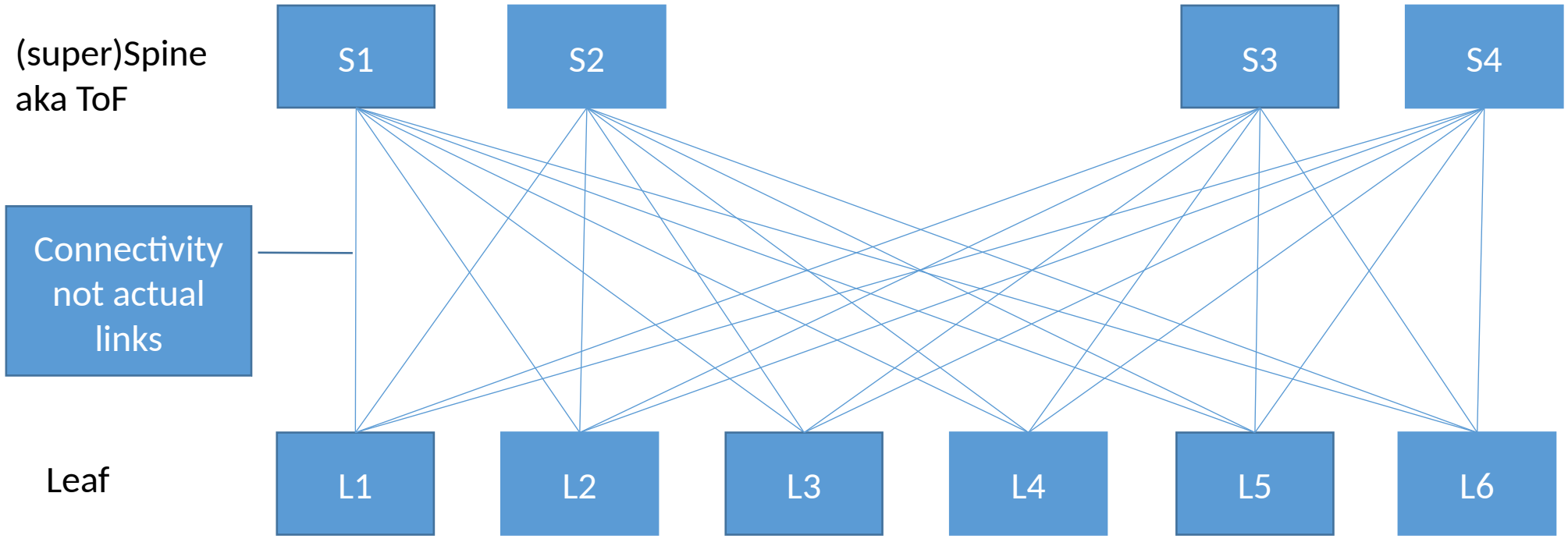
What we want to obtain: connectivity from any leaf to any leaf, and (lots of) well-balanced ECMP.

What we build: connectivity from any leaf to any spine node and then from any spine node to any leaf (logical Clos)

If possible we add redundancy in the connectivity to avoid fallen leaves, but this consumes ports

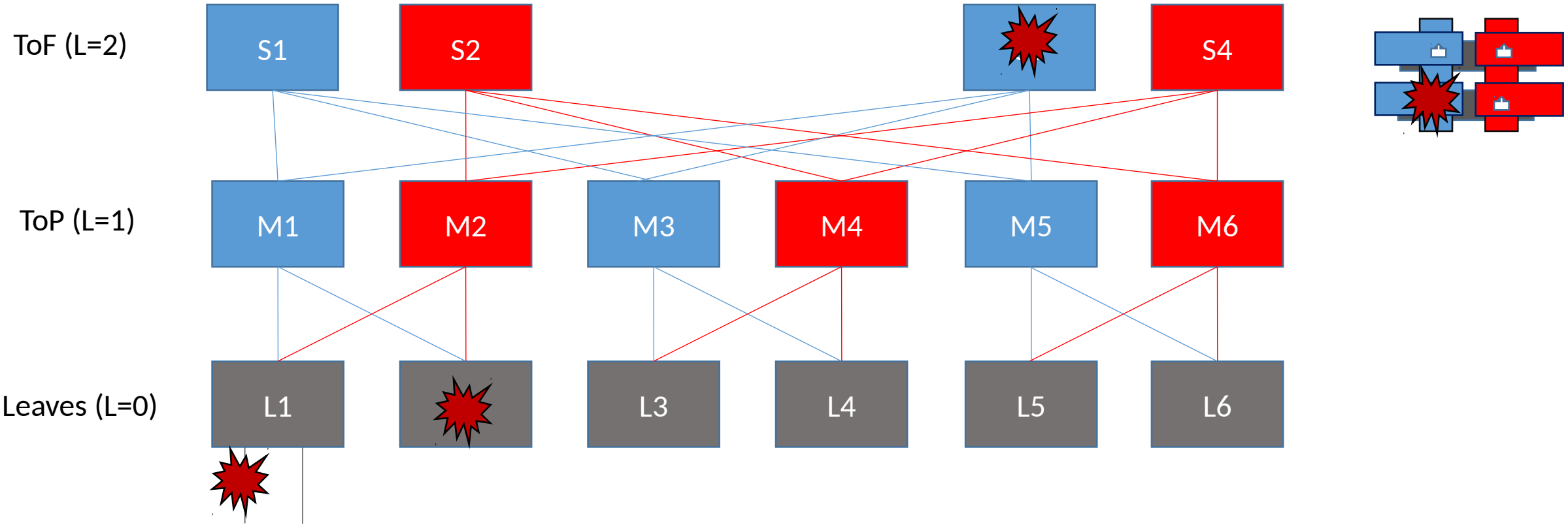
A **fallen leaf** is a leaf that is not connected to all spine nodes (connectivity below not ensured)



If the breakage is a southern link from a leaf Node going down, then connectivity to any node attached to the link is lost. There is no need to disaggregate since the connectivity is lost for all spine nodes in a same fashion.

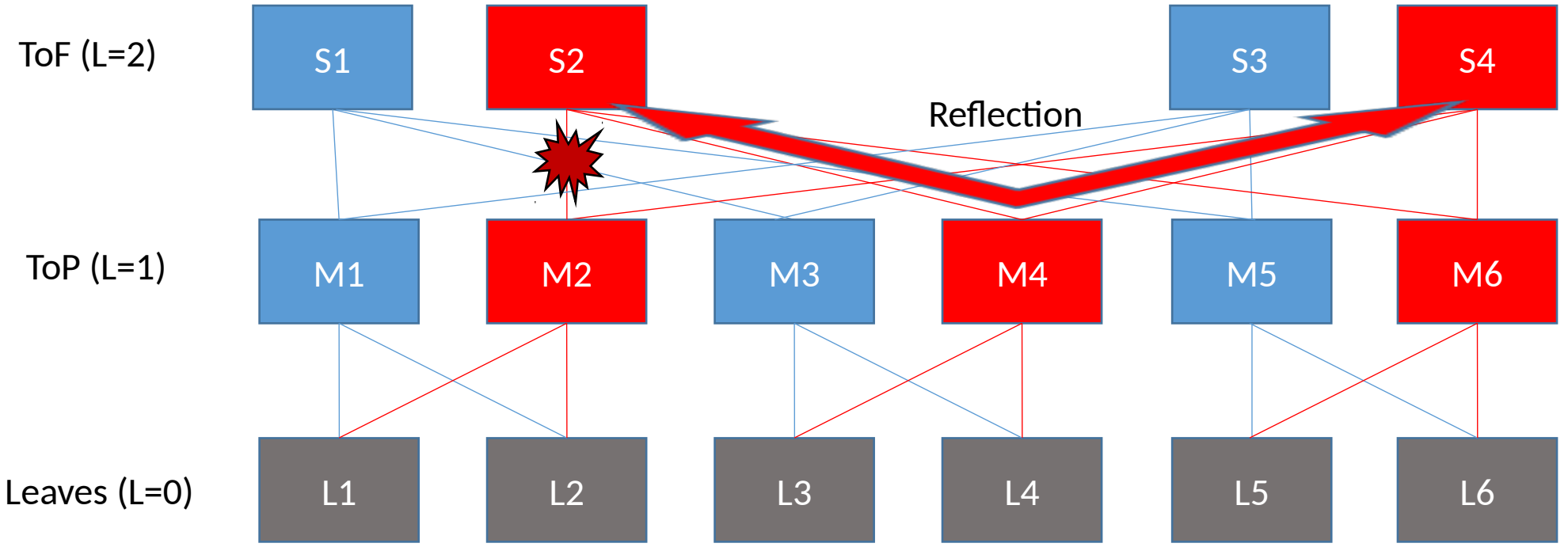
If the breakage is a leaf Node going down, then connectivity through that leaf is lost for all nodes. There is no need to disaggregate since the connectivity is lost for all spine nodes in a same fashion.

