ queues from edge router to core router, map code point to combination of physical and virtual queues, RED parameter like upper threshold, lower threshold and maximum packet dropping probability. This is the result after 40 seconds :

CP	*TotPkts*	*TxPkts*	*ldrops*	*edrops*
All	*249126*	*249090*	*21*	*15*
10	*150305*	*150300*	*0*	*5*
20	*98821*	*98790*	*21*	*10*

CP is code point, TotPkts is Total receiving packets, TxPkts: total sending packets, ldrops: number of dropped packets because of link overflow, edrops: number of dropped packets because of RED. In this scenario, the number of edrops packets is 5 for CP 10 and 10 for CP 20. We can also vary the value of Token Bucket, minimum threshold, maximum threshold, maximum packet dropping probability to monitor and calculate the number of early detect dropping packets.

Currently, NS have supported DiffServe for IP already but MNS have not supported DiffServ yet. So through my simulation in IP, we can prove the effect ion of DiffServ. It is an initialization for an integrated module DiffServ for IP in NS and MPLS in MNS. The new module will manipulate information from DSCP to label.

6. IMPLEMENTATION OF LINUX MPLS

Linux is one of the best networking operating systems and can operate as a router. It not only reduces hardware invention cost but also makes your network run stability. There is a variety of open source software on Linux like Zebra, Linux router project supporting multi routing protocols like RIP, OSPF, BGP … For a MPLS network, we must have some specific MPLS Cisco routers (platform at least 3500, IOS 12.3 or above). With Linux, after applying MPLS patch and changing network options, we just recompile the kernel, install MPLS Linux Project or RSVP-TE, which are supporting MPLS and integrated well with Zebra, IPsuite… It is really a useful tool for students to approach and research MPLS in case there is a lack of specific equipments. Moreover, there is a large community using MPLS Linux and willing to share their experience.

MPLS for Linux is distributed widely at [30]. It implements a forwarding plane and a set of signaling protocols for MPLS (RFC 3036) [19], including two main packets: mpls-linux and ldp-portable. It defines data structure ILM (Incoming Label Mapping) which contains label, opcode, FEC… to interpret incoming MPLS label. First, it extracts label from the top shim header, looks up label in ILM and applies different processing based on its different opcode like POP_AND_LOOKUP, POP_AND_FORWARD, NO_POP_AND_FORWARD, SEND_TO_RP…

7. CONCLUSION

As a conclusion, MPLS is a hybrid technology. It combines the best of IP and ATM. MPLS solves the problem of traffic engineering by combining constraint-based routing and explicit routing. Moreover, it can cooperate with and easy-to-upgrade from existing ATM networks. MPLS ensures the quality of service and utilizes network resources but the cost to deploy is inexpensive. MPLS is really a promising solution to apply for backbone network in Vietnam.

We have analyzed the main routing issues in MPLS networks including constraint-based routing, explicit routing, traffic engineering and fast reroute. MPLS is hard to approach because of specific infrastructure and network devices. Based on simulation in NS, We can illustrate that advanced routing in MPLS can make the best use of network resources. Through NS, we can illustrate all the theory without a big invention to build a real MPLS network. On the other hand, we ourselves can implement a simple MPLS networks base on MPLS Linux. I think this solution is a new promising way and suitable to approach MPLS in unprofitable organizations like university environment in Vietnam.

There are different ways to support quality of service. In this paper, we can conclude that MPLS supports DiffServ operations very well. With DiffServ, customers can use telephony and data services with point-to-point insurance for real-time applications like high quality videoconference, VoIP, e-learning. DiffServ-TE automatically chooses a route in pre-defined services like premium, gold, silver, bronze for each type of customer services.

Currently, MNS have not supported DiffServ yet. So our next goal is to build a new module that manipulates information from DSCP to label, so that NS can support MPLS DiffServ.

