April 2023

DETECTING AND REPORTING STACK SPLITS IN A STACKABLE SWITCH

Zach Cherian

Julie Quan

Lalitha Devi Appasani

Manish Jhanji

DETECTING AND REPORTING STACK SPLITS IN A STACKABLE SWITCH

AUTHORS:
Zach Cherian
Julie Quan
Lalitha Devi Appasani
Manish Jhanji

ABSTRACT

Stack splits are a commonly encountered failure for stackable systems within enterprise networks, where there is a possibility of multiple active stacks forming from the breakdown of a single stackable system. Proposed herein are techniques involving a management station-driven approach that provides for the ability to identify/enumerate stack splits and alert network administrators that appropriate actions to resolve failures can be taken. Further, techniques proposed herein may facilitate performing preventative actions by a network independently or/and in concert with a network adminstrator, in order to avoid network outages, chaos, and further network failures.

DETAILED DESCRIPTION

Stacking is a concept of grouping/configuring multiple switches into a single system such that there is a single control plane and distributed forwarding planes. Such a grouping of planes can appear as a single, large switch in the network. One advantage of stacking is that the stacks can be easily expanded or shortened by adding or removing single switches into a stack. Stacking also allows for the expansion of a network without disturbing existing switches, which can reduce cost, as well as management overhead.

Each switch has two ports located on the back panel that are used for stacking. The switches in the stack are connected via these stack ports using a stacking cable that forms a full ring. The stack port on the last switch can be connected to the first one to complete the ring and packets can be forwarded to all switches over the ring.

Many stackable switch architectures can allow up to 16 nodes to form a stack using a stack cable. However, a common problem that can be encountered in such architectures is that sometimes a stackable system can become broken in a manner that may result in multiple active stacks, which can be disastrous for the network.

For example, any failures on the switches in a ring of a switch stack can break the ring. In such cases, certain operations can be utilized to determine alternate path to reach the switches. However, failures on multiple switches in the stack may result in some switches not being able to see other switches in the same stack. This scenario results in a stack split in which the stack is split into multiple segments in which each segment may have one or more switches being identified as single stack and switches in one segment cannot see switches in the other segment. Thus, the problem with a stack split scenario is that the split stack cannot be identified from any of the switches that are/were part of the stack.

Such a split stack can result in network outages as the clients connected to some of the switches can lose connectivity. Other problems may also be caused, such as packet drops as the split stacks use the same stack and bridge Media Access Control (MAC) address or same management IP addresses.

This proposal provides various techniques through which detection of split-brain or split-stack situations can be provided from within the network and a management station vantage point so that a network administrator (admin) can take actions to resolve such situations.

Some stackable switches now have advanced telemetry reporting via operational models that cover various aspects of switch operational states, including stackable topology. For the techniques of this proposal, a discovery protocol can be utilized that discovers stack topology using broadcasts. The protocol can be provided on each switch in a stack and can be responsible for electing active and standby switches in the stack and assigning unique switch numbers for all switches in the stack. An active switch can be responsible for complete stack management and the IP address and MAC address of this switch is used as stack identity. With the help of underlying platform drivers, the exact connections between the stack ports are also known.

The following information from the stack devices can be used to identify a stack split:

- Stack MAC address of the stack;
- Management IP address of the stack that was configured;
- MAC address of each stack member;

- Reload reason of each of the stack members;
- Stack ring status (full/half);
- Stack port connections; and/or
- Number of members in the stack

Every stack will have one active switch that is the point of management for all switches in the stack and one standby switch that can take over when the active switch goes down, in which the standby switch can then continue to manage the stack.

In the event of a stack split, however, the active switch and standby switches may or may not be in the same segment. Thus, there are two possible scenarios that may be detected for a stack split, as follows:

1. Active and standby switches remain in same segment – In this scenario, the separated switches in a different segment will reload and form a new stack, as they have lost connectivity to the older active and standby switches. New active and standby switches will be elected for this stack.
2. Active and standby switches in separate segments – In this scenario the old standby switch becomes the active switch and the rest of the members join the stack. Only a new standby switch will be elected for this stack. In this case, the switches don't reload.

Figure 1, below, illustrates an example scenario that may occur for a switch stack.

