Traffic Engineering in MPLS Networks: ARS – An Adaptive Routing Scheme Based on Control Charts

Adriana de Andrade Oliveira [1], **Martín Gómez Ravetti** [1] , **Geraldo Robson Mateus** [1]

[1]Departamento de Ciência da Computação – Universidade Federal de Minas Gerais
Av. Antônio Carlos, 6627 – Campus Pampulha – Belo Horizonte, MG

dri@dcc.ufmg.br, martin@dcc.ufmg.br, mateus@dcc.ufmg.br

Abstract. *One of the modern trends of Internet design is that lower level transport protocols should be network conscious and adaptive. They should monitor network conditions and answer appropriately to changes in these conditions, which requires using network information (measurements) from the recent past to guide future behavior. This paper presents the design and evaluation of a new adaptive routing scheme for MPLS capable networks with load balancing and minimal drop rates and average packet delay goals. This paper also proposes and evaluates the use of control charts as a traffic change´s detection mechanism in IP networks.*

Resumo. *Uma das tendências modernas para a Internet é a necessidade de protocolos de transporte terem conhecimento da rede e serem adaptativos. Eles devem monitorar as condições da rede e responder de forma apropriada a mudanças nessas condições; o que requer o uso de informações da rede (medidas) do passado recente para guiar o comportamento no futuro próximo. Este artigo apresenta o desenho e avaliação de um novo esquema de roteamento adaptativo para redes MPLS com os objetivos de balanceamento de carga, minimização do atraso médio dos pacotes e minimização das perdas. O artigo também propõe e avalia o uso de cartas de controle como mecanismo de detecção de mudanças do tráfego em redes IP.*

1. Motivation

In an IP network, there seems to exist two issues that must be considered to guide the design of a routing mechanism. The first issue is that load balancing techniques are needed, since various links stay underutilized most of the time, while others are congestioned. The second issue is the highly dynamic nature of traffic demands. It is worth noting that these two goals change the concept of ”best-effort” routing used nowadays. Now, choosing a route for a flow means choosing the possibility of obtaining the best global performance for the network and not always the shortest path for that specific flow. Therefore, routing mechanisms should try to use all the idle resources while being very adaptive to traffic changes.

Due to its ability to control the routes of flows in a IP network, MPLS was chosen to be used in this work. MPLS [Awduche et al., 1999] is a packet label-based switching technique where packets are assigned a label which identifies the treatment the packets will receive in the network. It allows sophisticated routing control capabilities to be introduced into IP networks (explicit routing) and can help to build backbone networks that better support QoS traffic.

Hence, this work addresses the two issues listed above (lack of load balancing and the dynamic nature of traffic) to achieve QoS routing in the following way. To cope with lack of load balancing, it uses the explicit routing feature of a MPLS (Multi Protocol Label Switching) network, changing dynamically the routes of some LSPs (Label Switching Paths) in order to avoid congestioned links. An algorithm is proposed to deal with these rerouting needs. This algorithm sometimes chooses longer paths for a LSP (instead of the shortest path) in order to balance the overall links load and to allow a better use of the network resources.

To deal with the dynamic nature of Internet traffic, it is also proposed in this work a mechanism based on traffic monitoring and control charts. Since it is not scalable to distribute every change in the link states, the proposed scheme will try to figure out if and why the traffic changed: if the change in traffic is due to common variation or if it is due to unexpected facts, such as a link or a router failure. With this approach, a link state update will be triggered only when a subtle traffic change occurs. This approach reduces the volume of updates of traffic information on the network.

The proposed scheme has a twofold goal: detect as soon as possible a traffic change that indicates that the routing scheme currently used will probably fail in the future and, based on this information, reconfigure the routes in the network, reducing the number of packets dropped and minimizing the effects of using longer less congestioned routes on the average packet delay.

2. Quality of Service in IP networks

A number of techniques were proposed to provide QoS in IP networks. Among the most popular we can list Integrated Services (IntServ) [Seaman et al., 2000], Differentiated Services (DiffServ) [Blake et al., 1998], Constraint-Based Routing (CBR) [Apostolopoulos et al., 1999], and, more recently, Multi Protocol Label Switching (MPLS) [Boyle et al., 2002].

