# RDM TELIC (An Automated Traffic Engineering Algorithm by Link Coloring)


*By*

*Muhammad Salman Zafar*

**Department of Software Engineering**
**National University of Science & Technology Islamabad**
**April 2010**

**In the name of ALLAH, the most Beneficent, the most Merciful.**

# ABSTRACT

Traffic engineering is a mechanism for optimizing the performance of a network by dynamically analyzing, predicting and regulating the flow of data transmitted over that network. In the last decade many standards and protocols have been introduced in the same domain by Internet Engineering Taskforces (IETF). QOS is another factor that cannot be overlooked at the same time besides resource optimization. MPLS with Differentiated service provide the dual flavor by ensuring Quality of Service and efficient network utilization.

Traffic Engineering with Link Coloring (TELIC), proposed in year 2002, is an algorithm to automate the process of Traffic Engineering for MPLS aware Diffserv domain. TELIC assigns colors to links in the network domain and updates link colors on the basis of reservable bandwidth. IETF introduced bandwidth allocation models Maximum Allocation Model (MAM) and Russian Doll Model (RDM) in year 2004. MAM refers to segregation/isolation of class types with no channel pre-emption whereas RDM refers to the aggregation by allowing lower classes to use bandwidth of higher class on availability and pre-empted when required. TELIC automates Traffic Engineering process by class type segregation (just like MAM) through link colors with inherent feature of pre-emption. Pre-emption makes TELIC distinct from MAM however TELIC defines no rule for premium classes (i.e. EF, AF) to move on the same link.

Current work introduces enhanced version of TELIC, while making it compliant with RDM. This is done by establishing traffic mixing i.e. lower classes can share the bandwidth of higher classes. The induction of traffic mixing trounce the missing part of TELIC and introduces new rule set for conjunction limits, link colors and bandwidth constraints thus automate the process of Traffic Engineering.

*Dedicated to My Family*

# *Acknowledgements*

All thanks to Allah Almighty, the benevolent and compassionate, who blessed me with the power and capabilities and remained contented with all intricacies found during the successful completion of this project.

I am greatly thankful to my parents for there prayers and selflessness showed for me in my whole educational career. My brothers and sister, wife and children whose love and courage always show me light in difficult situations.

I do not wish to merely complete a formality when expressing my gratitude and admiration for my Ex-supervisor ***Dr. Shaleeza Sohail*** and my supervisor ***Dr. Aasia Khanum***. Apart from their skilled supervision and invaluable help in the project work, I would especially like to thanks ***Dr. Junaid Zubari*** (joint Supervisor) for his continued help throughout the project. At the same time I equally express my respect and regards to the all the committee members for their time and diligence with whom I had quite a few good discussions and their never ending queries sharpened my own concepts.

Finally, I express special thanks to my friends, especially Mr. Khurram Shahzad, for providing unconditional and continuous support during time of crisis.

Thanks to every body.

**Muhammad Salman Zafar**

# Table of Contents

## LIST OF FIGURES



## LIST OF TABLES



## LIST OF ALGORITHMS

# *Chapter 1*

## 1.1 Introduction

### 1.1.1 Abstract

Traffic engineering is a mechanism for optimizing the performance of a network by dynamically analyzing, predicting and regulating the flow of data transmitted over that network. In the last decade many standards and protocols have been introduced in the same domain by Internet Engineering Taskforces (IETF). QOS is another factor that cannot be overlooked at the same time besides resource optimization. MPLS with Differentiated service provide the dual flavor by ensuring Quality of Service and efficient network utilization.

TELIC, proposed in year 2002, is an algorithm to automate the process of Traffic Engineering for MPLS aware Diffserv domain. TELIC assigns colors to links in the network domain and updates link colors on the basis of reservable bandwidth. IETF introduced bandwidth allocation models MAM (Maximum Allocation Model) and RDM (Russian Doll Model) in year 2004. MAM refers to segregation/isolation of class types with no channel pre-emption whereas RDM refers to the aggregation by allowing lower classes to use bandwidth of higher class on availability and pre-empted when required. TELIC automates Traffic Engineering process by class type segregation (just like MAM) through link colors with inherent feature of pre-emption. Pre-emption makes TELIC distinct from MAM however TELIC defines no rule for premium classes (i.e. EF, AF) to move on the same link.

Current work introduces enhanced version of TELIC, while making it compliant with RDM (Russian Doll Model). This is done by establishing traffic mixing i.e. lower classes can share the bandwidth of higher classes. The induction of traffic mixing trounce the missing part of TELIC and introduces new rule set for conjunction limits, link colors and bandwidth constraints thus automate the process of Traffic Engineering.

### 1.1.2 Introduction

The penetration of broadband results in development of numerous internet applications. World is moving toward the centralization. Various application have been developed which are internet enabled. Some of them can process the request in batches like emails, however lot of application require real time updates some of the example of such applications are stock-exchange databases, online games, Business intelligent applications etc. This is not just stopped yet the entire telecommunication infrastructure (NGN) is already moved on IP network. Due to the intense usage of real time internet application it is the need of hour that connectionless networks should be designed in such a way that they work like connection oriented circuits with guaranteed bandwidth while providing compliance with Service level agreements (SLA's) and proper network utilization.

In order to cope up with the emerging and most demanding requirements on Internet different protocols, compression techniques and security architectures are developed and research is going for getting further mature results. IETF and different study groups are working on developing new protocols and standards in order to meet the requirement of real time applications. Among others, some protocol provide quantitative guarantee to the flows like RSVP (Resource Reservation Protocol) while some have addressed the qualitative guarantees by defining behavior aggregates i.e. Diffserv (Differentiated Services architectures), however MPLS bring the new concept in QOS frameworks which ultimately leads toward Traffic Engineering.

**1.2 Problem Statement**

It is aimed to develop an algorithm which automates the process of Traffic Engineering i.e. to optimize the links usage, router paths while serving user demands. The broader network can be divided into core domain and access domain. Core domain deal with the traffic aggregates, resulting in more predictive behavior and well-behaved network as compared to the access domain. On the other hand access domain has to provide real time usage to the users and have limited bandwidth pipes, which results in un-predictive behavior on the network creating lot of management headache for network

administrators. Users always expect to have well-behaved network for access domain as well.

In the class-based Internet, users are able to apply Diffserv tags to the packets that enter the network. The packets are treated as per the Diffserv class tags because each router in the network deploys CBQ (class-based queuing) or a variation of this scheme serving the premium packets first before the others. In class-based queuing, effectively a router would deal with separate queues, each confirming to its defined class of service. The incoming traffic is placed in the appropriate queue as per the service tags. Each class receives a minimum bandwidth, and if it exceeds its limits the traffic associated with the class is suspended in its queue until the class throughput is within the limits agreed upon.

MPLS is a switching technology using labels. In MPLS network, incoming packets are assigned a “label” by a LER (Label Edge Router) as according to their forward equivalence class (FEC). Packets are forwarded along LSP (Label switching Patch) where each "label switch router (LSR)" makes forwarding decisions based solely on the contents of the label, eliminating the need to look for its IP address. At each hop, the LSR takes off the existing label and applies a new label for the next hop. Next hop also decides how to forward the packet by reading just the label on the packet. These established paths, Label Switch Paths (LSPs) can guarantee a certain level of performance, to route around network congestion, or to create IP tunnels for network-based virtual private networks. In many ways, LSPs are similar to circuit switched paths in ATM or Frame Relay networks, except that they are not dependent on a particular Layer 2 technology.

In a Diffserv-enabled MPLS access domain, the constrained routing algorithms work in conjunction with MPLS to carve out tunnels that may contain uniclass traffic. The labels are distributed accordingly and transmission starts when the LSP tunnels are laid out. MPLS traffic engineering employs "constraint-based routing," in which the path for a traffic flow is the shortest path that meets the resource requirements (constraints) of the traffic flow. Tunnel paths are calculated at the tunnel head based on a fit between

required and available resources (constraint-based routing). The IGP automatically routes the traffic into these tunnels. Typically, a packet crossing the MPLS traffic engineering backbone travels on a single tunnel that connects the ingress point to the egress point. The main purpose of this work is to automate the process of traffic engineering in MPLS aware DiffServ domain.

## 1.3 Stakeholder Descriptions and Scope

Stakeholders are the network administrator, who implements the algorithm on the access domain.

### 1.3.1 Scope

#### 1.3.1.1 Functions

- ✓ Accept the traffic set.
- ✓ Accept the number of hops and links in the domain.
- ✓ Establish LSPs on the network on different suitable paths.
- ✓ Make TELIC compliant with RDM.

#### 1.3.1.2 Performance and constraints

- ✓ Should work on the static traffic set however it has capability to work with real time traffic.
- ✓ Domain considered in the project is configurable by network administrator on initial setup.
- ✓ Quickly and efficiently LSPs formalization of the given traffic sets on the sub domain from ingress to egress.
- ✓ Performance of the algorithm is increased multiple as compared to SHORTD algorithm.
- ✓ Enhanced admission control by the introduction of link coloring mechanism.
- ✓ Bandwidth to be allocated is taken in term of percentage for materializing the algorithm.

The novel-based console for testing the algorithm is developed in C++. At any point of time user can check the available bandwidth and can define the new domain and traffic set as per the requirement.

### 1.4. Objectives

The main Objective of the algorithm is to make TELIC complaint with RDM (Russian Doll Model), the detail of RDM can be found in subsequent chapters. The additional objective includes:

- ✓ Limiting the bandwidth of the higher class.
- ✓ Lower classes can use the bandwidth of higher classes.
- ✓ Conjunction Degree has been defined to simulate a controlled network in order to avoid delay and jitter on links.

#### 1.4.1 Requirements to Run the Algorithm

##### 1.4.1.1 Developer's Requirements

- ✓ Operating System

  Microsoft® Windows™, Linux, Unix
- ✓ Development Tools

  Microsoft® Visual Studio 6.0, and other C++ editor
- ✓ Other Software

  Microsoft Office.

#### 1.4.2 Algorithm Positioning

Numerous researches have been carried on the part of Traffic Engineering. One of the efforts was done in Year 2002 by introducing and algorithm for the automation of traffic engineering before the introduction of any bandwidth allocation model. Two models i.e. MAM and RDM were introduced in year 2004. The effort was done on automatic the existing Traffic Engineering algorithm making it compliant with RDM (Russian doll model).

One of the most important features of the project is automation of Traffic Engineering in order to create the programmed environment for network administrator for improving efficiency and SLA compliance.

# *Chapter 2*

## 2.1 Introduction

Traffic engineering is also known as Traffic Management. The methods of traffic engineering can be applied to all kind of networks, including the PSTN (Public Switched Telephone Network), LANs (Local Area Networks), WANs (Wide Area Networks), Cellular Networks, Proprietary Business and the Internet.

The theory of traffic engineering was originally conceived by A.K. Erlang, a Danish mathematician who developed methods of signal traffic measurement in the early 1900s. Traffic engineering makes use of a statistical concept known as the law of large numbers (LLN), which states that "As an experiment is repeated, the observed frequency of a specific outcome approaches the theoretical frequency of that outcome over an entire population". In telecommunications terms, the LLN says that the overall behavior of a large network can be predicted with reasonable certainty even if the behavior of any single packets cannot be predicted.

When the level of network traffic nears, reaches or exceeds the design maximum, the network is said to be congested. In a telephone network, traffic is measured in call seconds (CCS) or erlangs. One CCS is equal to 100 seconds of telephone time. One erlang is equal to one hour or 36 CCS of telephone time. In a congested network, one of three things can happen when a subscriber attempts to send a message or place a call:

- The user receives a busy signal or other indication that the network cannot carry out a call at that time.
- A message is placed in a queue and is eventually delivered according to specified parameters.
- A message is rejected, returned or lost.

When message queues become unacceptably long or the frequency of busy signals becomes unacceptably high, the network is said to be in a high-loss condition. A major objective of traffic engineering is to minimize or eliminate high-loss situations. In

particular, the number of rejected messages or failed call attempts should be as close to zero as possible. Another goal of traffic engineering is to balance the QOS (Quality of Service) against the cost of operating and maintaining the network.

## 2.2 Quality of Service Architectures

QOS for IP networks leads us to two different approaches. One is integrated service Architecture which is per flow oriented and the other Differentiated Service architecture which is class oriented. Intserv architecture uses Resource Reservation protocol (RSVP). Diffserv architecture uses DSCP (Differentiated service code point) for defining per hop behavior. Intserv can be used with Diffserv to provide scalability in the per-flow behavior of RSVP. MPLS is another most popular mechanism for the provision of QOS on the telecom networks. The details of above mentioned architecture is mentioned as under.