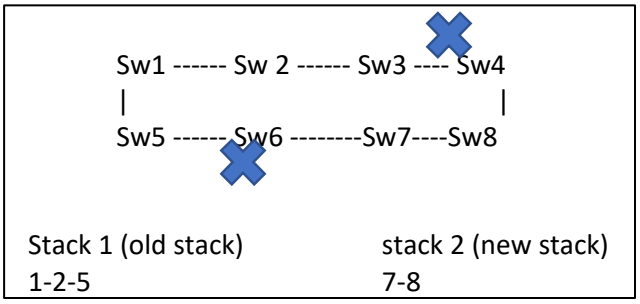

Figure 1 – Example Stack Split Scenario

As shown in the example of Figure 1, if a split occurs in the stack of 8 switches due to issues with switches 4 and 6, two separate stacks can form, one stack including switches 1, 2, and 5, and a second stack including switches 7 and 8.

Consider various operations that can be performed utilizing techniques of this proposal in order to detect and identify a stack split that may occur for the first scenario in which the active and standby switches remain in the same segment, as follows:

- The reload reason on the stack members in the new stack will be "Lost active and standby." For example, as all the switches in this segment reload and come back up and form new stack(s), the last reload reason will be "Lost active and standby."
- A duplicate IP trap can be seen for the management IP of the stack.
- Both stacks may be using the same IP address as their management as they share same operating system configuration, unless the switches have acquired an IP address using Dynamic Host Configuration Protocol (DHCP).
- A reduced number of stack members can be detected in both of the stacks as stack member count reduces as the stack is split into two or more stacks.
- The stack ring status can be detected as Half in both of the stacks. For example, the stack ring is broken so the status changes to Half from Full ring (even when there are 2 or more segments).
- Stack port connections remain same in both the stacks. For example, there is no change to the physical connections of the switches, so connections remain the same except for the removed switches.

Further, consider various operations that can be performed utilizing techniques of this proposal in order to detect and identify a stack split that may occur for the second scenario in which the active and standby switches end up in different segments, as follows:

- The new stack will have the same stack MAC address as that of the older stack but it will be marked as a foreign MAC address as MAC address does not belong to any member switch in the stack.
- A duplicate IP trap can also be detected for the management IP address of the stack as both stacks will use the same IP address as their management IP address as they share the same operating system configuration.

- A reduced number of stack members can be detected in both of the stacks as stack member count reduces as the stack is split into two or more.
- The stack ring status can be detected as Half in both of the stacks as the status changes to Half from Full ring when the stack ring is broken.
- Stack port connections may remain same in both the stacks. There is no change to the physical connections of the switches, so connections remain same except for the removed switches.

Advantageously, the techniques of this proposal can be adopted by any controller/management station that manages networks and is receiving telemetry information and/or can query switches for such telemetry information.

Upon detecting a stack split, various quick remedial actions that can be performed by a network controller/management station can include shutting down one or more split segments that earlier were part of the same single stack. A self-healing technique, as discussed in further detail, below, may be utilized to automatically mediate and contain a failure. This can allow a network to remain in operation without chaos due to multiple segments each declaring themselves as L2/L3 peers to other switches/routers in the network. The self-healing may then be followed up with resolution type activities at the controller using trouble-ticketing workflows, such as fixing/replacing faulty stack-cabling connecting the isolated segments or other faulty hardware modules.

The automatic self-healing technique can be used to prevent network chaos without restoration to the pre-split state of a split stack. To facilitate such a technique, the stack or each of the stack split segments, as the case may be, can advertise their presence via Layer 3 (L3) advertisements provided for the connected uplinks (e.g., connected to upstream networks and the management station) to an upstream node towards an enterprise core. The advertisements can include unique identifying stack node information (e.g., MAC address) that allows a receiving entity to form a view of the stack or be aware of segments but continue to map the segments to the intact stack that existed before the failure/split.

This scheme involves the cooperation of an upstream switch which can be referred to as a StackMonitor (SM) service node that exists outside of the stack and is to reflect or otherwise acknowledge the L3 advertisements back to the downstream switches. In one

instance, such a host endpoint feature can be implemented within a switch offering the SM service.

Recall that this proposal involves scenarios in which there can be two or more failures in a stack ring leading to stack split segments where the ability to communicate between all member nodes (full mesh) is no longer working and only member nodes that are part of the same surviving segment have a full mesh communication capability. Thus, an external entity, such as an SM service node, needs to be involved to facilitate communications between surviving stack split segments.