IntServ and DiffServ were the two first attempts of the Internet community to support QoS in the Internet. The IntServ solution weakness is its lack of scalability, since the resource reservation process is made hop by hop. Diffserv relies on traffic conditioners sitting at the edge of the network to perform this QoS function: traffic classification, marking, shaping and policing. But, the DiffServ model does not attempt to guarantee a level of service. It rather strives for a relative ordering of aggregations such that one traffic aggregation will receive better or worse treatment relative to other aggregations. Admission control at the boundary does not consider the availability of resources in the Diffserv network region along a specific path.

In [Awduche, 1999], the applications of MPLS to traffic engineering in IP networks are discussed, since traffic engineering in conventional IP networks is a challenging problem. MPLS is crucial to load balancing since in an IP network without MPLS capabilities it is a challenge to control the flow's routes. With the MPLS explicit route capacity, the flow does not have to follow the shortest path route defined for it anymore.

Constraint-Based Routing (CBR) is a process that is able to find paths that are subject to multiple quantitative as well as qualitative constraints. Some of several ongoing works are already proposing mechanisms to combine some of the techniques described. CBR, for example, is an important tool to be used in conjunction with MPLS for arranging how traffic flows through the network and improving its utilization.

Although all proposed techniques allow an increase in the quality of services provided, they do not address the problem of dynamically balancing the traffic on the network to achieve congestion control.

Our work address the problem of dynamically balancing the traffic on the network to achieve congestion control. It tries to adapt the routing scheme of a MPLS network based on a traffic changes detection tool (Control Charts), which will be the key aspect for the dynamic behavior of our scheme.

This paper is organized as follows. The next section discuss some present related work. Section 3 proposes a first adaptive scheme that consider the network traffic conditions to change LSP (Label Switching Paths) routes dinamically, focusing on the load balancing of the network. In the Fourth Section, another adaptive scheme is presented, based on shortest paths computation. The following section describes control charts. Sixth Section presents some experiments, which were executed to show how control charts can be used to trigger some modifications on the routing scheme. Concluding remarks are discussed in the last section.

3. Related Work

To address the problem of dynamically balancing the traffic on the network, the important issue to be considered is the setup of the LSPs routes in the MPLS networks and also the frequency that the process is repeated. Our work is a contribution in this direction, since it tries to make use of alternative paths when congestion occurs in a very dynamic way. The pre-established LSP routes of our algorithm can be computed using the techniques proposed by [M. Chatzaki and Courcoubetis, 1999].

In [M. Chatzaki and Courcoubetis, 1999], they work to enhance MPLS with resource allocation capabilities. Their approach is related to themes like admission control and route selection. Their work is similar to ours in the sense we also do route selection, but they face a scalability problem in the paper, due to their proposed method to calculate the costs of accepting a new flow on the network. Our works is different in the sense that we do not classify flows and we do not deal with admission control. We only rearrange the flows to be in a more balanced way.

One of the most cited congestion control schemes, called MIRA (Minimum Interference Routing Algorithm) [Kodialam and Lakshman, 2000], is based on a dynamic online path selection algorithm. The main weakness of this scheme are the computational complexity necessary to implement it, the unbalanced network utilization for some network topologies and the fact that it does not take into account the current traffic load in routing decisions.

It is presented in [Salvatori and Batiti, 2003] a load-balancing scheme for a MPLS network through a local search algorithm. The idea is to minimize the congestion of the network by performing local modifications. For each tentative move, the most congested link is located and one of its crossing LSPs is rerouted along an alternative path. The main differences between their scheme and ours are that they search *the best* alternative path and also the way they trigger the rerouting procedure.

In our scheme, the algorithm stops when it finds *the first* alternative path that satisfies the demand and has a number of hops that does not exceed the shortest path by a threshold.

