MLAG on Linux

Scott Emery, Wilson Kok

Cumulus Networks Inc.
180 E. Dana Street, Mountain View, CA
scotte@cumulusnetworks.com
wkok@cumulusnetworks.com

Abstract

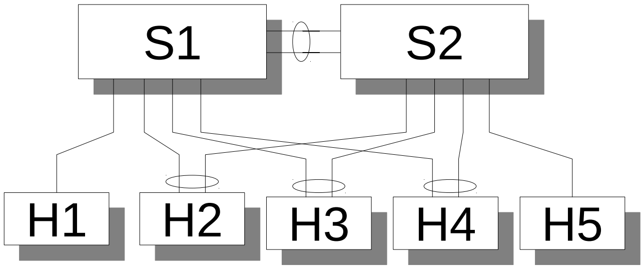
MLAG is a networking technology which allows for increased redundancy and bandwidth in layer 2 networks. This paper begins with an overview of MLAG, the problems it solves, and the common use cases. This leads to the important design considerations and caveats of a properly functioning MLAG implementation, especially with respect to MAC address learning and packet forwarding. This requires additional capabilities to be added to the Linux kernel bridging and bonding drivers for proper MLAG operation. An example of a recent implementation of MLAG on a Linux system is used to describe the types of data which must be synchronized between bridges and the interactions with other system components, such as the spanning tree daemon.

Keywords

MLAG – Multi-chassis LAG
LAG – A link aggregation group
ISL – Inter-switch link
Bond – The term used in this paper for a LAG, EtherChannel, Port Group, Trunk, or other words used to describe this same concept.
Singly connected node – a node that is connected to a single switch
Dually connected node – a node that is connected to two different switches
MLAGd – MLAG daemon

Introduction

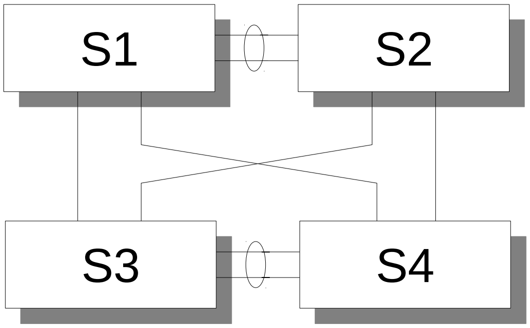
Multi-Chassis Link Aggregation, or MLAG, is a bond where at least one of the bond partners has member ports on multiple physical systems. This is illustrated in the following diagram.



In this diagram H2, H3, and H4 are each configured with a 2-member bond. However, unlike a traditional bond, the other end of each bond member is connected to a different physical device, S1 and S2.

In a network implementing an MLAG there are no assumptions or additional requirements for the bond partners implementing a traditional bond, H2, H3, and H4. They could be running any OS and are not required to run any particular protocols. However, if they choose to run the Link Aggregation Control Protocol, LACP, then the protocol must function properly. From the point of view of H2, H3, and H4 they have a 2-member bond connected to a single partner device. This means that they can use either or both bond members and can distribute traffic across the two members in any manner they desire. This paper assumes that the devices implementing MLAG, S1 and S2, are running Linux with the enhancements described.

Another common example of MLAG is one that is formed between a pair of switches connecting to another pair of switches as shown in the figure, below.



In this example, switches S1 and S2 form a logical switch and so do switches S3 and S4. Between the two logical switches is a single, four-member bond.

In this document, we will focus on the description of these two common examples. The extension of the model to support other cases may require additional mechanisms and considerations but by no means is it precluded by the proposal presented in this document.

Control plane model

The MLAG control plane functions include but are not limited to the followings:

- identify member interfaces of the MLAG across different physical nodes

- verify if the member interfaces' configuration and runtime parameters satisfy the criteria for joining the MLAG and becoming operational
- set up the data plane based on the forwarding rules for MLAG operational interfaces as well as non-MLAG singly connected interfaces
- synchronize MAC address database and IGMP group membership information to reduce flooding of traffic
- detect various link failure and node failure scenarios and ensure minimum traffic disruption and maintain loop free topology at all times

While it is possible to implement the entire MLAG control plane in the kernel, we believe it is simpler and more robust to delegate most of the control plane functions to a protocol daemon, i.e. MLAG daemon, while using the kernel to enforce correct default interface state. This model is analogous to the kernel bridge driver working in tandem with user space Spanning Tree daemon such as mstpd.