The L3 advertisements and other exchanges can be provided as an extension to existing stack discovery protocol utilized to determine stack topology that supports User Datagram Protocol (UDP) communications in order to utilize anycast transmissions. Figure 2, below, illustrates example details that may be associated with realizing the self-healing technique of this proposal for a campus network.

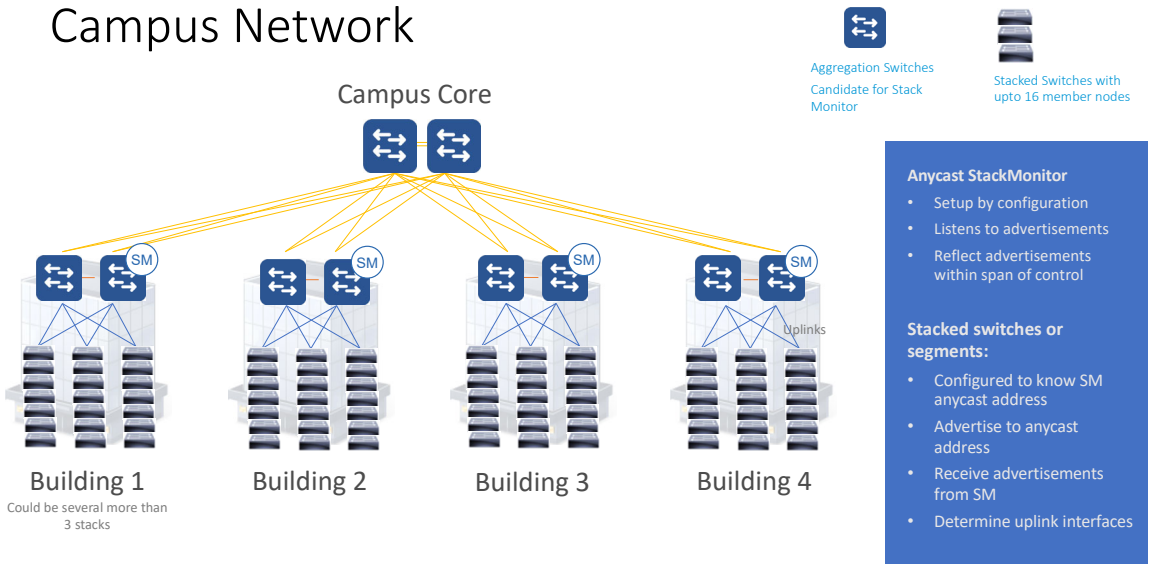

Figure 2: Example Self-Healing Scenario Involving a Campus Network

The upstream StackMonitor (SM) node can be setup through configuration and the SM service can be listening on an anycast destination address. Standard routing configurations can be used to propagate routes to these anycast SM services. The SM can unicast an acknowledgement back for every advertisement.

Further, stacks can be configured to know the anycast SM address. The advertisements from stacks in the access layer can be periodic and low frequency (e.g., once an hour or the like). These advertisements can also be used by the stack and the upstream SM node to determine their respective reachable exit interfaces. Within an enterprise network there can be multiple SM service nodes - ideally these would be switches in the distribution/aggregation layers. Each SM node may have a span of control (e.g., downstream stacked switches) in the access with which the SM node communicates.

The SM node switch can effectively build a view of the stacks and stack-segments from the advertisements received. The SM node is therefore aware of all the stacks and segments in its span of control. Essentially, the SM service maintains “directed group memberships” for a stack.

Further, SM listening to an anycast address offers resiliency and conserves the need for IP addresses. For example, if an SM service node should go down, other SM nodes can receive advertisements from that span of control and build out new stack views.

In this proposal, the management loopback interface (used to connect to the management station or a node that has L3 connectivity to an upstream SM) on a stack is to be configured to acquire an IP address via DHCP. Once a failure occurs, and a split-stack segment realizes its membership has reduced it must release its DHCP IP address and try to acquire an IP address again. Once an IP address is acquired, gratuitous Address Resolution Protocol (ARP) advertisements are to be announced on uplink connections, which may allow all the split-segments to continue to have L3 connectivity via unique IP addresses to/from an SM node.

Following a failure/stack split, all segments can immediately announce their stack node memberships and status (e.g., active_split, quiescent_split, etc.). Initially, all segments are quiescent_split. The old IP address can be specified as one of the fields in new advertisements being sent out in order to serve as an additional field that can be correlated to the previously intact stack.