In their work the rerouting procedure is triggered when the set up of a new LSP causes the detection of network congestion (when only $x\%$ residual bandwidth is left on some link). The impact of the parameter x over the algorithm behavior is cited by them as future work. Our scheme only tries to reroute a LSP if a new demand arrives and it is not able to find a route.

Our scheme will also use control charts to detect traffic changes on links. If a traffic change is detected, some actions are triggered to evaluate the load of the network and choose the best routing policy to be used in the current conditions: rerouting or not.

4. A Naive Adaptive Routing Scheme

4.1. IP with OSPF x IP over MPLS

To study the problem of routing in IP networks a series of experiments were conducted. Firstly, we were trying to figure out the mayor diferences between IP networks with the basic OSPF protocol and IP networks over MPLS. In order to evaluate the performance of OSPF x MPLS networks, two network configurations were investigated. Configuration in this work means the routing protocols and type of routers used.

The first configuration simply uses the OSPF algorithm and does not contain MPLS capable routers; it is the routing scheme used by standard IP networks *(OSPF)*, which does not take into account the load at the link to route the traffic, and normally routes the flows through the shortest path.

The second configuration, called *MPLS*, establish routes for each LSP in the beginning of the simulation but does not change its configuration due to changes in traffic load. The routes for the LSPs are not based only on shortest paths but in load balancing as well. When the network is congested, the LSP established a priori can help to balance the load. But, if the network is not highly loaded, the use of the LSP in a longer route can lead to a higher delay.

OPSF and MPLS were evaluated with three different sample networks. Several simulations were carried out, with lightly and heavily-loaded links to determine the impact of each different routing approach on the packet delay and packet drop rate.

In the first sample network, there are two routes for a traffic flow, but the difference in hops between them is four hops. In the second sample network, there are three routes for a traffic flow, but the difference in hops between them is 2 and 1 hops. In the third sample network, there are two routes for a traffic flow, and the difference in hops between them is only 1 hop.

All networks show congestion in some links, but this congestion ceases in a short interval. We analyzed the impact of this change in traffic in the routing schemes used. To simulate traffic variation, some traffic sources, which in the beginning go through the shortest path, are finalized.

The performance metrics used were the number of packets lost and the average packet delay, which is defined as the metric for the network with OSPF configuration over the metric for the network with MPLS configuration.

Topology	OSPF	MPLS
1	$3,7\%$	0
2	$3,7\%$	0
3	$62,0\%$	0

Table 1: Packets Dropped Rate OSPF X MPLS

Table 1 shows the rate of packets dropped for the three sample networks simulated. As we expected, the performance of OSPF quickly deteriorates as the network load grows high. At high utilization, when there is contention for bandwidth, the drop rate increases. The MPLS solution avoids the more congested path, decreasing the drop rate. As can be noted from Table 1, the improvement on the drop rate is significant.

The other side of the coin is the effect of the use of MPLS on the delay. Table 2 shows the packet delay time ratio for the MPLS, which is defined as the average packet delay time for the network running with OSPF over the packet delay time for the same topology running with MPLS. The delay increases as expected since the LSPs are using longer routes, different from the shortest more congestioned ones. But, the increase in delay is compensated by the decrease in the number of packets lost.

Topology	OSPF/MPLS
1	0.56
2	0.81
3	3.69

Table 2: Average Delay/Packet OSPF and MPLS

The good surprise is that the use of longer explicit routes does not hurt the delay time so much in the case of a congestioned network, as the case of the third example. The reason stems from the fact that the time spent in queues or being re-send are higher than the time spent going through the longer route.

It was show in this experiment that MPLS is crucial to load balancing since in an IP network without MPLS capabilities it is a challenge to control the flow's routes. With the MPLS explicit route capacity, the flow does not have to follow the shortest path route defined for it.

4.2. Pure MPLS x Adaptive MPLS

Afterwards, we have developed a scheme for changing the routes of some LSPs on a MPLS capable network, based on link utilization, with the goal of balancing the traffic in the network.

The load status at each link is represented by its utilization. The scheme was built on top of NS (Network Simulator) and was implemented through *scripts* that included commands for monitoring network links. One function of the script was to collect load information at each link. In order to reduce message traffic in the network, a route is changed only when its link utilization varies significantly. In other words, an interval of utilization variation was defined for the links such that the router only transmits a message demanding the route modification when the link utilization variation exceeds the established interval.