MLAG Daemon

In order to facilitate communication between the MLAG peers a daemon runs on each device. The daemon:

- Communicates with the peer MLAG daemon
- Retrieves and modifies kernel state, including the FDB, the MDB, LACP partner information, and VLAN configuration
- Communicates with the user-space Spanning Tree daemon

Inter-switch link (ISL)

The two physical switches in a logical node need to be interconnected by a physical link or bond. We will refer to this connection as the ISL (inter-switch link) throughout this document.

The ISL is used by the MLAG daemons to exchange management traffic regarding the control of the MLAG operations. The ISL is also part of the L2 topology, carrying protocol traffic such as Spanning Tree PDUs and IGMP reports, as well as data traffic in the case where there are singly connected nodes, either by configuration or due to link failures.

MLAG identification and formation

An MLAG physical node needs to identify the links that are connecting to it's peer physical node in order to form a LAG.

MLAG-enabled attribute and MLAG id

Each physical node within a logical node assigns an id to the interface by configuration. The physical nodes in the same logical node exchange information about their MLAG member interfaces. In the simplest case, the logical nodes can declare interfaces with the same MLAG id to successfully form an MLAG without any verification.

Alternatively, other criteria can be used, such as using LACP to confirm the identity of the peer logical node, before declaring the MLAG being formed and operational. The criteria may even include configuration consistency checks on the interfaces across the physical nodes, e.g. ensuring they have the same VLAN membership.

Here, we propose that an MLAG member connection be configured as a Linux bond on each physical node, and the kernel bonding driver carries an MLAG-enabled attribute on the bond. The attribute is crucial in allowing the kernel to properly enforce the default interface state and traffic behavior before the interface is deemed MLAG operational by the MLAG daemon.

The MLAG id may also be carried as a bond attribute in the bonding driver, though we believe this is not strictly necessary and can be maintained by the user space MLAG daemon.

MLAG in 802.3ad mode

When an MLAG is in 802.3ad mode, LACP is running between physical nodes of the two logical nodes. Using the example of a dual homed host connecting to two switches, the host is running two LACP sessions, one with each switch. In order for the two links to aggregate under the same bond, the two switches need to present the same LACP system ID to the host. The LACP system ID is directly derived from the bond interface mac address. As such, this can be simply achieved by setting the same interface mac addresses on the bonds for the MLAG on both switches.

Here we propose that the system ID can be de-coupled from the bond MAC address. This allows the LACP system ID to remain static even when bond membership and MAC addresses change.

Note that the two switches should only be setting the common system ID on the bonds when they are ready to operate in MLAG mode. For example, during boot up, before the two MLAG daemons can communicate with each other, the bonds should not be set with the common ID. Failure to do so can lead to data path loops and other traffic problems.

MLAG Data Plane Model

A dual homed node may send and receive traffic from either or both the the links to the switches. It is important to enforce the following traffic forwarding rules:

- the same packet is not delivered to the node more than once via different paths that constitute the same MLAG
- packets sourced from a dual homed node should not be delivered back to the node via a different path

- loops must never be formed

In order to enforce these rules, the following mechanisms are used.

Duplicate filtering

A filtering mechanism that is enforced when packets egress the ISL. A packet that traverses the ISL can be forwarded to single homed nodes but dropped if it is to be delivered to dual homed hosts. In Linux, ebtables is one possible option to implement that behavior. An ebtables rule matching the ISL as input interface and the output interface to the dual homed host can be installed to drop packets. When the dual homed host becomes single homed, the ebtables rule can be removed. Using ebtables, however, may not scale, as one rule is required per output interface, and in the non VLAN-aware bridge model, there can be many VLAN devices on each output port that belongs to many bridges. One simple way to solve this problem is to introduce the bridge port attributes of 'ISL' and 'dual-connected' to allow the bridge driver to make filtering decision based on them.

Interface States

When an MLAG-enabled bond is created the bonding driver places the bond in the protocol-down state. This new interface state causes the member interfaces to be carrier down. This state is necessary because it signals to the partner at the other end of the member link that the link is not available for use. When the MLAG daemon takes control of the bond, it will change the state to dormant. The dormant state ensures that only protocol traffic e.g. LACP PDUs can pass and not general data packets. When the MLAG control plane determines that the MLAG is formed and this MLAG member bond interface can start forwarding traffic, the MLAG daemon will clear the dormant state.

