Saturday, March 29. 2014

About link aggregation, water and buckets

Despite popular belief ethernet networks aren't lossless, on the long way between the TCP/IP stack from one side to the TCP/IP-stack of the other side there is a lot that can happen to the data. Datacenter Bridging is a mechanism to put some kind of losslessness on the network needed for shoehorning storage protocols on it (FCoE). That said most of the time ethernet networks appear as lossless however it's not guaranteed.

But that are a lot of different stories i won't write about today. That said you can make some errors in configuration that make ethernet networks more lossy than necessary and those can even haunt you in relatively simple configuration like a link aggregation.In order to do link aggregation you have to do distribute the traffic on the members of the aggregation. That's obvious. Link aggregation could simply do "round robin" which would ensure that all links are nearly equally loaded. However there is a problem and that starts with the standards related to Link aggregation. The respective standards specifically mandates that parts of a conversation shall not be reordered by link aggregation. And that the point where round robin is out of the game. Let's assume you have a conversation between A and B. There are two paths ab1 and ab2. Let's assume to transfer the first frame of the communication via path ab1, with round robin the second frame goes via port ab2 ... now let's assume that there something not okay with this path ... it needs longer to process the frame. In the meantime you've send the third frame via path ab1, there are no delays so it could arrive before the other. You don't want that ... because out-of-order frames and reordering them are a major nuisance to communication.

How do you prevent this and ensure that each frame of a communication goes over the same path while still distributing the load? That's quite simple. You take information from the communication that enables you identify conversations. A coarse way to it, could be taking the mac addresses of the communication partners do some math on it, do a modulo by the number of interfaces on it and take the n-th interface of the trunk where n is the result of the modulo. By doing so every conversation from one mac address to another is always transported via the same cable. Of course it does it for all conversation between the both mac addresses, but the requirement "No conversation should use multiple cables" is fulfilled. Or let's say it this way: The system views all the traffic between the both mac addresses as one conversation.

As this isn't optimal other distribution algorithms were invented using more data in the frame to distribute the traffic. Like IP numbers, VLAN-ID, source port number, destination port numbers. Everything where you can assume when it's equal it's the same conversation.

This is important in other regards, too. One ever reoccurring question of customer is "Hey, i bundled 2 10GbE interface into a trunk and i just get roughly 10 GBit/s worth of traffic through it with NFS despite all tuning between one client and a server". The problem is: Solaris just opens one TCP/IP connection for the communication between an NFS client and a NFS server by default. Everything is multiplexed on this connection. So you have one conversation and the load distribution of link aggression is always using the same wire for this conversation. Thus obviously your limit is the capacity of one wire. However it's easy to change. You can change rpcmod:clnt_max_conns in /etc/system to allow more connections than just one. Usually i set it to 8 or so. Then you have multiple connections, thus multiple conversations, thus more than just the bandwidth of one link.

The other point where this gets important is the explanation why the interfaces of a link aggregation aren't equally loaded at link aggration. Customers are asking me "Why is interface 1 loaded by 25 percent and interface 2 by 75 percent? I want to see equal distribution. Is this a bug in link aggregation?" No, it's not a bug. It's standard-compliant link aggregation perfectly working. The load distribution is directly dependent to the traffic you are putting on the line. If 75 percent of the traffic is on one conversation and 25 percent of traffic is on all other conversations the least skewed distribution you can see on a two interface aggregation is the distribution 75% to 25% in the case all conversations of the 25 percent part is on one interface, the 75% on the other. While a really pathological case it's even possible that even when you have many small conversations all traffic goes to one interface because the load balancing algorithm always computes the same interface number out of the packets (but that's really a corner case). That said most of the time you see a skewed distribution in link aggregation on the interfaces you have aggregated.

However, when you start go get near the maximum bandwidth of your trunk you may run in a strange issue. I recently moved data close to 40 GBit/s in a 4 interfaces trunk of 10 GbE interface. However i saw dropped packets in my UDP benchmark and discarded packets on one port of the switch on the ingress side ... dang. Each interface on it's one was perfect ... 10 Gbit/s without a single packet dropped, when using DLMP ... no packet dropped. Just when using link

aggregation with LACP, i saw this discarded packets. WTF?

The reason is actually quite simple and obvious as soon as you think about it. When doing link aggregation all the path down between two nodes each node has a independently working frame distributor (the thingy on the sending switch that distributes the load). Each one is calculating it's own load distribution decision. Now lets step back and think about an example: Let's assume you have 4 buckets with 10 litres of water. Each bucket can hold exactly this 10 litres of water. Now you redistribute that water on four new buckets. The first bucket gets 10 litres. the second one as well. Out of some reasons the third bucket just get 8 litres. So you have 12 litres for the fourth bucket. You just get 10 into the bucket and perhaps you have a little buffer bucket at the point you redistribute the water in the buckets, however when this one is full, you spill water.

In our network the discarded frames are the spilled water. Okay, the analogy doesn't fit completely but over all it should hold the water

It can happen when you load the lines to the max and the load distribution isn't equal on the ingress and the egress side . When your link aggregation distribution mechanism is very good (like almost absolutely equal distribution) and the next distribution algorithm is very bad at this (like totally skewed to one interface) the probability is high, that you overload one channel.

What's the consequence of it? When doing LACP near the aggregated line speed of all channels ensure that the distribution mechanism on the all components on the path between your two nodes is as similar as possible. Using L4,L3,L2 (load distribution based on port numbers, ip numbers and mac addresses) as the link aggregation on the server and doing just L2 distribution (load balancing just based on mac addresses) on the switch will very likely lead to a situation where the combination of all conversations will be distributed onto several interfaces to the switch and thus will likely lead to a traffic stream between two servers that is bigger than just interface. But the L2 distribution will try to egress all this packets on only one interface. Trying to push in 40 GBit/s into a switch and just allowing 10 GBit/s to flow out, will at first overwhelm the buffers in the switch and than lead to discarded packets.

So the basic tip is: Keep the load distribution algorithms with load balancing as similar as possible. On the other hand: I tend to plan link aggregations with one interface more than needed. So at first i have still enough bandwidth, when one interface fails and at second i have some additional headroom on the lines as the complete traffic is distributed about more interfaces than necessary.

Posted by Joerg Moellenkamp in English, Technology at 09:59

Isn't DLMP only one path for a given Mac-Address (vNIC) active at a time vs LACP all path's are active (depending on distribution)?
Anonymous on Apr 2 2014, 16:03