The third configuration *(Adaptive MPLS — AMPLS)* is then a modification of the second one due to the presence of this link utilization monitoring feature, which acted with the following goals:

- avoiding the use of longer routes in the case of a less loaded network
- providing load balancing in the case of a congested network, using alternative routes, not in the shortest path.

The adaptive routing scheme (AMPLS) and the pure MPLS approach were evaluated with the same three sample networks described before.

As opposed to fixing routes for the LSPs during all the simulation, it was allowed that an LSP changed routes based on a traffic change. Dynamic traffic changes demand a re-evaluation of the best routes. We examined a scenario in which the traffic demands changed and the network adapted itself to the situation.

Table 3 shows that without the adaptive scheme, the degradation of performance for less congested networks will happen, since the LSP routes established in the beginning are not well suited to accommodate the traffic change, in this case the traffic was lower than in the moment the

Topology	MPLS/AMPLS
1	1.32
2	1.23
3	1.70

Table 3: Average Delay/Packet MPLS and AMPLS

LSPs were set up. The better example is the third topology where the traffic sources continue to send the packets through the longer route.

The AMPLS scheme exhibited the best performance. The reason stems from the fact that it took into consideration information about the traffic load (e.g. critical links) to make better routing decisions. It is also important to mention the adaptive nature of this scheme, since it utilizes alternative LSP routes, depending on the network utilization.

5. Adaptive Routing Scheme with Path Selection Strategy (ARS)

The experiments described in the previous section showed the importance of using alternative paths to load balance the traffic in MPLS networks and the importance of monitoring the traffic. It encouraged the authors to investigate the problem of alternative path selection in a MPLS networks in the moment a rerouting is needed and the also traffic changes point detection problem.

In this section, it will be presented a new algorithm for the alternative path selection. Later, the problem of traffic changes point will be addressed.

The path selection algorithm work as follows: firstly, it tries to find a route for a request that just arrived in the network. Three situations can happen:

- there is no route
- a very long route is found
- a route with maximum D hops difference of the shortest route is found

If no route is found, meaning that the request will be rejected, or if a very long route is found, the algorithm tries to reallocate some LSP, i.e., it will try to find a LSP that can be rerouted to leave space for the new request that just arrived.

```
// D is some distance to the shortest-path
route = alocateOneRoute(new LSP, demand);
if (route does not exist
or
    route > minRoute+D)
    find the congestioned link in
      the shortest path route;
    do
       for each LSP that crosses this link;
       if (LSP.demand > newLSP.demand)
          find a newroute such as
            newroute >= LSP.route +D;
    while (not found and there are LSPs);
    if found() changeRoutes;
```

Table 4: Routing Scheme - Path Selection Strategy

To choose a LSP to be rerouted, the algorithm search for the most utilized link in the shortest path route of the request that just arrived. Then it checks for each LSP that crosses this link if this LSP has a demand greater than the demand of the new request and if there is an alternative route for this LSP.

We will try to reroute just one LSP. The algorithm stops when it finds one LSP that satisfies the constraints or when no LSP is found.

In order to reduce message traffic in the network, a LSP route is changed only when a new request will be discarded otherwise or when the change of number of hops for the new one is considerable, which means that the overall delay would be reduced.

5.1. Simulation and Experiments of ARS

The proposed scheme was built on top of NS (Network Simulator).

5.1.1. Validate

In order to validate our scheme (ARS), we reproduced the first set of experiments described in [Salvatori and Batiti, 2003]. They also proposed a path selection strategy and they compare their algorithm DYLBA with MIRA [Kodialam and Lakshman, 2000]. In their experiments they propose a network topology 5.1.1, which is a special case of the concentrator topology.

