

Do balance-alb and balance-tlb support fault tolerance?

Bonding Mode 5 (balance-tlb) works by looking at all the devices in the bond, and sending out the slave with the least current traffic load. Traffic is only received by one slave (the "primary slave"). If a slave is lost, that slave is not considered for transmission, so this mode is fault-tolerant.

Bonding Mode 6 (balance-alb) works as above, except incoming ARP requests are intercepted by the bonding driver, and the bonding driver generates ARP replies so that external hosts are tricked into sending their traffic into one of the other bonding slaves instead of the primary slave. If many hosts in the same broadcast domain contact the bond, then traffic should balance roughly evenly into all slaves.

If a slave is lost in Mode 6, then it may take some time for a remote host to time out its ARP table entry and send a new ARP request. A TCP or SCTP retransmission tends to lead into ARP request fairly quickly, but a UDP datagram does not, and will rely on the usual ARP table refresh. So Mode 6 is fault tolerant, but convergence on slave loss may take some time depending on the Layer 4 protocol used.

If you are worried about fast fault tolerance, then consider using Mode 4 (802.3ad aka LACP) which negotiates link aggregation between the bond and the switch, and constantly updates the link status between the aggregation partners. Mode 4 also has configurable load balance hashing so is better for in-order delivery of TCP streams compared to Mode 5 or Mode 6.

If this bond will be bridged to virtual machines, then you cannot use Mode 5 or Mode 6 due to MAC rewriting behaviour of both modes under certain conditions, and doubly so due to the ARP intercept behaviour of Mode 6.

All modes 0 to 4 will work with VM bridges, but 0 (round-robin) and 3 (broadcast) are probably not suitable for most workloads, definitely not for TCP and SCTP streams. All modes 0 to 4 require switch config, except Mode 1 (active-backup).

Revision #2

Created 2021-03-24 19:24:17 UTC by Sam

Updated 2021-03-24 19:26:21 UTC by Sam