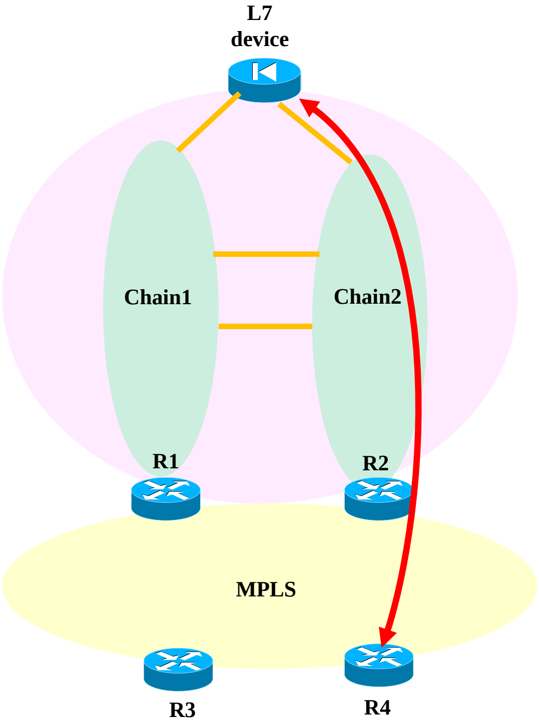
1 Traffic balancing through network chains

In the past I've been designing routing for internet access architectures for both mobile and fixed access. Almost always you have to deal with L4/L7 devices that require symmetrical routing, together with public pools that can't be natted. When it's not a firewall, it's a Deep Packet Inspection device (to limit p2p traffic). As traffic grows, it goes like this:

1. you setup 2 chains that backup each other (cheaper approach respect to having a lot of peripherical chains)
2. you setup 4 chains that backup each other, two by two
3. you setup N+1 backup chains, i.e. N active chains and one empty chain that is used only in case of backup or maintenance reasons on an active chain
4. you move the chains near the end-systems, thus scaling horizontally

This was not the scope of this post, but consider the following network scenario:



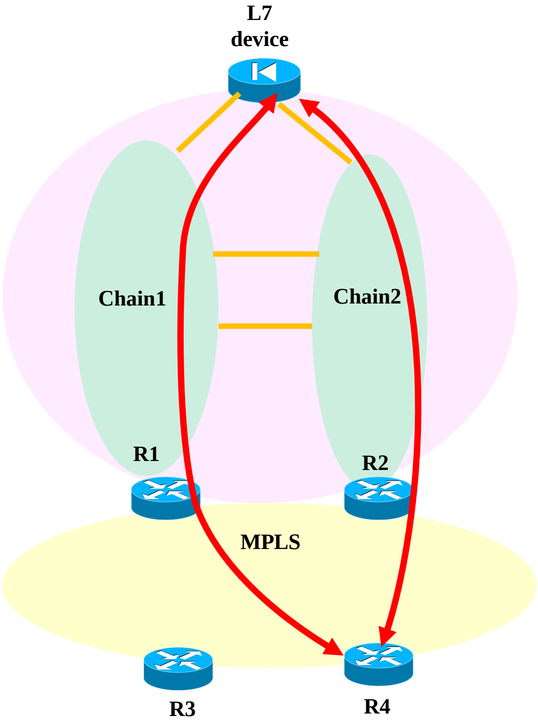
The two depicted chains are l3 chains, they are a layer3 backup to each other. Suppose now there is a strange failure, not a normal one, that causes the traffic to be dropped **even though dynamic routing stays up everywhere**. It's a so called '**black holing**' event. When this kind of things happen, the day after there will be someone arguing: why don't we balance the traffic over the two available chains ? should we manage it in this way, the next time a 'black holing' happens, only 50% of the traffic would be impacted ! (these are the same guys that would suggest you to go for static routing after a problem with dynamic routing, and would suggest you to go for the latter after a problem with the former approach, thus shifting from one approach to the other right after every service outage).

Maybe there are some university studies about the psychological impact of a service outage on the customer base: supposing there is such a strange failure once every month (you'd rather investigate why it happens, instead of thinking about changing your routing architectures), we would averagely impact half times the active chain and half times the backup chain. Thus this study would claim that 100% of impact once every two months is worse than having 50% of impact once every month. I've never heard of such a study anyway.