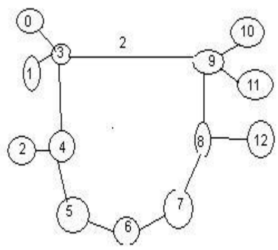


Figure 1: Concentrator Topology

They first consider the set-up of three LSPs with the same bandwidth. The result of MIRA is the following:

Setup order	Bdwth	Ingress	Egress	Route
LSP1	1Mbps	2	12	4-3-9-8
LSP2	1Mbps	1	11	3-9
LSP3	1Mbps	0	10	blocked

Table 5: MIRA's results

In this first example, the MIRA scheme failed to route the third LSP. By using their scheme (DYLBA), the set-up of the three LSPs produces the following results:

Setup order	Bdwth	Ingress	Egress	Route
LSP1	1Mbps	2	12	4-3-9-8
LSP2	1Mbps	1	11	3-9
LSP1	1Mbps	2	12	4-5-6-7-8
LSP3	1Mbps	0	10	3-9

Table 6: DYLBA's results

Our scheme (ARS) managed to route all LSPs and also provided the advantage of not requiring the rerouting of any LSP.

Setup order	Bdwth	Ingress	Egress	Route
LSP1	1Mbps	2	12	4-3-9-8
LSP2	1Mbps	1	11	3-9
LSP3	1Mbps	0	10	3-4-5-6-7-8-9

Table 7: ARS's results

5.1.2. Performance

Another experiment was made to check the performance of the network using our scheme (ARS) versus pure MPLS. The performance metrics used were the number of packets dropped and the average packet delay.

Table 8 shows the results of ARS and Pure MPLS in the same network topology used in the previous subsection. It shows the rate of packets dropped. As we expected, the performance of Pure MPLS quickly deteriorates as the network load grows. At high utilization, when there is contention for bandwidth, the drop rate increases. The ARS solution avoids the more congested path, decreasing the drop rate. As can be noted from Table 8, the improvement on the drop rate is significant.

Sch	Pkts	% Drop,	Avg Hops	Avg Delay
MPLS	1118	22.5	2.29	0.14
ARS	1128	0	4.62	0.11

Table 8: Pure MPLS x ARS

The other side of the coin is the effect of the use of longer paths on the delay. As we can see, the average number of hops increases, but the increase in delay is compensated by the decrease in congestion, queues and retransmission.

In the second example, we examine the impact caused on the network by the introduction of the adaptive routing scheme. As opposed to shortest paths routes for the LSPs, the adaptive scheme sometimes chooses longer ones. We examine a scenario in which the traffic demands are not so high and the network adapts itself to the situation.

Sch	Pkts	% Drop,	Avg Hops	Avg Delay
MPLS	850	1.41	3.10	0.14
ARS	852	1.40	3.10	0.14
ARS-Big D	860	0	4.80	0.098
DYLBA				0.090

Table 9: Schemes Comparison

Table 9 shows the rate of packets dropped for the sample network simulated, for the various schemes studied. As we expected, the performance of ARS MPLS is similar to Pure MPLS when the network is not congestioned, since there is no shortest path to avoid. Besides, the ARS without hop restrictions (Big D) managed to avoid the small part of the network that was congested, increasing the average number of hops but not the average delay. It is interesting to note that the DYLBA results show a similar average delay for the longer route (LSP1), although it has a superior computational cost than the ARS proposed.

6. Traffic Changes Detector

There is much interest in using network measurements for both modeling and operational purposes. Measurements are inherently bound to present. However, the modeling and operational uses of these measurements are only successful if they are good predictors of the future or give a good idea of what is happening and the actual tendency. In this work, the traffic change detector will be used to trigger the load balancing mechanisms when some unusual event happens. We will work with XBar-Charts and EWMA-Charts, which are described in the following sections. These charts will be used to check traffic tendency.

6.1. XBar-Charts

Shewhart's statistical process control (SPC) [Hansen, 1963] is a methodology for charting the process (XBar-Charts) and quickly determining when it is out of control.

A series of rules exist that are used to detect conditions in which the process is behaving abnormally to the extent that an out of control condition is declared. In this work, we used the extreme point condition test, where a point is either above the upper limit or below the lower limit.