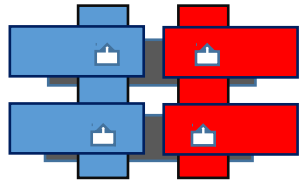
If the breakage is a ToF Node going down, then northern traffic is routed via alternate ToF nodes in the same plane and there is no need to disaggregate routes



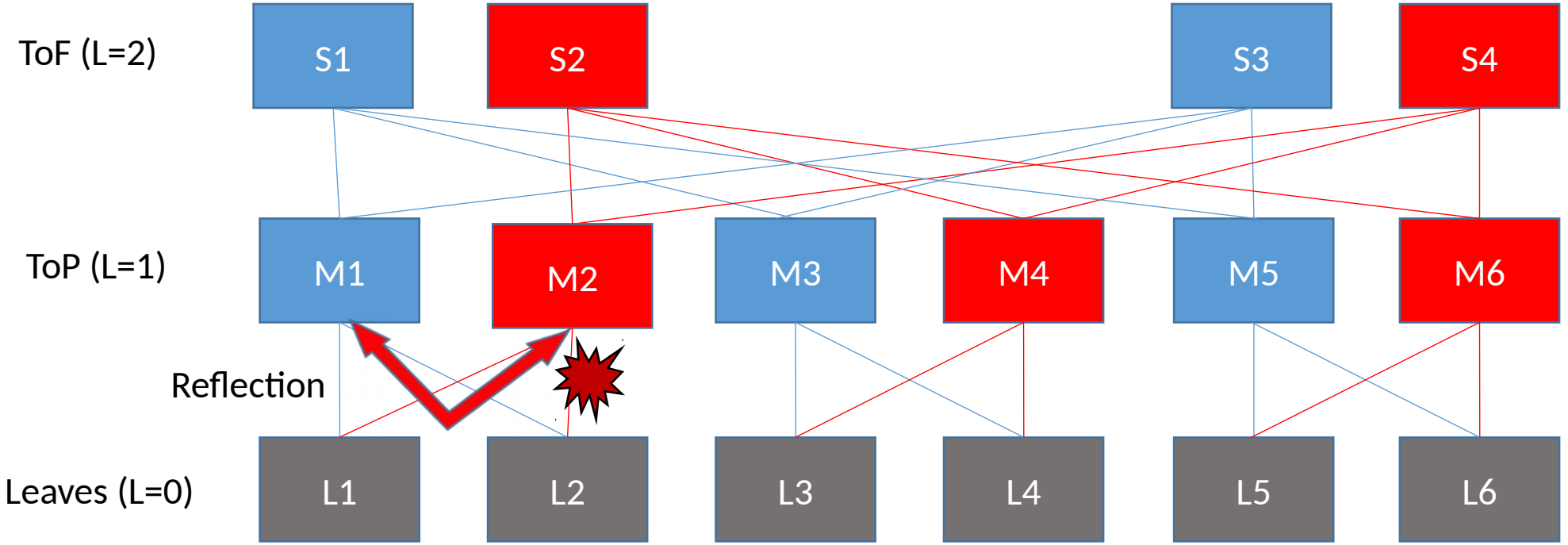
In a general manner, the mechanism of non-transitive positive disaggregation is sufficient when the disaggregating ToF nodes collectively connect to all the ToP nodes in the broken plane. This happens in the following case:

- If the breakage is the last northern link from a ToP node to a ToF node going down, then the fallen leaf problem affects only The ToF node, and the connectivity to all the nodes in the PoD is lost from that ToF node. This can be observed by other ToF nodes within the plane where the ToP node is located and positively disaggregated within that plane.

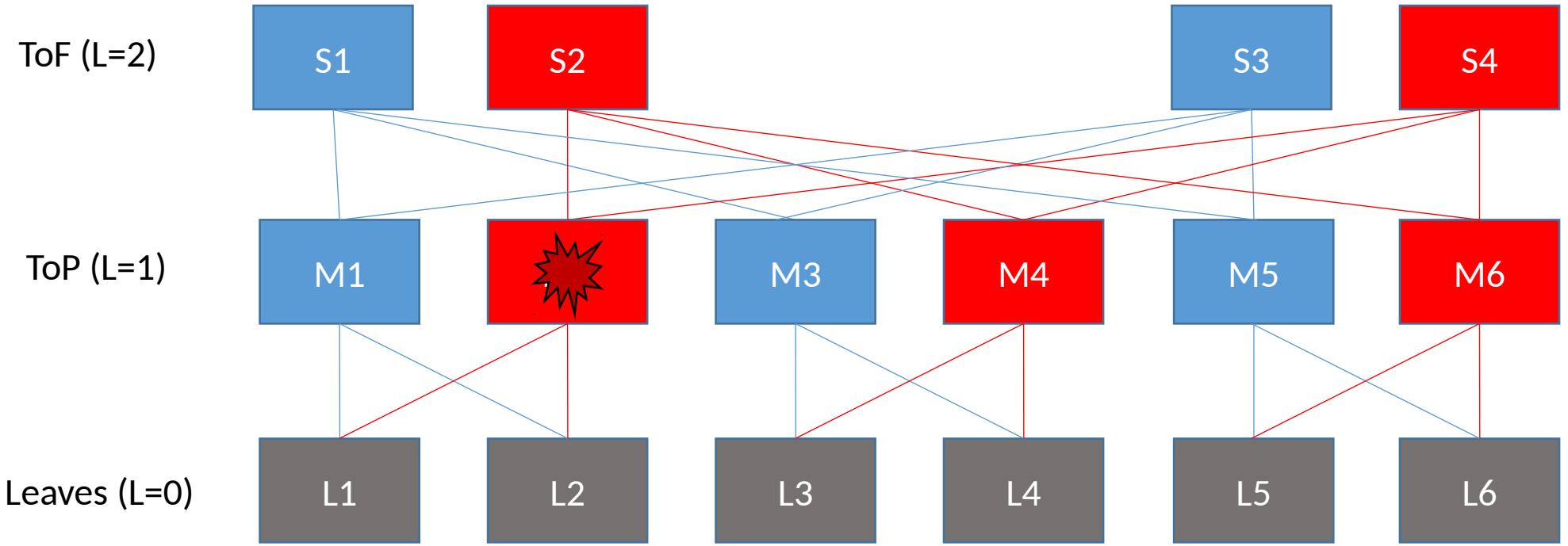




If the breakage is the last northern link from a Leaf node within a plane - there is only one such link in a maximally partitioned fabric - that goes down, then connectivity to all unicast prefixes attached to the Leaf node is lost within the plane where the link is located. Southern Reflection by a Leaf Node - e.g., between ToP nodes if the PoD has only 2 levels - happens in between planes, allowing the ToP nodes to detect the problem within the PoD where it occurs and positively disaggregate. The problem can be observed by the ToF nodes in the same plane through the flooding of N-TIEs from the ToP nodes, but the ToF nodes need to be aware of all the affected prefixes for the negative disaggregation to be fully effective. The problem can also be observed by the ToF nodes in the other planes through the flooding of N-TIEs from the affected Leaf nodes, together with non-node N-TIEs which indicate the affected prefixes. To be effective in that case, the positive disaggregation must reach down to the nodes that make the plane selection, which are typically the ingress Leaf nodes, and the information is not useful for routing in the intermediate levels



If the breakage is a ToP node in a maximally partitioned fabric - in which case it is the only ToP node serving that plane in that PoD - that goes down, then the connectivity to all the nodes in the PoD is lost within the plane where the ToP node is located - all leaves fall. Since the Southern Reflection between the ToF nodes happens only within a plane, ToF nodes in other planes cannot discover the case of fallen leaves in a different plane, and cannot determine beyond their local plane whether a Leaf node that was initially reachable has become unreachable. As above, the problem can be observed by the ToF nodes in the plane where the breakage happened, and then again, the ToF nodes in the plane need to be aware of all the affected prefixes for the negative disaggregation to be fully effective. The problem can also be observed by the ToF nodes in the other planes through the flooding of N-TIEs from the affected Leaf nodes, if there are only 3 levels and the ToP nodes are directly connected to the Leaf nodes, and then again it can only be effective it is propagated transitively to the Leaf, and useless above that level.



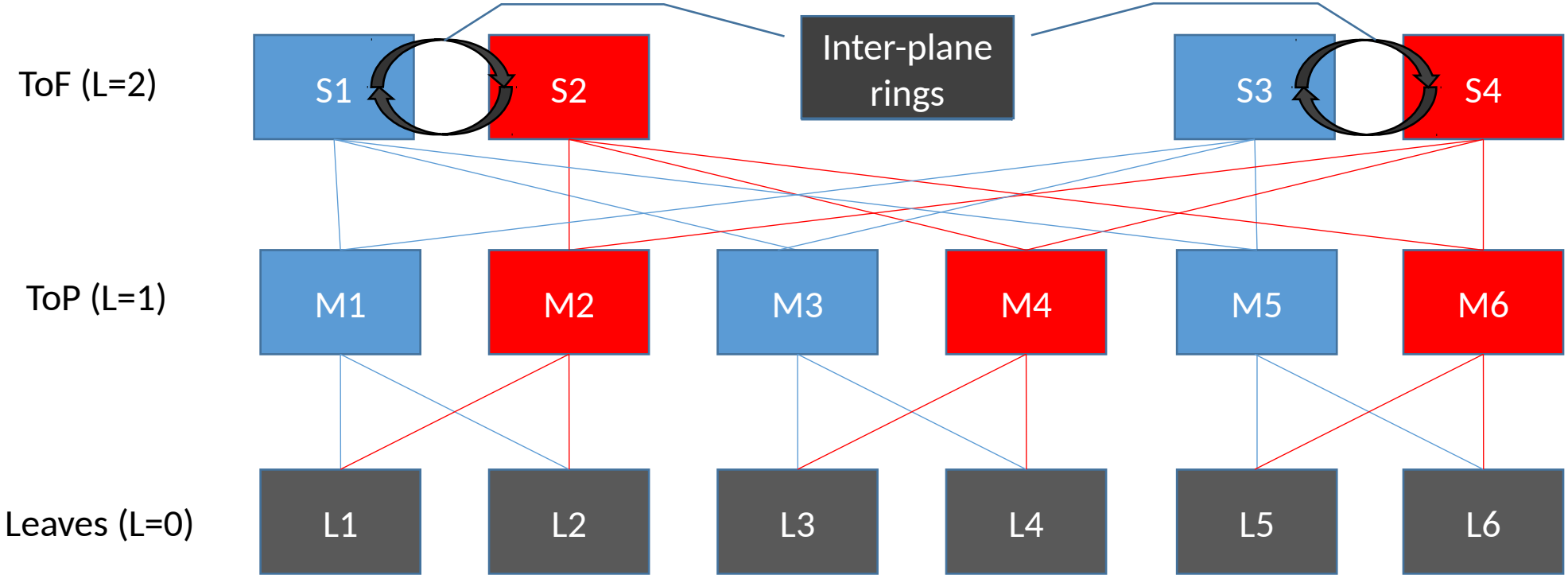
In case of a partitioned ToF, RIFT can use inter-plane rings to connect the dots between planes

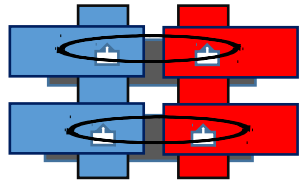
S1 and S2 synchronize prefix-related information (Non-Node N-TIES typically from leaves) over their ring