As soon as a split status is detected by an SM node for an advertisement, the SM node will unicast-acknowledge such advertisements. The SM node will also send full information of the segments of a previously intact-stack to all the segments of the

previously intact-stack that it knows about. Figure 3, below, is a call flow illustrating various operations as discussed above involving a StackMonitor.

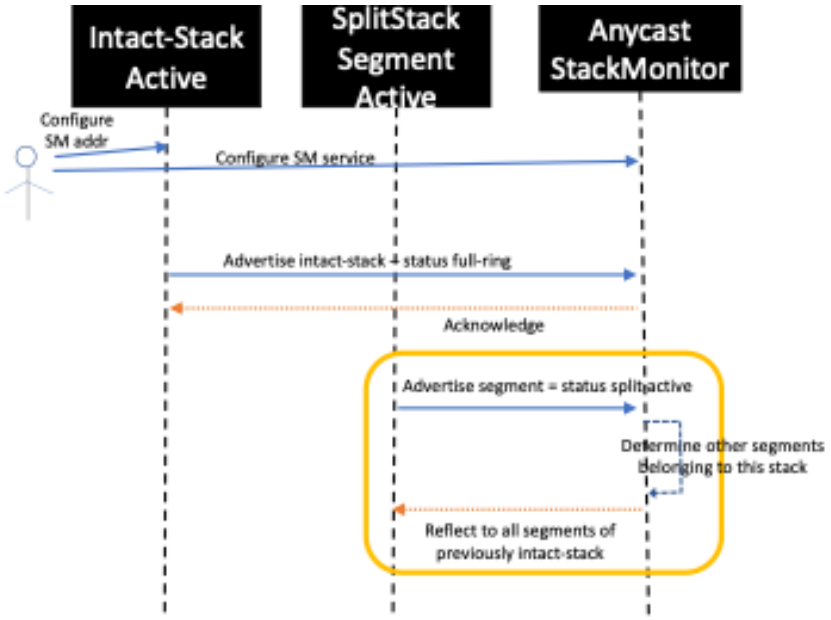

Figure 3: Example Operational Call Flow

During operation, in accordance with techniques of this proposal, a multi-level criteria can be employed at member nodes in order to determine which split segments should become quiescent until inter-stack link connections are restored. In one implementation, switch priority, size (number of ports) of a split segment, importance of ports (e.g., Power over Ethernet (POE) priority, trusted ports, etc.) and other information can be used as part of a policy that influences a scoring algorithm that can be used to maximize critical operating network infrastructure and minimize blast radius and disruption due to stack splits.

This score can be published as a field in the advertisements and can be computed for each member node of a stack while the original stack is intact and kept ready. The score can also be communicated in a full-mesh manner between member nodes while the original stack is intact. In some instances, the original Active switch can have bonus points added to its score so that post-split, if it has survived, the segment containing the old Active switch wins, to allow for L2/L3 peering continuity.

In such an implementation, all the stack segments can continue to report their stack node memberships and status (e.g., active, quiescent) to management stations in addition to an SM. The stack segments can also continue to use unique identifying stack node information as discussed earlier, which can allow a receiving SM service to continue to map the stack segments to the intact-stack before the failure or split.

Using the scores advertised by other segments, one segment can transition to an active_split status. Any segment(s) late to the party, even if they have a higher score, will lose their claim to active_split status. Effectively, an expedited variant of the discovery protocol can be employed in this election except that it is to be performed over L3 and an SM node is involved. The wait time (Tw) can be configured to be sufficiently small so that peering protocol relationships to the active_split segment do not time-out, when possible.

Further, the quiescent segments should effectively isolate themselves from L2/L3 protocol peering networks so that they do not respond to peering (forwarding) protocol messages and cannot forward data traffic from/to downstream clients. Thus, network-healing can be provided such that split segment does not disrupt the network but is reachable. The quiescent_split stack may still respond to link local protocols, such as Link Layer Discovery Protocol (LLDP), and perhaps to management traffic. This technique should work even if there are stacks in the aggregation or distribution layers of the network that continue to make use of the same anycast SM services. Even if a split stack is detected by the split-segments locally within the network itself, it is still important for the management station to detect the split and alert an operator to the split for recovery actions.

Accordingly, techniques provided herein facilitate a management station-driven approach that provides for the ability to identify/enumerate stack splits and resulting segments and alert network administrators so that appropriate actions to resolve failures can be taken. Further, techniques provided herein allow preventative actions by a network in order to avoid network outages, chaos, and further network failures.