### 2.2.1 Resource Reservation Protocol (RSVP)

RSVP was originally designed with multicasting applications in mind. Current Internet multimedia applications such as Real audio-video, *vic* (video-conferencing tool), *vat*/*rat* (audio-conferencing tools) have more receivers than senders. For example, a NASA shuttle launch is viewed worldwide over Mbone (multicasting backbone). Typically, unicast communication is handled as a degenerate case of multicast. RSVP has been designed to accommodate heterogeneous receiver systems i.e. Receiver Oriented. The receiver-oriented design caters for diverse receiver requirements. Say, for example, in a multicast session with multiple senders, one receiver could be interested in a particular sender whereas another receiver could be interested in all senders. A receiver may modify its requested QoS anytime. This can also happen in response to a sender's modification of its traffic characteristics (TSpec). A new sender can start sending to a multicast group and may need a larger reservation. Also, a new receiver joining a multicast group may request a different QoS.

#### 2.2.1.1 Reservation Methods

Reservation style indicates to the network element that an aggregation of reservation request is possible for a multicast group. Resource reservation controls how much

bandwidth is reserved, whereas reservation filter determines the packets that can make use of this reservation. RSVP supports three styles of reservation.

- ✓ Wildcard Filter
- ✓ Shared Explicit
- ✓ Fixed Filter

#### 2.2.1.1.1 Wildcard Filter

The wildcard-filter (and shared explicit) style reservation is suitable for multicast sessions. In wildcard-filter sources are not likely to send information at the same time. Typically, audio applications are suitable for this style since only a limited number of participants can converse with each other simultaneously. A reservation slightly exceeding the requirements for a single speaker (for over speaking and interjections) will be sufficient for this style.

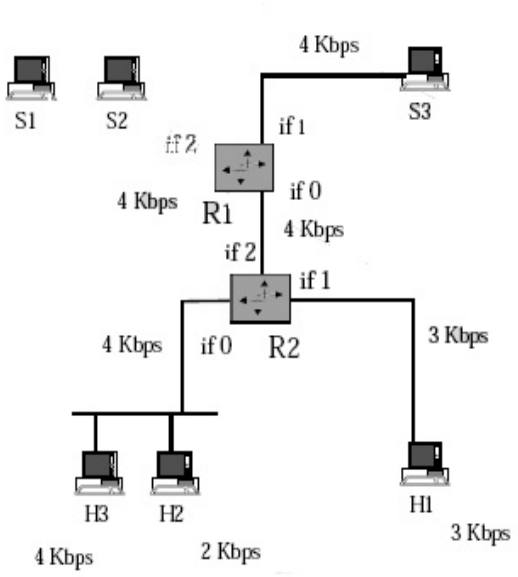


***Figure 2.1 (Wildcard Filter) [10]***

Multiple senders are required to coordinate the use of shared bandwidth. RSVP protocol doesn't take care of the conference control and floor control issues. We look at the wildcard-filter style using an example in Figure 2.2. These examples use a rate of Kbps for simplification (in reality token bucket parameters are used). The example uses a multicast session with three senders S1, S2, and S3 and three receivers H1, H2, and H3.

The senders S1 and S2 as well as receivers H1 and H2 are shown to be on a LAN segment capable of implementing traffic priority schemes. Following are the requirements of receivers:

Reservations for H2 and H3 are merged—4 Kbps (if0, R2). Another request comes from H1 on if1 at R2 for 3 Kbps. R2 sends a merged request via (if2, R2) of 4 Kbps. It is the larger of (if1, R2) 3 Kbps and (if0, R2) 4 Kbps. Router R1 forwards a 4 Kbps request on if1 and if2 (to S1, S2, and S3). An important point to note here is that the source is not identified and that merger of requests at routers does not use the sum of the incoming requests, but takes the larger of the two values.

#### 2.2.1.1.2 Shared Explicit

The shared-explicit-filter style reservation is similar to wildcard-filter, with the only difference that senders are identified. The reservation is shared among all senders in the list.

Figure 2.3 shows an example of the shared-explicit-filter style reservation. In this case, following are the requirements of receivers:

- ✓ H1 wants to reserve 1 Kbps for S1 and S2.
- ✓ H2 wants to reserve 3 Kbps for S1 and S3.
- ✓ H3 wants to reserve 2 Kbps for S2.

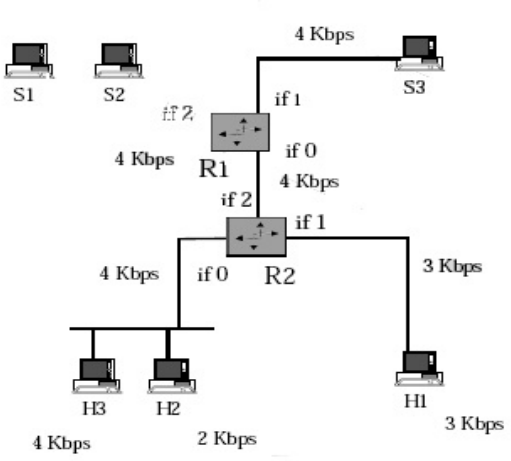


***Figure 2.2 (Shared-Explicit Filter) [10]***

A reservation for sources S1, S2, and S3 from H2 and H3 on (if0, R2) is merged to 3 Kbps. Another request comes from H1 on if1 of R2 for 1 Kbps to S1 and S2. The requests on if0 and if1 of router R2 are merged and forwarded on if2 as 3 Kbps for S1, S2, and S3. The requests received on if0 of router R1 are forwarded as follows:

- ✓ on if2 3 Kbps for S1 and S2.
- ✓ on if1 3 Kbps for S3.

#### 2.2.1.1.3 Fixed Filter

The fixed-filter style reservation is suitable for applications such as videoconferencing, where one window is required for each sender and all these windows need to be updated simultaneously. Fixed-style reservation requires that receivers identify the source from which they want to receive the reservation along with the bandwidth required. Bandwidth is not shared (between sources), since reservations are made for a particular source.

Figure 2.4 shows how the fixed-filter style reservation can be used. Following are the requirements of receivers:

- ✓ H1 wants to reserve 3 Kbps for S1 and 4 Kbps for S2.
- ✓ H2 wants to reserve 2 Kbps for S1 and 2 Kbps for S3.
- ✓ H3 wants to reserve 1 Kbps for S1.

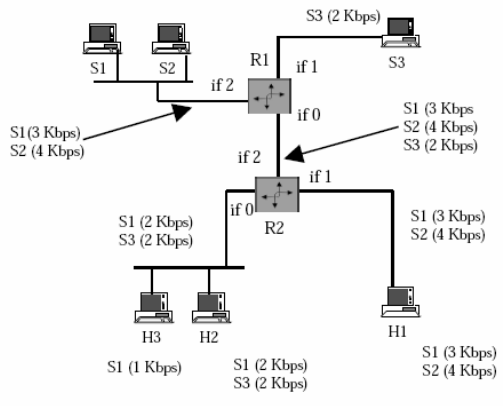


***Figure 2.3 (Fixed Filter) [10]***

The reservation for source S1 from H2 and H3 is merged to 2 Kbps at (if0, R2). The reservation for source S3 of 2 Kbps from H2 arrives at (if0, R2). Another request comes from H1 on if1 of R2 for 3 Kbps to S1 and 4 Kbps to S2. The requests on if0 and if1 of router R2 are merged and forwarded on if2 as follows:

- ✓ Kbps for S1;
- ✓ Kbps for S2;
- ✓ 2 Kbps for S3.

The requests received on if0 of router R1 are forwarded as follows:

- ✓ on if2 3 Kbps for S1 and 4 Kbps for S2;
- ✓ on if1 2 Kbps for S3.

#### 2.2.1.2 RSVP Messages

RSVP has many message types, the two most important are *Path* and *Resv* messages. Other message types include reservation confirmation message, error report messages and reservation and path teardown messages. RSVP messages travels hop-to-hop. The next hop is determined by the routing table. The router maintains where the messages came from and maintain their states which is also called route pinning.

### 2.2.2 Differentiated Service Architecture

Diffserv is the architecture for providing scalable service differentiation on the Internet [9]. A “Service” defines some significant characteristics of packet transmission in one direction across a set of one or more paths within a network. Diffserv is scalable because the packets entering a network are classified only at the ingress routers. The remaining routers in the network provide treatment to these packets according to their classification. The architecture of DiffServ is shown in the below mentioned diagram.

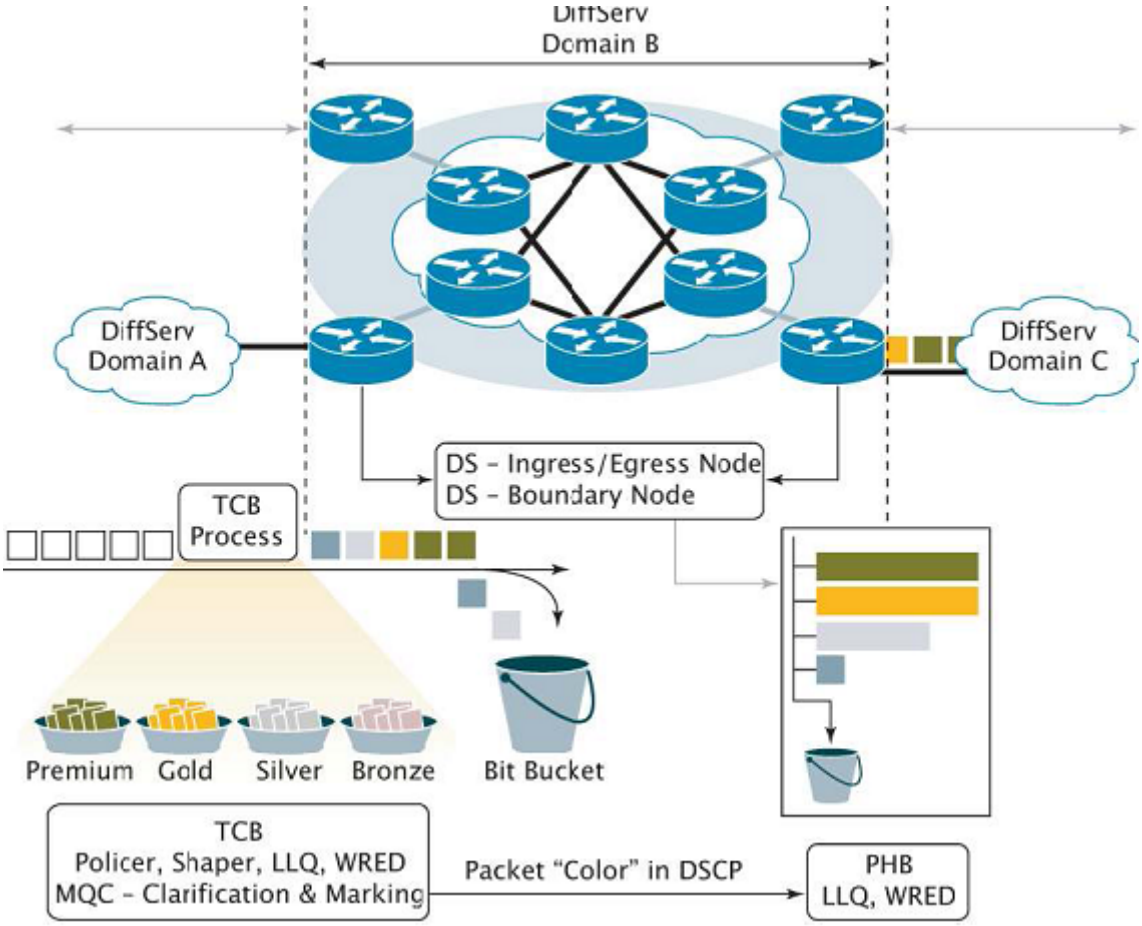


***Figure 2.4: Differentiated Service Architecture [9]***

The packets are classified by marking their DS field in the IP header. Normally this is termed as marking the Differentiated Services code point (DSCP). Packets having the same DSCP are treated as packets belonging to a particular class of traffic. Different per hop behaviors are than applied to different classes of traffic. Per-hop behaviors are defined to permit a reasonably granular means of allocating buffer and bandwidth resources at each node among competing traffic streams [9]. There have been different types of per hop behaviors defined. IETF is trying to standardize two type of per hop behaviors.

I. Assured Forwarding (AF) PHB

II. Expedited Forwarding (EF) PHB

#### 2.2.2.1 Assured Forwarding (AF) PHB

Assured Forwarding (AF) PHB group is a means for a provider DS domain to offer different levels of forwarding assurances to IP packets received from a customer DS domain [10]. Assured Forwarding (AF) PHB group provides forwarding of IP packets in N independent AF classes. Within each AF class, an IP packet is assigned one of M different levels of drop precedence. An IP packet that belongs to an AF class I and has drop precedence j is marked with the AF code point, where 1 <= I <= N and 1 <= j <= M [10]. Currently, four classes (N=4) with three levels of drop precedence in each class (M=3) are defined for general use. A Diffserv node implementing AF PHB must implement all four AF general classes. Furthermore, that node must allocate configurable and minimum amount of forwarding resources to each implemented AF class [10]. It has to be noted that in case of congestion the packets of a class having high drop probability would be dropped in favor of packets of that class having low drop probability.

