

Analysis of Multilayer Traffic Engineering strategies for optical networks

Student: Kimberley Vanvaeck
 Promoter: Mr. Joeri Gerrits
 Academic year: 2004 - 2005

Index

1. Introduction.....	2
1.1. Choice of the subject.....	2
1.2. Hanoi University of Technology	2
1.3. The project and its goals	2
1.4. Objectives	3
2. Introduction to Optical networks.....	4
2.1. The need for a faster medium with a higher bandwidth	4
2.2. Optical media.....	4
2.3. Multimode and single-mode fiber.....	5
2.4. Wavelength Division Multiplexing.....	5
2.5. Optical Cross Connects.....	6
3. Evolution of optical networking	7
3.1. Intelligent optical networks.....	7
3.2. Multilayer networks.....	7
3.3. Traffic Engineering and MPLS.....	8
3.4. Multilayer Traffic Engineering.....	10
4. Analysis of MTE strategies	12
4.1. The network topology	12
4.2. A reactive strategy	13
4.2.1. <i>The strategy</i>	13
4.2.2. <i>Analysis by means of an example scenario</i>	13
4.2.3. <i>Window size</i>	19
4.2.4. <i>Conclusion</i>	19
4.3. A proactive strategy	20
4.3.1. <i>The strategy</i>	20
4.3.2. <i>Analysis by means of an example scenario</i>	23
4.3.3. <i>Conclusion</i>	29
4.4. General conclusion.....	29
5. Problems	31
6. Conclusion and results.....	32
7. Word of thanks.....	33
8. References.....	34
8.1. Bibliography	34
8.2. Software	34
Appendix I: List of frequently used abbreviations	35

1. Introduction

1.1. Choice of the subject

During their last year of studies in Applied Computer Sciences, students at EhB must realize a final work. The goal of the final work is to research and resolve a problem, as well as to learn new techniques. This does however not always have to be accomplished in Belgium.

A few years ago, I already knew that I wanted to do part of my studies in a foreign country, to meet other cultures, to experience what it would be like to stay there for a longer period and to cooperate in a project there. When I was asked if I was interested in working on a research project on optical networks in Vietnam, this seemed like the ideal occasion to do so. Also, given my interest for networking, the subject of optical networks really attracted me.

I received a scholarship from the Flemish Interuniversity Council and at the 1st of March 2005, I went to Hanoi, the capital of Vietnam. There I participated in a research project at Hanoi University of Technology for one month.

1.2. Hanoi University of Technology

Hanoi University of Technology (HUT) is based in Hanoi, Vietnam. It was the first national university of technology, established in 1956. Over nearly half a century, HUT has been continuously growing. At present, HUT has fourteen faculties and about 1800 employees, with 1500 in the teaching staff. There are more than 35000 undergraduate students being trained at HUT, as well as 2000 graduate students.

1.3. The project and its goals

My research work took place at the faculty of Electronics and Telecommunications and was part of the project “Multilayer Traffic Engineering: Performance Evaluation”.

The project started on December 2004, in cooperation with Vrije Universiteit Brussel (VUB), and runs over three years. At the current stage of the project, information related to Multilayer Traffic Engineering (MTE) is being collected and important parameters to think about when evaluating MTE strategies are being determined.

I will get back to this later, with a more technical explanation of what Multilayer Traffic Engineering actually is. Generally speaking, the entire project concerns the management of bandwidth and the routing of data on optical networks.

The next step in the project is to develop a new integrated MTE strategy which is a trade-off between several aspects such as network stability and capacity efficiency.

1.4. Objectives

Before going to Vietnam, no exact objectives were set yet, as the project had already been running for a considerable amount of time and several people were working on it, the objectives could be set after I had arrived at HUT.

From previous research it had become clear that choosing the right strategy is often more difficult than applying it.

At HUT I did research on different MTE strategies. First I had to read papers and search for information on the Internet, to learn about the subject of Multilayer Traffic Engineering. Then I had to develop some strategies and examine which influence they would have on an example network, where the main traffic was Internet based. Next, I had to simulate these with a program called Glass, to check if my findings were correct.

2. Introduction to Optical networks

2.1. The need for a faster medium with a higher bandwidth

Within time, the amount of traffic on networks and the speed at which the traffic has to be transmitted always becomes higher. The deployment of optical fiber (see 2.2) and Wavelength Division Multiplexing (see 2.4) has proven to be a good solution for this.

2.2. Optical media

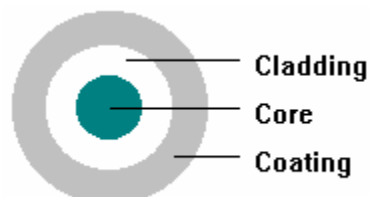
Optical networks use fiber optic cable to transmit data. This type of cable contains a bundle of glass or plastic threads, called fibers, each of which is capable of transmitting messages modulated into light waves. Fiber optics provide a much greater bandwidth (the amount of data that can be transmitted in a fixed amount of time) than metal cables. They are also thinner and lighter, and less susceptible to interference. In optical networks, light pulses are used to transmit information down fiber lines instead of using electronic pulses to transmit information down copper lines. Data can be transmitted digitally, unlike the electronic pulses on copper lines.

To generate these light pulses, there is a transmitter on one end of a fiber line. The transmitter converts an electrical analog into a corresponding optical signal. The source of the optical signal will be a LED when using multimode fiber, or a laser when using single-mode fiber (see 2.3). Both produce infrared light, invisible to the human eye.

At the other end of the fiber, there is a receiver. The receiver can be compared to a photoelectric cell in a solar powered calculator: light received is converted into electricity. It converts the light pulse back into the original electrical signal which first entered the transmitter at the other end of the fiber. When the signal is once again transformed into voltage changes, it can be sent over copper wire to a computer, a switch, a router, etc.

A fiber consists of:

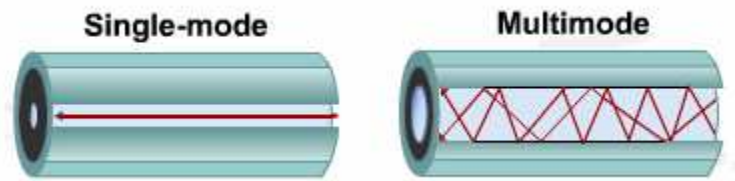
- The core: this is the light transmission area of the fiber, either glass or plastic.
- The cladding: a reflective material, to cause reflection within the core so that light waves are transmitted through the fiber.
- The coating: usually multi-layers of plastic applied to preserve fiber strength, absorb shock and provide extra fiber protection.



2.3. Multimode and single-mode fiber

Multimode fiber is optical fiber that is designed to carry multiple light rays (or modes), each at a slightly different reflection angle within the core. Multimode fiber transmission is used for relatively short distances because the modes tend to disperse over longer lengths.

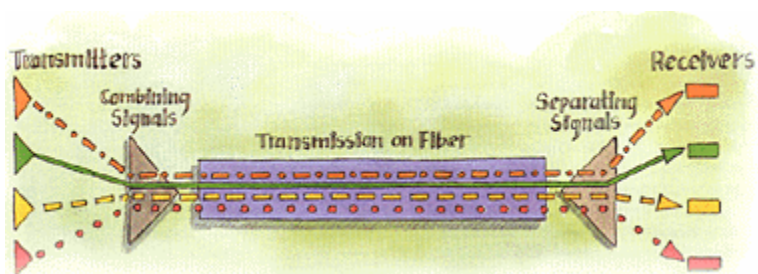
Single-mode fiber is optical fiber that is designed for the transmission of a single ray (mode) of light and is used for long distance signal transmission. It consists of the same parts as multimode fiber, but the light ray travels through a smaller core.



2.4. Wavelength Division Multiplexing

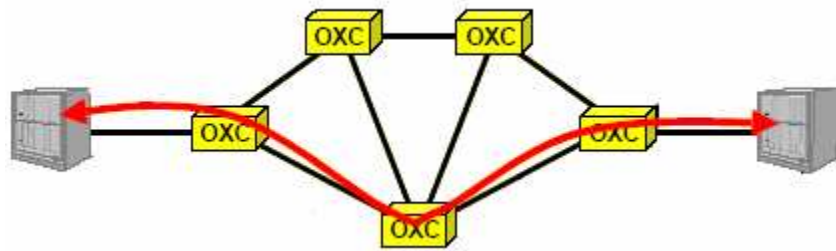
An optical cable typically consists of a number of individual fibers, where 4, 12, 24 and 48 or more fibers in a cable are common. It would however not be very cost-effective to install a new optical cable with even more fibers, when all the installed fibers have been used. It would be more interesting to put more signals on the same fiber instead. There are different ways to transport more data on a single fiber. One would be Time Division Multiplexing (TDM), where many signals of the same type are multiplexed together electrically before they are converted into light pulses and transmitted. Another technique is Wavelength Division Multiplexing (WDM). WDM bundles several colors (invisible to the human eye) together on a fiber, each of them representing a signal.

WDM basically works like this: there are several transmitters, each generating laser light with a slightly different color (wavelength). These colors are sent into a prism. The prism combines the waves into a single beam. At the other end of the fiber there's another prism, which separates the single beam back into the original colors. It works just like separating the white light from the sun into a spectrum of colors.



2.5. Optical Cross Connects

On one single fiber, there can be several lambdas. A lambda is a wavelength channel, which distinguishes itself by the color of light. Furthermore, such a network exists of Optical Cross Connects (OXC). OXCs are devices that can switch optical signals between different optical fibers, without the need for conversion to electrical signals. They are comparable to software controlled switches, which connect two wavelength channels. If the OXCs are well configured, a light path can be established between any two nodes, by connecting several wavelength channels.