Same for S3 and S4. As a result

⇒ A breakage is detected by ToF nodes in the plane where it happens

⇒ Those nodes associate the breakage with fallen leaves, prefix N-TIEs seen over the Ring but not within plane



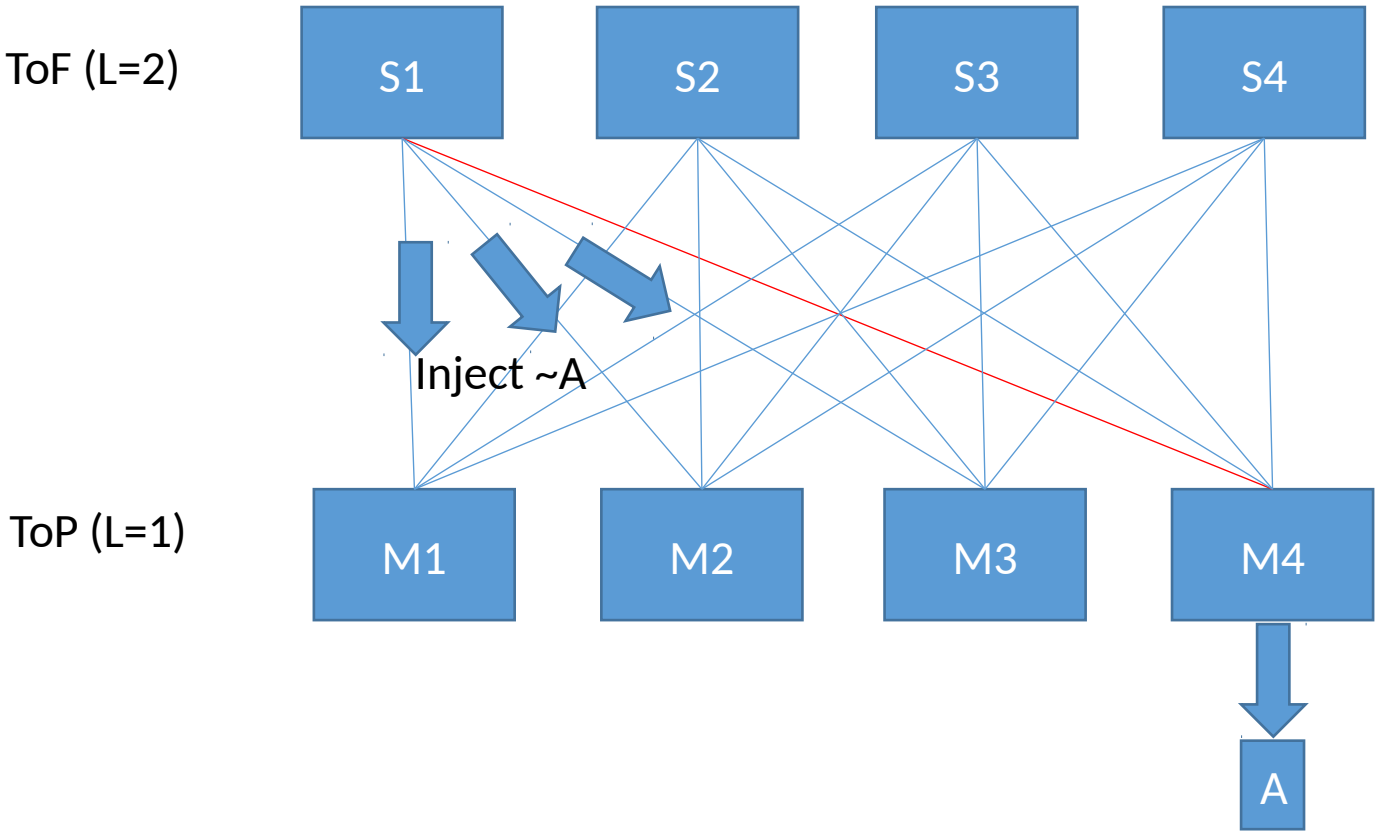


The process of negative disaggregation is as follows:

S1 figures that A exists and is not reachable. As a consequence S1 injects a new negative route to all of its children.

Upon that message the children install a route to A via all the parents from which they did not receive a negative route to A (that's S2, S3, and S4 here). This is 3 messages instead of 12 for the same route information

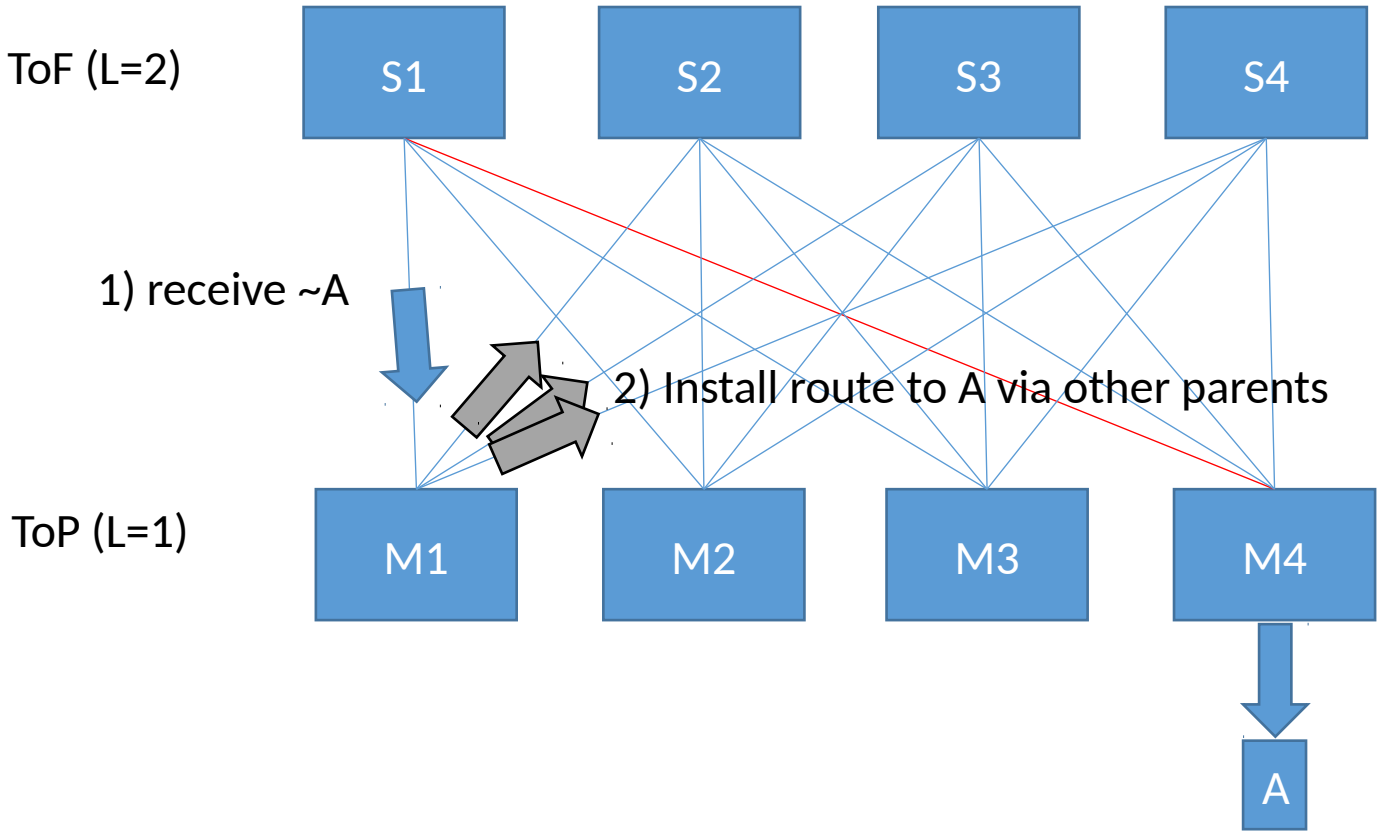
Transitive disaggregation operation: an intermediate node (e.g., M1) propagates the negative advertisement when it has it from all its parents, IOW as a consequence of receiving one from a last parent.

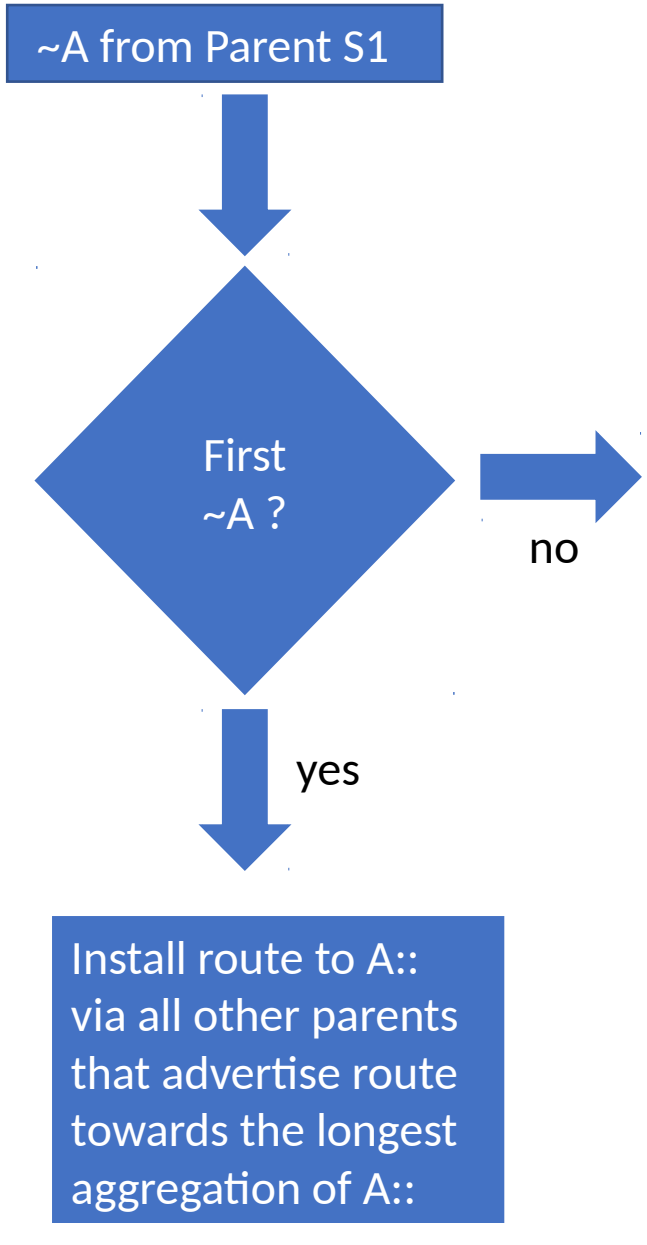


(perspective of M1). M1 gets the ~As in order.