#### 2.2.2.2 Expedited Forwarding (EF) PHB

The intent of the EF PHB is to provide a building block for low loss, low delay, and low jitter services [11]. Usually there are two types of delays that occur in the network, Propagation delays on wide area links and queuing delays on switches and routers [11]. Propagation delays are a fixed property of the topology and hence have to be lived with however queuing delays can be minimized. In order to minimize the queuing delays it has to be made sure that the arrival rate of EF marked packets to an interface is less than their service rate at that interface regardless to the load of other packets at that interface. Hence EF defines a PHB in which it is guaranteed that EF packets will receive service at or above a configured rate. This also minimizes jitter and can limit the maximum delay experienced by a packet to a desired level.

#### 2.2.2.3 Traffic Conditioning

DS boundary node performs traffic conditioning. A traffic conditioner typically classifies the incoming packets into pre-defined aggregates, meters them to determine compliance to traffic parameters (and determines if the packet is in profile, or out of profile), marks them appropriately by writing/re-writing the DSCP, and shapes (buffers to achieve a target flow rate) or drops the packet in case of congestion. Figure 2.6 illustrates the typical traffic conditioner at the edge of a DS-domain. A DS Internal node enforces the appropriate PHB by employing policing or shaping techniques, and sometimes re-marking out of profile packets, depending on the policy or the SLA.

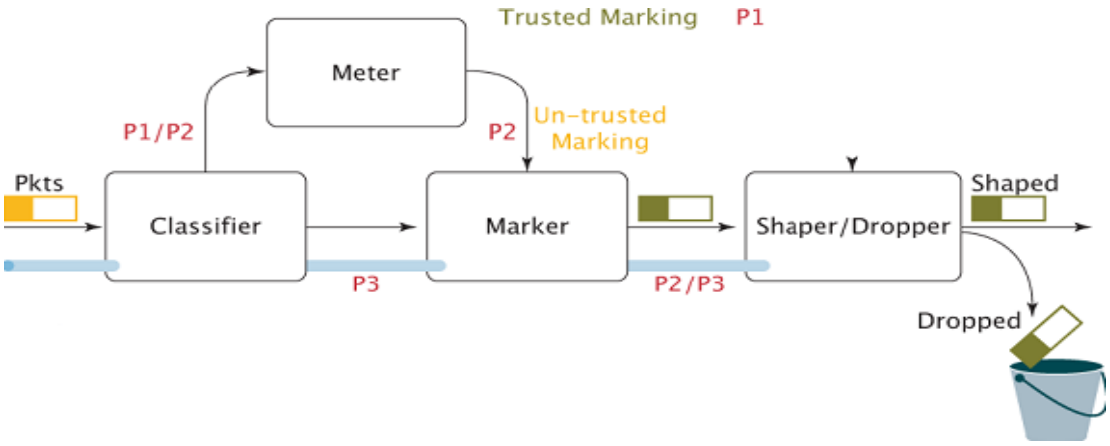


***Figure 2.5 Traffic Conditioning Block (TCB) [9]***

### 2.2.3 Multi Protocol Label Switching (MPLS)

Multiprotocol label switching (MPLS) is a versatile solution to address the problems faced by present-day networks—speed, scalability, quality-of-service (QoS) management, and traffic engineering. MPLS has emerged as an elegant solution to meet the bandwidth-management and service requirements for next-generation Internet protocol (IP)–based backbone networks.

MPLS is an Internet Engineering Task Force (IETF)–specified framework that provides for the efficient designation, routing, forwarding, and switching of traffic flows through the network. MPLS performs the following functions:

- specifies mechanisms to manage traffic flows of various granularities, such as flows between different hardware, machines, or even flows between different applications
- remains independent of the Layer-2 and Layer-3 protocols
- provides a means to map IP addresses to simple, fixed-length labels used by different packet-forwarding and packet-switching technologies
- interfaces to existing routing protocols such as resource reservation protocol (RSVP) and open shortest path first (OSPF)
- supports the IP, ATM, and frame-relay Layer-2 protocols

In MPLS, data transmission occurs on label-switched paths (LSPs). LSPs are a sequence of labels at each and every node along the path from the source to the destination. LSPs are established either prior to data transmission (control-driven) or upon detection of a certain flow of data (data-driven). The labels, which are underlying protocol-specific identifiers, are distributed using label distribution protocol (LDP) or RSVP or piggybacked on routing protocols like border gateway protocol (BGP) and OSPF. Each data packet encapsulates and carries the labels during their journey from source to destination.

#### 2.2.3.1 LSRs and LERS

The devices that participate in the MPLS protocol mechanisms can be classified into label edge routers (LERs) and label switching routers (LSRs). An LSR is a high-speed router device in the core of an MPLS network that participates in the establishment of LSPs using the appropriate label signaling protocol and high-speed switching of the data traffic based on the established paths.

An LER is a device that operates at the edge of the access network and MPLS network. LERs support multiple ports connected to dissimilar networks (such as frame relay, ATM, and Ethernet) and forwards this traffic on to the MPLS network after establishing LSPs, using the label signaling protocol at the ingress and distributing the traffic back to the access networks at the egress. The LER plays a very important role in the assignment and removal of labels, as traffic enters or exits an MPLS network.

#### 2.2.3.1 Labels and Label Bindings

A label, in its simplest form, identifies the path a packet should traverse. A label is carried or encapsulated in a Layer-2 header along with the packet. 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. The label values are of local significance only, meaning that they pertain only to hops between LSRs.

Once a packet has been classified as a new or existing FEC, a label is assigned to the packet. The label values are derived from the underlying data link layer. For data link layers (such as frame relay or ATM), Layer-2 identifiers, such as data link connection identifiers (DLCIs) in the case of frame-relay networks or virtual path identifiers (VPIs)/virtual channel identifiers (VCIs) in case of ATM networks, can be used directly as labels. The packets are then forwarded based on their label value.

Labels are bound to an FEC as a result of some event or policy that indicates a need for such binding. These events can be either data-driven bindings or control-driven bindings. The latter is preferable because of its advanced scaling properties that can be used in MPLS.

The generic label format is illustrated in *Figure 2.6*. The label can be embedded in the header of the data link layer (the ATM VCI/VPI shown in *Figure 2.7* and the frame-relay DLCI shown in *Figure 2.8*) or in the shim (between the Layer-2 data-link header and Layer-3 network layer header, as shown in *Figure 4*).

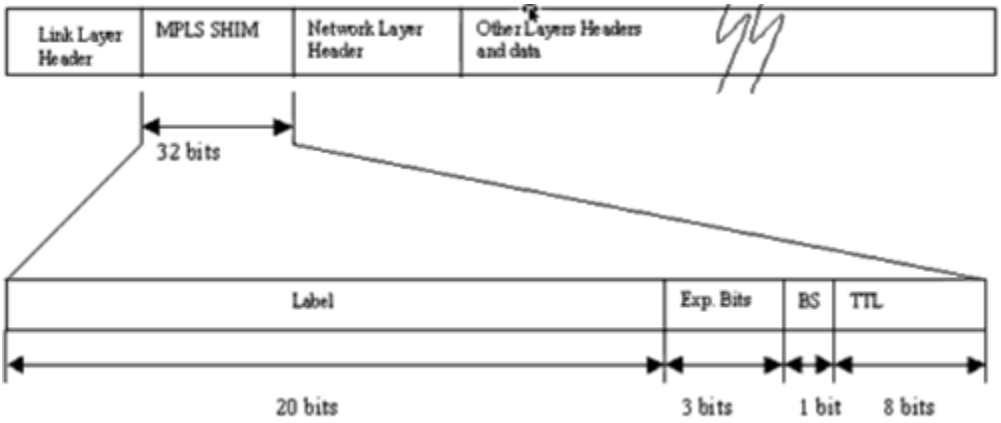


**Figure 2.6. MPLS Generic Label Format [4]**

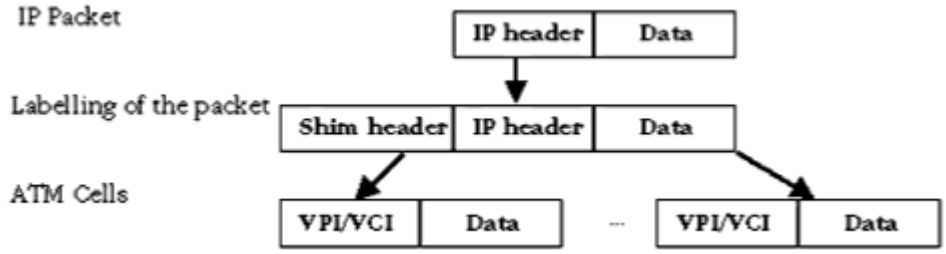


***Figure 2.7. ATM as the Data Link Layer [4]***

### 2.2.3.3 MPLS Operation

The following steps must be taken for a data packet to travel through an MPLS domain.

1. label creation and distribution
2. table creation at each router
3. label-switched path creation
4. label insertion/table lookup
5. packet forwarding

The source sends its data to the destination. In an MPLS domain, not all of the source traffic is necessarily transported through the same path. Depending on the traffic characteristics, different LSPs could be created for packets with different CoS requirements.

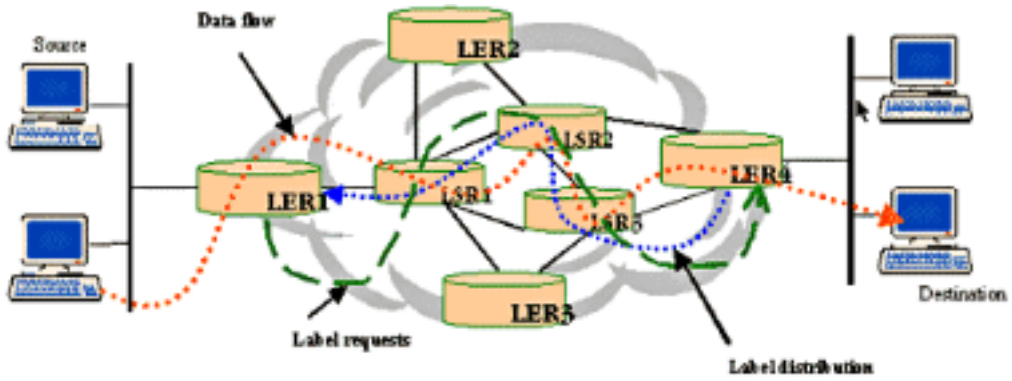


*Figure 2.8 LSP Creation and Packet Forwarding through an MPLS Domain [4]*

### 2.3 Diffserv aware MPLS domain

Diffserv and MPLS working together resolve the IP quality problem. Diffserv uses the IP TOS (type of service) field to classify traffic into different classes at the boundary node to provide QoS. MPLS also classifies traffic into different FECs with which it can provide QoS. Table 2.1 shows the similarities between Diffserv and MPLS. MPLS networks support Diffserv by mapping Diffserv Behavior aggregates onto LSPs. The DSCP of a packet determines the behavior of the nodes and MPLS label of a packet determines the route of the packet. MPLS Diffserv network combines these to features best match traffic engineering and QoS.

When a Diffserv packet arrives into a MPLS network, ingress LSR examines the TOS field of IP datagram to check the Diffserv information (DSCP) [6]. The incoming traffic is mapped to appropriate LSP.

| Sr # | Similarities between Diffserv & MPLS |
|---|---|
| 1. | Complexity is pushed to edge routers. |
| 2. | Classification of traffic at edge routers |
| 3. | Labeling of packets after classifying them |
| 4. | Transit routers treat packets according to the labels |
| 5. | Labels are short and of fixed length |
| 6. | Aggregation support |

**Table 2.1 Similarities and differences of MPLS & Diffserv**

MPLS can map Diffserv traffic to MPLS traffic in several ways. Multiple BAs can be mapped to single LSP or a single BA is mapped to single LSP. When multiple BAs are

mapped to a single LSP, Exp field in MPLS is used to specify PHB. This method is called EXP-Inferred-PSC LSP (E-LSP). When a single BA is mapped to a single LSP, it is Label-Only-Inferred-PSC LSP (LLSP).

- ✓ **E-LSP:** EXP field of MPLS header (3 bits) is used to specify BAs. Label can be used to make a forwarding decision and EXP field can be used to determine how to treat the packet.
- ✓ **L-LSP:** A separate LSP can be established for a single FEC BA combination. In this case, the LSR can infer the path as well as treatment of the packet from the label of the packet. The EXP field encodes the drop precedence of the packets.

Though E-LSP is very useful in a network with limited number of traffic classifications (less than or equal to 8), along with increasing number of traffic classification, E-LSP is not going to serve the purpose. L-LSP is the answer for MPLS Diffserv with many types of PHBs defined. Using different trade-off and combinations of techniques can solve the scalability problem of L-LSP. [12].

The problem with E-LSP is that availability of only three bits allows representation of 8 BAs for a given FEC. This is not useful when more than 8 BAs are defined. Though L-LSP supports arbitrarily large number of PHBs, the problem is scalability. In a network with different LSPs for the different BAs increases the number of labels a LSR has to maintain. With increasing number of PHBs, maintaining that amount of labels can become a problem. [12].

# *Chapter 3*

## 3.1 Introduction

Traffic Engineering is the process where data is routed through the network according to a management view of the availability of resources and the current and expected traffic. The class of service and quality of service required for the data can also be factored into this process.