3. Evolution of optical networking

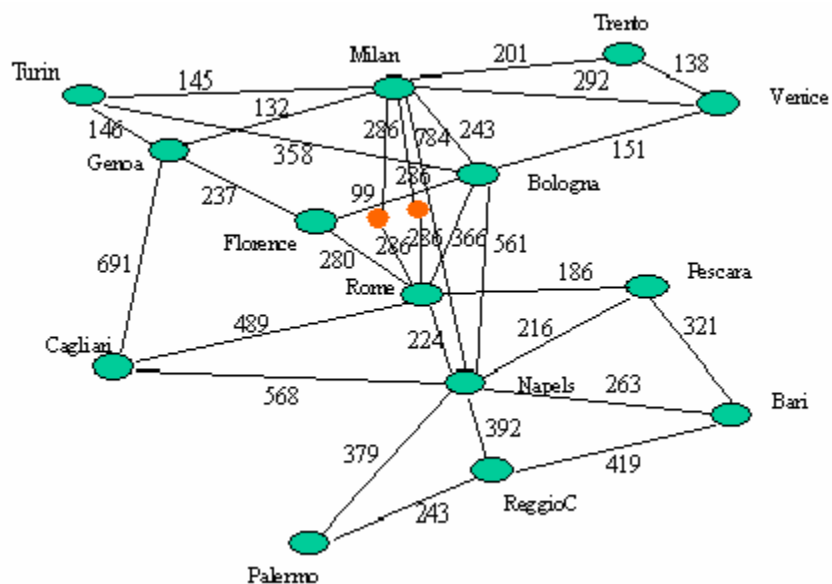
3.1. Intelligent optical networks

The popularity of the Internet also causes another problem: the traffic patterns are very dynamic, they change often. Therefore it is important for an optical network to react quickly to these changes. And that is exactly the problem nowadays: due to the manual intervention of the network operator, it can take up to a few weeks, or even months, to provision a light path. Nowadays, research is being done on the development of Intelligent Optical Networks (IONs), in order to solve this problem. The purpose of Intelligent Optical Networks is to bypass the manual intervention by providing a distributed control plane able to automatically process requests from the clients to establish light paths.

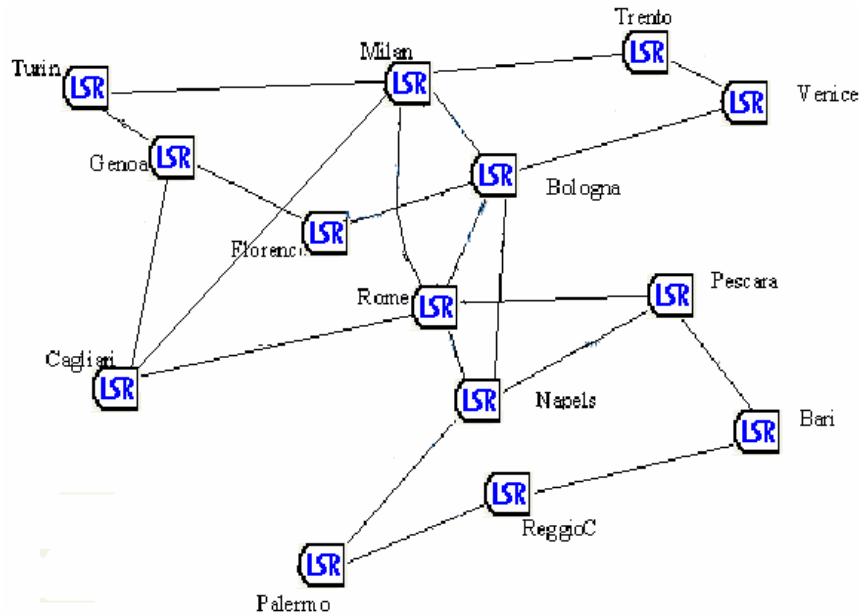
The ITU has developed a standard for these IONs: ASON (Automatic Switched Optical Network)

3.2. Multilayer networks

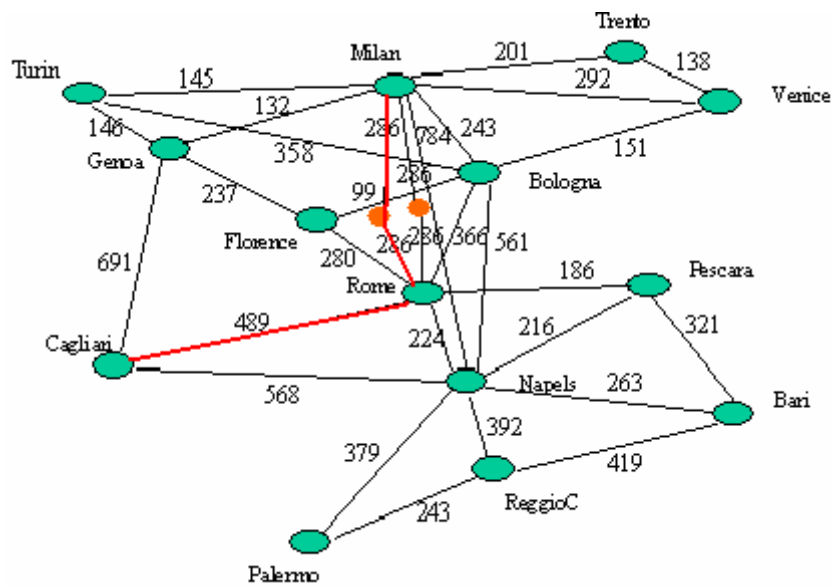
This takes us to multilayer networks. The use of ION functionality in an IP-over-OTN (Optical Transport Network) allows us to dynamically reconfigure the logical IP network when needed. Logical links can be added where the capacity of existing links seems insufficient, and they can be removed when they are barely used or not used at all. For every logical link, there is a light path in the optical layer. The logical topology is virtual: it contains real routers, but there are no actual cables or any other kind of links between these. Two nodes in this layer can be directly connected, even though there exist no direct links in the physical layer. In this case, they will cross several OXCs. For example, during my research, I used the following physical topology:



It is however perfectly possible to create the following topology in the logical layer:



The light path to provide the link between Milan and Cagliari can run like this:



3.3. Traffic Engineering and MPLS

Simple routing strategies may cause congestion in parts of the network, while other parts remain underused. One of the goals of Traffic Engineering (TE) is route traffic over the network in such a way that the network load is divided more evenly and to reach a certain

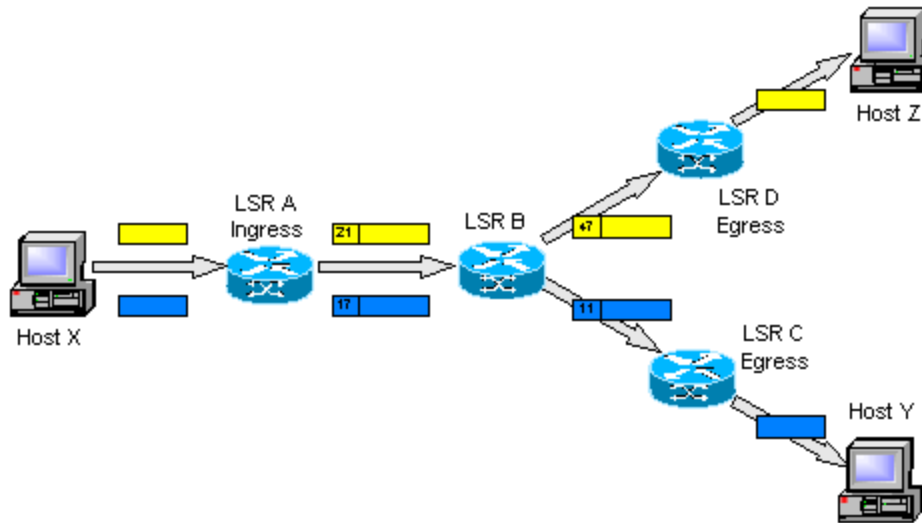
Quality of Service (QoS). The QoS depends on the purpose of the network and the priorities of its users. In some cases, the aim is optimal use of all resources (bandwidth on logical links), even if this possibly slows down the network. Existing light paths will then be used whenever possible.

In other cases it is more important to constantly provide a minimum of capacity, even if this causes resources to be wasted.

MPLS (Multi-Protocol Label Switching) is a protocol independent system for Traffic Engineering. It was once developed to offer faster packet forwarding than traditional IP routing, but because of the progress of router hardware, its speed has become less important. It was the flexibility of MPLS which made it a standard for modern networks to reach QoS and for optical networks.

Traditional IP networks are connectionless. When a packet is received, the router determines the next hop using the destination IP address on the packet alongside information from its own forwarding table. The router does not know which routers the packet will travel next. Packets which belong to the same traffic flow do not necessarily take the same route. MPLS flows on the other hand, are connection-oriented and packets are routed along pre-configured Label Switched Paths (LSPs). Routing is done by Label Switched Routers (LSRs).

When MPLS is used on an IP-network, the first LSR looks at the IP address of the destination. It then selects the LSP and tags the packet with a label in order to identify the LSP. When the next router receives the packet, it uses this label to identify the LSP to which the packet belongs. It then looks up the LSP in its own forwarding table to determine the best link over which to forward the packet, and the label to use on this next hop. At each LSR it passes, the packet receives a new label. A different label is used for each hop, chosen by the router performing the forwarding operation. This allows the use of very fast and simple forwarding engines, as the router can select the label to minimize processing. If each LSP would receive its own fixed label, it would be difficult to guarantee that the label is unique on each LSR. Giving the labels a unique numbering in a distributed environment and making sure they are unique for every router in the network, would be a lot more complex than to just make sure the numbers are unique on one router. Therefore, the routers themselves choose the labels, only having to take into account which labels are in use by themselves. This is not so complex, as they can easily look this up in a so called "swapping table" containing the numbers used for labeling. The disadvantage is that the labels have to be changed every time, on every router. The LSR connected to the destination (or destination network) removes the label and routes the packet out of the MPLS network.

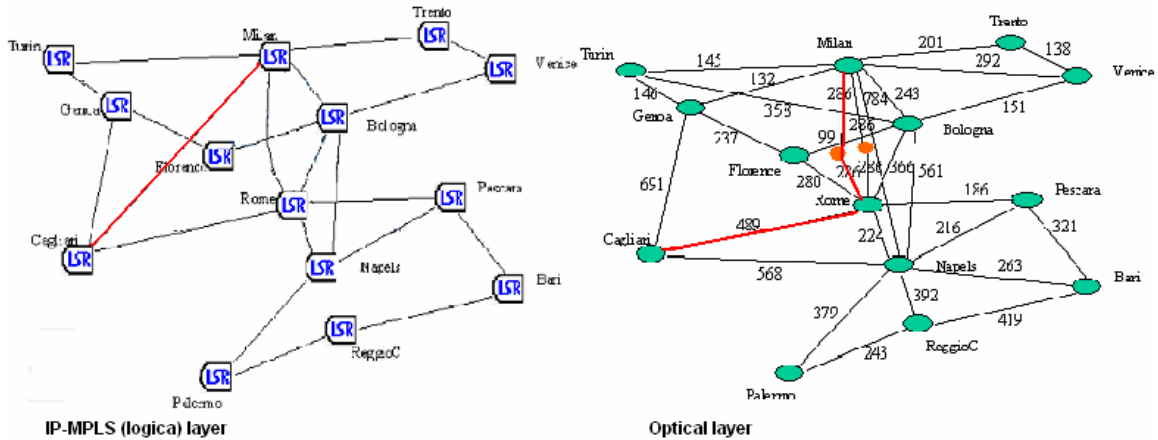


MPLS only defines the forwarding mechanism and needs two other protocols to establish the LSPs: a routing protocol and a signaling protocol. As a routing protocol, an interior gateway protocol such as OSPF (Open Shortest Path First) or IS-IS (Intermediate System-to-Intermediate System) is commonly used. Normally, these protocols only distribute the network topology, but Traffic Engineering extensions to these protocols can be used, which make them able to also distribute the QoS requirements, so that routers can take these into account. The signaling protocol informs the LSRs which labels and links to use for each LSP. When Traffic Engineering is required, RSVP-TE (Resource Reservation Protocol) can be used, or CR-LDP (Constraint-based Label Distribution Protocol). When no Traffic Engineering is needed, LDP (Label Distribution Protocol) can be used.

