Executive Summary

OVS bonding is supported in both kernel and DPDK datapaths, optionally with LACP support. The table below summarizes the support of different modes.

	Linux Kernel Bonding	OVS bonding
active-backup	supported	supported
balance-slb	supported equivalent mode: balance-xor (mode 2) + L2 hash	supported

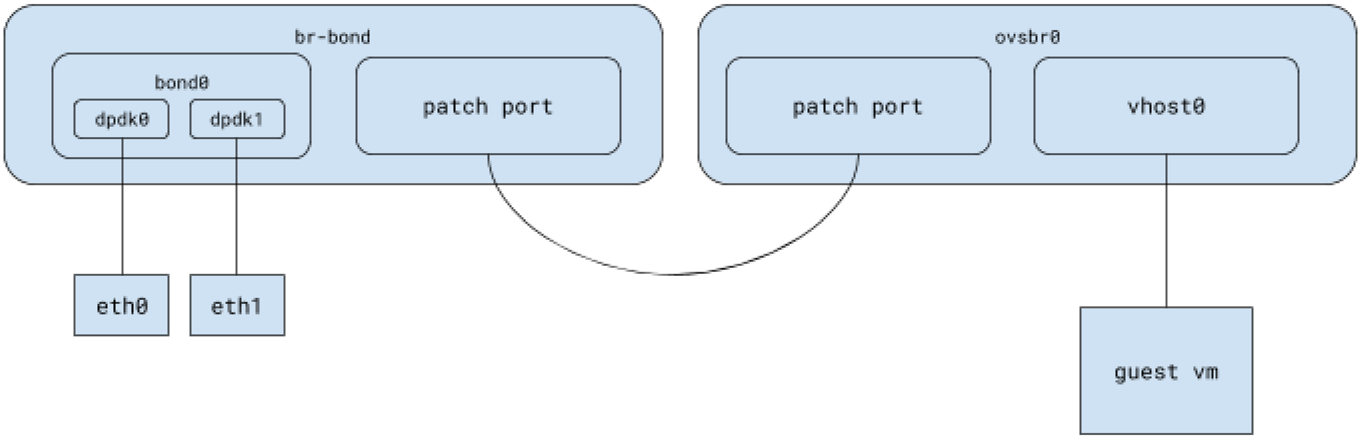
These modes were tested with testpmd generating 2 millions pps into an OVS bridge, with 64k UDP flows. A script running on the server let the two links flap randomly, while the receiving server checks that the throughput doesn't drop.

The software versions used in the tests are OVS 2.11 with DPDK 18.11 from FD-19A release

```
# cat /etc/redhat-release
Red Hat Enterprise Linux release 8.0 (Ootpa)
# uname -r
4.18.0-78.el8.x86_64
# ovs-vswitchd -V
ovs-vswitchd (Open vSwitch) 2.11.0
DPDK 18.11.0
```

Details of the setup we used

Whenever the control plane is Neutron, OVN, or another solution, the OVS bonding needs a particular setup to work well. Traditionally the bonding port was added directly in a bridge, e.g. br-int or br-ext, along with all the other interfaces.
The problem with this setup is that OpenFlow exposes numbers for the interfaces rather than ports, there is no simple way to output packets on a bonding.
We found a simple workaround for this, which is creating a bridge with only the bonding in it, and connecting it via a patch port. The bonding bridge needs to have the action=NORMAL OpenFlow rule to work effectively.
In this way, packets can be sent to the bonding using the patch port OpenFlow number.
The setup should look like this:



The bonding types we used are active-backup if the target is just redundancy, and balance-slb if the target is increased throughput, both with and without LACP.

Related BugZillas:

#1465551 802.3ad support for dpdk bond (NIC bonding with OVS-DPDK)
#1591216 [RFE]:Add support OVS-bond with LACP on RHOSP environment
#1651155 [RHOSP 13] OVS support for LACP bonds

Other non production ready alternatives

In the OVS scenario, there are also other bonding solutions other than the one we described above.
We tried those alternatives too, but they didn’t satisfy the quality, stability or performance requirements we expect for a production software.

- OVS bonding in balance-tcp mode
    - requires LACP, thus adds complexity to the setup
    - the recirculations needed for L4 hashing hurts performances
    - tech-preview
- DPDK PMD bonding
    - code is too unstable even for a test environment
    - poorly tested
    - very few people are working on it upstream
    - OVS upstream developers have no interest in maintaining it