IPvlan

Eric Dumazet (edumazet@google.com)
Mahesh Bandewar (maheshb@google.com)

IPvlan (instead of MACvlan) : Why?

- Switches may apply policies to disable CAM-table overflow essentially allowing a mac-address per incoming frame per port
- Excessive use of mac-addresses per NIC on host may put the host in promiscuous mode.
  - Might have impact if the connected switch starts throwing all multicast traffic as well.
- Other alternatives are not really well performing
  - NAT
  - Forwarding

IPvlan: Development Challenges, Choices

- Integrate changes into macvlan itself
  - Proved difficult without any compromises
- Broadcast / Multicast tweaks
  - Turn-off broadcast if IPv4 is not used.
  - Use multicast filter in decision making while forwarding to slave
- Communication with Master (mostly in *init_ns*)
  - macvlan solves this with hairpin-mode support from connected switch
  - IPvlan can’t do this (same L2 address)!

IPvlan: Development Challenges, Choices (continue…)

- To communicate with host the setup need to put host on the same IPvlan bridge as rest of the slaves.
- Alternatively there are patches to the stack to inject routes in the master’s namespace to redirect packets. These were viewed as too intrusive.

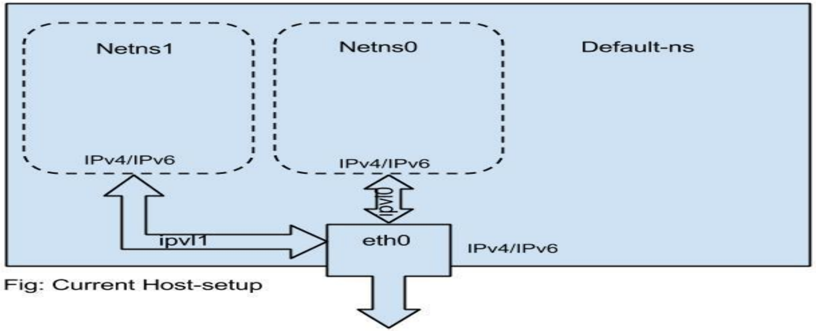


Fig: Current Host-setup

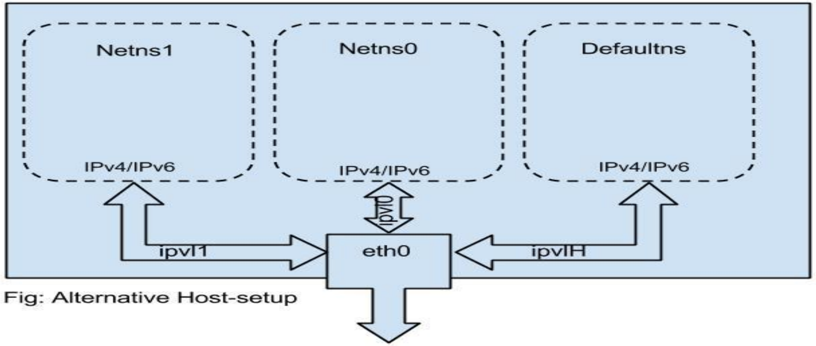


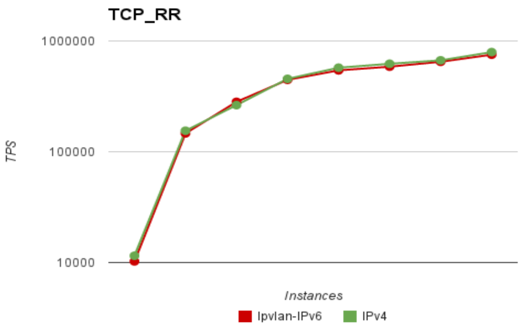
Fig: Alternative Host-setup

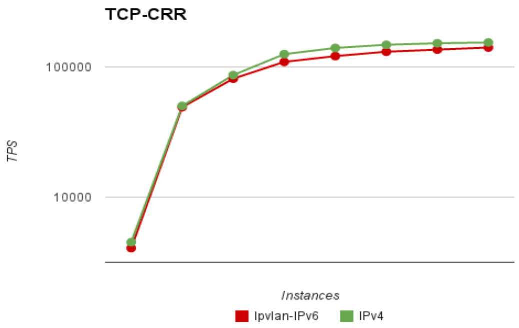
IPvlan: Use Cases

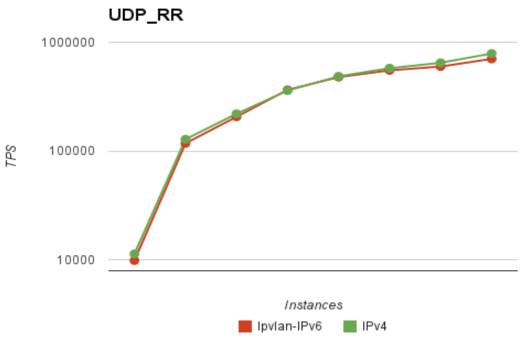
- IPvlan in L2 mode [slave(s) assigned to namespace(s)]