3.4. Multilayer Traffic Engineering

If congestions occur in the logical layer, traffic can be rerouted with regular Traffic Engineering in the IP-MPLS layer. For optical networks, this will be the logical layer (hence virtual). For example, if the aim is to keep the load on all links under 70%, TE would route a new traffic flow of 30% in such a way that it does not use any links with a load of more than 40%. This may not be sufficient, due to capacity shortage in the logical layer.

A solution is to reconfigure the logical topology, by setting up or tearing down light paths in the optical layer. This is called Multilayer Traffic Engineering (MTE). Often, there are several possible paths in the optical layer to set up a light path. For the light path in the following example, to establish a link between Milan and Cagliari, one alternative would be to run through Turin and Genoa, instead of Rome.



When to reconfigure the logical network depends on events on the network and the chosen strategy. Reaching a certain QoS is not the only thing to keep in mind: it is also important to ensure the network is not being reconfigured too often, as this may cause interruptions and instability. Because of that, it is important to choose a good window size for measuring the load on the network. The window size is the period of time in during which received packets and their size are being measured. (More about window size in 4.2.3)

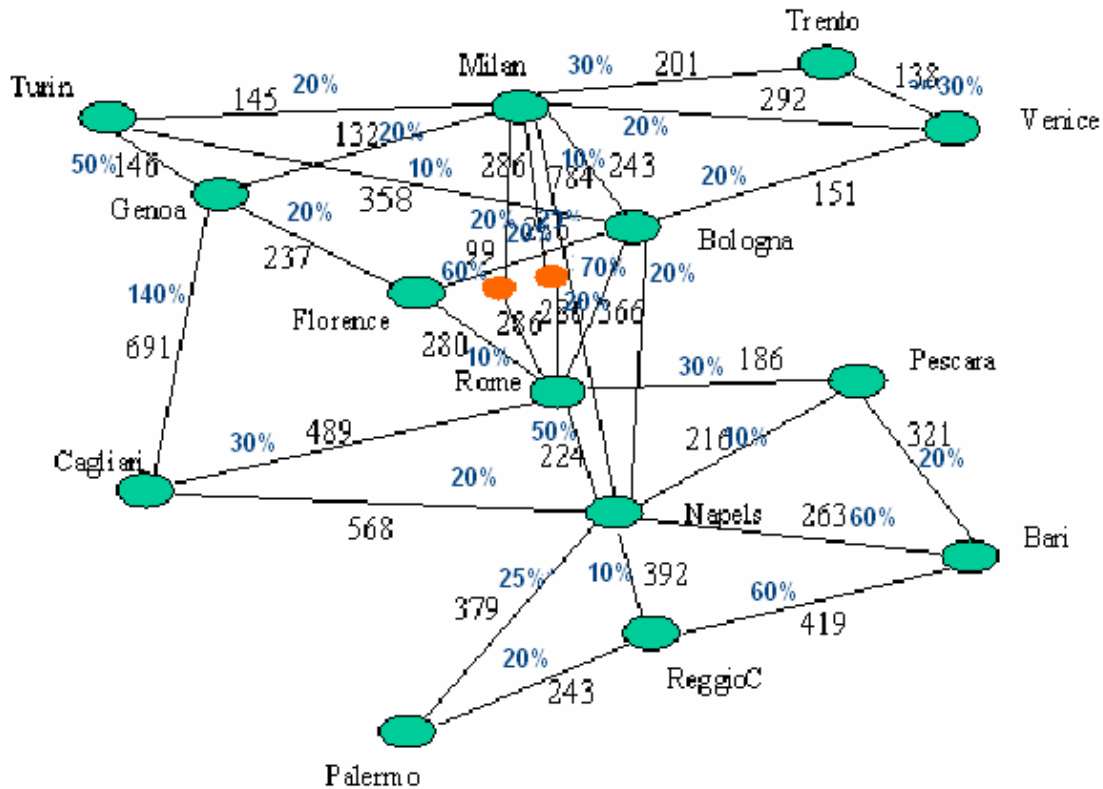
Another difficulty is to avoid that links become overloaded, while also making sure not to provide too many superfluous light paths which may remain underused for most of the time.

We can distinguish two kinds of strategies: proactive and reactive. In the reactive approach, one only intervenes when a problem has occurred, while proactive strategies are designed to try and avoid problems. Proactive strategies would be most effective when future traffic and changes can be predicted. This is not very realistic with Internet traffic.

4. Analysis of MTE strategies

4.1. The network topology

The topology onto which I applied some MTE strategies (albeit in theory for now), looks like this:



The black lines are the physical links, the fiber optic cables. The numbers in black are the distances and the percentages in blue represent the initial traffic load on the links.

Between Genoa and Cagliari, you see a traffic load of 140%. This already points out the benefits of IONs: imagine that a high bandwidth use like this arises very unexpectedly, and manual intervention is needed to provide an extra light path. Congestion would occur, and remain existant until this has happened. Queues would become longer, data would be buffered (which is an extra ballast for the routers) and data may get lost. Even in situations where it is possible to predict situations of sudden high bandwidth usage like this, but generally there is a lower bandwidth usage on this link, there would be a problem: it is possible to provide two light paths between Genoa and Cagliari, but most of the time, when bandwidth usage is normal, this extra resource (light path) would be wasted.

We assume there will be mainly Internet traffic on this network, which implicates that the traffic pattern and the load are hard to predict. The links are bidirectional and the

bandwidth of each wavelength channel (λ) is 2.5 Gbps. As each link has the same amount of bandwidth, the usage of an LSP will be indicated as percentages in the rest of this document (100% is the maximum load of a link).

4.2. A reactive strategy

4.2.1. The strategy

Because Internet traffic is hard to predict, a reactive strategy can be interesting, as it only takes action after certain situations (such as congestion) have been detected. Of course, the disadvantage is that problems are only solved after they have occurred, instead of avoiding them. It is also undesirable that one change in the logical topology trigger many others, causing too many reconfigurations and hence interruptions.

When additional traffic is created between two routers, and somewhere in between a router discovers that one of its outgoing links is getting congested, it has to suggest the routers causing the additional traffic to setup a new link in the logical layer, which results in a new light path in the physical layer. The cost of this link should be set to a lower number than the cost of other routes to the same destination, in order to attract traffic to the new link. Traffic on underused links should (if possible) be rerouted over another link, so that underused links can be removed.

When T_{cong} (load) on a link is higher than T_{high} (maximum load on a link), a new link will be setup, and when T_{cong} on a link is higher than T_{low} (minimum load on a link), the link will be removed and the traffic will be rerouted. It is important to choose the right values for T_{high} and T_{low} , so that a small increase of traffic on a link does not trigger reconfigurations, and so that one reconfiguration does not cause too many others.

The values of T_{high} and T_{low} depend on the required QoS. Here we want to avoid links which are very poorly used, but also avoid too high loads on a link, to prevent buffering and possible loss of packets.

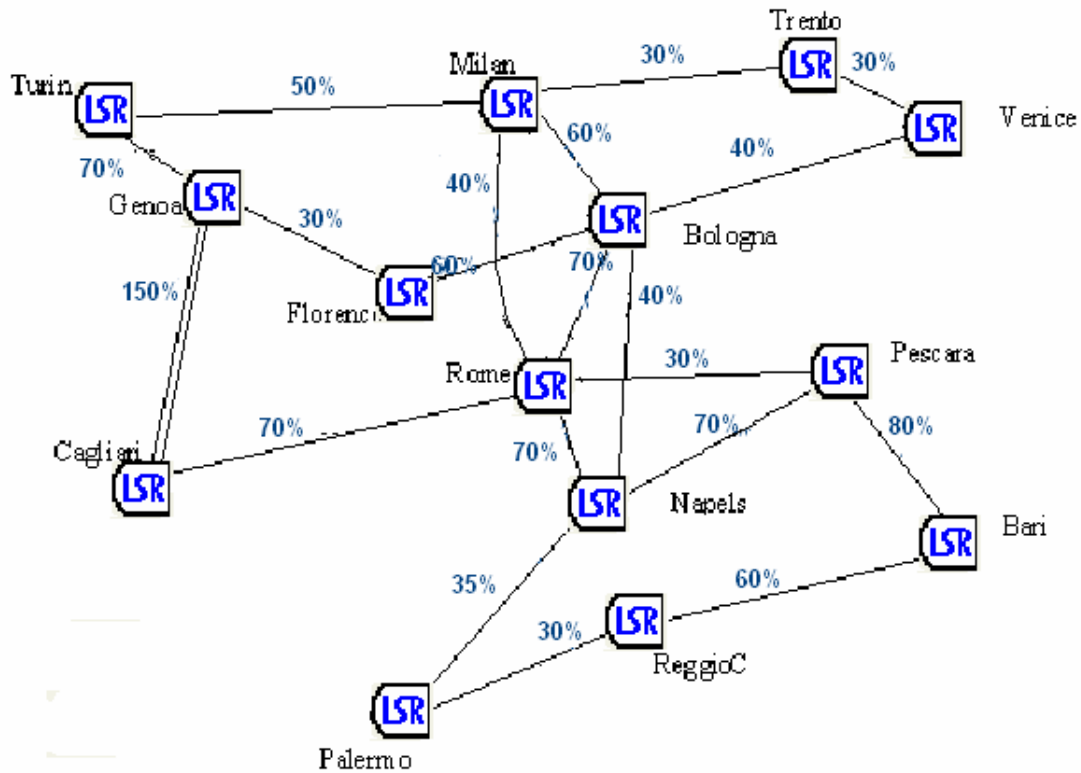
The following values seem reasonable:

- $T_{\text{high}} = 80\%$
- $T_{\text{low}} = 30\%$

T_{mid} (the ideal average load) is then 55%.