Traffic Engineering may be under the control of manual operators. They monitor the state of the network and route the traffic or provision additional resources to compensate for problems as they arise. Alternatively, Traffic Engineering may be driven by automated processes reacting to information feedback through routing protocols or other means.

## 3.2 MPLS Traffic Engineering

MPLS traffic engineering provides an integrated approach to traffic engineering. With MPLS, traffic engineering capabilities are integrated into Layer 3, which optimizes the routing of IP traffic, given the constraints imposed by backbone capacity and topology. MPLS traffic engineering employs "constraint-based routing," in which the path for a traffic flow is the shortest path that meets the resource requirements (constraints) of the traffic flow. In MPLS traffic engineering, the flow has bandwidth requirements, media requirements, a priority versus other flows, and so on. It automatically establishes and maintains the tunnel across the backbone, using RSVP. The path used by a given tunnel at any point in time is determined based on the tunnel resource requirements and network resources, such as bandwidth.

Available resources are flooded via extensions to a link-state based Interior Protocol Gateway (IPG). Tunnel paths are calculated at the tunnel head based on a fit between required and available resources (constraint-based routing). The IGP automatically routes the traffic into these tunnels. Typically, a packet crossing the MPLS traffic engineering backbone travels on a single tunnel that connects the ingress point to the egress point.

MPLS traffic engineering is built on the following IOS mechanisms:

- Label-switched path (LSP) tunnels, which are signaled through RSVP, with traffic engineering extensions. LSP tunnels are represented as IOS tunnel interfaces, have a configured destination, and are unidirectional.
- A link-state IGP (such as IS-IS) with extensions for the global flooding of resource information, and extensions for the automatic routing of traffic onto LSP tunnels as appropriate.
- An MPLS traffic engineering path calculation module that determines paths to use for LSP tunnels.
- An MPLS traffic engineering link management module that does link admission and bookkeeping of the resource information to be flooded.
- Label switching forwarding, which provides routers with a Layer 2-like ability to direct traffic across multiple hops as directed by the resource-based routing algorithm.

As mentioned above, the goal of traffic engineering is to find a path in the network that meets a series of constraints. Thus, these constraints need to be taken into account in calculating path to the destinations. Some of these constraints are the bandwidth requested for a particular LSP (for example, 10Mbs from source x to destination y), 2) the administrative attributes (“colors”) of the links that the traffic is allowed to cross (for example, no low-latency links, where low-latency links are marked with a particular administrative attribute), 3) the number of hops that the traffic is allowed to pass , 4) the priority of this LSP when compared to other LSPs (for example, one out of eight possible priority levels). Other constraints are also possible. Calculating a path that satisfies these constraints requires that the information about whether the constraints can be met is available for each link, and this information be distributed to all the nodes that perform path calculation. This means that the relevant Once this information is available, a modified version of the shortest-path-first (SPF) algorithm, called constrained SPF (CSPF), can be used by the ingress node to calculate a path that complies with the given constraints.

Conceptually, CSPF operates in the same way as SPF, except it first prunes from the topology all links that do not satisfy the constraints. For example, if the constraint is

bandwidth, CSPF prunes from the topology links that don't have enough bandwidth. Figure 6 shows a network topology and two LSPs with bandwidth requirements. Once the LSP A-C is set up, no resources for the LSP B-C are available along the shortest path, the links on the shortest path are pruned from the topology and CSPF picks the alternate path as the best available.

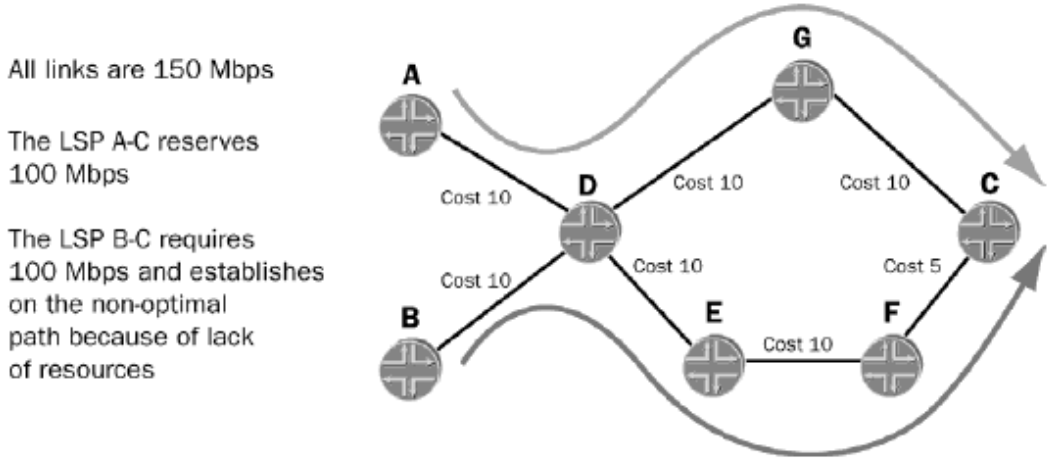


***Figure 3.1 : LSP can take the longer path when shortest not available [3]***

Finally, after a path has been successfully calculated, MPLS forwarding state is established. As the path is set up, the available resources are updated at each node and the other nodes are informed of the changes through the IGP.

The only limitation with MPLS Traffic Engineering is one could not differentiate **customer traffic based on Class-of-Service.** However the SLA compliance issue can be sort out by introducing class types in MPLS network, which can also be termed as DS-Traffic Engineering. DS-TE can happen in MPLS aware Diffserv domain which not only increasing the scalability but also cover the limitation of Class of service.

### 3.3 Traffic Engineering for MPLS aware Diffserv Domain

Differentiated Services (DiffServ) enables scalability in network designs with various classes of service. MPLS traffic engineering (TE) facilitate resource reservation, fault-

tolerance, and optimization of transmission resources. MPLS DiffServ-TE combines the advantages of both DiffServ and TE. The result is the ability to give strict Quality of Service (QoS) guarantees while optimizing use of network resources. The QoS delivered by MPLS DiffServ-TE allows network operators to provide services that require strict performance guarantees such as voice and to consolidate IP and ATM/FR networks into a common core.

DiffServ-TE supports a maximum of eight TE-classes, TE0 through TE7, which can be selected from the 64 possible CT-priority combinations via configuration. At one extreme, there is a single CT with eight priority levels, very much like the existing TE implementation. At the other extreme, there are eight distinct CTs, with a single priority level. Figure 3.2 shows the 64 combinations of class type and priority, and a choice of eight TE-classes.

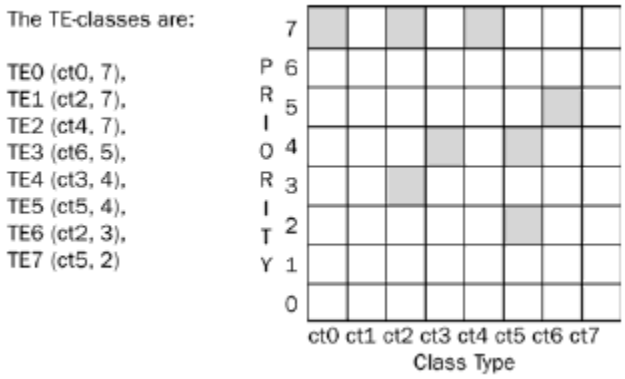


***Figure 3.2 Multiple 64 Class types with different priorities [3]***

After the path is calculated, it is signaled and admission control and bandwidth accounting are performed at each hop. The CT information for an LSP is carried in the new Classtype object (CT object) in the RSVP path message, and specifies the CT from which the bandwidth reservation is requested. Two rules ensure that it is possible to deploy DiffServ-TE incrementally in the network: 1) the CT object is present only for

LSPs from CT1 through CT7 (if the CT object is missing, CT0 is assumed), and 2) a node that receives a path message with the CT object and does not recognize it rejects the path establishment. These two rules ensure that establishment of LSPs with per-CT reservation is possible only through DiffServ-TE. DiffServ provides the correct scheduling behavior for each type of traffic. The combination of DiffServ and per-CT traffic engineering ensures strict service guarantees.

Several Constraint routing algorithms are introduced and has been worked on by in the and different algorithms have been worked out by IETF and many universities. Some the work not only focused on the engineering the traffic on MPLS-Diffserv domain but also the major object was the automation of the engineering process. The work is based on the below mention Algorithms.

### 3.3.1 Traffic Engineering with Link coloring (TELIC)

In MPLS aware Diffserv domain TELIC objective is to establish LSPs for incoming traffic trunks with service awareness [2]. The constraint based routing is used to establish label switching paths. Each LSP request specifies the amount of bandwidth and FEC. TELIC seek the widest path for establishing LSP on the domain based on the link colours. Silver, green, white, yellow and red are the link colours which are being updated on the basis of bandwidth utilization. Initial states of all the links are the silver and green out of which silver is taken as low-delay links that are most suited for premium traffic. Silver links are converted to white and green are converted in yellow and finally red. Based on the information in the requested traffic trunk algorithm tries to locate LSP that best meets the using subgraph in the domain [1]. On determining LSP it is registered in Master LSP table and the link colour is updated which is also marked in master LSP table for having convenience and fast response to next request. The algorithm used by TELIC is shown below in the Table 3.1.

| **TRAFFIC ENGINEERING WITH LINK COLORING (TELIC)** |
|---|
| **Input:** |
| A graph consisting of N nodes and M links, with each link specified as with available bandwidth B, delay D, reliability R and a color C |
| **Output:** |
| An LSP between the designated ingress router and the egress router satisfying the minimum cost criteria and meeting the FEC criteria with the color of the most congested link in the new LSP |
| **Algorithm Steps: (TELIC reads the domain topology from a file before starting the listed steps)** |
| (1) Read the next request |
| (2) Determine the FEC of the request |
| (3) If it is EF, route the LSP with a subgraph that includes silver, white and green links in that order |
| (4) If it is AF, route the LSP with a subgraph that includes green, yellow white in that order |
| (5) If it is DF, route the LSP with a subgraph that includes red, yellow and green links in that order |
| (6) Output the LSP, store it in the LSP table in the ingress router and reduce the available bandwidth in each link of the new LSP by subtracting the allocated bandwidth from Bj for each j |
| (7) Update the colors of the links included in the new LSP as per the color table in the ingress node |

***Table 3.1 : Algorithm of TELIC [1]***

When TELIC is implemented in a Diffserv-aware MPLS domain, it can achieve the following targets:

- Balance the load across the domain
- The congested or heavily utilized links are avoided
- In Diffserv coded flows, the EF traffic is allocated LSP's that avoid links being used by other traffic
- Every flow gets its fair share of the network resources.
- If the network is heavily overloaded, best effort traffic will pass through most congested links.

Figure 3.3 shows an MPLS domain having single source destination pair ISP topology with 9 intermediate routers, two edge routers and 16 links that can be used between the ingress and the egress node.

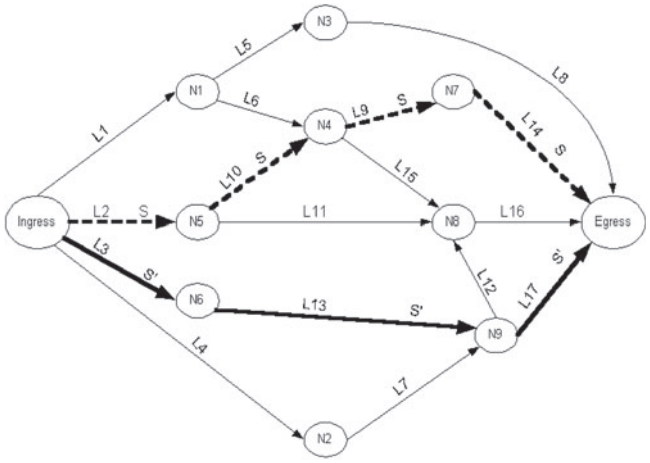


***FIGURE 3.3. MPLS domain with various links and nodes [1]***

Initially, two paths are marked silver for QOS sensitive traffic (S and S') and the remaining links are all marked green. The assured service FEC uses a subgraph consisting of green, yellow, white and silver links and the best-effort FEC uses a subgraph consisting of red, yellow and green links. Within the subgraph, TELIC looks for a minimum-cost path, where the cost of the *i*th path is defined as:

$$C_i = \sum^{J=1-n} (1/B_J + D_J) * R_J$$

Where *Ci* = cost of the *i*th path that has *Ni* hops, *BJ* = available bandwidth on link *J*, *DJ* is the delay for link *J*, and *RJ* = reliability for link *J* (ISP-controlled and link medium-dependent factor with smaller values for silver and white links)[2].