Step 1: upon first ~A, say coming from S1:

- Select parents advertising reachability to the longest known aggregation that encompasses A, typically the default route
- Install a more specific route towards A via each of them but S1
- Remove route to S1 for negative routes to prefixes nested in A:: (recursive)





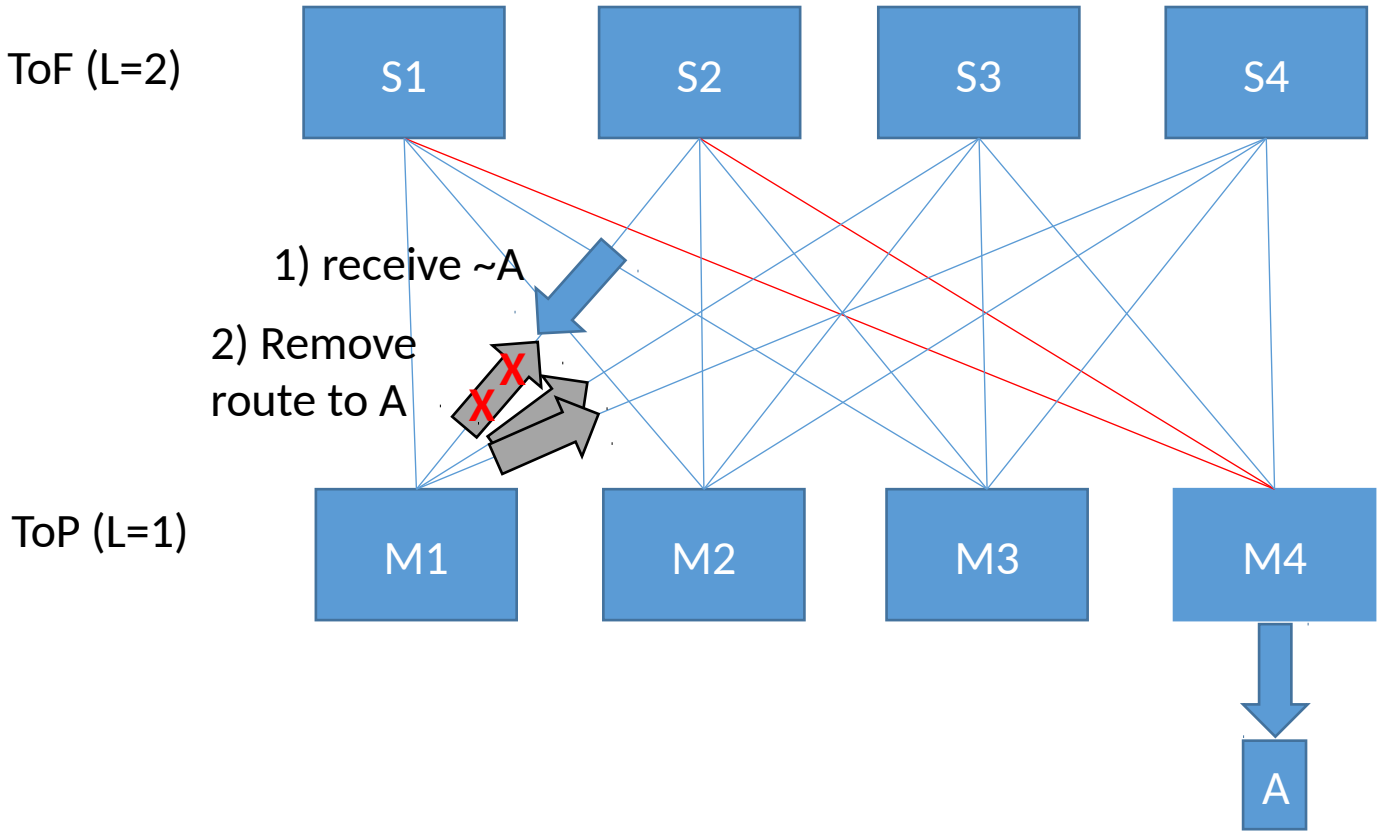
(perspective of M1).

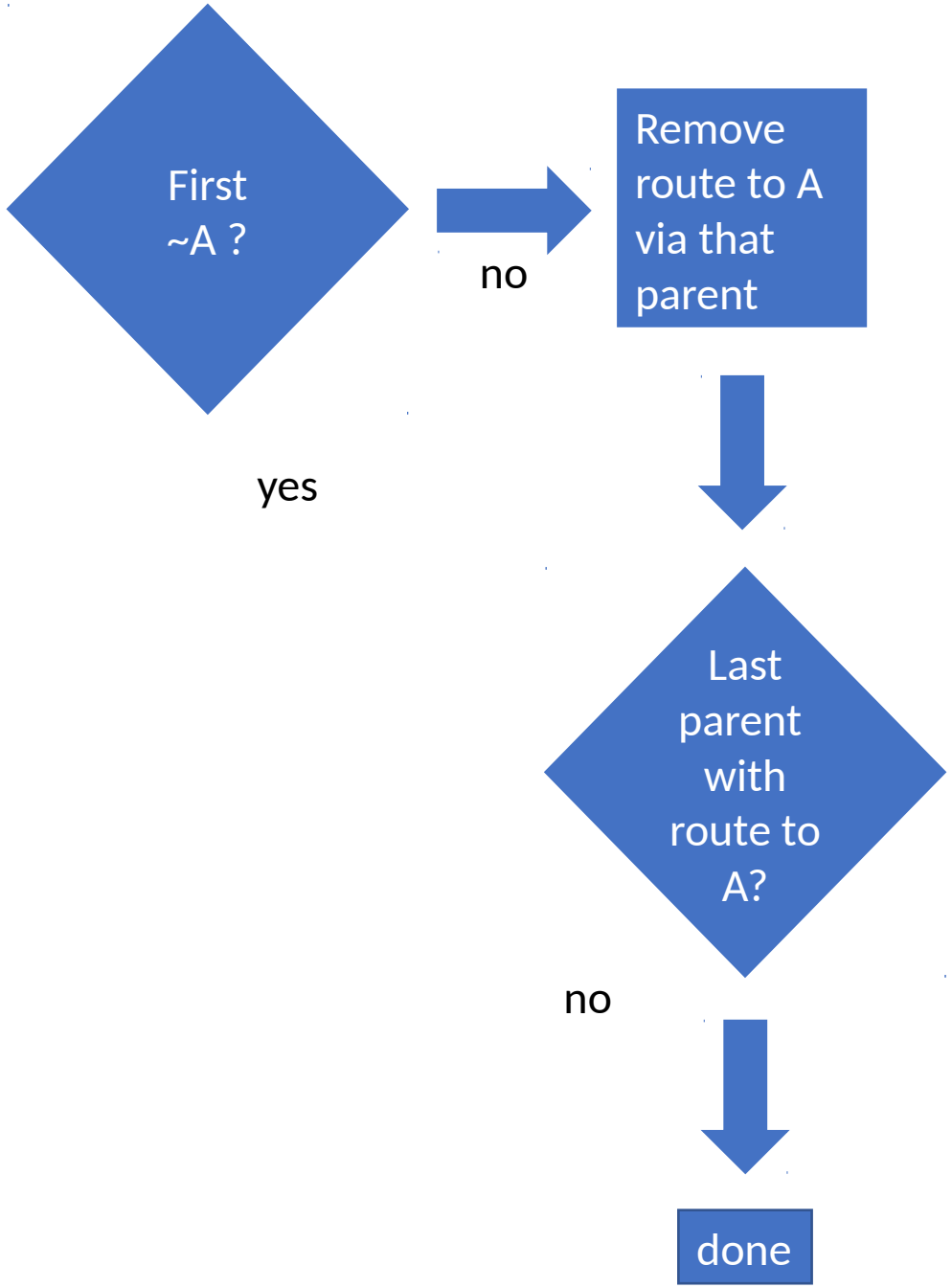
M1 gets the ~A advertisements in some order, uncontrolled.

Step 2: second ~A, coming from S2:

Route to A:: and nested negative prefixes via parent S1 are removed. This is not the last parent with a route to A, so do nothing else.

Next, a third ~A is received from say, S4. Same thing, routes via S4 for A:: and nested negatives are removed.