But is this really true ? Is it really like this ? From RFC 1925, truth number 8:

```
(8)  It is more complicated than you think.
```

Balancing algorithms are usually proprietary and there's not so much you can do to tweak them. Sometimes you can change a port-channel balancing algorithm, and you do it **per flow**. The more parameters are considered (for example tcp/udp source/destination ports), the better will probably be the results, and the better traffic will be balanced over the odd or even available links and paths.



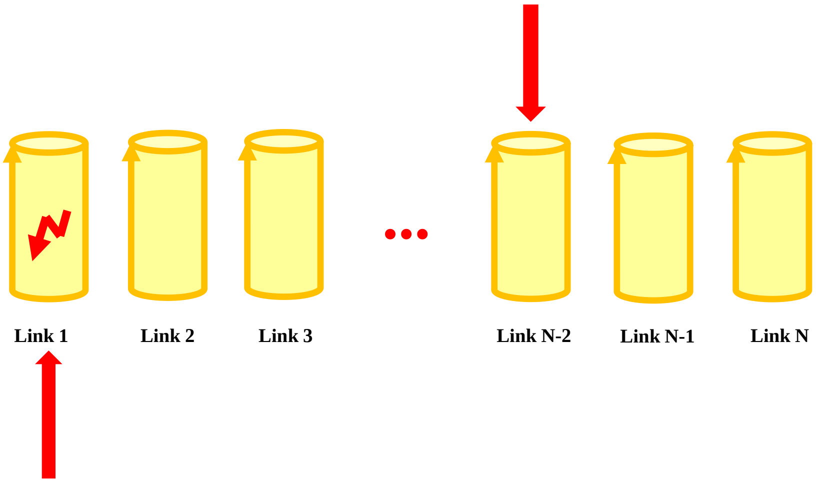
The above approach is not so much different (let's suppose the firewall is connected through a vpc): in real life, you will not be able to have traffic exiting and returning through the same links on a chain of routers/switches. If you have **two routing paths** (this is important, we know a firewall could PAT the traffic and do the trick …) and you balance traffic between them, you can't ensure that routing is symmetrical, and a single tcp/udp flow will follow exactly the same links in both directions. This means the following, thinking about the above picture, and supposing you go for the balanced chain approach:

1. a certain **flow** could have upload traffic through chain 2, and download traffic through chain 2
2. a certain **flow** could have upload traffic through chain 2, and download traffic through chain 1
3. a certain **flow** could have upload traffic through chain 1, and download traffic through chain 2
4. a certain **flow** could have upload traffic through chain 1, and download traffic through chain 1

If you have a traffic black holing scenario on chain 2, only in case 4 traffic will NOT be affected. This means that averagely speaking, if balancing works per flow and traffic is fairly balanced, every time there is a black holing in one of the two chains, **75% of the traffic is affected**.
So the question now is … do you really prefer to have a 100% outage once every two months, rather than a 75% outage once every month ?

1.1 Port-channel and link degradation

Moving on with the game (because it's more a probability exercise rather than a networking topic), let's suppose we have a port-channel with N links between two routers. Let's suppose that one link for some strange reasons has problems, and starts dropping traffic (or a high percentage of packets) even though it's still up and participating into the bundle (lacp 'keepalive' packets are fine). How many flows flowing through this bundle will experience problems ? let's suppose that traffic flows through the bundle in both directions, even though we don't know on which links.



The number of flows that will be **discarded** is the following:

$$\frac{2*N-1}{N^2}$$

For N = 1, the output is 1 → 100% of the traffic will experience problems. For N=2 (which is the double chain example we made before), the output is 0.75 → 75% of traffic will be dropped. How did we get the above equation ? the denominator is easy: you have N different available links for upload traffic, and for EACH of them you have N different links for download traffic, for a total of N^2 choices. This represents the total number of ways you can distribute upload/download traffic. How many different traffic distributions are there, causing flow issues ? the constraint, is that there must be at least one direction of the flow through the broken link. Looking to the above figure, if you keep the upload fixed, there are N different choices for download traffic. If you keep the download fixed on the broken link, you can have N-1 choices for the upload traffic (minus 1 because you have already considered the upload and download both flowing through the broken link). Thus you get the $2*N-1$.