The shortest-distance algorithm does not consider the *D* and *R* factors for computing an LSP. It takes into account the residual bandwidth available on each link and it considers the whole domain for allocating a requested LSP tunnel [2]. We may express the total distance of a *k*-hop path *p* as the sum of this inverse value:

$$Dist_{(P)} = \sum^{J=1-k} (1/n_J)$$

Where dist(*p*) = total distance for the *k*-hop path *p*, and *nj* = residual bandwidth of the link *j*. TELIC distribute the domain into subgraphs for avoiding sharing of links between EF and best-effort traffic. Therefore, TELIC works on routing DF traffic on the links which either not carry EF traffic or the link is not conjusted. The congestion on the link is determined by the color of the link. The link goes toward the red color the more congestion it is. Thus link coloring mechanism not only is used to avoid congestion but also take care of conjunction on the link.

#### 3.3.1.1 TELIC Results

The performance of the TELIC has been analyzed for different traffic sets on different domain. The traffic set was considered static and the domain is defined. The results are drafted through configuring the algorithm on the network simulator and it is found that the allocation mechanism is much better than for SHORTD algorithm. the conjunction factor results for all the three domains when different traffic sets are processed. The traffic set 1 has EF service request for 20% of the overall bandwidth demand and the remaining bandwidth requests contain only 4% AF and 96% DF requests. The ratio of AF requests is increased linearly until the last traffic set, in which about 92% of the non-EF demand is in the AF.

| S. No. | Domain | TELIC | SHORTD |
|---|---|---|---|
| 1 | ISP | 0 | 306 |
| 2 | MP | 0 | 104 |
| 3 | MIR | 4 | 34 |

***Table 3.2 Conjunction Factor Values[2]***

| S. no. | Domain | TELIC | SHORTD |
|---|---|---|---|
| 1 | ISP | DF = 80 | DF = 10 |
| 2 | MP | DF = 80 | 0 |
| 3 | MIR | EF = 20<br>DF = 250 | EF = 0<br>DF = 230 |

***Table 3.3 Rejected bandwidth values [2]***

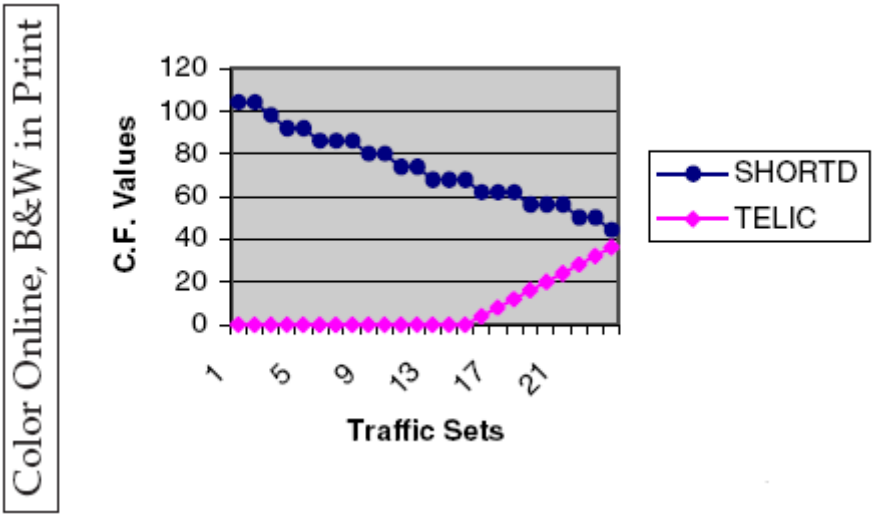


*Figure 3.4 Conjunction factor for MP Domain [2]*

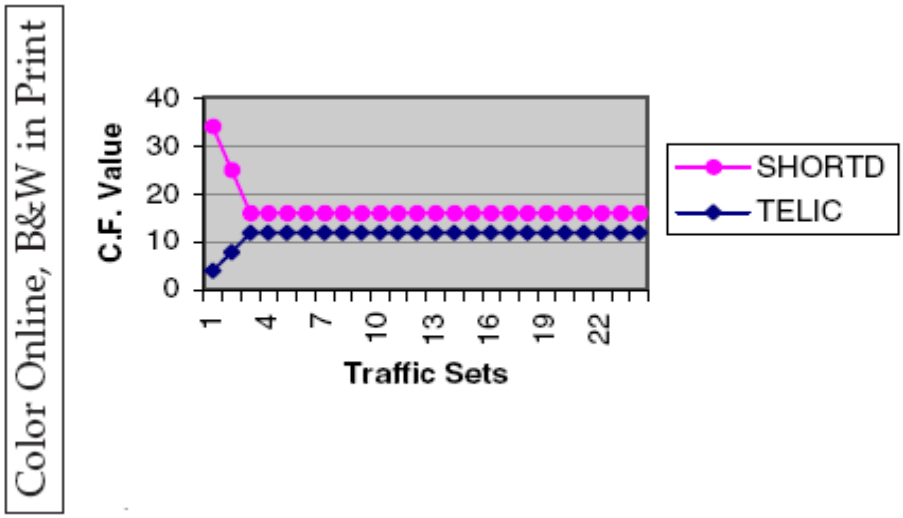


*Figure 3.5 Conjunction factor for MIR Domain [2]*

It is obvious from the results that TELIC minimizes the conjunction factor whereas SHORTD routing results in large values for the conjunction factor. It may be mentioned here that the conjunction factor has been computed from the premium traffic perspective. In other words, this value shows the degree of merging of other traffic with the premium

traffic on the links within the domain under observation. Thus it is a very important parameter for the ISP who would like to charge the customer for the premium service and also provide the expected premium service to the customer. From the graphs of CF values, it can be seen that CF for SHORTD is too high for the sets in which best-effort traffic dominates the non- EF portion. On the other hand, TELIC remains successful in minimizing CF and thus outperforms SHORTD by a large margin. As the ratio of AF service class is increased, TELIC and SHORTD produce closer results. This is due to the fact that the AF class traffic is regulated and thus it can share links with the EF traffic without jeopardizing the EF performance. From the table of rejected bandwidth, it is seen that mostly the best-effort traffic requests get rejected. However, it is one aspect in which TELIC's performance can be improved. In the next section, we describe the DFTS algorithm developed for increasing the allocation ratio for DF traffic. This in itself can work as a load-balancing algorithm in a domain.

### 3.3.2 DFTS

This section presents a new load-balancing algorithm, named Default Forwarding Trunk Splitting (DFTS), which splits the first and subsequent rejected DF trunks across the domain into several smaller trunks and routes them on separate preferred links. Classical max–min rate-based allocation strategy has been regarded as a fair and efficient technique for flows that have certain minimum rate requirements and peak rate limitations. DFTS does not use max–min strategy because of the fact that that the best-effort traffic does not have a minimum rate requirement. In addition, the rate-based allocation does not consider the class of service. DFTS is developed and integrated with TELIC and SHORTD for increasing the DF allocation of these algorithms. There are some features that differentiate DFTS from the existing works:

- DFTS splits the rejected DF request based on the paths available in the domain, minimum link capacity in each path and the classification of each path as QoS, non-QoS and mixed.
- It tries to improve the DF allocation without increasing the conjunction factor by a large amount.

- It can handle the request partially instead of discarding the whole request.
- It brings the rejected paths (with insufficient link capacity) under consideration in a certain order.

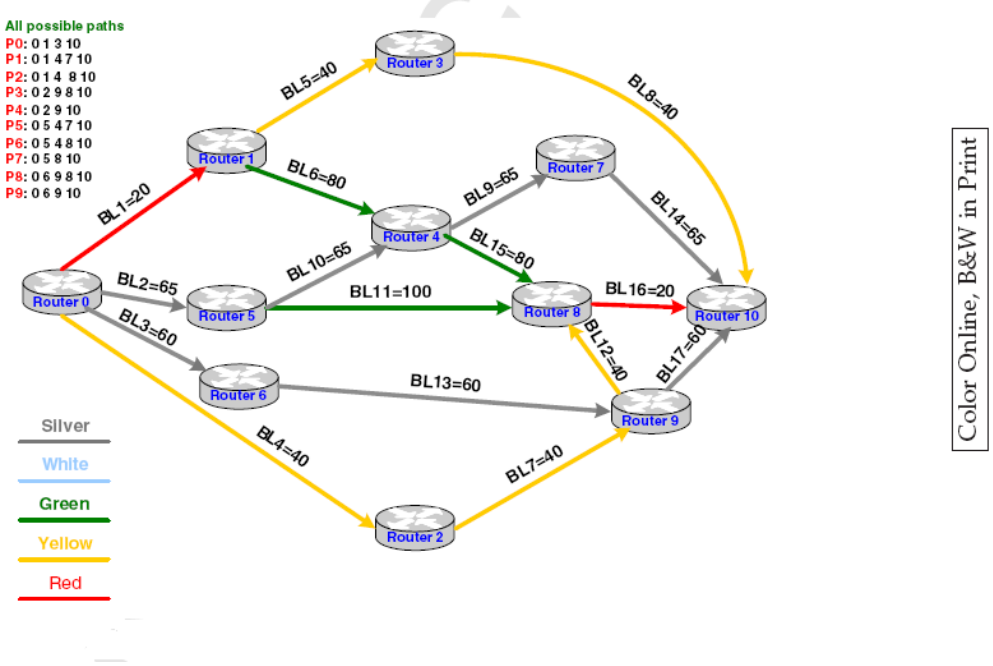


***Figure 3.6 An Example ISP domain [2]***

We consider a simple network in order to introduce the DFTS algorithm. This network has three disjoint paths 1, 2 and 3 from ingress to egress (we assume that the network topology is irrelevant). Their available bandwidths are 10, 30 and 20 Mbps, respectively. If a traffic trunk arrives with 35Mbps, none of these paths can handle this request. DFTS can solve this problem by splitting up the request into two separate LSPs: 10Mbps and 25Mbps. The first and second split trunks utilize paths 1 and 2, respectively. Now, assume that another traffic trunk of 40 Mbps arrives. In this case, again none of the above paths can handle this request, for two reasons. Firstly, path 1 is totally utilized. Secondly, the request exceeds the capacity of both paths 2 and 3. However, DFTS offers another solution. Instead of discarding the whole request, the algorithm partially handles this request. In other words, it satisfies the request partially with the available bandwidth.

DFTS splits up the request into three separate LSPs: 5Mbps, 20Mbps and 10Mbps, respectively. First and second split trunks utilize paths 2 and 3 and the third one is discarded.

### 3.3.1.1 DFTS Results

TELIC and SHORTD's performance with and without the DFTS algorithm is measured by using it on traffic requests that arrive at networks built with ISP topology as seen in Figure 12. Figures 13 and 14 illustrate a comparison between TELIC and SHORTD in terms of DF allocated bandwidth and total enhanced bandwidth with ISP domain.

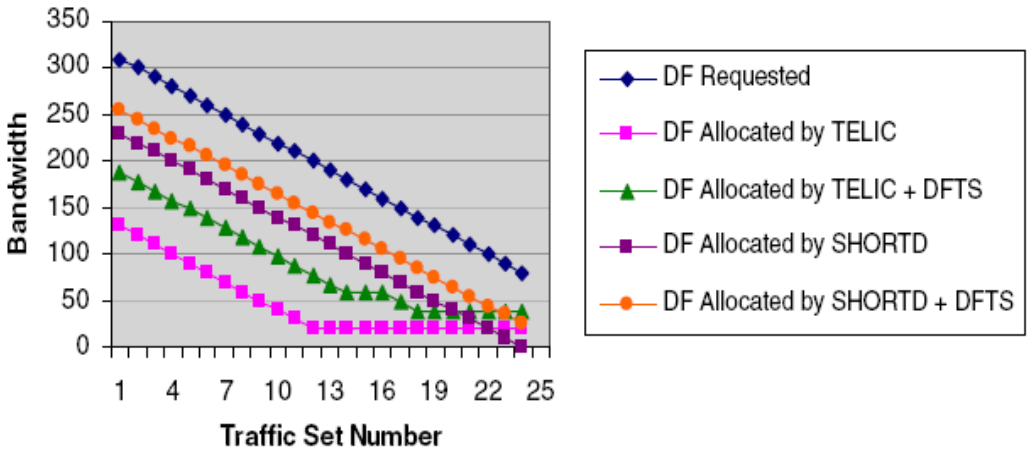


*Figure 3.7 Allocated BW with different algorithms in ISP domain [2]*


*Figure 3.8 Total enhanced BW with different algorithms in ISP domain [2]*

As seen in Figure 13, TELIC maintains a constant gap between request and allocation. This gap is due to the fact that some links become congested quickly and there is no possible path in the domain under consideration to the egress without those links. TELIC with DFTS manages to allocate 32% of the DF requests that were earlier rejected in the initial part of the curve. Later, the DF allocated bandwidth decreases because of the increase in AF requests. SHORTD sustains a steady gap between request and allocation and this gap is reduced with the application of DFTS by 25 Mbps. Thus, DFTS increases the allocation ratio for DF requests while controlling the CF value through path classification.

### 3.4 Bandwidth Constraint Model

One of the most important aspects of the available bandwidth calculation is the allocation of bandwidth among the different CTs. The percentage of the link's bandwidth that a CT (or a group of CTs) may take up is called a bandwidth constraint (BC). RFC 3564 defines the term "bandwidth constraint model" to denote the relationship between CTs and BCs.