  - Setup is similar to several hosts connected to an external switch.
  - This is very similar to macvlan bridge mode.
  - Each slave can operate on it’s own [IP, routing etc.]
  - Suited in trusted environment.

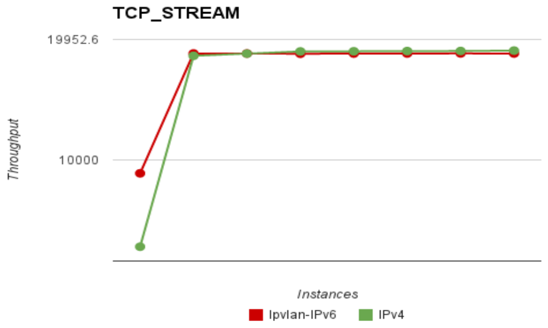
- IPvlan in L3 mode [slave(s) assigned to namespace(s)]
  - All slaves receive only unicast. Master handles rest.
  - Each slave relies on the routing from the masters’ namespace.
  - Tinkering with IP inside namespace may lose connectivity.
  - Suited in non-trusted environment.

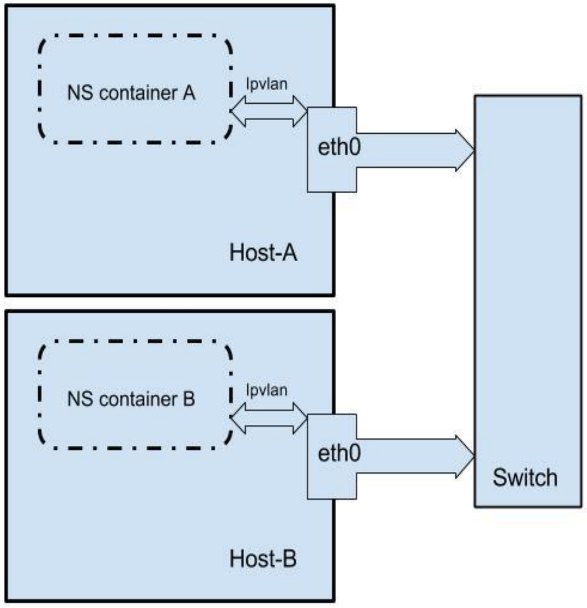
IPvlan: Performance comparison











IPvlan Setup

IPvlan: Future Enhancements

- Defer multicast / broadcast traffic processing.
  - Deferring will boost unicast throughput and improve latencies
  - Doesn’t need to process these in fast-path.
- ARP filters.
  - Applicable to IPv4 only
  - Probe into the ARP and forward only when slave may have interest.
  - Will save on packet duplication when slaves are many.
- Tap type slave device interface for Kvm/Qemu.
  - Similar to macvtap

IPvlan: Future Enhancements (continue…)

- Offload XMIT to hardware if NIC supports it.
  - Similar to few current Intel NICs for macvlan.
  - Can’t use just L2 and logic needs to include L3 (possibly more?)
- Enhanced L3 mode to enable communication with master.
  - L3 mode does not deal with broadcast and multicast.
  - If master has to communicate with other slave, then should use same bridge.
  - Nominate one of the slaves to receive broadcast / multicast
- Userspace support
  - Docker
  - CRIU

IPvlan: Questions? Comments?