The equation could be generalized to M = number of links causing a black hole, and N = total number of links in the bundle, with the obvious constraint $1 \leq M \ \leq N$:

$$\frac{M(2*N-M)}{N^2}$$

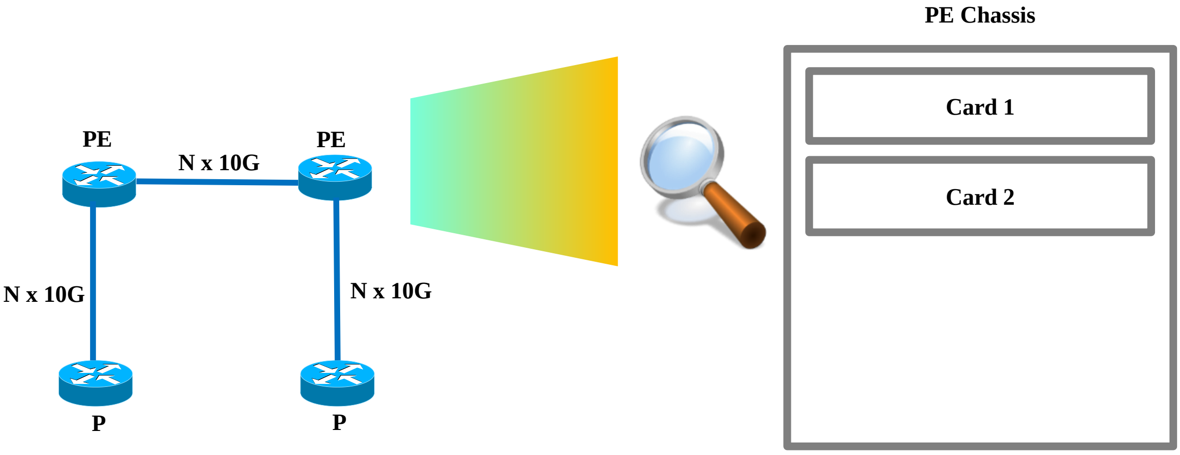
The output is the previous one for M=1, while it is 1 for M=N. For M = N-1 (all but 1 link have problems), the output is $\frac{N^2-1}{N^2}$ ... in this case, traffic should flow in both directions through the only left good link, thus there is just one chance out of all the possible upload/download distributions of traffic.

Next post about the “Monty Hall” problem (maybe).