#### 3.4.1 Maximum Allocation Model

The most intuitive bandwidth constraint model maps one BC to one CT. This model is called the maximum allocation model (MAM) and is defined in [DSTE-MAM]. From a practical point of view, the link bandwidth is simply divided among the different CTs, as illustrated in Figure 3.9.

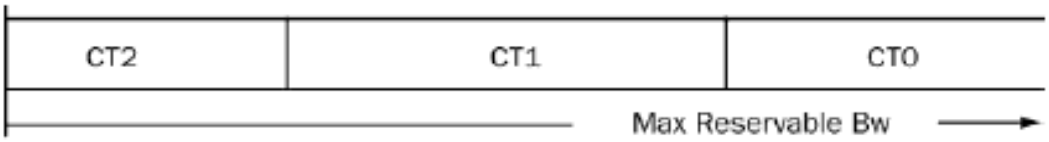


***Figure 3.9 bandwidth allocation model MAM [8]***

The problem with MAM is that because it is not possible to share unused bandwidth between CTs, bandwidth may be wasted instead of being used for carrying other CTs. Consider the network shown in Figure 3.10. In the absence of voice LSPs, bandwidth is available on all the links on the shortest path, but this bandwidth cannot be used for

setting up another data LSP. The second data LSP is forced to follow a non-optimal path, even though bandwidth is available on the shortest path. On the other hand, after both data LSPs have been set up, if a voice LSP needs to be established, bandwidth is available for it on the shortest path.

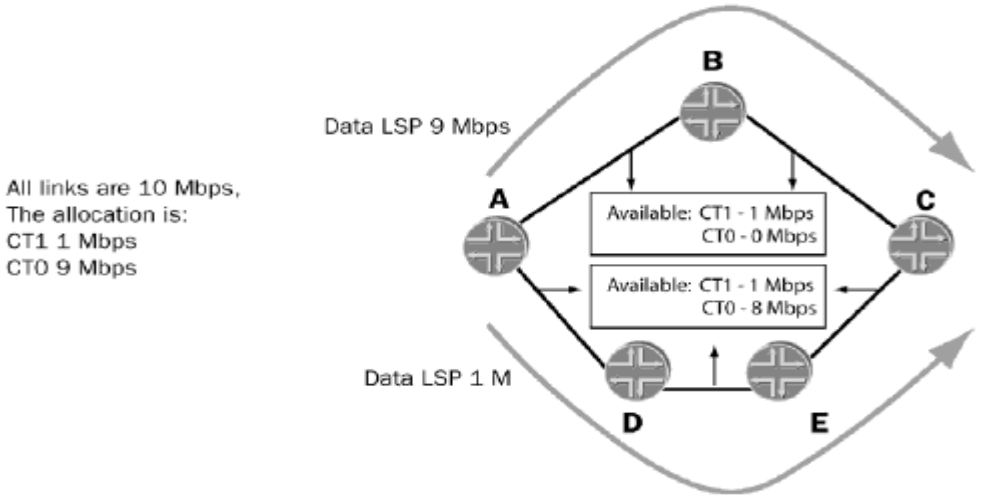


***Figure 3.10 : Even if no voice LSP in the domain, no data packet can occupy it.***[3]

The benefit of MAM is the class type isolation, thus no priorities need to be configured on the network.

### 3.4.2 Russian Doll Model (RDM)

The Russian dolls bandwidth allocation model (RDM) defined in [DSTE-RDM] improve bandwidth efficiency over the MAM model by allowing CTs to share bandwidth. In this model, CT7 is the traffic with the strictest QoS requirements and CT0 is the best-effort traffic. The degree of sharing varies between two extremes. At one end of the spectrum, BC7 is a fixed percentage of the link bandwidth that is reserved for traffic from CT7 only. At the other end of the spectrum, BC0 represents the entire link bandwidth and is shared among all CTs. Between these two extremes are various degrees of sharing: BC6 accommodates traffic from CT7 and CT6, BC5 from CT7, CT6 and CT5 and so on. This model is very much like the Russian doll toy, where one big doll (BC0) contains a smaller doll (BC1) which contains a yet smaller doll (BC2), and so on.

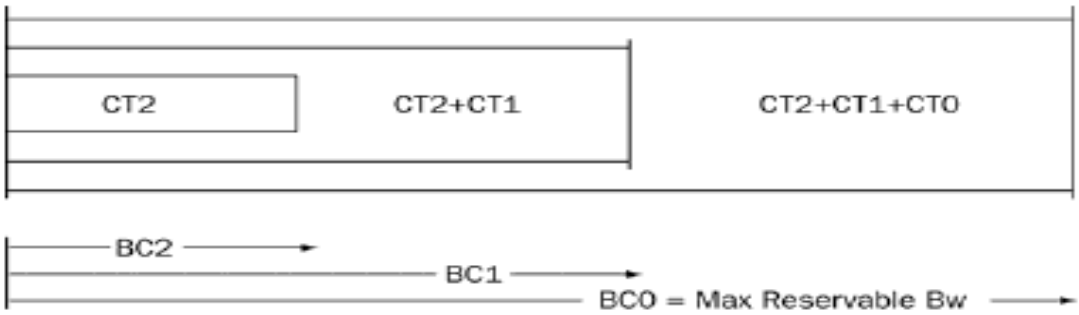


***Figure 3. 11 Bandwidth allocation model for RDM [8]***

The advantage of RDM relative to MAM is that it provides efficient bandwidth usage through sharing. Consider the network from Figure 3.12, which carries voice traffic and data traffic. The total bandwidth available on each link is 10Mbs. 1Mbs is allocated to BC1 and 10Mbs are allocated to BC0. What this means is that each link may carry between 0 and 1Mbs of voice traffic and use the rest for data. Assuming that a data LSP is already established over the path A-B-C, in the absence of voice traffic, a second data LSP can be established to take advantage of the unused bandwidth. Another useful property that is achieved through sharing is cheap over provisioning for real-time traffic. Since the extra bandwidth can be used by other types of traffic, allocating it to the real-time class does not affect the overall throughput of the network.

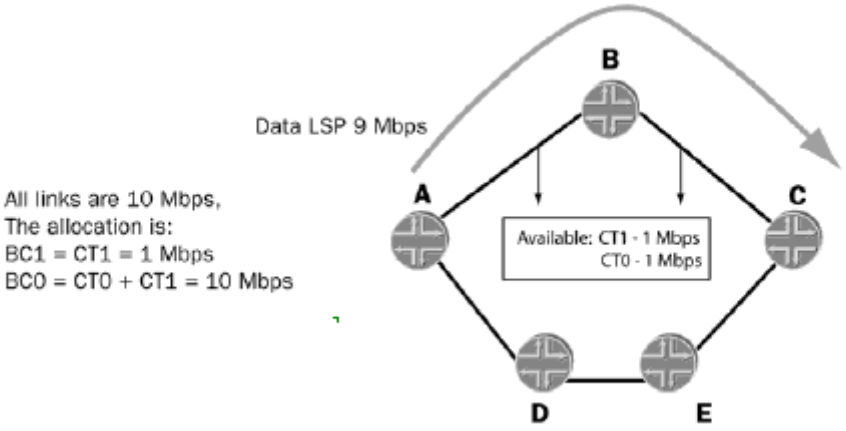


***Figure 3.12 : data packets can move on the link in absence of voice LSPs [3]***

The drawback of RDM is class type isolation; pre-emption is required to guarantee the bandwidth to the premium traffic. In RDM priorities of different class types are used for preempting the low priority classes.

# *Chapter 4*

## 4.1 Background and Introduction

This section provides a new algorithm for automating the process of traffic engineering. The algorithm is the enhancement in existing TELIC by implementing a standard Bandwidth Constraint Model RDM thus making TELIC compliant with RDM. RDM Compliant TELIC not only work on the compliance issue but also achieve the un address parts of TELIC by allocating Assured Forwarding on Silver links and introducing the mechanism for controlled pre-emption of best effort traffic.

Bandwidth allocation models are already discussed in previous chapter; briefly there are two bandwidth allocation models for MPLS aware differentiated service domain which are Maximum allocation model (MAM) and Russian Doll Model (RDM). Channels are distributed on the link for different class of traffic in MAM resulting in the under used link, however MAM bears the advantage of available bandwidth of premium classes. RDM behaves like Russian dolls i.e. a small doll inside a big doll and so on. RDM has the advantage of the use of bandwidth by lower classes on availability and pre-empted when required. RDM however have an over-head of preemption but link utilization increased as compared to MAM.

RDM TELIC is an automation of traffic engineering process which is based Russian Doll Model. Some features which differentiate RDM TELIC from the existing work are.

- ✓ Bandwidth allocation procedure is changed and now it is based on RDM
- ✓ In TELIC AF not being allocated on Silver links which is removed in the current work resultantly improves the allocation of AF on multiple sub-domains
- ✓ TELIC pre-empt DF request on arrival of any premium request even if the room for that request is unavailable. Current algorithm introduces controlled pre-emption of best-effort traffic.
- ✓ Conjunction degree of links is introduced with threshold values in order to avoid congestion on the link.

## 4.2 Algorithm Working Principles

RDM TELIC work on the principle of RDM i.e each class is given a particular bandwidth percentage which can be used by lower classes on availability. The limits are defined by network administrator on the time of setting up the domain. These values are re-configurable based on the ratio of traffic, link available, quality of the links and signed SLAs. The basis algorithm is defined in the table 4.1 below.

| **RDM TRAFFIC ENGINEERING WITH LINK COLORING (TELIC)** |
|---|
| **Input:** |
| A graph consisting of N nodes and M links, with each link specified as with available bandwidth B, delay D, reliability R and a color C |
| **Output:** |
| An LSP between the designated ingress router and the egress router satisfying the minimum cost criteria and meeting the FEC criteria with the color of the most congested link in the new LSP |
| **Algorithm Steps: (TELIC reads the domain topology from a file before starting the listed steps)** |
| (1) Read the next request |
| (2) Determine the FEC of the request |
| (3) If it is EF, check for the available limit route on the subgraphs that includes silver, white and green links in order. If the limit is available, then check if it is used by lower classes , if yes then go for preemption and route the LSP with a subgraph that includes silver, white and green links in that order other queue the request at the lower position. |
| (4) If it is AF, check for the available limit route on the subgraphs that includes green, yellow , white & silver links in order. If the limit is available, then check if it is used by lower classes, if yes then go for preemption and route the LSP with a subgraph that includes silver, white and green links in that order other queue the request at the lower position. |
| route the LSP with a subgraph that includes green, yellow white in that order |
| (5) If it is DF, route the LSP with a subgraph that includes red, yellow and green links in that order |
| (6) Output the LSP, store it in the LSP table in the ingress router and reduce the available bandwidth, increasing the CD of the link . |
| (7) Update the colors of the links included in the new LSP as per the color table in the ingress node |

**Table 4.1 :** Algorithm of RDM TELIC

Conjunction Degree of each of the link is identified on finding the sub graphs. Traffic is passed on the sub domain having low conjunction Degrees. CD is also set on empirical

values. As the bandwidth of the lowest class is not consumed by higher class so it may not be considered in calculating the conjunction degrees. The formula for the conjunction degree is updated and mentioned below.

$$CD = CD2 + CD1 \qquad \text{----------} \quad (4.1)$$

Where CD is the degree of the conjunction of the link however CD2 refer to the shared bandwidth of EF by ***AF and DF*** and CD1 is used as the Shared bandwidth of AF by ***DF***. In order to have the controlled conjunction the threshold are define again on the basis of reservable bandwidth. The formula for the thresholds can be found as under.

$$CD(CT\ Threshold) = Coef./100 \times Max\ Reservable\ BW \quad \text{----------} \quad (2)$$

Where CD(CT Threshold) is the threshold limit of each of the class type which can be calculated individually , Coef is the co-eficient based on the empirical values which can be set by domain administrator. The sum of all the threshold give the threshold of the link. Now it is easy to calculate the Conjunction Factor of the domain, which is the sum of the conjunction degree of entire link. The conjunction factor for the domain consisting of n links can be defined as

$$F_{(c)} = \sum^{i=1-n} (CDi) \qquad \text{----------} \quad (3)$$

Where $F_{(c)}$ = Conjunction factor of the domain under consideration and CD(i) refer to conjunction degree of the link. The Objective of the RDM TELIC is to make TELIC compliant with Russian Doll Model with minimum conjunction. The conjunction at certain level is also handled by link colours.

## 4.3 Implementation

The algorithm is developed in C++

### *4.3.1 Implementation Pseudo Code*

The program runs with menu as shown in below mentioned Figure 4.1.