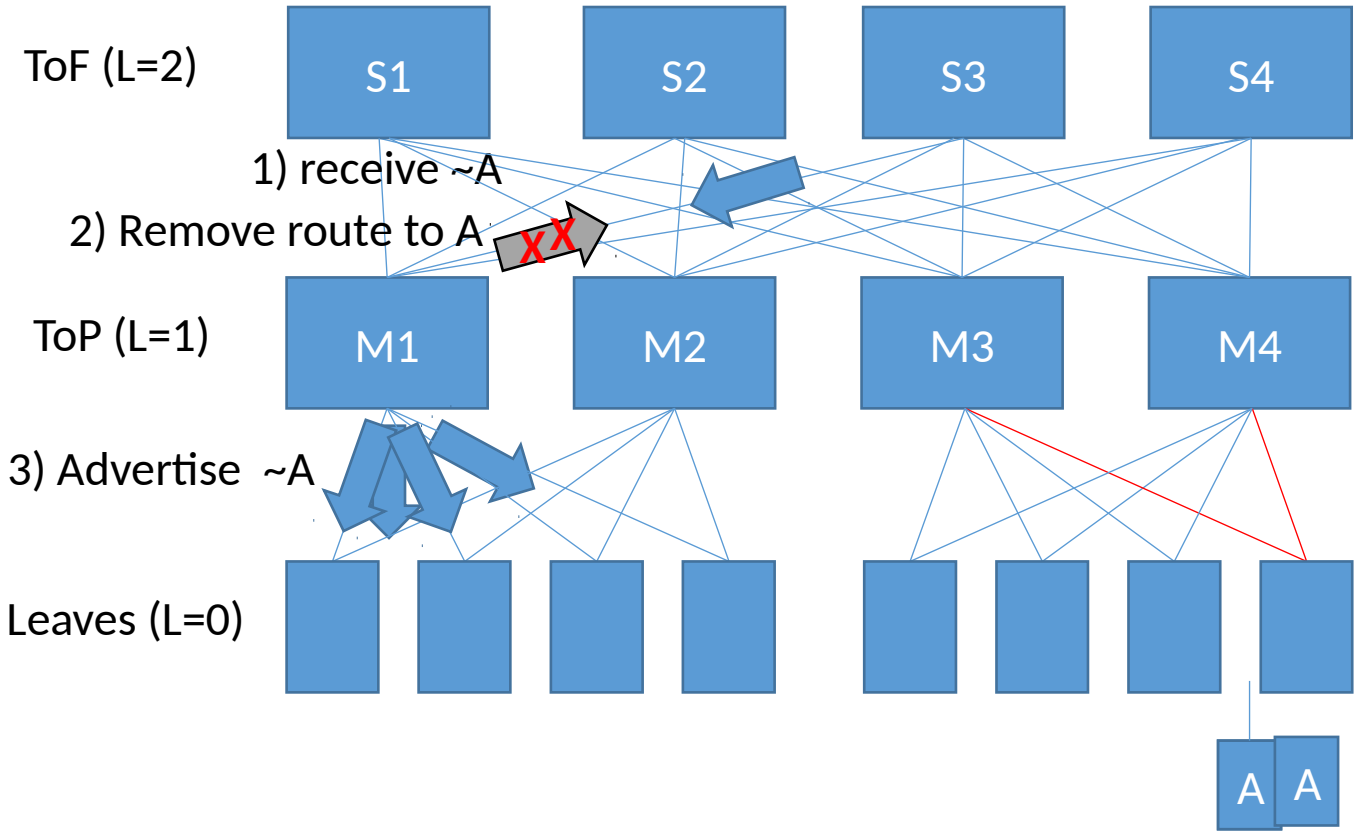


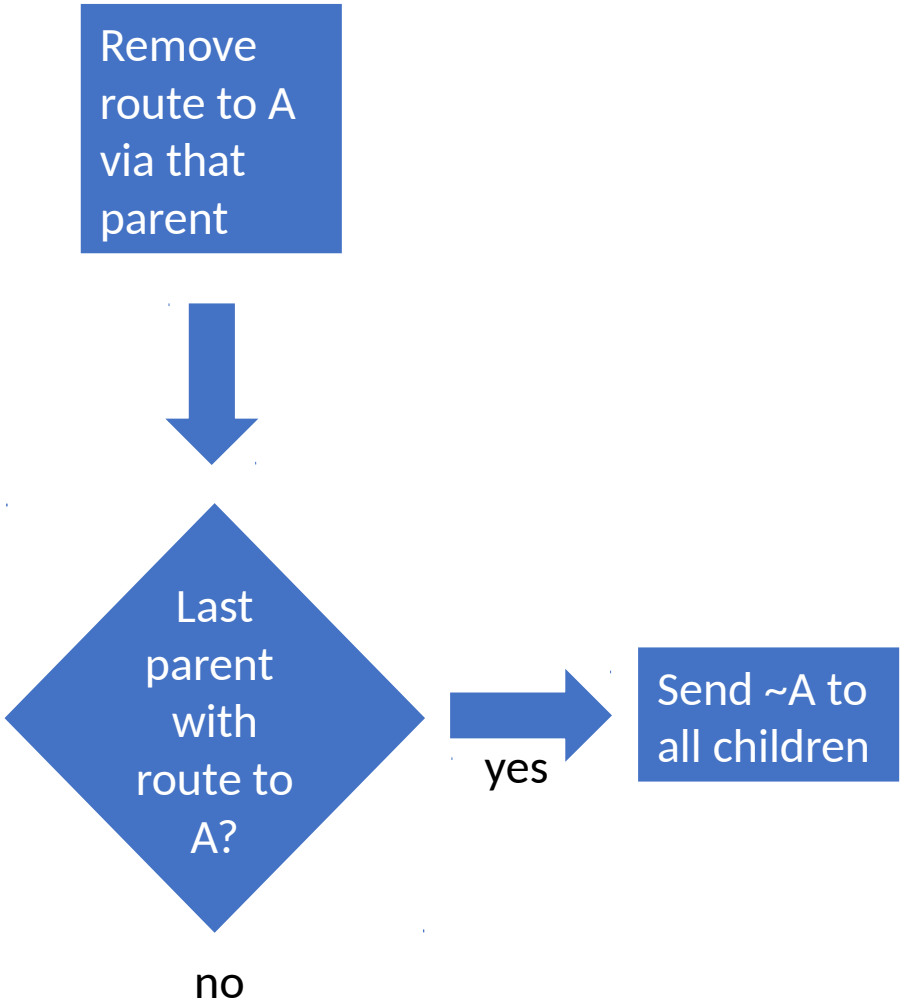
(perspective of M1).

Step 3: fourth ~A, coming from S3:

Route to A via parent S3 is removed.

This was the last parent advertising reachability to the longest known aggregation that encompasses A, so transitively send a ~A to all children





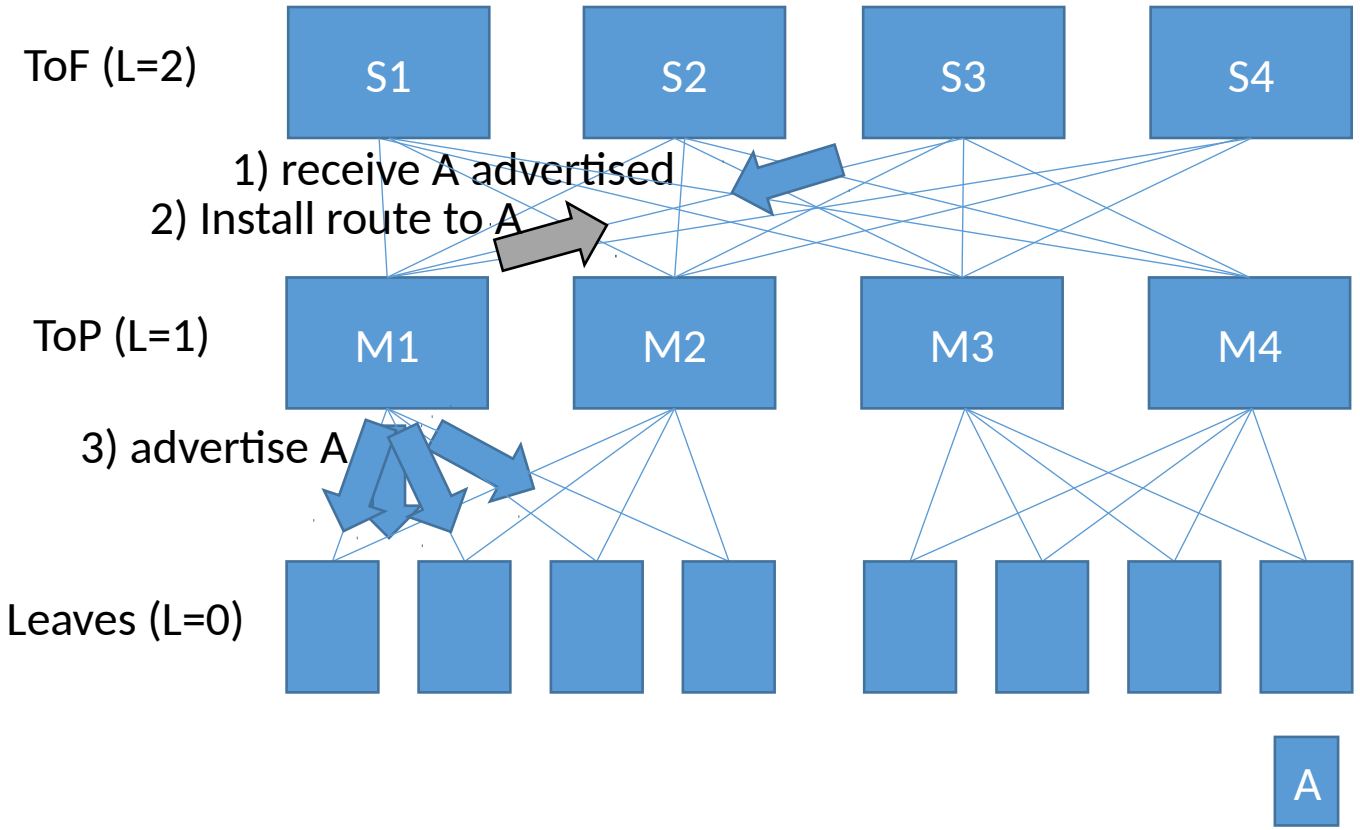
(perspective of M1).

