DSA switches: domesticating a savage beast

Vladimir Oltean
Distributed Switch Architecture: an overview

- Framework for managing Ethernet switches

- Driver writer FAQ: should I write a plain switchdev or a DSA driver for my hardware?
  - switch supports direct packet I/O => plain switchdev
  - switch supports indirect packet I/O through an Ethernet port of the host => DSA
Summary of changes to DSA

• Separation between control plane and data plane packets
• Support for unoffloaded upper interfaces
• RX filtering
• Support for cross-chip bridging in more varied topologies
  • Disjoint trees
  • “H” trees
The data plane and the control plane

- Different classes of switches
  - Fully managed
  - Unmanaged
  - Lightly managed

- tag_8021q can help with the unmanaged and lightly managed switches, but only until the bridge claims the VLAN table
  - must teach the bridge about data plane packets
  - TX forwarding offload: extend skb->offload_fwd_mark for TX
Offloading software upper interfaces

- Offloading support added for LAG and HSR/PRP
- Repaired the software fallback which got broken when DSA was integrated with switchdev
  - API added to switchdev for drivers to explicitly declare that they offload a bridge port
- FDB isolation is still an issue
RX filtering

- No RX filtering for standalone ports (IFF_UNICAST_FLT), just bridged
- Tell the hardware which addresses must be filtered towards the host
  - Assisted learning on the CPU port replaces hardware source address learning, sniffs switchdev FDB events on foreign interfaces
  - Port MAC addresses, the bridge device MAC address are offloaded by switchdev as local/permanent FDB entries
- Still cannot remove CPU port from the flooding domain of user ports
  - DSA interfaces might be bridged with foreign interfaces
  - Bridge with upper interfaces might become promiscuous (no IFF_UNICAST_FLT)
Switch topology changes

• Daisy chains have been the norm, but crazy people always like to “innovate”

• Changes at the cross-chip notifier level to support bridging between DSA user ports in other circumstances
Disjoint trees
H trees
Conclusion

• Transforming the wide variety of DSA switches into something that is compatible with the network stack’s expectations requires a good amount of creativity
• The risk is that we might shoehorn them into something that departs from the use case they were intended for
• For the finer points, be sure to read the full paper with the same name!
  • https://linuxplumbersconf.org/event/11/contributions/949/