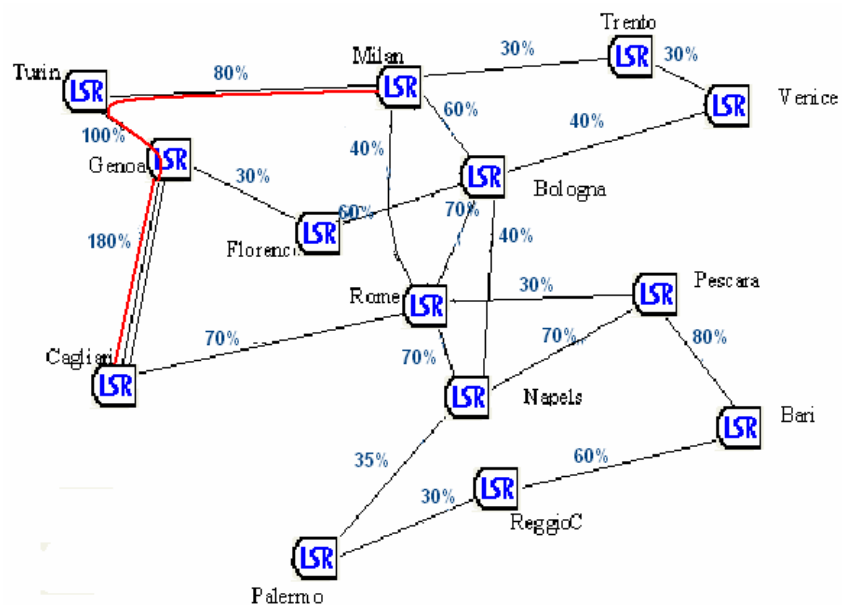
4.2.2. Analysis by means of an example scenario

To meet these conditions, the initial logical topology cannot simply follow the physical topology. To start, it is devised as follows:



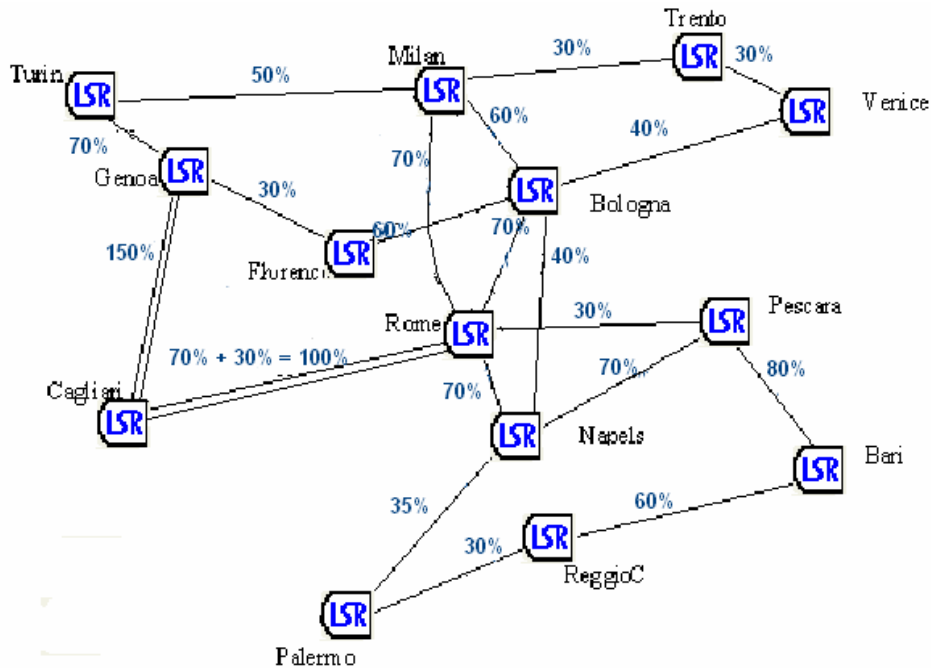
Genoa – Cagliari, with a load of 150%, also meets these conditions as it uses two links with a load of 75% each.

Now let's say an additional traffic flow of 30% from Milan to Cagliari is being established as follows:

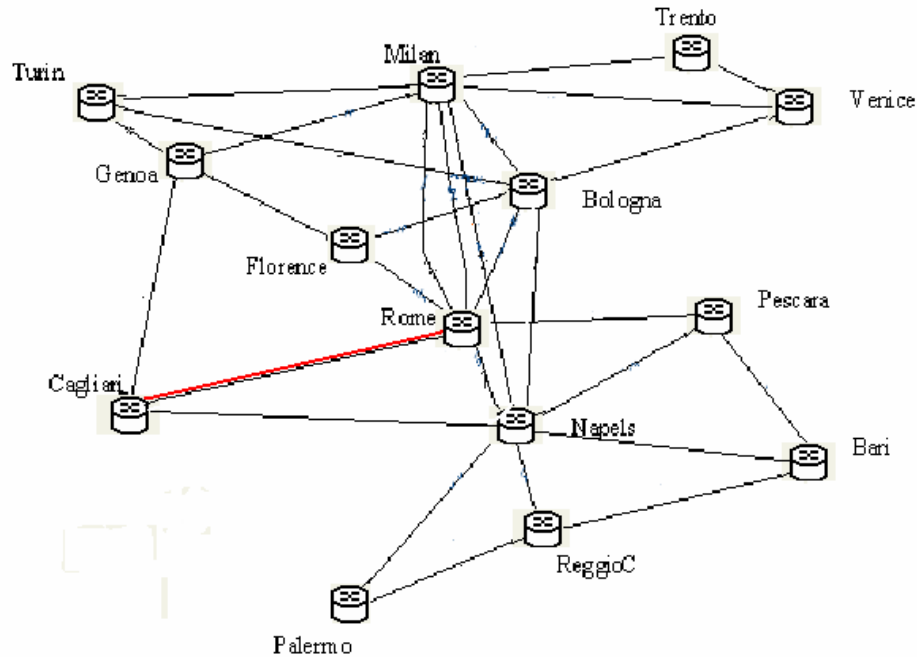


$T_{cong} > T_{high}$ between Turin and Genoa as well as between Genoa and Cagliari.

Turin will notice the congestion on the outgoing link to Genoa, and suggests that new logical link should be established. A direct link could be setup between Milan and Cagliari, but as there is still bandwidth left on the link from Milan to Rome, we only need a new link from Rome to Cagliari. The cost of this new link will be set lower than the other ones, so that traffic will be attracted to this new link.

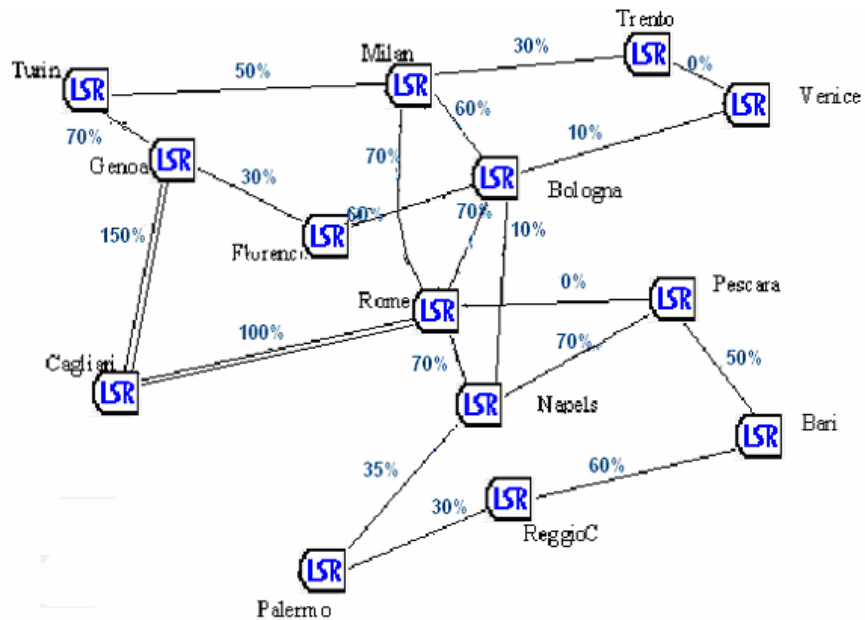


The network will use RSVP-TE as a signaling protocol for light paths, to allow the reservation of network resources. The light path in the optical layer will, in the first place, take the shortest path, but should, when there are several options with the same hop count, be established on the least congested path. However, here we have only one direct path. Hence, the light path will run like this in the optical layer:

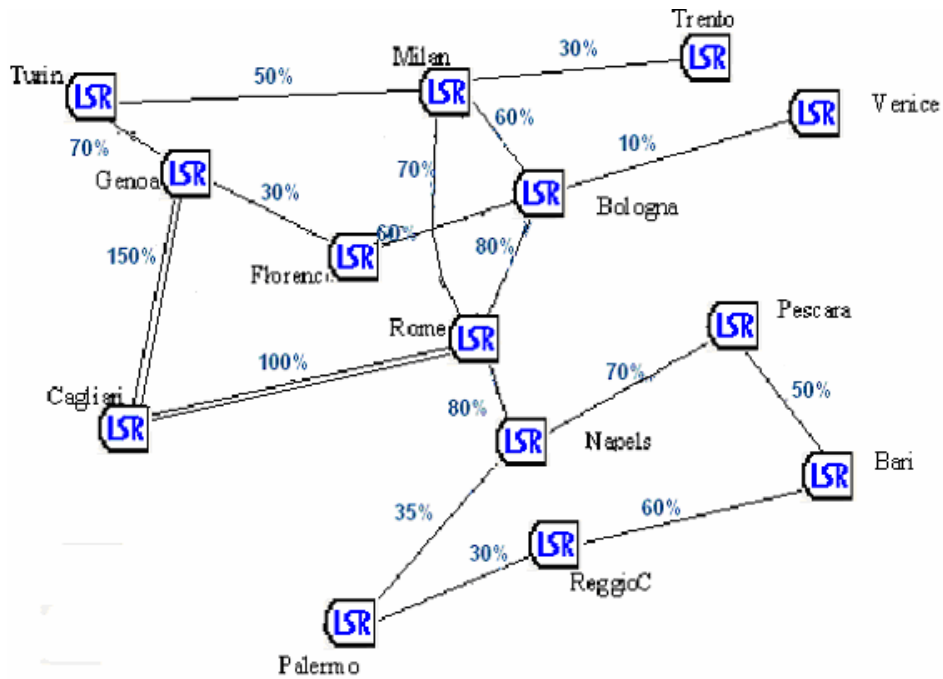


Until now, this strategy seems to work quite well. No links are underused or congested, and the average link load is 53,41%.