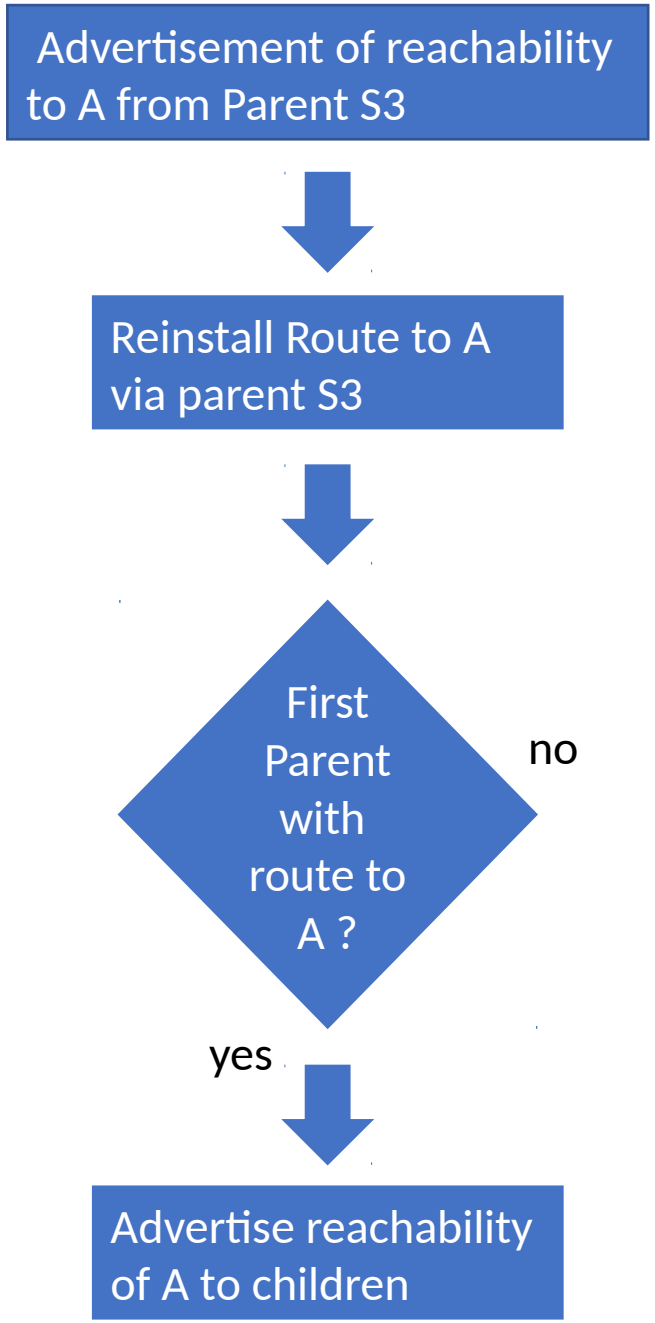
Link comes back up, information spreads in uncontrolled order

Step 4: A advertisement coming from S3 (really, undoing of ~A)

Route to A via parent S3 is reinstalled. Negative routes for prefixes nested in A:: are completed to add S3 as a feasible successor (recursive).

Since A is now reachable, M1 send advertises reachability to A again to children (again, as an undoing of ~A)

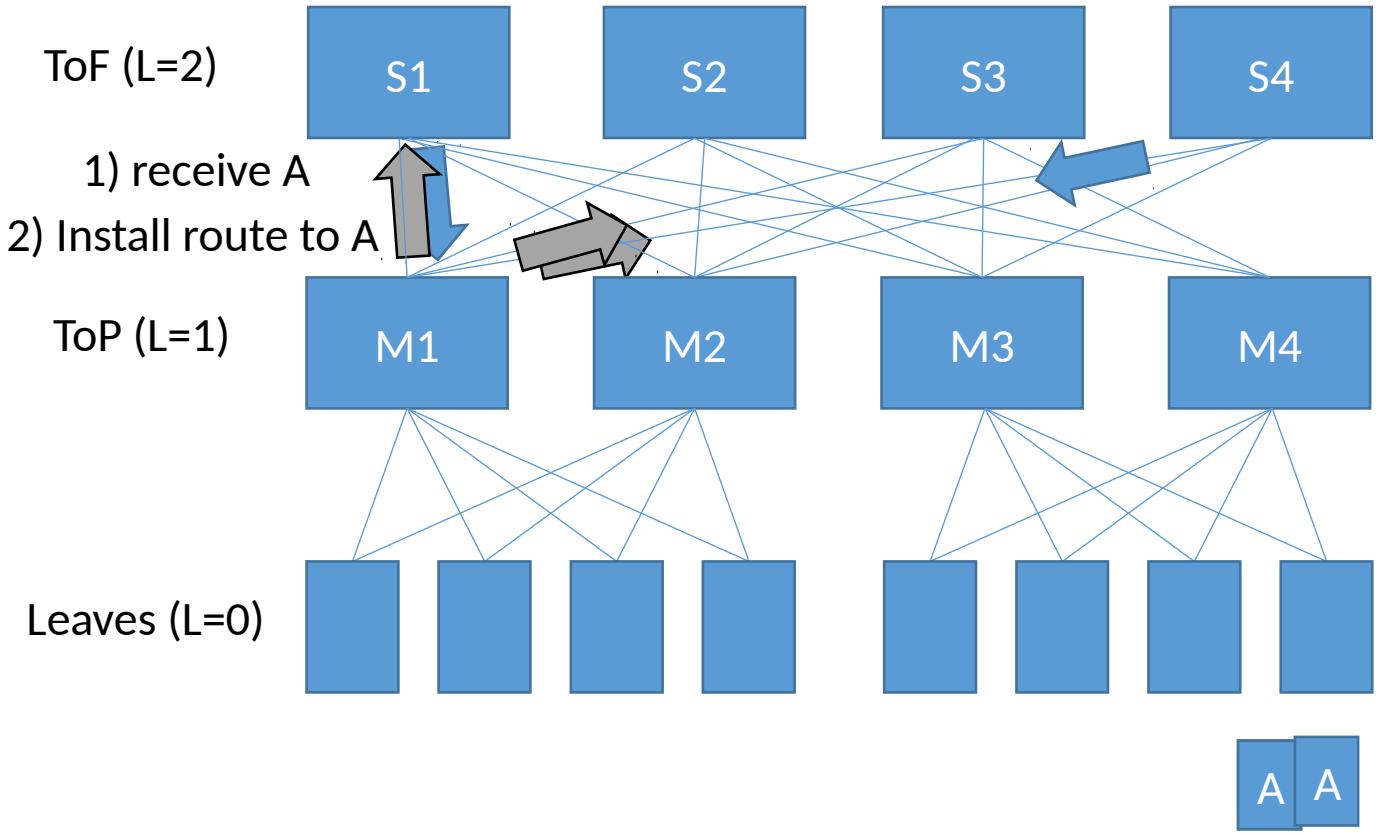


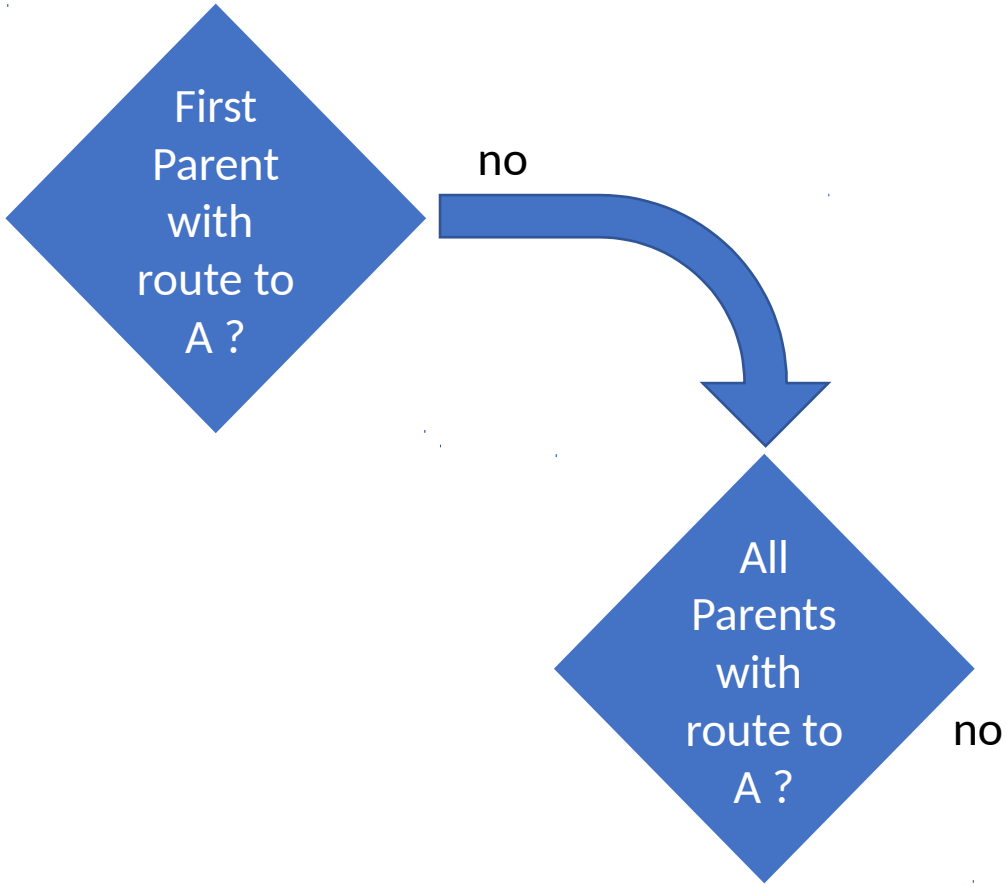


(perspective of M1).