Split brain handling

In the case of a split brain situation, e.g. the ISL is down, all MLAGs need to be broken up in order to avoid loops and incorrect traffic behavior. For 802.3ad mode MLAGs, it is possible to change the LACP system ID on one side to cause the LACP peer to disaggregate the bond and keep one aggregator inactive. However, for non 802.3ad MLAGs, this is not an option. A more robust and universal way to achieve this is to allow only links on one switch to be operational and keep the links on the other switch carrier down. This is achieved with the new protocol-down interface state. The reason for keeping the links carrier down is such that in the non-802.3ad mode the link peer does not attempt to send traffic which is only to be discarded, thus causing a traffic black-hole.

Since disabling links is considered a drastic, but necessary, requirement, care should be taken to make certain that the logical switch has entered a split-brain situation. It is recommended that heartbeat mechanisms are used on multiple paths between the peer switches. Only when the ISL goes down and at least one heartbeat is still active are the switches in split-brain mode.

Note that single homed connection does not require any change when in split-brain mode.

MAC address and IGMP states Management

The following measures are crucial to the operation of MLAG in terms of robustness and traffic performance and convergence:

- A dual homed node can send packets with the same source MAC address on any or both the links to the two switches. The packets may also cross the ISL. This can produce the following problems.
  - when the MAC address is only learned on one switch, traffic destined to this address and sourced from the other switch will be flooded until the node sends packets with the same address towards that switch also, and there is no guarantee when this will happen
  - the source MAC address can move between the ISL and the MLAG, worst case on a per packet basis, causing constant MAC moves and potentially out-of-order packets

The proposed solution is for the MLAG daemons on both switches to synchronize their MAC address databases and disable MAC address learning on the ISL.

Similarly, IGMP reports and queries may be seen by only one of the switches. The MLAG daemons can synchronize IGMP group membership and router ports information so that multicast traffic distribution can converge faster.

Spanning Tree operations with MLAG

Spanning Tree ensures a loop free Layer 2 topology. In an MLAG environment where a pair of switches operate logically as a single switch from the forwarding perspective, some modifications are needed for Spanning Tree to operate correctly. More importantly, in the case of link or node failures or MLAG transitions, Spanning Tree must be able to detect and break loops.

The approach presented here requires the minimum amount of Spanning Tree state synchronization between the MLAG switch pair. Each switch calculates the topology independently based on its BPDU exchange with its neighboring nodes, with the following additional logic:

- Spanning Tree identifies which is the ISL
- a BPDU received on a root port needs to be processed and also relayed across the ISL to it's MLAG peer switch.
- The relayed BPDU needs to carry information about the ingress port. In case the BPDU is received on a dually connected link, the link identification should allow the peer MLAG switch

to match it up to the local interface that is part of the MLAG, making the BPDU appear to be coming on the MLAG from the peer logical node. The simplest way to do that without introducing tunnel encapsulation is to overwrite the BPDU source MAC address with the LACP system id of the peer logical node. In case the MLAG is non-802.3ad, the source MAC address can be encoded with the MLAG ID that is common between the two switches.

- Spanning Tree does not originate BPDUs on the MLAG member link if the switch MLAG role is secondary.
- Spanning Tree sends and receives BPDUs on singly connected links as normal, even if the switch MLAG role is secondary.
- The MLAG switch pair uses a common bridge id when generating BPDUs. The MLAG switches stop using the common bridge id when MLAG is no longer operational.

Other Considerations

MLAG Control Traffic

The MLAG daemons exchange information regarding the MLAGs and interfaces. The daemons should also be sending periodic keep-alives to ensure the liveness of the peer such that it can react quickly in case of the peer not alive. This traffic crosses the ISL, and it is important to ensure that the traffic:

- is given high priority versus normal data traffic
- is independent of the network topology even when the ISL is determined to be part of a Spanning Tree loop and is blocked

To achieve the first goal, the management traffic can be assigned a more preferable traffic class and obtain preferential treatment in queueing and scheduling.

To achieve the second goal, we propose the ISL be configured as a VLAN on the ISL that is not part of any bridge.

Acknowledgements

The authors wish to thank Cumulus Networks which funded the development of the MLAG implementation used as the basis of this paper.