References:

[1]. UYLESS BLACK, ***MPLS Label Switching Network***, *PRENTICE HALL*, 2002

[2]. VIVEK ALWAYN, ***Advanced MPLS Design and Implementation***, *Cisco Press*, 2002

[3]. SEAN HARNEDY, ***MPLS Primer***, *PRENTICE HALL*, 2002

[4]. BRUCE DAVIE, YAKOV REKHTER, ***MPLS Technology and Application***, Academic Press, 2000

[5]. JAMES REAGAN, ***CCIP MPLS Study Guide***, *CIBEX*, 2002

[6]. STEPHEN THOMAS, ***IP Routing Essential and Architecture***, *WILEY*, 2002

[7]. Bassam Halabi , ***Internet Routing Architecture***, *Cisco Press*, 1997

[8]. ROBERT N. MYHRE, ***CCNP 2.0 routing***, *PRENTICE HALL*, 2001

[9]. JEFF DOYLE, ***TCP/IP Routing Vol I***, *Cisco Press*, 1998

[10]. ***Cisco Network Academy Program***, *Cisco Press*, 2001

[11]. Christopher Y.Metz, ***IP Switching***, *McGraw-Hill*, 1999

[12]. Peter Tomsu, ***MPLS-based VPN***, *PRENTICE HALL*, 2002

[13]. Andrew S. Tanenmaum, ***Computer Networks***, *PRENTICE HALL*, 1996

[14]. Douglas E. Comer, **Internetworking with TCP/IP: Principles, Protocols and Architecture**, PRENTICE HALL, 2000

[15]. Thomas M. Thomas II, **Cisco Internetwork Design**, MacGraw-Hill, Technical Expert, 2001

[16]. "**Multiprotocol Label Switching Architecture**", RFC 3031.

[17]. "**Requirements for Traffic Engineering Over MPLS**", RFC 2702.

[18]. "**Constraint-based LSP setup using LDP**", RFC 3212.

[19]. "**MPLS Loop Prevention Mechanism", RFC 3063**.

[20]. "**RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209**

[21]. "**Constraint-Based LSP Setup using LDP", RFC 3212**

[22]. "**A Framework for QoS-based Routing in the Internet**", RFC 2386.

[23]. "**An Architecture for Differentiated Services**", RFC 2475.

[24]. "**MPLS Support of Differentiated Services**", RFC 3270.

[25]. "**A provider Architecture for Differentiated Services and Traffic Engineering**", RFC2430.

[26]. http://www.iec.org/online/tutorials/mpls/index.html

[27]. http://www.isi.edu/nsnam/ns/

[28]. http://perform.wpi.edu/NS/
[29]. http://flower.ce.cnu.ac.kr/~fog1/mns/mns2.0/source/mns_v2.0.tar
[30]. http://mpls-linux.sourceforge.net/
[31]. http://www.lathspell.de/linux/uml/
[32]. http://charlie.it.uts.edu.au/~tkaphan/xtn/capstone/
[33]. http://www.mplsrc.com
[34]. http://www.cisco.com/warp/public/104/
[35]. http://www.convergedigest.com/tutorials/
[36]. http://cell-relay.indiana.edu/mhonarc/mpls/
[37]. http://www.juniper.net/techpubs/
[38]. http://ww.ietf.org
[39]. http://www.nanog.org/mtg-0402/soricelli.html
[40]. http://web.mit.edu/6.263/www/EJLW01.pdf
[41]. http://flower.ce.cnu.ac.kr/~fog1/mns/mns2.0/doc/MNS_v2.0_arch.pdf

Author's Profile

TRAN CONG HUNG was born in VietNam in 1961
He received the B.E in electronic and Telecommunication engineering with first class honors from HOCHIMINH university of technology in VietNam, 1987.
He received the B.E in informatics and computer engineering from HOCHIMINH university of technology in VietNam, 1995.
He received the master of engineering degree in telecommunications engineering course from postgraduate department HaNoi university of technology in VietNam, 1998.
He has recently finished his Ph.D at HaNoi university of technology in VietNam, 2004.
His main reseach areas are B– ISDN performance parameters and measuring methods.
Currenthy, he is a lecturer, deputy head of Faculty of Information Technology II and head of section Network & Data Transmission in Posts and Telecoms Institute of Technology (PTIT), in HOCHIMINH City, VietNam.

Author's Profile

NGUYEN HOANG THANH was born in VietNam in 1981
She has received the B.E in Information Technology from PTIT, 2003 with graduation thesis entitled "Advanced Routing in Multiprotocol Label Switching Networks". She won the Consolation Award of Students' Science Research Movement given by Minister of Education and Training, 2003, First Prize in Students' Science Research Movement, Posts and Telecommunications Institute of Technology, 2002. Science Research project: Managing Network Devices remotely based on Dynamic Agent, Voice portal using VXML (or Accessing internet by telephone), 2003, Implementation a Local Exchange of 4 subscribers, 2001. Currently, she is a teaching assistant of Faculty of Information Technology II and in section Network & Data Transmission in Posts and Telecoms Institute of Technology (PTIT), in HOCHIMINH City, VietNam.