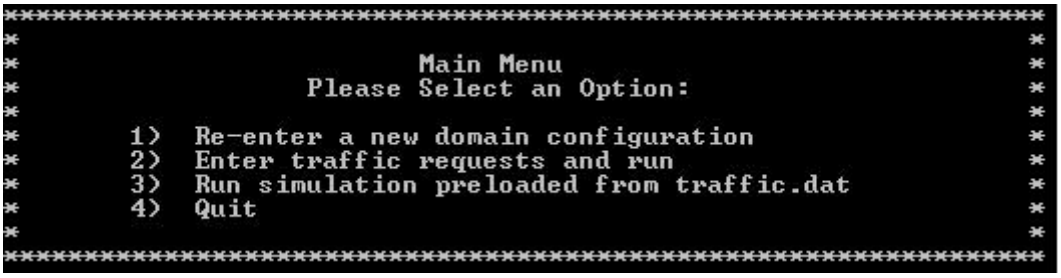


**Figure 4.1: Main Menu of the Program**

***Option 1***

Option 1 selection on the main menu starts reading the domain configuration with number of nodes and links in the domain. We have to mention the node to node link by specifying the color which can either be silver or green as mentioned in below mentioned figure.

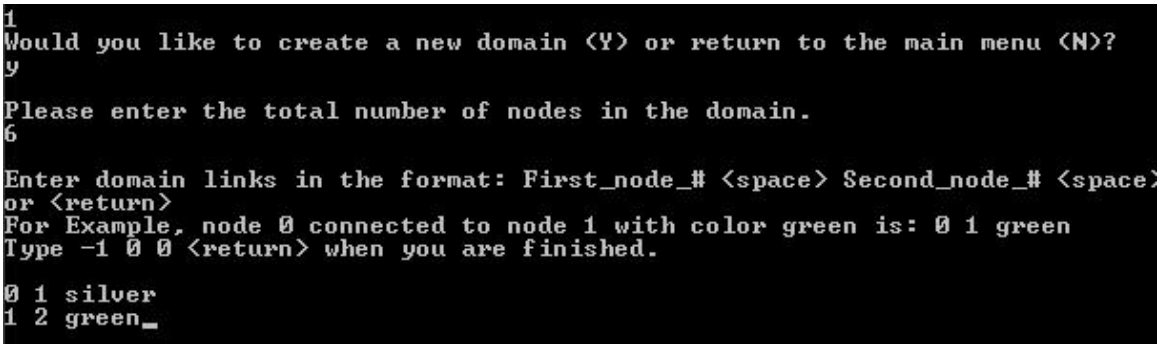


**Figure 4.2: Selecting Option 1 for creating new domain**

The domain is created using alterdomain method and the created domain is saved in domain.dat which is placed in the working directory of the program.

***Option 2***

Option 2 selection on the main menu starts reading the traffic request on the ingress node and places it in traffic.dat. The file traffic.dat is on the root directory of the program. The Figure 4.3

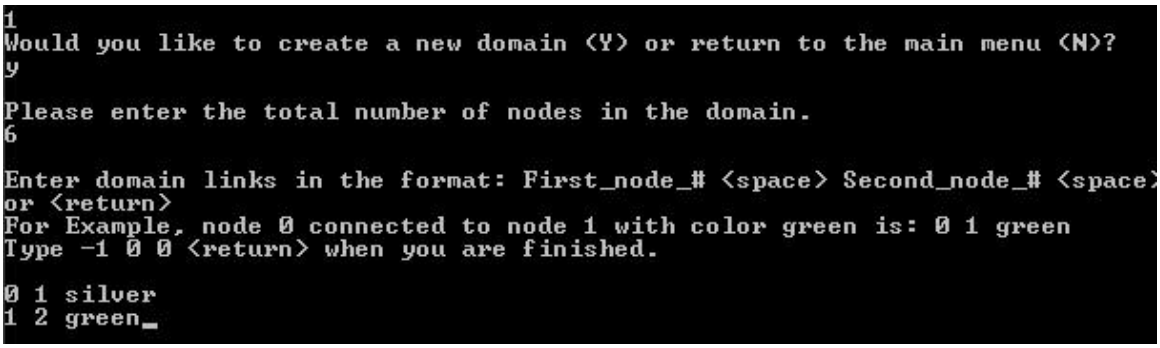


**Figure 4.3: Option 2 Selection for reading the traffic request of the file**

The method used to read the traffic request is **readtrafficrequest** which is in the main method of network.cpp.

***Option 3***

Option 3 selections on the main menu starts reading the domain configuration from domain.dat created through option 1 and traffic request from traffic.dat. The ultimate simulation is represented through this menu selection as show below.

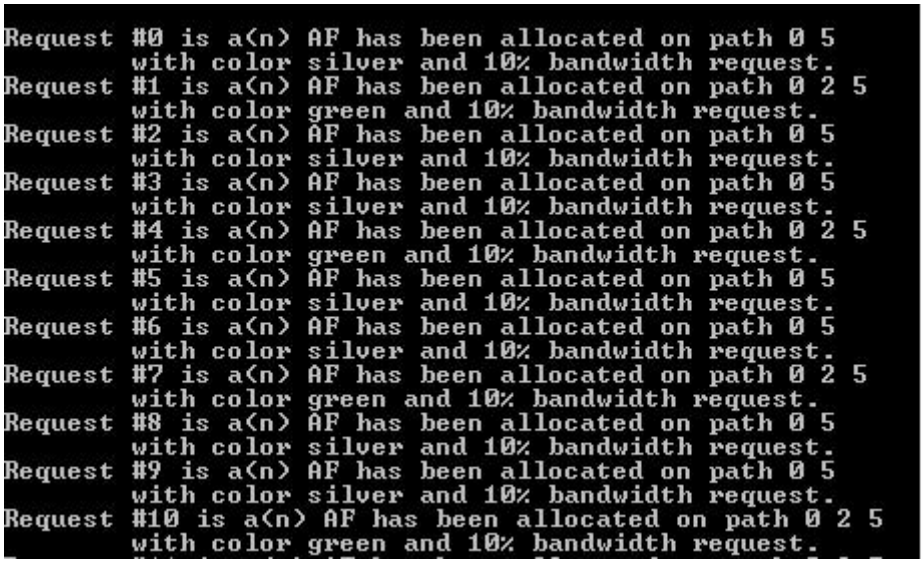


**Figure 4.4: Option 3 Selection for simulating the LSP creation.**

The created LSPs are shown with their subgraph. The method used to create LSP is ProcessLSPrequest. The step by step flow of the program is discussed below.

1. **reloadDomain** …. Reset the domain object to original configuration…. Mean reads the domain.dat and allocate resources to request i.e.

a. reading the domain.dat
b. getting total number of nodes
c. creating links between nodes as directed containing source -> destination, color of the link, cost associated (default 1) , weight(bandwidth associated ( for silver and green its is 100 and for other it is based on LIMITS like Silver_LMT,YELLOW_LIMIT,GREEN_LIMIT. ( these limits are threshold that on acquiring bandwidth they can move to next level until to red.

2. **processLSPQueue(domain obj)….** Allocate the shortest path … the flow is
   **a.** The copy of the domain object is created so that the original path will not get disturbed.
   **b.** Make an array representing the LSP Table depending on the traffic request.
   **c.** Store the traffic.dat in the struct name queue (lspclass, bandwidth) for each traffic request until the MAX_TRAFFIC allowed.
   **d.** Set the requested traffic queue as mentioned above and set all the remaining with '0'
   **e.** Setting the priority Index of the queue according to the class requested and then inserts it into the LSP table after creating the sub graphs and finding the shortest paths**.** The EF sub graphs found on the basis of the color of the link, first of all silver then silver and white if not found then green if not found then removing DF traffic and even if its not found then the not satisfaction is displayed.
   f. For the AF sub graph are also found on the priority starting from green sub graph
3. If the paths are created successfully and LSP has also been created then the status of the links/domain can be printed on request…
4. On another request, the domain object is again reloaded to get in to the original shape.
5. In this way the colors of the links are changed are changed and TELIC has been implemented.

### 4.4 Results & Discussion

The Performance of RDM TELIC is measured on the basis of static traffic sets on different types of domain i.e. single path SP, Multipath MP, Several paths, Irregular several paths, Fish and Duck. The results are compared with TELIC. It has been found that the performance is improved in the case of allocation, rejection and pre-emption of AF and DF traffic and the missing part of the TELIC is also addressed. Different domains used for conducting experiments are shown in below mentioned Figure 4.5 (a, b& c) respectively.

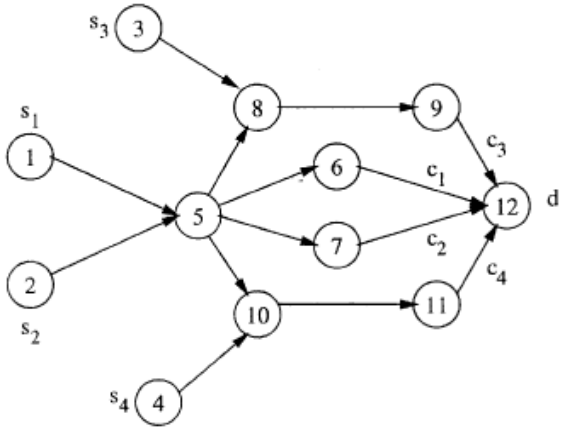


**Figure 4.5(a): Fish Topology**

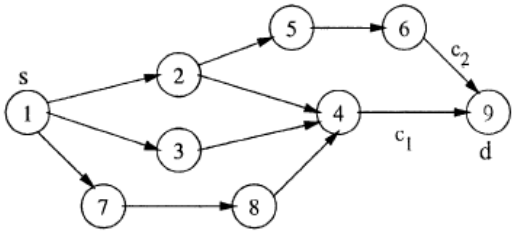


**Figure 4.5 (b) : Duck Topology**

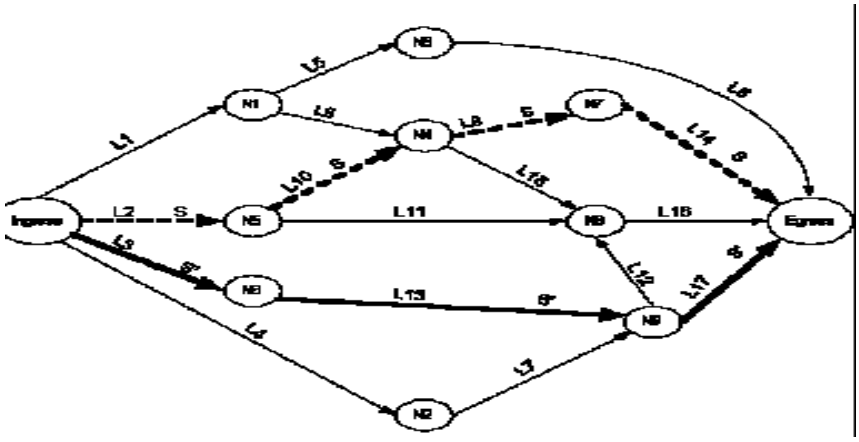


**Figure 4.5 (c ) : Irregular Several Paths (ISP)**

The experiment has been carried out starting 80-20 split of requested traffic set in which 20% of the total traffic was requested by EF and the remaining 80% further split into 70-30 by assigning 30 % to AF. The ratio of different traffic changed while increasing EF traffic trend. The experiment has been carried on around 25 traffic sets while applying on multiple domains as shown in above mentioned Figure 4.5. The results drafted on the basis of experiments can be found below.

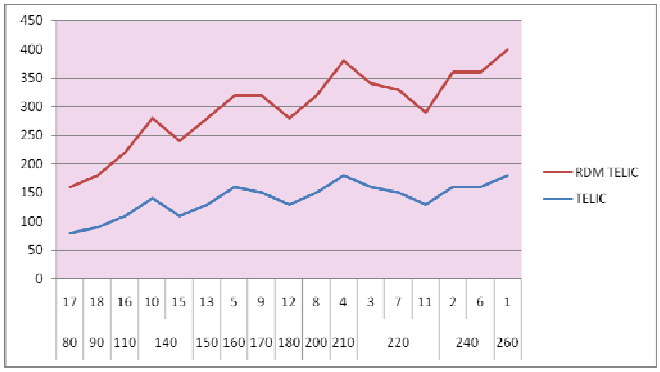


**Figure 4.6 (a): AF Allocation Requests on Fish and Allocation Trend**

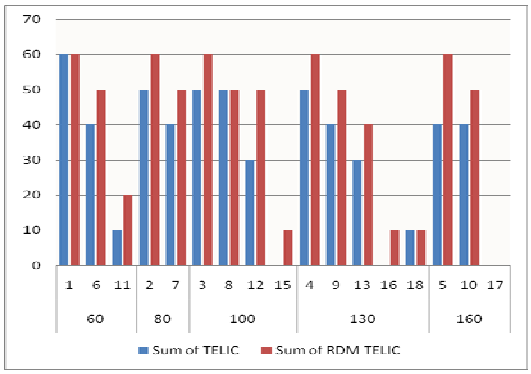


**Figure 4.6 (b): AF Allocation Requests on Duck and Allocation Trend**

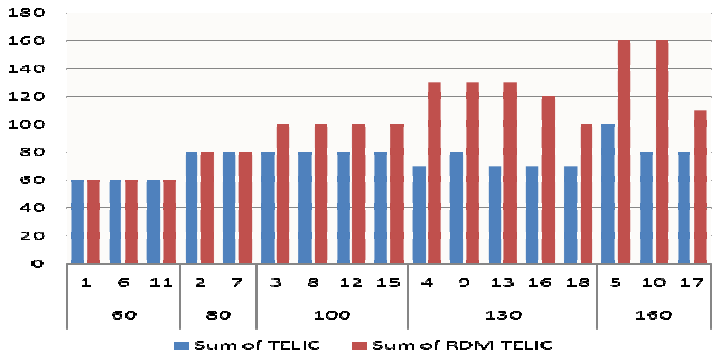


**Figure 4.6 (c): AF Allocation Requests on ISP and Allocation Trend**

It has been found that requests EF i.e. Premium Traffic has been allocated 100 % in both of the cases however the allocation in the case AF traffic has been increased in RDM TELIC as compared to TELIC which is shown in the above mentioned graph.