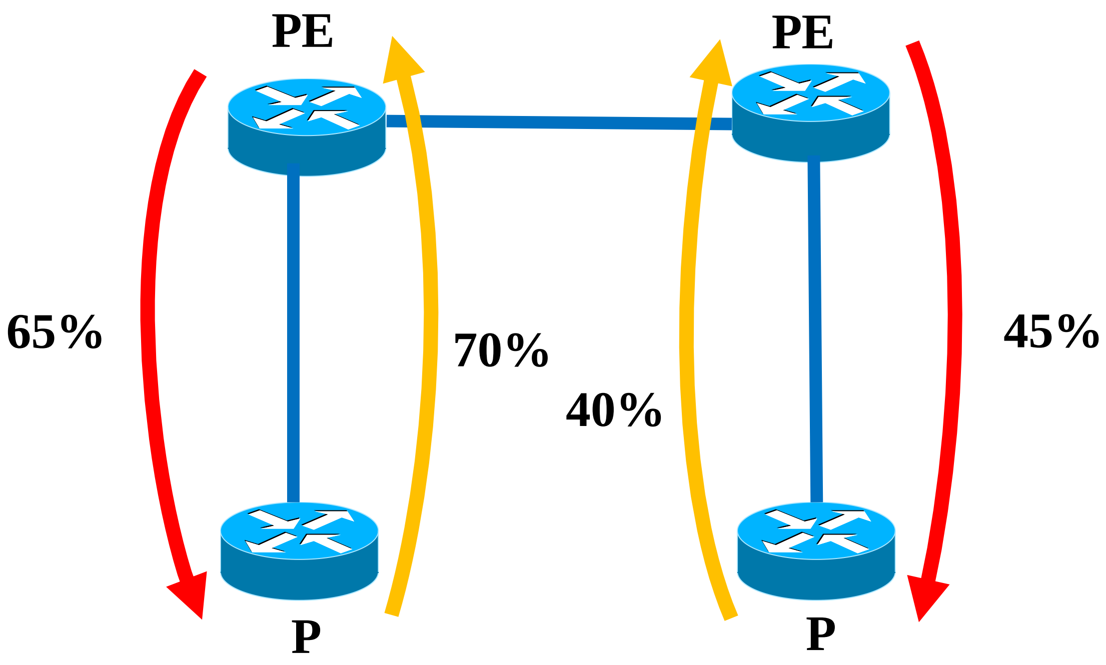
2 Port distribution on line cards

Another form of ‘balancing’ I’ve been discussing in the past, is how to distribute links on a router through the available line cards. Usually this is a problem when you have a new network or you have just replaced old P/PE routers with new ones, with a lot of more interfaces. You will usually have just two line cards, over which you will have to distribute your core links. The question is:

1. is it better to distribute half of the links on one line card, half on the other or …
2. is it better to put ALL the links on the same line card



Apparently, it looks like you have ‘more redundancy’ if you go for the first choice: if you have a line card failure, you still have 50% of the bandwidth available. Unfortunately, traffic is not always fairly balanced on the available links, nor it is always so easy to move traffic from one side to the other. For example in the following case:

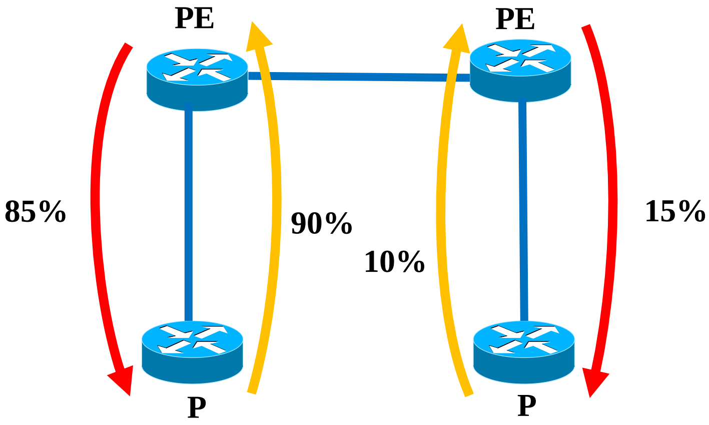


- in case of a card failure on the left PE/P bundle, you loose **15%** of traffic on the PE→P direction, you loose **20%** of traffic on the P→PE direction
- in case of a card failure on the right PE/P bundle, you loose **no traffic**

Of course we are excluding manual or automatic intervention (through 'complex' external SDN systems through PCEP protocol, segment routing and traffic engineering, so on and so forth …). What happens in case ALL the links have been connected on the **SAME line card** ?

- in case of a card failure on the left PE/P bundle, you loose **10%** of traffic on the PE→P direction, you loose **10%** of traffic on the P→PE direction
- in case of a card failure on the right PE/P bundle, you loose **10%** of traffic on the PE→P direction, you loose **10%** of traffic on the P→PE direction

So in this case the two solutions are probably equivalent or at least very similar. Approach number 2 becomes more convenient ONLY in case the network is **VERY unbalanced**:



Approach 1:

- in case of a card failure on the left PE/P bundle, you loose **35%** of traffic on the PE→P direction, you loose **40%** of traffic on the P→PE direction
- in case of a card failure on the right PE/P bundle, you loose **no traffic**

Approach 2:

- in case of a card failure on the left PE/P bundle, you loose **no traffic**
- in case of a card failure on the right PE/P bundle, you loose **no traffic**

Beware that connecting all the above bundles' interfaces distributing them on two line cards, thinking that you can configure the 'min-link' command, is not a choice … this would imply that loosing one line card you would loose ALL your bundles, thus isolating the router from the network …

Real life can be more complex than this, you could have three or even more possible links/bundles, for example on the P routers of your core network, where redundancy is even more important. If you have a cube in your core network (quite a standard topology for an mpls core network), every P router would have three different bundles. Probably you would also have at least three different line cards. In general you can't provide a rule for all the possible configurations. Let's say that in case you have 4 or more line cards, the best thing you can do is distributing your links belonging to the same bundle as much as you can, in case you have two line cards, don't discard the possibility of assigning all the links belonging to the same bundle to the same line card, i.e. ensuring 100% of bandwidth and traffic re-routing in case of a line card failure.