Step 5: A advertisement coming from S1 and then S4 (really, undoing of ~A)

- Route to A via those parents is reinstalled, and recursively for nested negative routes.
- There are other parents that poisoned A (S2 here), so do nothing else



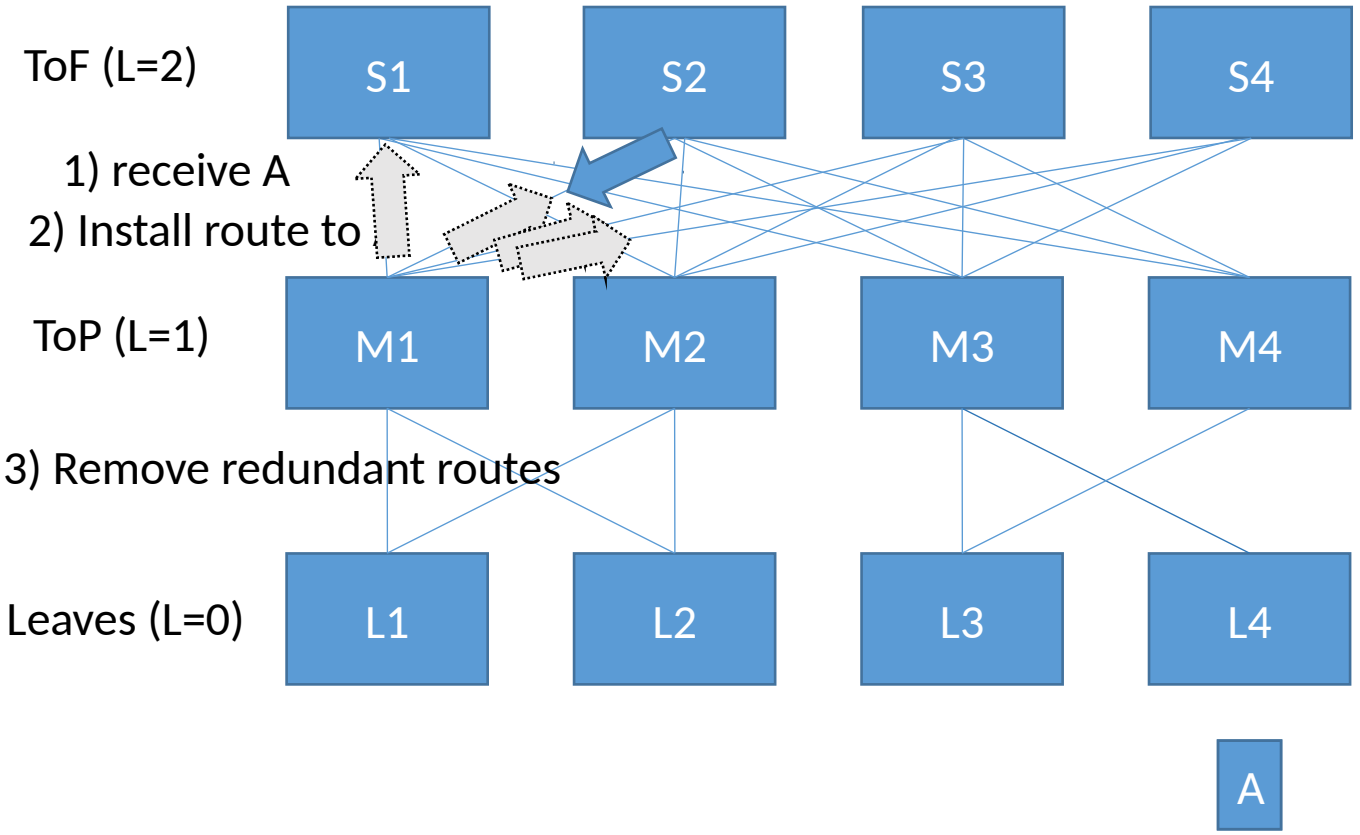


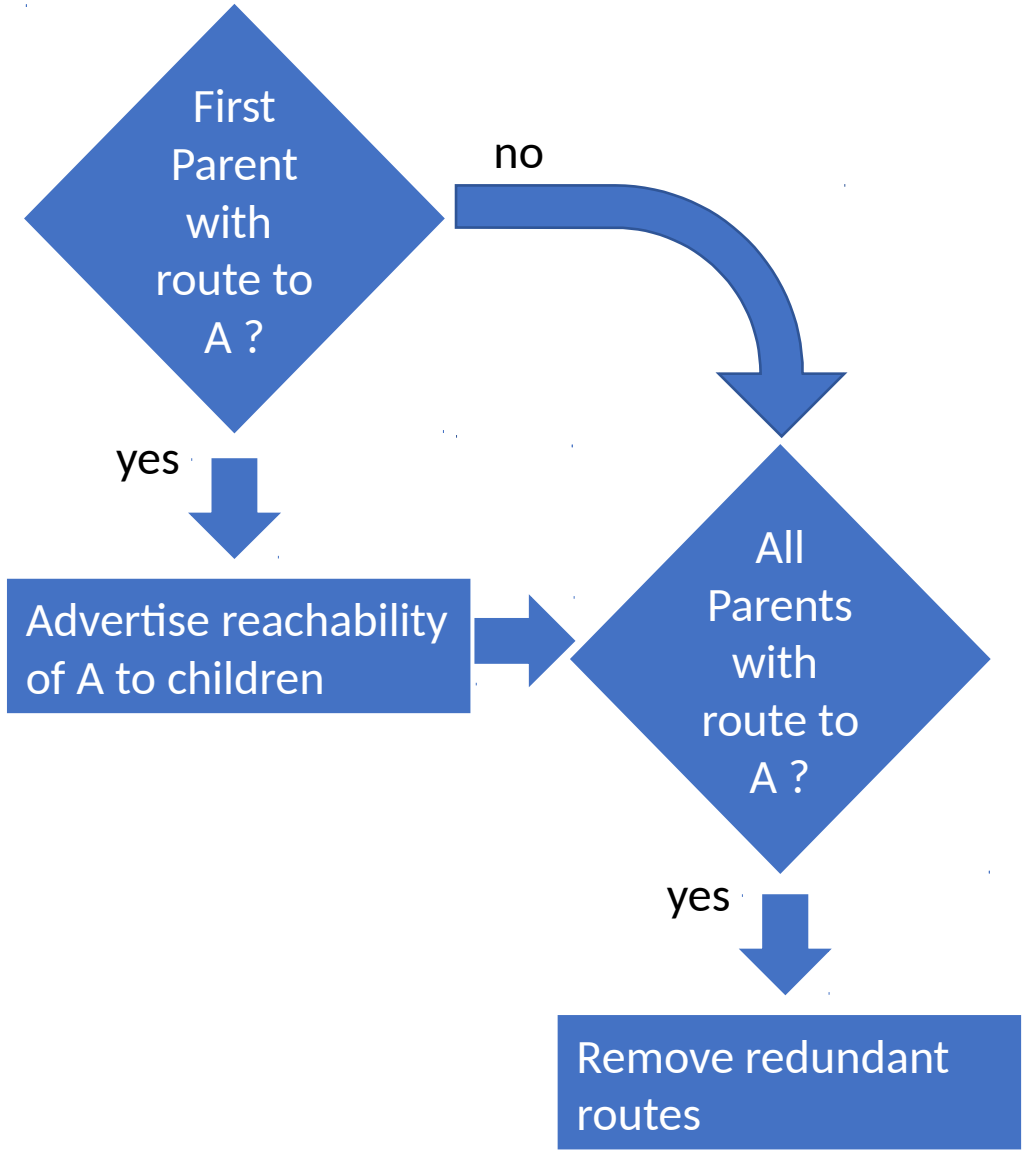
(perspective of M1).

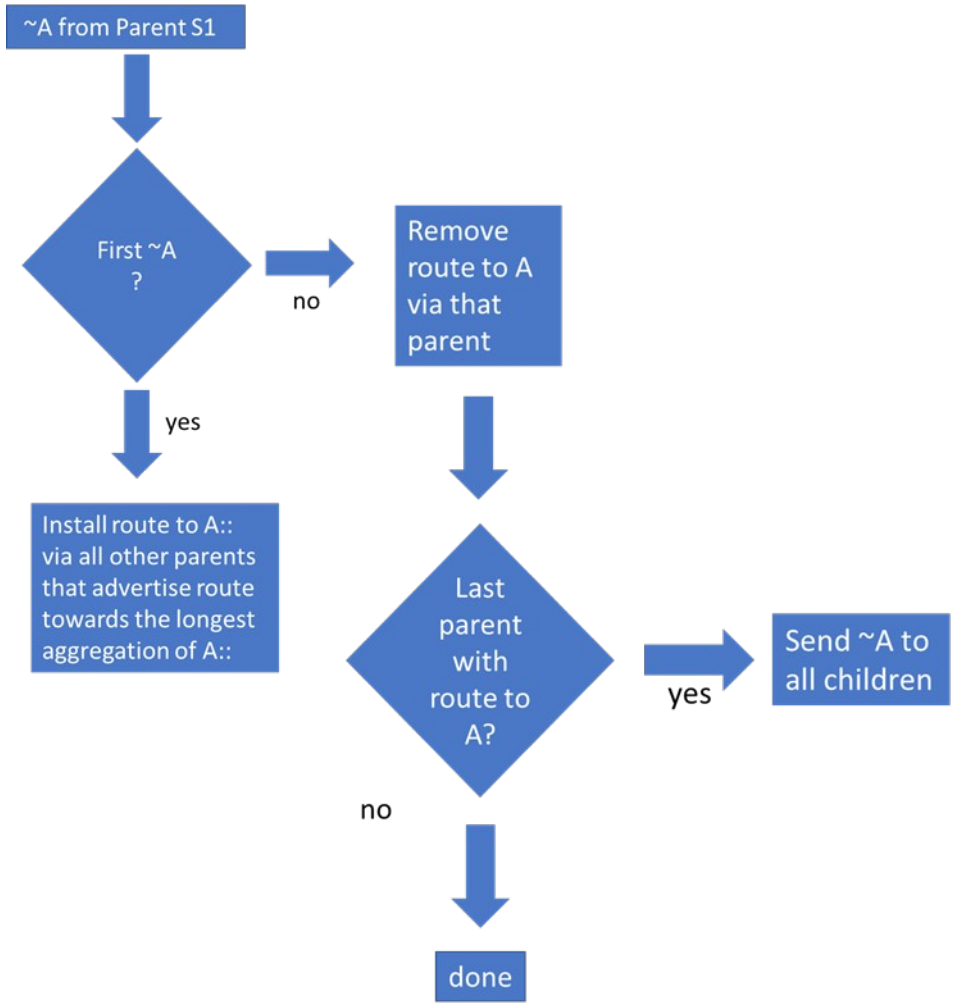
Step 6: A advertisement coming from S2.