The allocation in the case of best-effort traffic is also goes on increasing direction as the bandwidth requested by DF is increased. The below mention graph shows the comparison of allocation by TELIC and RDM TELIC.

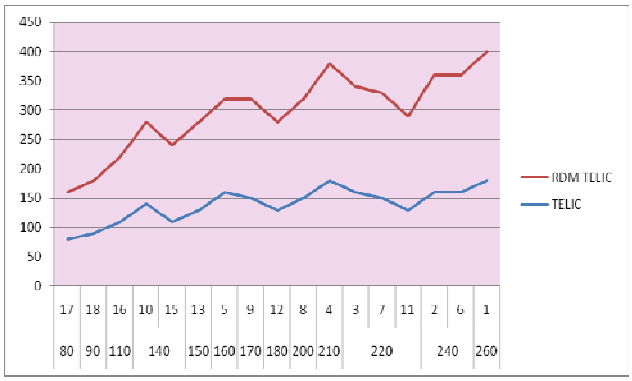


**Figure 4.7 (a): DF Allocation Requests on Fish and Allocation Trend**

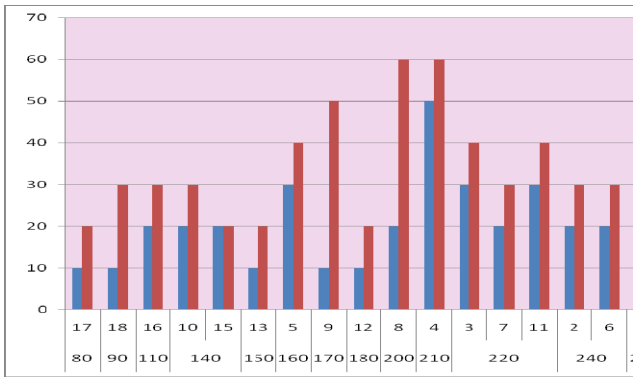


**Figure 4.7 (b): DF Allocation Requests on Duck and Allocation Trend**

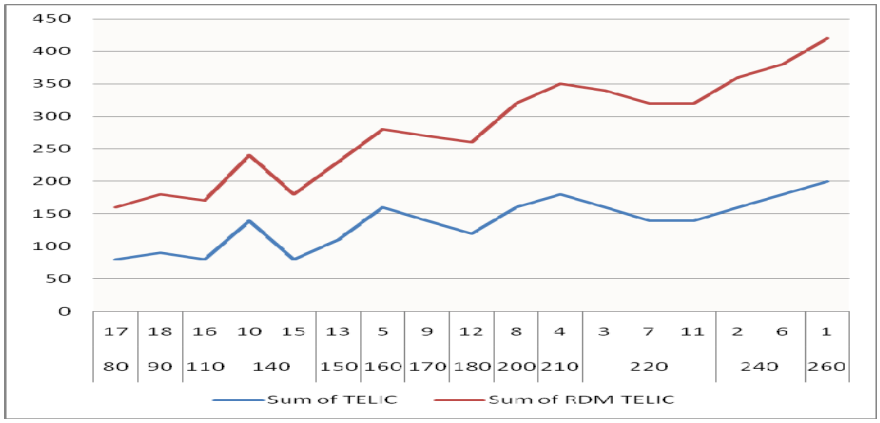


**Figure 4.7 (c): DF Allocation Requests on ISP and Allocation Trend**

As the allocation for AF and DF in the case of RDM TELIC increased as compared to TELIC, this increase in allocation has reverse effect on rejection of bandwidth of AF and DF respectively. The graph below represents the rejection trend of AF and best effort traffic.

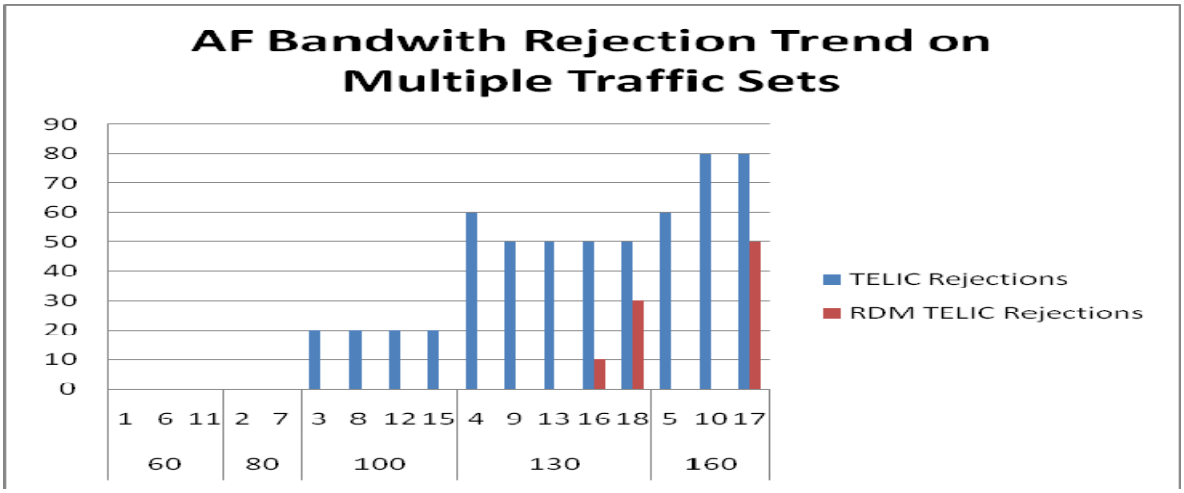


**Figure 4.8: AF Class Bandwidth Request and Rejection Trend**

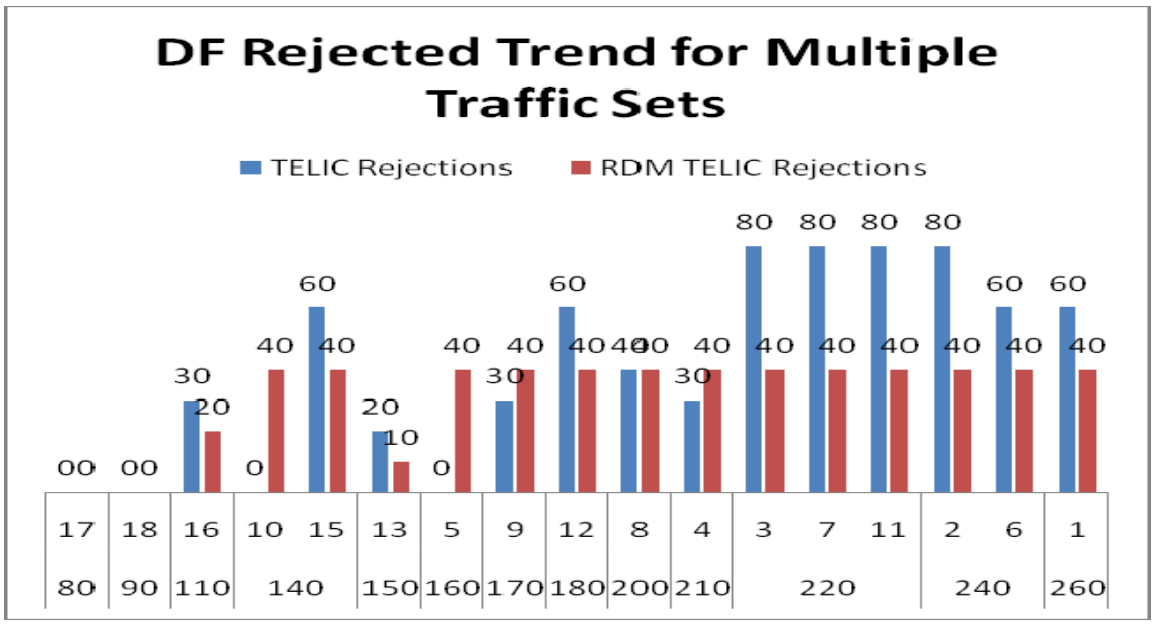


**Figure 4.9: DF Class Bandwidth Request and Rejection Trend**

It is evident from above mentioned figure 4.3 and 4.4 that the rejection for AF and best effort traffic on the links has been reduced in RDM TELIC. Preemption for the best effort traffic is also reduced in RDM TELIC as shown in the figure 4.5.

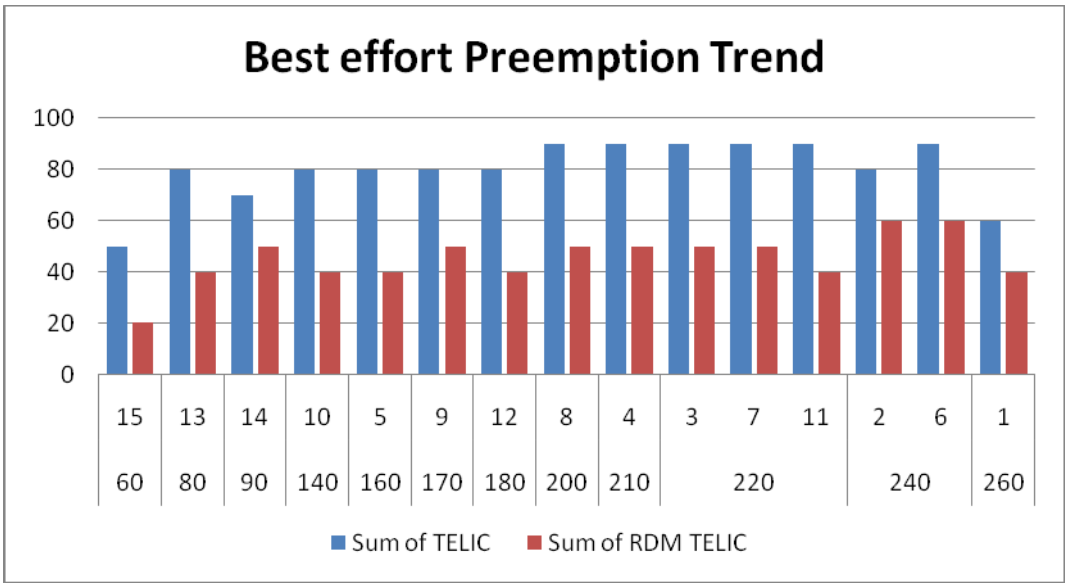


**Figure 4.10: Best Effort Class Bandwidth Request and Preemption Trend**

## 4.5 Conclusion and Future Work

RDM TELIC has been proposed. The implementation of the algorithm is done in C++ on the existing work of TELIC. The results are analyzed on multiple domains with different traffic sets.

Future work may include generating the dynamic traffic set through an application i.e. TS Generation Tool which supports multiple Class of Service (COS). LSP holding time may be incorporated to entertain further requests after rejection to compute the delay and improvement of the algorithm performance. This may be added with simulation on Network simulator.

**References and Bibliography**


[1] Dr. Junaid Zubari, An Automated Traffic Engineering Algorithm for MLS DIFSERV Domain 2002.

[2] Dr. Junaid Zubari, Nabeel Ali Al-Baloch, Noval Scheme for Traffic Engineering in access domains.

[3] C. Semeria "Traffic Engineering for the New Public Network" , *White Paper*, Juniper Network, 2000

[4] Xipeng Xiao, Alan Hannan, Brook Bailey, Traffic Engineering with MPLS in Internet 1998.

[5] Dr. Junaid Zubari, Emerging Methods for Voice transport over MPLS networks.

[6] Asha Rahul Sawant, Jihad Qaddour, MPLS DiffServ Combined approach

[7] Tomi Solala, A Framework of Integrated Service over Diffserv Network 2000.

[8] F. Le Faucheur, Ed., RFC 4125(MAM),4126(MAM Comparison),4127(RDM), Bandwidth allocation Models and comparison in Diffserv aware MPLS Traffic Engineering June, 2005

[9] S. Blake, D. Black, M. Carlson, "*An Architecture for Differentiated Service*," RFC 2475, December 1998

[10] B.Davie, A.Charny, J.C.R.Bennet,"*An Expedited Forwarding PHB (Per-Hop Behavior),"* RFC 3246, March 2002

[11] Artur Ziviani, Jose F.de Rezende, "Towards a Differentiated Services Support for voice traffic"

[12] Gonzalo Camarillo, Routing Architecture in Diffserv MPLS networks, Advanced Signaling Research Laboratory, Ericsson, FIN-02420 Jorvas, Finland