

This doc was last updated: 23rd Sep 2017

During [knet upstream presentation](#) at [HA Summit 2017](#), Perry M. asked why knet is not using vxlan encapsulation or vxlan in general as transport mechanism.

I could not answer the question right away as I was not very familiar with the vxlan technology. I spent a bit of time looking at vxlan and while there are similarities with knet, the 2 technologies are very different.

TL;dr: the vxlan and knet don't share the same use cases and vxlan is not completely fit for the HA use case.

vxlan is an extension to vlan, designed to transport **only ethernet frames** between nodes, using UDP technologies. It does **not provide** direct (see below for longer explanation) redundant communication between nodes nor encryption or compression.

Knet is designed to transport **all kind of data** (including ethernet) between nodes, using UDP/SCTP technologies. **It does provide direct** (again see below) redundant communication between nodes, with encryption (FIPS compliant) and compression.

=== stop reading here if that was enough ===

Based on presentations from Cisco and Juniper and other blogs/tutorials, I found out that it is possible to add redundancy and encryption to vxlan, but it does require a set of external tools (bonding/teaming, bridging, ipsec, BGP/OSPF, VRRP/keepalived) and depending on the network topology, also network equipment capable to understand vxlan. Perhaps there are better ways, but google was not super helpful in this specific area (beside displaying some rather complex and scary data center diagrams).

The vxlan stacking, to achieve the same feature set as knet, appears to be much more complex and more difficult to configure and potentially debug in case of problems. knet is all integrated in one solution. If anyone has better pointers to those architectures/setups, I am more than happy to review this statement and see how that could work for HA :-)

Knet was designed to perform super fast failover when detecting connectivity issues. Vxlan needs to rely on protocols that are much slower to respond to failures. This could cause longer outages in HA environments.

Vxlan is a recognised RFC standard, knet is not (yet?). This is clearly a negative point for knet, on the other side, knet does not use any non-standard technologies either. Knet run on top of UDP/SCTP/IP. Knet encapsulation does not require network equipment support at all.

Feature comparison (based on HA use case):

	knet	vxlan
Link heartbeat	yes	no
Link failover/redundancy	yes	no (1)
PMTUd	yes	no
Encryption	yes	no (2)
Scalability	yes (3)	yes
Compression (4)	yes	no
Data types	all (5)	ethernet only
Debugging/logging (6)	centralized per-node	unknown
Network overhead (bytes/pckt) (7)	26	50

- 1) Requires external tools.
- 2) Can be achieved via ipsec.
- 3) HA use case does not require the same level of scalability as vxlan. Knet is designed to scale up to 64K nodes, but not tested at that level yet.
- 4) Not strictly required for HA use case but hey, i have to make knet look good right? ;)
- 5) This requires a bit of explanation. Currently corosync uses knet directly to transport all HA traffic. It is possible for corosync to share it's knet setup to the host via [libtap](#) and expose network interfaces (similar to vxlan) to all the applications on the host itself. This would allow transporting Ethernet traffic over knet in a corosync based HA environment. The downside here is that there is a minimal latency overhead for applications using the tap device exported by corosync/knet due the extra packet memcpy between kernel and userland. Vxlan is probably faster in that respect as the implementation is all in kernel.
- 6) Assuming redundancy is a strict requirements for HA, knet provides all the logs of the network status in a per-node centralized fashion, making debugging much simpler. Vxlan (with the external tools) requires parsing more logs and matching them to get to the bottom of any issues.
- 7) Without encryption.

Conclusion:

One of the most obvious questions would be: can we reuse knet technologies to make vxlan fit for HA? Most likely yes. Layering knet health checks and control to drive vxlan can be done, but there is still an issue around the datatypes that knet can transport and vxlan cannot.