Next, we omit some existing traffic flows from Bari to Rome and from Naples to Trento, causing the load to decrease with 30% like this:

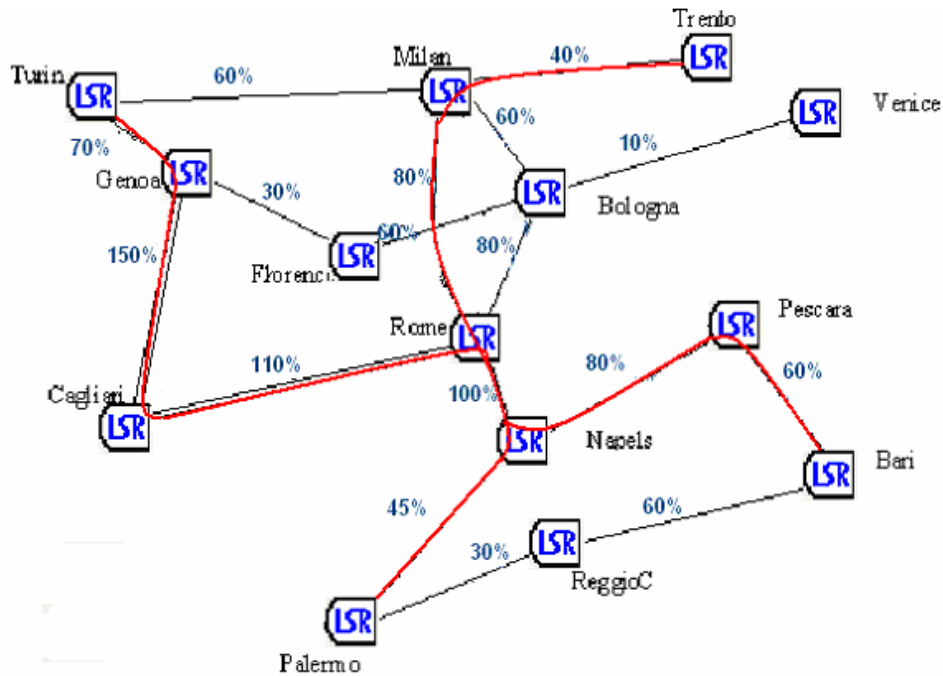


Several links are now underused and the average link load only amounts to 33,86%, which is far beneath T_{mid} . Underused links are removed where possible and the remaining traffic is rerouted. The result is the following:

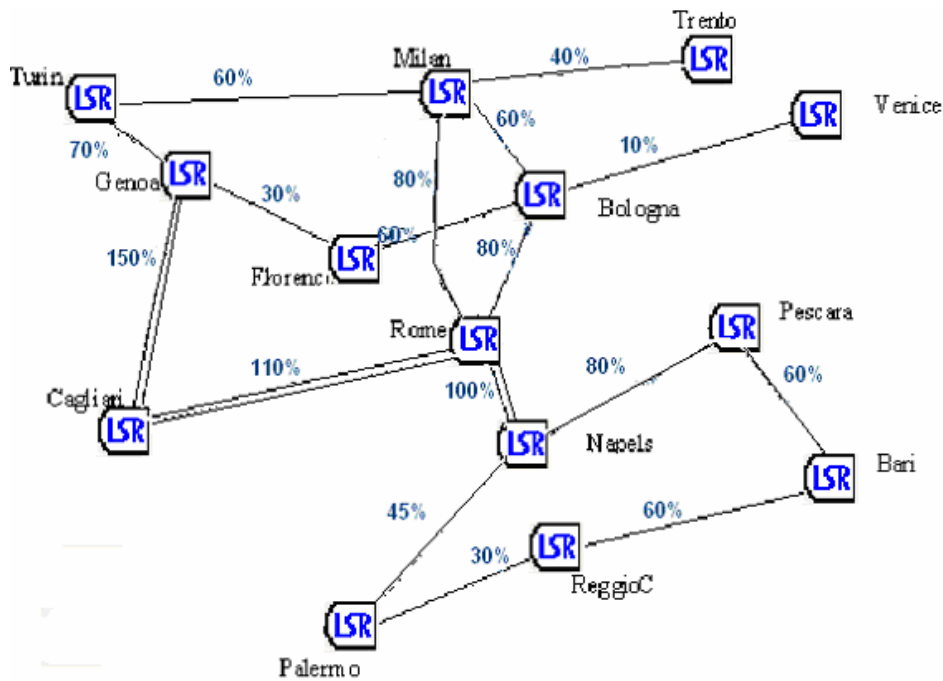


The average link load is now 54,47%. In spite of this, there remains one heavily underused link between Bologna and Venice, and two links which are being pushed to their limits (Rome – Bologna and Napels – Rome), with 80%.

Now, we establish two new LSPs with a load of 10%: between Turin and Palermo, and from Trento to Bari, this causes the link Rome – Napels to become congested:



The common path for both traffic flows is Rome – Napels. As the need for bandwidth is not even higher than T_{low} for these traffic flows, and they cause no other links to become congested, it would not be beneficial to setup direct light paths for these. In this case, it is better to setup a second link between Rome and Napels. The shortest physical path for this is directly from Rome to Napels. The logical topology then looks like this:



The average link load is now 56,25%.

4.2.3. Window size

The load on the logical links is calculated by the LSRs. In practice, the traffic patterns change constantly and the need for bandwidth can turn rapidly. Adjusting the logical topology for or very brief network overloads is not a good idea, as it only causes instability. That is why it is important to leave a well chosen amount of time between these calculations.

The window size is the period during which the number of packets received and their size are being measured, before the traffic demand (average for this period) is calculated with this formula:

$$D = \frac{S * N_p}{T_{ow}}$$

- D: traffic demand
- S: size of the packet
- N_p: number of packets received during the observation window
- T_{ow}: size of the observation window

To improve the QoS, we need to choose a good observation window size. If the window size is small (0,1s to 0,5s), it will harm the network stability, and if it is large (3s and more) then the calculated traffic demand is not accurate enough, which will harm the network reliability. This could be a problem for example when a link is congested most of the time, but the traffic flow causing this congestion drops every so often. If the link load is extremely low at these moments, the calculated average traffic demand may seem ideal. If this situation persists, the network may never be reconfigured at all, which renders the entire strategy useless. Large window sizes may be useful on networks with a rather stable network load. However, this is not the case with this network topology, transporting mostly Internet traffic, which is rather unpredictable. In order not to cause reconfigurations of the logical topology for very small traffic bursts, which may occur with Internet traffic, the window size should not be too small either. A window size of 1s could be a good compromise. This is of course difficult to test without installing and using the network in practice.

4.2.4. Conclusion

The general performance of this strategy seems good, as the average link load after applying the strategy is always near 55%, which was intended. The downside is that there are still remain underused links. There is however few we can do about this, when the

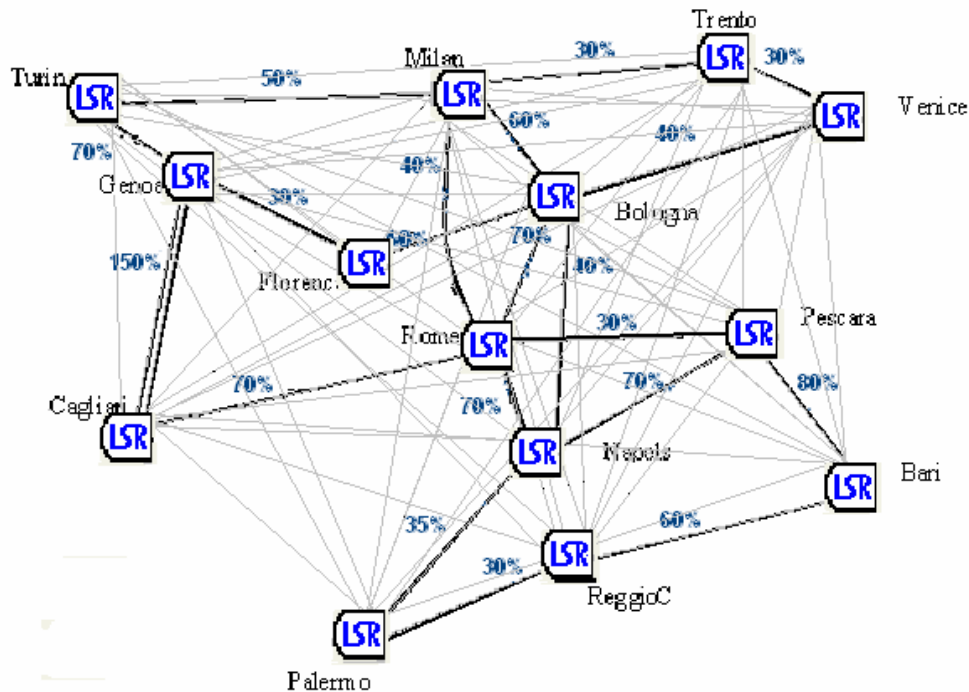
traffic cannot be rerouted. A disadvantageous consequence with any reactive strategy is of course that problems are not being prevented, but solved instead.

4.3. A proactive strategy

4.3.1. The strategy

Instead of waiting until optimizations are triggered by events (congestion, underused links), like with the reactive strategy, we could also use a proactive strategy, which continuously tries to optimize the routing.

First, we assume that all traffic is routed over a fictive full-mesh logical network, along the shortest path. This does not mean that all of these light paths actually have to be available.



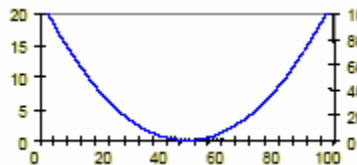
The lines in gray represent the fictive full-mesh. The black lines are the links which already have a light path. For this I used the same topology as I used in the previous strategy, as it suits the conditions (same as in the reactive strategy):

We want the load on links to be between 30% and 80%, and the average link load we want to achieve is 55%.

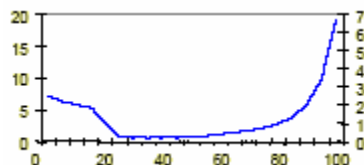
The cost to use a link depends on the load of that link, but also on the flow that is being routed, because if the load of the new flow would be too high, this could lead to less optimal situations. Hence, costs have to be recalculated for every flow routed. In general,

highly loaded links should be assigned a high cost, as well as links in the fictive topology which do not yet exist as light paths. Average loaded links which can still take some extra traffic should be given a lower cost. Lightly used links should also be assigned a higher cost, to avoid that many of them will be established.