The XBar-chart is computed from *m* data points, which are ordered in time. These points are in fact the mean of *n* data points measured. The first task to compute the control limits is to find the population mean (X) and the mean range (R). The mean range is estimated based on the range of *m* subgroups.

$$\overline{\overline{X}} = \frac{\overline{X}_1 + \overline{X}_2 + ... + \overline{X}_m}{m} \tag{1}$$

$$\overline{R} = \frac{R_1 + R_2 + ... + R_m}{m} \tag{2}$$

The limits are calculated as follows:

$$LowerLimit(LL) = \overline{\overline{X}} - A_2(n)\overline{R} \tag{3}$$

$$UpperLimit(UL) = \overline{\overline{X}} + A_2(n)\overline{R} \tag{4}$$

where A_2 values depend on n, as shown in Table 10.

N	2	3	4	5	7	9
A2	1.880	1.023	0.729	0.577	0.419	0.337

Table 10: Some A_2 values

6.2. EWMA-Charts

An EWMA (Exponentially Weighted Moving Average) Chart is used when it is desirable to detect out-of-control situations very quickly (Montgomery, 1990). It has a built in mechanism for incorporating information from all previous data, weighting the current information with a higher weight.

The main advantages of EWMA charts are: they detect out-of-control conditions more quickly than XBar charts and this detection can be done by using only one rule, being within or outside the 3-sigma limits. The EWMA chart worst disadvantage is that it is more difficult to construct.

The EWMA statistic at time *t* is computed recursively from individual data points, which are ordered in time. The first EWMA statistic is the average of historical data and is the centerline for the control chart.

$$Xhat(0) = \frac{\overline{X}_1 + \overline{X}_2 + ... + \overline{X}_m}{m} \quad (5)$$

$$Xhat(i) = \lambda \overline{X}_i + (1-\lambda) Xhat(i-1) \quad (6)$$

The upper (UL_i) and lower (LL_i) limits are calculated as following:

$$Sigma = \frac{R}{D2(n)} \quad (7)$$

$$F_i = \sqrt{(\frac{\lambda}{1-\lambda})(1-(1-\lambda)^{2i})} \quad (8)$$

$$LowerLimit(LL_i) = Xhat(0) - \frac{3Sigma}{\sqrt{n}} F_i \quad (9)$$

$$UpperLimit(UL_i) = Xhat(0) + \frac{3Sigma}{\sqrt{n}} F_i \quad (10)$$

where D_2 values depend on n, as shown in Table 11.

N	3	5	7	9
D2	1.693	2.326	2.704	2.970

Table 11: Some D_2 values

7. Control Charts Experimental Results

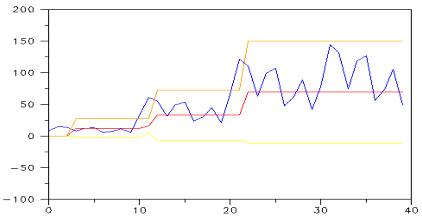
In this work, to test the use of control charts for traffic changes detection, it was generated some synthetic data, based on distributions functions such as exponential, poisson and lognormal. Three traffic scenarios were generated, synthetically:

- First Scenario: Increasing Traffic - in the beginning the traffic followed a distribution with mean 10 Mbps. Then the mean would increase abruptly to 40 Mbps, 80 Mbps and 95 Mbps. It is worth to emphasize that the control charts should be able to detect 4 major changes in this scenario.
- Second Scenario: Decreasing Traffic - in the beginning the traffic followed a distribution with mean 95 Mbps. Then the mean would decrease to 80 Mbps, 40 Mbps and finally 10 Mbps. Also in this scenario, the control charts should be able to detect 4 major changes.
- Third Scenario: Alternate Traffic - in the beginning the traffic followed a distribution with mean 10 Mbps, which increase to 60 Mpbs, decreased to 30 Mbps and again increased to 70 Mpbs after a while. In this scenario, the control charts should be able to detect 4 major changes.