The most intuitive idea would be to use a parabolic cost function. A parabolic function may for example look like this when graphically represented:



However, previous research has pointed out that a cost function that differs from a real parabolic function by an elevated plateau for very lightly loaded links a quite wide plateau with a low cost for reasonable loaded links and a very high cost for heavy loaded links gives the best results. A parabolic function may look like this:



A function that may work for all links with a link load that is higher than the Light Load threshold (30%) and that will not become congested when routing the new flow would be the following:

$$\text{Cost} = (\text{Link Load Percentage} + \text{Traffic Demand Percentage}) / 2$$

- For example: $\text{Cost} = (32 + 15) / 2 = 47 / 2 = 23,5$
- Another example, where the route would be less preferable: $\text{Cost} = (55 + 20) / 2 = 75 / 2 = 37,5$

In order to get plateaus, I would round the results of the previous formula downwards by 10 like this:

- $\text{Cost} = (32 + 15) / 2 = 47 / 2 = 23,5 \rightarrow 20$
- $\text{Cost} = (55 + 20) / 2 = 75 / 2 = 37,5 \rightarrow 30$

The lowest possible cost would be 10, which would therefore be the ideal route to choose.

The highest cost we want to allow on a route is 80%. The cost of a route which can not take any more traffic would be at least (if the traffic load were 1) 40:

- $\text{Cost} = (80 + 1) / 2 = 81 / 2 = 40,5$

To get plateaus, we will round up the cost of all links which would become overloaded if they would transport the new traffic flow. For example, 40,5 becomes 50, 55 becomes 60, etc. We want them to be very high however, as otherwise the sum of the costs of a route consisting out of several, average loaded links, may be higher than cost of an overloaded one. So we add 150:

- $\text{Cost} = (80 + 1) / 2 = 81 / 2 = 40,5 \rightarrow 50 + 150 = 200$

As we want to prevent any more traffic on these routes, the cost of links in the fictive full-mesh which do not yet exist as light paths, should have a lower cost. We'll assign them a cost of 81, plus 21 for each extra hop. These numbers have been calculated to avoid that direct links crossing multiple routers would be preferred over shorter links which have a higher chance of being used by other flows as well. For example:

- $\text{Cost} = 81 + 21 + 21 = 123$

Using links with a load below the Light Load threshold (30%) would be uninteresting unless the new traffic flow has a rather high load. We'll assign them a cost of 100, minus the sum of the current link load and the traffic demand, and we then round down. An example of the cost formula would be:

- $\text{Cost} = 100 - (25+13) = 100 - 38 = 62 \rightarrow 60$

This causes that, in case the new traffic flow is high enough, using an underused link becomes more interesting.

The numbers can be rounded (the normal way) when needed.

We get the following algorithm to calculate the cost of a link:

```
int getLinkCost()
{
    float llp, tdp, T_high, T_low, tmp;
    int cost, extra_hops;
    bool exists;

    // llp = Link Load Percentage
    // tdp = Traffic Demand Percentage
    // exists = existant light path or not
    // extra_hops = OXCs crossed between two end points setting up a new light path
    // T_high = maximum allowed link load, 80% in this scenario
    // T_low = minimum link load aimed for

    if(exists)
    {
        if(llp < T_low)
        {
            // If the link load is less than T_high, we are talking about an underused link. We
            // assign them a cost of 100, minus the sum of the current link load and the traffic
```

```

// demand, and we then round down.

    tmp = 100 - (llp + tdp);
    tmp/=10;          // We want to round by 10, but the floor function rounds
                    // decimal numbers. So we divide the number by 10, to
                    // multiply it again later.
    cost = (int)(floor(tmp)*10);
}
else if((llp + tdp) > T_high)
{

// If the load after establishing the new traffic flow would be higher than T_high
// this would become an overloaded link. We want to assign it a very high cost.
// We take the sum of the Link Load Percentage and the Traffic Demand
// Percentage, divide it by 2, round it upwards by 10 and then add 150, to get the
// link cost.

    tmp = (llp + tdp) / 2;
    tmp/=10;
    cost = (int)((ceil(tmp)*10)+150);
}
else
{

// In all other cases, we are talking about a normally used link, which should be
// assigned a reasonable cost.
// We take the sum of the Link Load Percentage and the Traffic Demand
// Percentage, divide it by 2, and round this downwards by 10, to get the cost.

    tmp = (llp + tdp) / 2;
    tmp/=10;
    cost = (int)(floor(tmp)*10);
}
}
else
    cost = 81 + (extra_hops * 21);

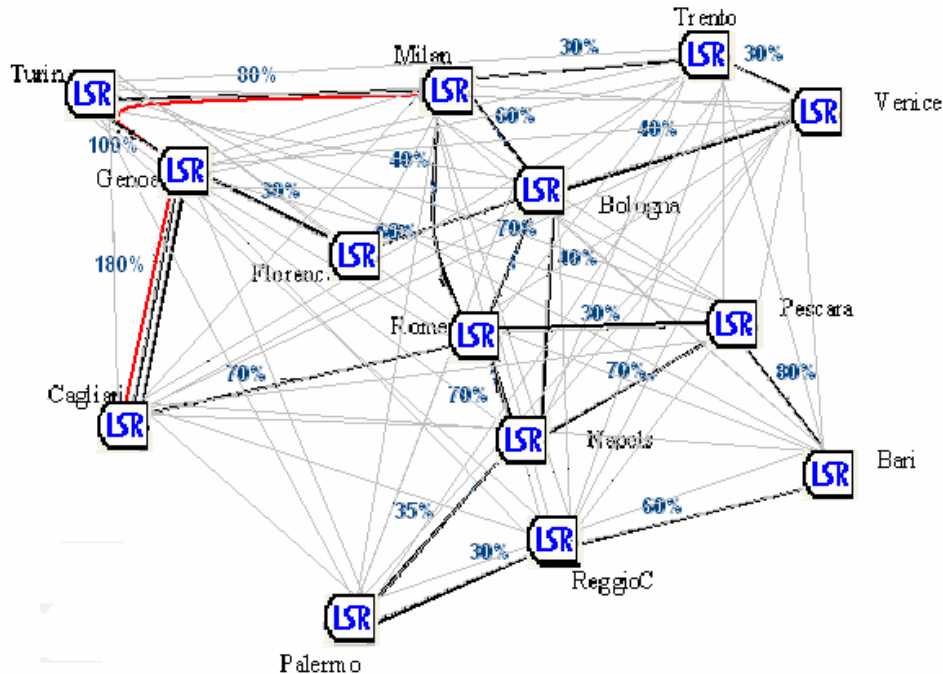
return cost;
}

```

4.3.2. Analysis by means of an example scenario

In order to be able to compare, I'll use the same scenario as for the reactive strategy. The difference is that now, instead of happening and being corrected, congestions will be prevented.

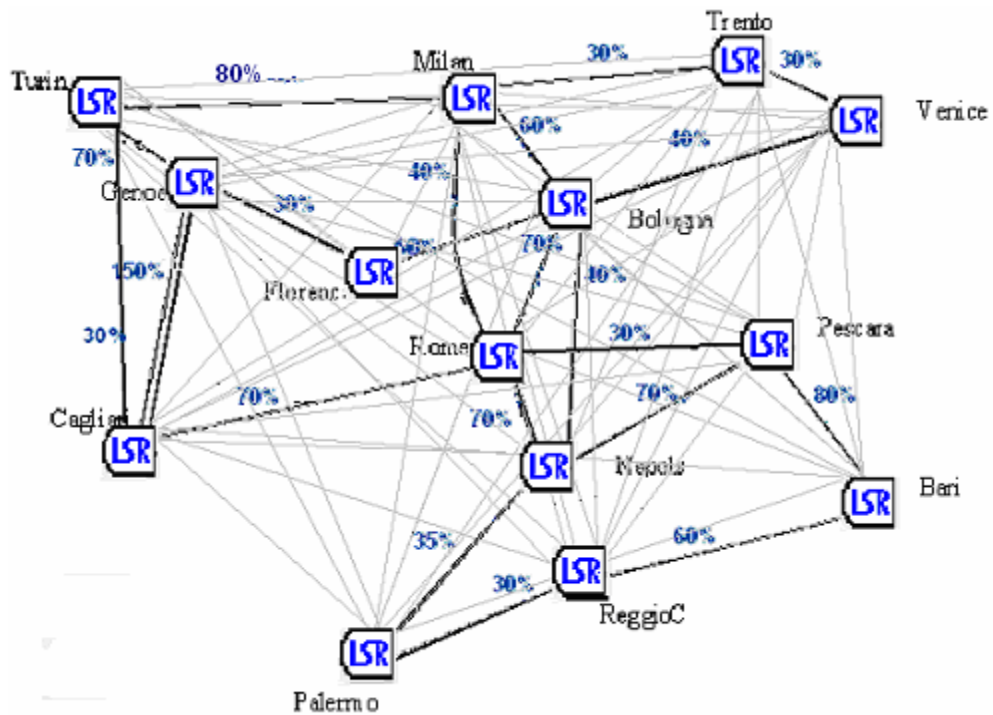
So first, there is an additional traffic flow of 30% from Milan to Cagliari. Without traffic engineering, the following would happen:



Two links would become congested. However, with this MTE strategy, the new traffic will take another route, based on their cost. There are several routes from Milan to Cagliari:

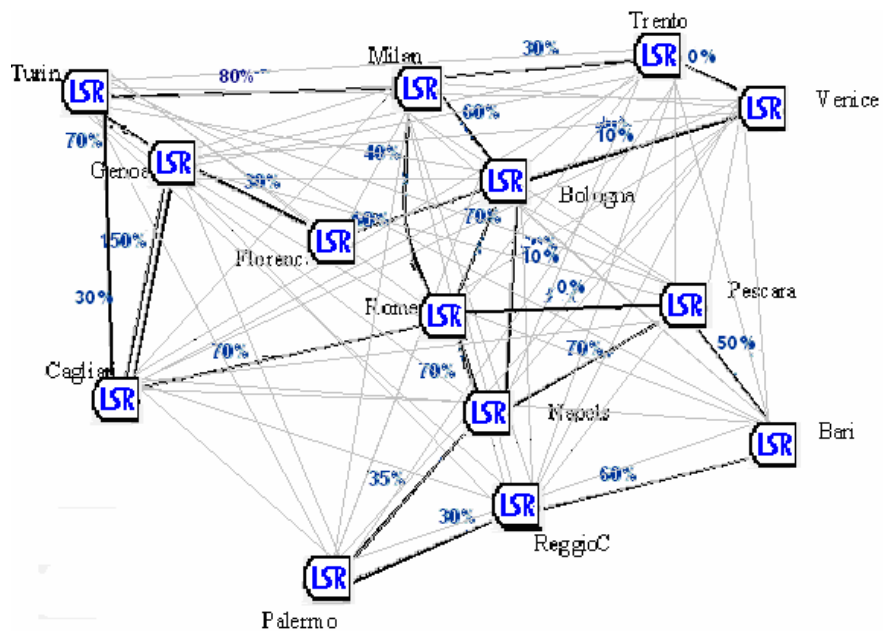
- A direct link, which does not yet exist as a light path (crossing Bologna and Rome):
 - Cost = 81 + 21 + 21 = 123
- Through Turin and Cagliari:
 - Cost for Milan – Turin = $(50 + 30) / 2 = 80 / 2 = 40$
 - Cost for Turin – Genoa = $(70 + 30) / 2 = 100 / 2 = 50 \rightarrow 50 + 150 = 200$
 - Cost for Genoa – Cagliari = $(75 + 30) / 2 = 105 / 2 = 52.5 \rightarrow 60 + 150 = 210$
 - Total cost: $40 + 200 + 210 = 450$
- Through Rome:
 - Cost for Milan – Rome = $(40 + 30) / 2 = 70 / 2 = 35$
 - Cost for Rome – Cagliari = $(70 + 30) / 2 = 100 / 2 = 50 \rightarrow 50 + 150 = 200$
 - Total cost: $35 + 200 = 235$
- Through Turin, with a link that does not yet exist as a light path to Cagliari:
 - Cost for Milan – Turin = $(50 + 30) / 2 = 80 / 2 = 40$
 - Cost for Turin – Cagliari = 81
 - Total cost: 121

So, the route which will be chosen is the last one.



The average link load is now 53,41%.

Next, the traffic demand from Bari to Rome and from Napels to Trento decreases with 30%:

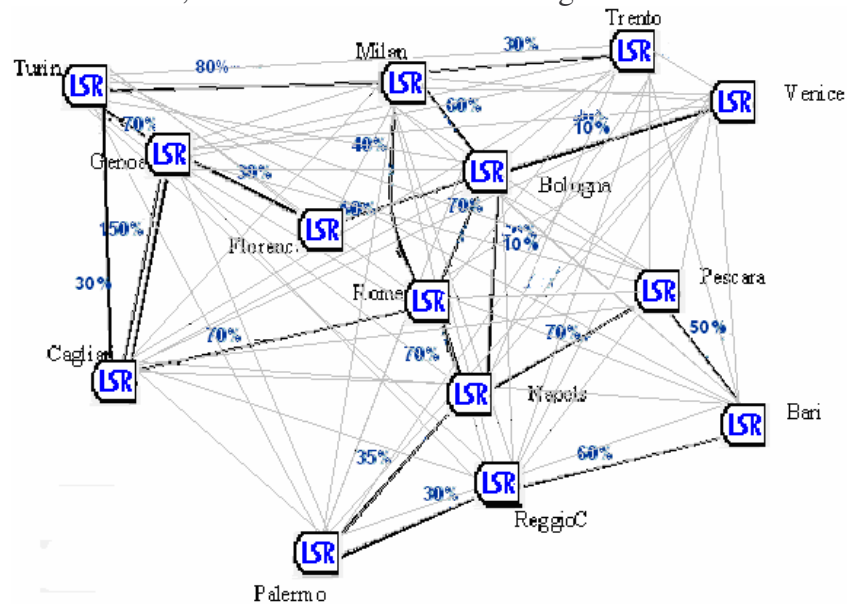


Some links will become underused:

- Rome – Pescara: 0 %

- Bologna – Napels: 10%
- Bologna – Venice: 10%
- Venice – Trento: 0%

Rome – Pescara and Venice – Trento will be torn down. Here, we do not reroute any traffic. However, the cost for both Bologna – Napels and Bologna – Venice will be very high for new traffic flows, unless the new flow has a high bandwidth demand.



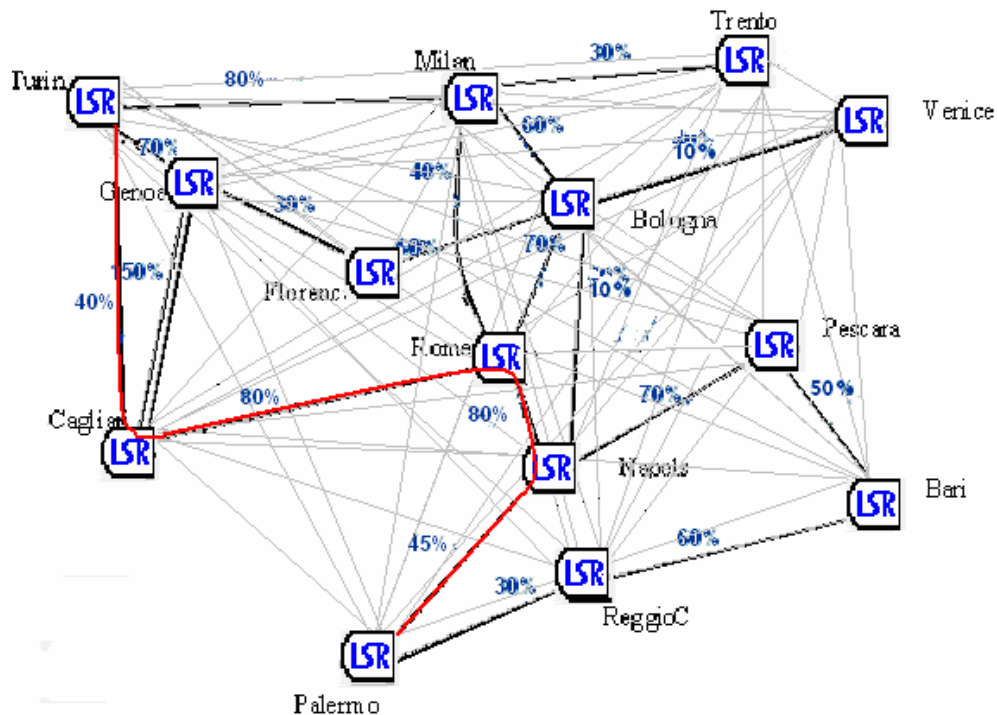
The average link load is now 51,25%.

Now, the traffic flows from Turin to Palermo and from Trento to Bari both increase with 10%. Instead of rerouting the flows after they have possibly caused problems, like with the reactive strategy, they will be drawn to the route with the lowest cost. Possible paths to take for the flow from Turin to Palermo are:

- Through Cagliari, Rome and Napels:
 - Cost for Turin - Rome = $(30 + 10) / 2 = 50/2 = 25 \rightarrow 20$
 - Cost for Cagliari – Rome = $(70 + 10) / 2 = 80/2 = 40$
 - Cost for Rome – Napels = $(70 + 10) / 2 = 80/2 = 40$
 - Cost for Napels – Palermo = $(35 + 10) = 45/2 = 22.5 \rightarrow 20$
 - Total cost = $20 + 40 + 40 + 20 = 120$
- Through Milan, Rome and Napels:
 - Cost for Turin – Milan (link which cannot take any more traffic): $(80 + 10) / 2 = 90/2 = 45 \rightarrow 50 + 150 = 200$
 - Cost for Milan – Rome: $(40 + 10) / 2 = 50/2 = 25 \rightarrow 20$
 - Cost for Rome – Napels = 40
 - Cost for Napels – Palermo = 20
 - Total cost = $200 + 20 + 40 + 20 = 280$
- Through Genoa, Cagliari, Rome and Napels → Useless, since the cost of Turin – Genoa is already higher than Turin – Cagliari directly. The rest of the costs would be the same as in the first proposed route.

- An alternative to the second route (through Milan) would be to go to Rome through Bologna, but the cost for this would also be higher than to do this directly.
- Through Milan, Bologna and Napels:
 - Cost for Turin – Milan = 200
 - Cost for Milan – Bologna = $(60 + 10) / 2 = 70 / 2 = 35 \rightarrow 30$
 - Cost for Bologna – Napels (underused link) = $100 - (10 + 10) = 100 - 20 = 80$
 - Cost for Napels – Palermo = 20
 - Total cost = $200 + 30 + 80 + 20 = 330$
- Another way to get to Bologna would be through Genoa and Florence, but this would obviously not be interesting either:
 - Cost for Turin – Genoa = $(70 + 10) / 2 = 80 / 2 = 40$
 - Cost for Genoa – Florence = $(30 + 10) / 2 = 40 / 2 = 20$
 - Cost for Florence – Bologna = $(60 + 10) / 2 = 70 / 2 = 45 \rightarrow 40$
 - Cost for Bologna – Napels (underused link) = $100 - (10 + 10) = 100 - 20 = 80$
 - Cost for Napels – Palermo = 20
 - Total cost = $40 + 20 + 40 + 80 + 20 = 200$
- New light paths could be set up, but, passing so many hops, the cost for this would be a lot higher then some of the previous ones.

The best (cheapest) route seems to be through Cagliari, Rome and Napels:

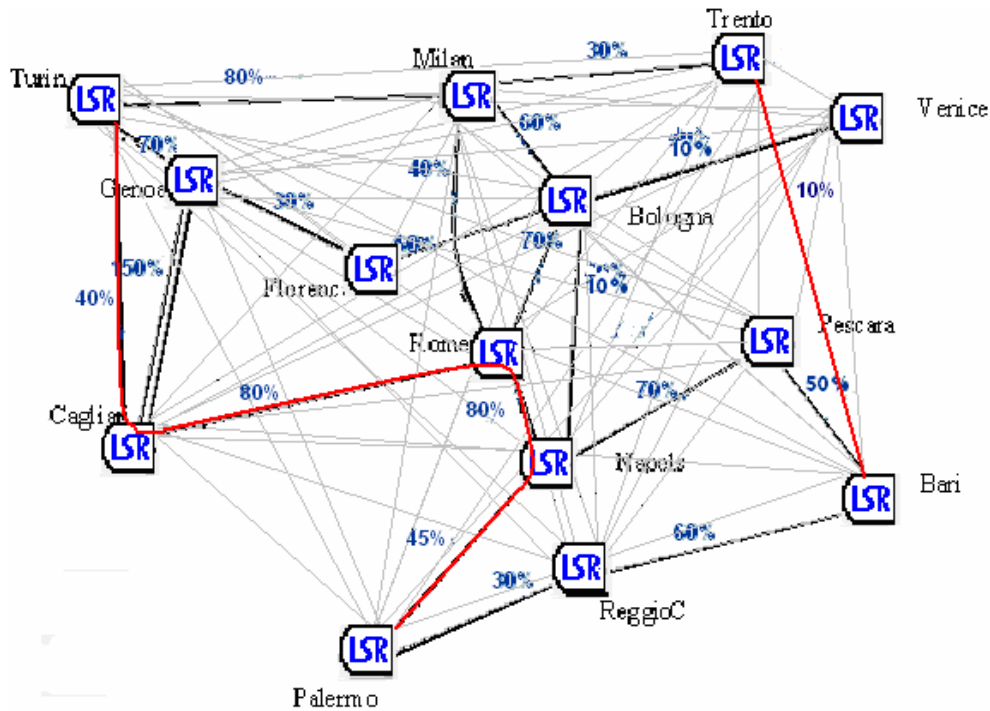


Possible paths to take for the flow from Trento to Bari are:

- Through Milan, Rome, Napels and Pescara:

- Cost for Trento – Milan = $(30 + 10) / 2 = 40 / 2 = 20$
- Cost for Milan – Rome = $(40 + 10) / 2 = 50 / 2 = 25 \rightarrow 20$
- Cost for Rome – Napels = $(80 + 10) / 2 = 90 / 2 = 45 \rightarrow 50 + 150 = 200$
- Cost for Napels – Pescara = $(70 + 10) / 2 = 80 / 2 = 40$
- Cost for Pescara – Bari = $(50 + 10) / 2 = 60 / 2 = 30$
- Total cost = $20 + 20 + 200 + 40 + 30 = 310$
- Through Milan, Bologna, Napels and Pescara:
 - Cost for Trento – Milan = $(30 + 10) / 2 = 40 / 2 = 20$
 - Cost for Milan – Bologna = $(60 + 10) / 2 = 70 / 2 = 45 \rightarrow 40$
 - Cost for Bologna – Napels = $100 - (10 + 10) = 100 - 20 = 80$
 - Cost for Napels – Pescara = $(70 + 10) / 2 = 80 / 2 = 40$
 - Cost for Pescara – Bari = $(50 + 10) / 2 = 60 / 2 = 30$
 - Total cost = $20 + 40 + 80 + 40 + 30 = 210$
- Through Milan, Rome, Bologna, Napels and Pescara:
 - Cost for Trento – Milan = $(30 + 10) / 2 = 40 / 2 = 20$
 - Cost for Milan – Rome = $(40 + 10) / 2 = 50 / 2 = 25 \rightarrow 20$
 - Cost for Rome – Bologna = $(70 + 10) / 2 = 80 / 2 = 40$
 - Cost for Bologna – Napels = $100 - (10 + 10) = 100 - 20 = 80$
 - Cost for Napels – Pescara = $(70 + 10) / 2 = 80 / 2 = 40$
 - Cost for Pescara – Bari = $(50 + 10) / 2 = 60 / 2 = 30$
 - Total cost = $20 + 20 + 40 + 80 + 40 + 30 = 230$
- Setting up a direct link, we would have to cross at least 4 hops (Milan, Rome, Napels and Pescara), which would cost $81 + 21 + 21 + 21 = 144$
- We could also use the first route, but setting up a new light path between Rome and Napels:
 - Cost for Trento – Milan = $(30 + 10) / 2 = 40 / 2 = 20$
 - Cost for Milan – Rome = $(40 + 10) / 2 = 50 / 2 = 25 \rightarrow 20$
 - Cost for Rome – Napels (new link) = 81
 - Cost for Napels – Pescara = $(70 + 10) / 2 = 80 / 2 = 40$
 - Cost for Pescara – Bari = $(50 + 10) / 2 = 60 / 2 = 30$
 - Total cost = $20 + 20 + 81 + 40 + 30 = 191$

It seems that in this case, the best route would be a new direct link between Trento and Bari, physically crossing Milan, Rome, Napels and Pescara.



The average link load is now 51,19%

4.3.3. Conclusion

I think the overall performance of this strategy is also quite good, and again we're nearly reaching the T_{mid} which we were aiming for. The average link loads reached with the reactive strategy were a little better (closer to 55%) though. There are still some underused links, and a proactive strategy like this, does not even attempt to reroute the traffic that is on them and remove them. It is possible that the link will be used at a certain moment, by a traffic flow with a high demand, but that is rather difficult to predict. The advantage of this strategy is however that congestion is being prevented. It seems that a hybrid strategy (both proactive and reactive) may be the solution for this network topology.

4.4. General conclusion

In order to avoid that too many underused links are being remained, the proactive strategy could be combined with the reactive one. The reactive strategy should only take care of underused links, as congested links are not supposed to be established at all.

When rerouting the traffic from underused links and removing them, there are two problems:

- Constantly rerouting traffic could cause network instability.

- Due to the proactive strategy, it is well possible that after some time, the underused link would become used by a traffic flow with a high bandwidth demand. If underused links are immediately removed, there is no chance for that to happen and possibly, a new light path will have to be set up for the new traffic flow.

For those reasons, and because it is difficult to predict future traffic, it may be useful to keep the underused links for a while before removing them. The strategy should keep an eye on the number of new traffic flows per 30 seconds. If this number is rather high (for example more than 4) the underused links should be kept. In case of a rather stable traffic pattern (less than 4) the underused links should be removed. The underused link we had left between Bologna and Venice can't be removed however, as it would render Venice unreachable (unless through links in the fictive full mesh, which do not yet exist as a light path), and there is no way to reroute the traffic.

5. Problems

During the theoretical analysis of the MTE strategies, I did not encounter any real problems, but normally, after doing that, I was going to simulate the strategies in GLASS. GLASS is an optical simulation framework which simplifies the evaluation of routing, restoration and signaling protocols in an optical or Internet environment. It allows researchers to study the behaviour of algorithms and protocols without the need to build a real network.

First, I learned to use the program by reading the documentation and looking at the example simulations which came with Glass. This program was still in the beta-phase (still in development, not completely operational yet) and it soon became clear that it was not exactly stable yet: the program regularly crashed after running a simulation or when opening a file. Also, many settings related to the setting up of LSPs had to be specified manually in the DML (Data Modeling Language) file in which a simulation was saved, as it could not be done in Glass-TSC (the graphical environment). The documentation did not seem entirely finished either.

To simulate multilayer topologies, light paths also had to be defined as an LSP (crossing OXCs instead of LSRs) in GLASS, although they are not really LSPs. Then you could setup a real LSP onto them, by specifying a tunnel ID (the tunnel ID specifying the light path to use). The problem was that this way, you could only travel one hop far on the logical topology. I only managed to setup LSPs travelling a direct light path. I found no information on how to forward LSPs between LSRs in the documentation. In the example simulations which came with GLASS, I found no multilayer simulations other than some where LSPs travelled a direct link between two LSRs. As no one at HUT had used GLASS before and hence could not help, I contacted one of the developers of GLASS and posted my question on a newsgroup, but I did not get any answers which solved the problem.

6. Conclusion and results

During my research at HUT, I certainly learned a lot about optical networking, the problems encountered and the solutions which are being worked on, such as Intelligent Optical Networks and Multilayer Traffic Engineering. Although rather theoretical, I found the subject very interesting to research and work on, as MTE might play such an important role in future networking.

From my research I can conclude that on networks with highly unpredictable traffic patterns, like from Internet traffic, will need an MTE strategy which combines a proactive and a reactive approach. (as described in 4.4)

Unfortunately, due to the problems I encountered with GLASS, I was not able to simulate the MTE strategies I developed. This was a bit disappointing, as I would have liked to see the strategies in action. Of course, this is the risk of doing research: you don't know what the results are going to be.

From the problems with Glass, I can also conclude that there may be a need for more advanced and stable simulation software to do research on MTE.

Aside from gaining technical knowledge, I also had the chance to get to know the Vietnamese culture and to take part in an internationally oriented project. To me, this gives an asset to my final work and it is an experience which I will never forget.

7. Word of thanks

Finally, I would like to address a word of thanks to Erasmushogeschool Brussel and the Flemish Interuniversity Council, for giving me the chance to go to Hanoi and work on this project. Furthermore, I would like to thank the following people for helping me realize this final work:

- Mr. Joeri Gerrits, promoter of the final work
- Mr. Jean-Pierre Roose from the office of internalization at EhB
- Dr. Tran Thi Ngoc Lan, supervisor of the project at HUT

8. References

8.1. Bibliography

Data Connection Computer Technology, “Introduction to Multi-Protocol Label Switching (MPLS)”, <http://www.dataconnection.com/mpls/whatis.htm>

Ghent University - IMEC, Department of Information Technology, “Multi-layer Traffic Engineering in Data-centric Optical Networks, Illustration of concepts and benefits”

Paola Iovanna, Roberto Sabella, and Marina Settembre, Ericsson Lab Italy, “A Traffic Engineering System for Multilayer Networks Based on the GMPLS Paradigm”

Ghent University - IMEC, Department of Information Technology, “Influence of the observation window size on the performance of multi-layer traffic engineering”

8.2. Software

GLASS: GMPLS Lightwave Agile Switching Simulator, <http://dns.antd.nist.gov/glass>

Appendix I: List of frequently used abbreviations

OTN: Optical Transport Network: network which uses optical media for data transport

TE: Traffic Engineering: technique used to route traffic over the network route traffic over the network in such a way that the network load is divided more evenly and to reach a certain Quality of Service (3.3)

MTE: Multilayer Traffic Engineering: traffic engineering method which takes into account both the logical and the optical layer of the network (3.4)

WDM: Wavelength Division Multiplexing: modulation technique used on optical networks, which transmits each of several data streams onto a different part of the light spectrum (2.4)

OXC: Optical Cross Connect: devices that can switch optical signals between different optical fibers, without the need for conversion to electrical signals (2.5)

ION: Intelligent Optical Network: optical network designed to bypass the manual intervention of the network operator when setting up or tearing down light paths (3.1)

IP: Internet Protocol

QoS: Quality of Service

MPLS: Multi-Protocol Label Switching: protocol independent system for Traffic Engineering (3.3)

LSP: Label Switched Path: pre-configured path for MPLS traffic flows (3.3)

LSR: Label Switched Router: router which routes traffic along Label Switched Paths on the logical layer (3.3)

RSVP-TE: Resource Reservation Protocol with Traffic Engineering extensions: signaling protocol which informs the Label Switched Routers which labels and links to use for each LSP (3.3)

T_{cong}: load on a link (4.2.1)

T_{high}: maximum load allowed on a link (4.2.1)

T_{low}: minimum load aimed for on a link (4.2.1)

T_{mid}: ideal link load average on the logical network (4.2.1)