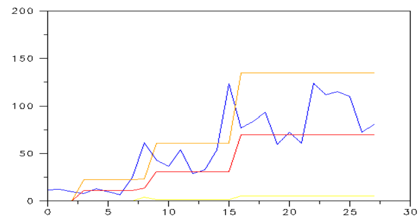
Several experiments were run to check if the control charts would be able to detect these traffic changes. It can be seen in Figure 2 the influence of the sample size for the XBar-chart in the first scenario. We evaluated the results for different values of the *n*: 5, 7 and 9. In all cases the Xbar-char was able to detect the traffic increase, but with n=9, the limits are closer to the reality.

Several experiments were run using different values for λ in EWMA-Charts: 0.10, 0.20 and 0.30. Bigger λ values represent more importance for current data. It is expected the EWMA-chart to be more conservative for small $lambda$ values.

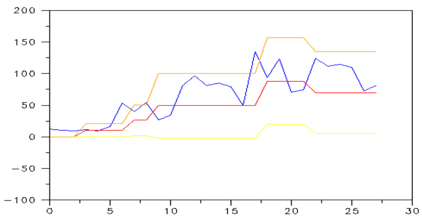
Figure 2: XBar Charts



(a) n=5



(b) n=7



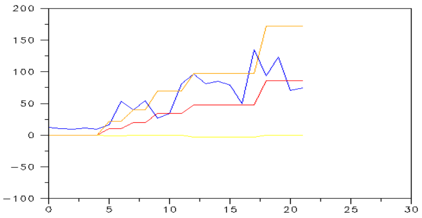
(c) n=9

If we compare XBar-charts and EWMA-charts, it is noticed in Figure 7 that the EWMA-chart reacts quickly, but it must be clear that it did not happen as much substantial changes as noticed by the EWMA-charts. The synthetic traces were generated creating four major traffic changes.

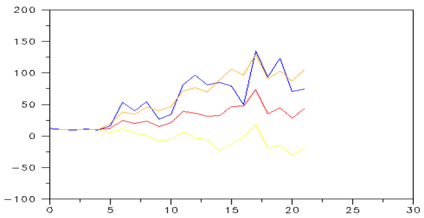
Table 12 shows the number of recalculations made by the two approaches. The recalculations occurs when an out-of-control condition was noticed, that is, the data is outside the 3-sigma region of XBar and EWMA-charts. The charting process was run again and therefore we could check if the new chart limits would confirm the tendency of the traffic change.

λ	n	m	scenario	EWMA	XBar
0.10	9	3	1	17	3
0.10	9	5	1	16	4
0.20	9	3	1	11	3
0.20	9	5	1	13	4
0.30	9	3	1	6	3
0.30	9	5	1	9	4

Table 12: Number of Recalculations



(d) xbar



(e) ewma

Figure 3: XBar Charts vs EWMA Charts - Scenario 1

The number of recalculations for EWMA-charts is much bigger than the number of recalculations for XBar-charts. But it can be seen in Figure 7 that the EWMA-chart notice small changes in data. Hence, one can choose the better chart for his/her application. If quickly changes are very important, the EWMA-chart is more appropriate, but its cost (processing and communication of changes) is higher. XBar-charts detects major trends at a lower cost.

Some results will be presented for the second scenario in Figure 7. It could be noticed in Figure 7 that the EWMA-chart recomputes its values and follows the traffic nature. The XBar-chart was not able to detect the traffic decrease.

In this work, the rule used in the experiments to detect out-of-control situations, in the XBar-chart, was the extreme point condition, cited earlier. Despite being easier to calculate, XBar-charts have more rules to test in order to detect abnormal situations than EWMA-charts. The results found for the second scenario show the need of implementing these other rules.

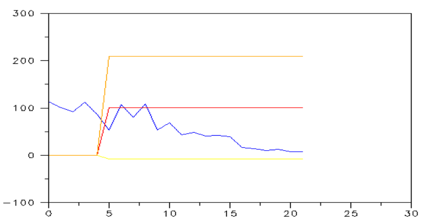
8. Traffic Changes Detection and ARS

After discussing how the control charts work, we will propose a new adaptive routing scheme for MPLS networks based on traffic change´s detection. The path selection algorithm proposed previously will be combined with the control charts in this new scheme.

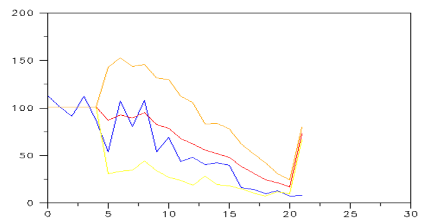
Table 13 gives the pseudo-code of the monitoring part of the proposed scheme. The rerouting part was shown in Table 5.

The first step of the proposed scheme is the calculation of the K shortest disjoint paths for a request. We work with pre-computed alternative routes, using Dijkstra algorithm. In the experiments we have made, the alternative routes are disjoint. This issue will be studied latter on. The pre-calculation of routes will help the rerouting step and the path selection algorithm.

The control charts are also computed and will be used to monitor each ingress router. If a traffic change is detected, the control charts is updated. If the utilization of a link, reaches a dangerous limit, the algorithm tries to reroute some LSPs, avoiding the congested link. It is a preventive approach since it tries to leave more bandwidth for future requests.



(a) xbar



(b) ewma

Figure 4: XBar Charts vs EWMA Charts - Scenario 2

```
for each possible LSP
  calculateKMinRoutes();
for each link
  calculateControlChart();
forever
  if monitorLink detects traffic has changed
     updateControlChart();
     if (link.utilization > %X . link.capacity)
        for each possible LSP
            calculateKMinRoutes(Avoiding Link);
```

Table 13: Routing Scheme - Monitoring

9. Concluding Remarks

Routing mechanisms for MPLS networks carry out critical functions for the performance of these networks. They take on responsibility for resource management and load balancing. In this work, we have developed a routing scheme for a MPLS capable network, based on traffic changes detection and alternative path selection, with the goal of balancing the traffic in the network to avoid congestion and achieve better resource utilization.

It was observed that with the proposed scheme we have obtained substantial reduction on the number of packets dropped, without imposing severe penalties on the average packet delay. In concluding, we have shown that the proposed adaptive routing scheme improves the overall performance of the network and reduces the packet losses caused by congestion.

Since the modern trend in routing is to be network conscious and adaptive, we have shown that control charts could be used to monitor network conditions and allow the routing protocol to respond appropriately to them. Two different types of control charts were evaluated and several experiments were run on synthetic data to show the main advantages of each type of control chart. EWMA-charts should be used when it is necessary to react quickly to small changes. XBar-charts should be used when major changes should be detected. Its main advantage is its simplicity and low overhead.

Although the results are encouraging, there are several aspects of the proposed scheme for which additional work is needed. Two of the more important are the simulation of control charts together with the path selection algorithm and also the study of the initial computation os LSPs. In this work the possible paths were configured to be the K shortest disjoint paths.

References

Apostolopoulos, G., Williams, D., Kamat, S., Guerin, R., Orda, A., and Przygienda, T. (1999). QoS Routing Mechanisms and OSPF Extensions.

Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and McManus, J. (1999). Requirements for traffic engineering over mpls.

Awduche, D. O. (1999). MPLS and Traffic Engineering in IP Networks. pages 42–47.

Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and Weiss, W. (1998). An architecture for differentiated services.

Boyle, J., Gill, V., Hannan, A., Cooper, D., Awduche, D., Christian, B., and Lai, W. (2002). Applicability statement for traffic engineering with mpls.

Hansen, B. L. (1963). *Quality Control: Theory and Applications*. Prentice-Hall.

Kodialam, M. and Lakshman, T. V. (2000). Minimum Interference Routing with Aplications to MPLS Traffic Engineering.

M. Chatzaki, S. Sartzetakis, N. P. and Courcoubetis, C. (1999). Resource Allocation in Multiservice MPLS.

Salvatori, E. and Batiti, R. (2003). A Load Balancing Scheme for Congestion Control in MPLS Networks.

Seaman, M., Smith, A., Crawley, E., and Wroclawski, J. (2000). Integrated service mappings on ieee 802 networks.