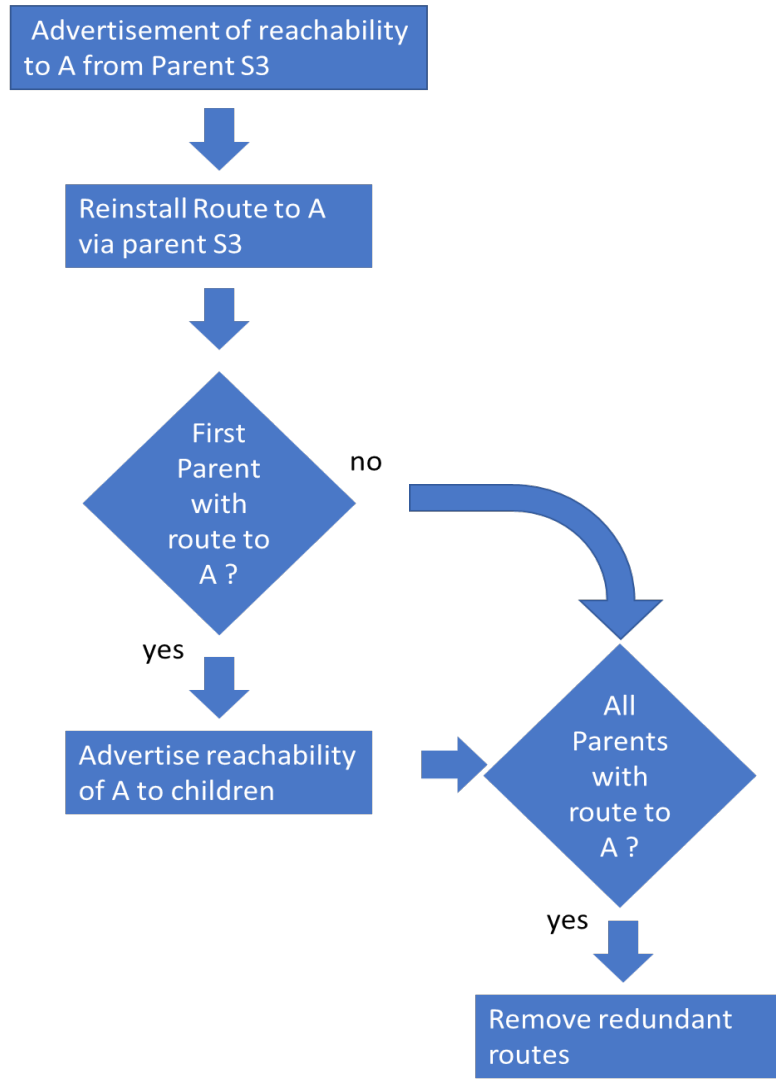
Route to A via those parents is reinstalled.

There are other parents that poisoned A, so do nothing else

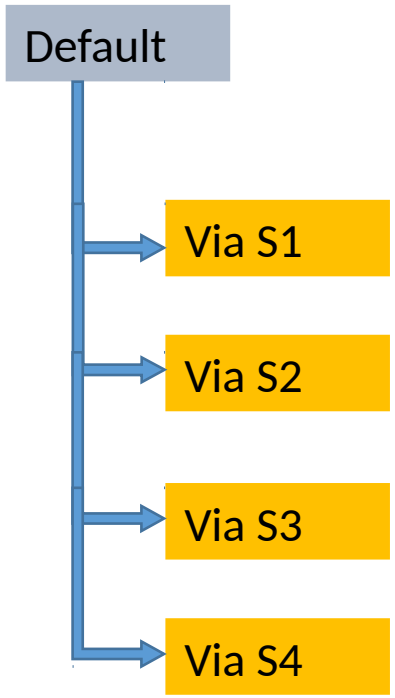




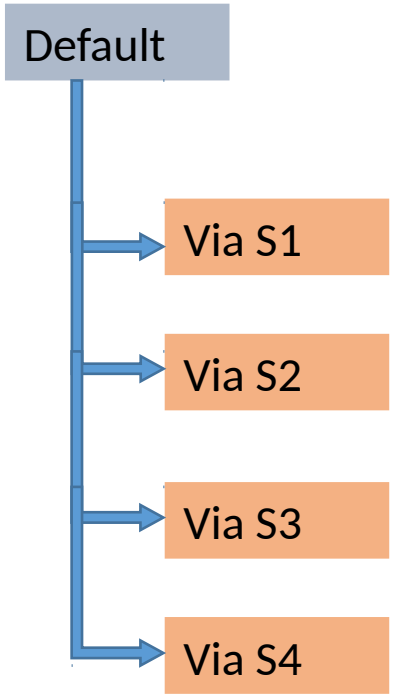

~A from Parent S1
First ~A ?
no
Remove route to A via that parent
yes
Install route to A:: via all other parents that advertise route towards the longest aggregation of A::
Last parent with route to A?
yes
Send ~A to all children
no
done


Advertisement of reachability to A from Parent S3
Reinstall Route to A via parent S3
First Parent with route to A ?
no
yes
Advertise reachability of A to children
All Parents with route to A ?
yes
Remove redundant routes

Routing table north, only default to start with

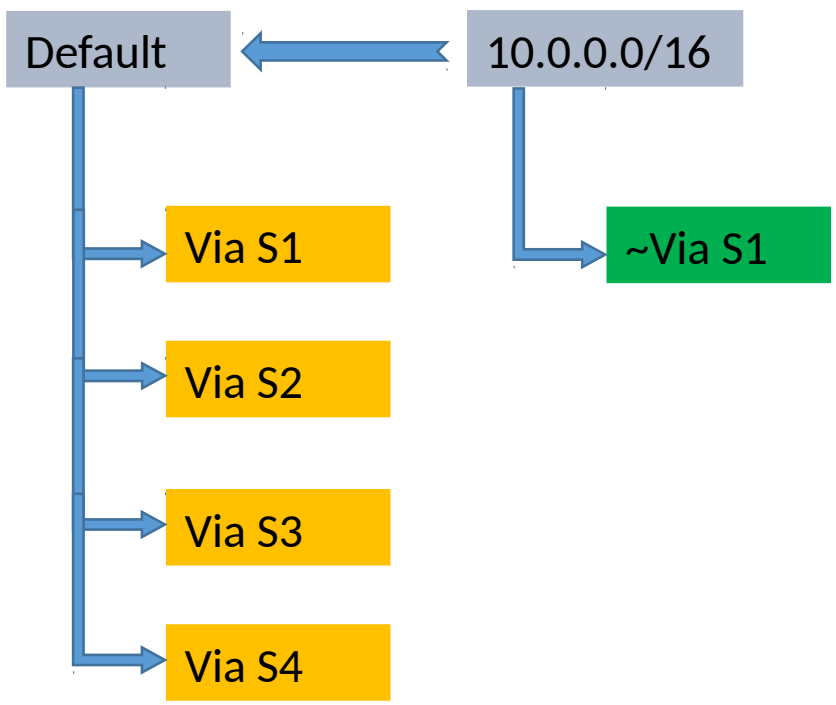


RIB

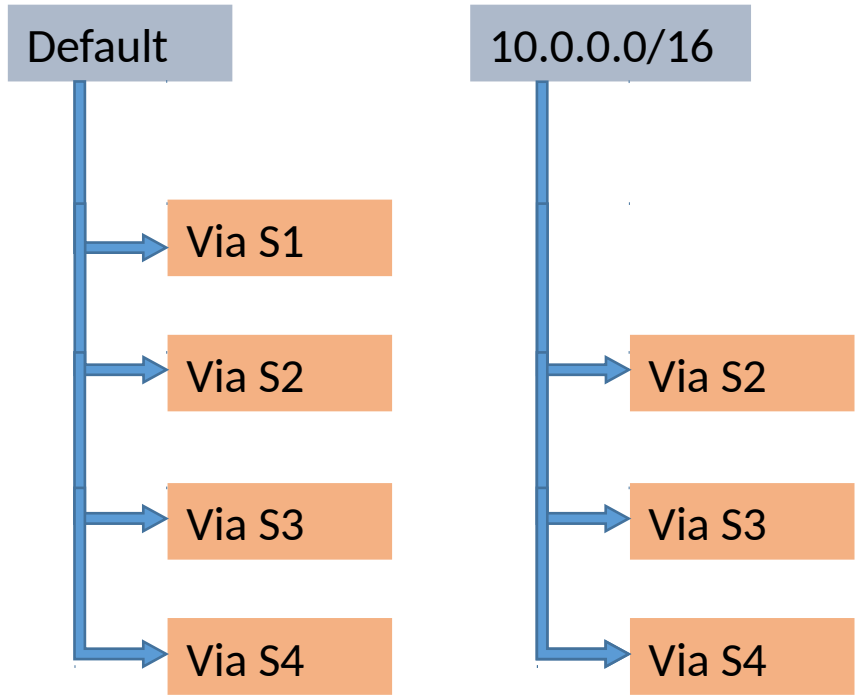


FIB

Getting a negative for 10.0.0.0, installing matching routes in FIB

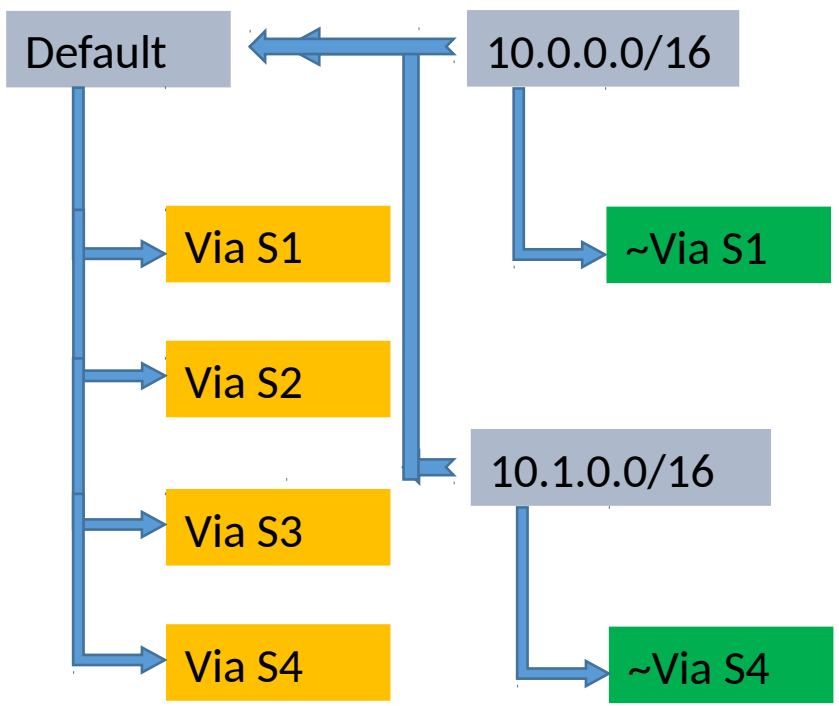


RIB

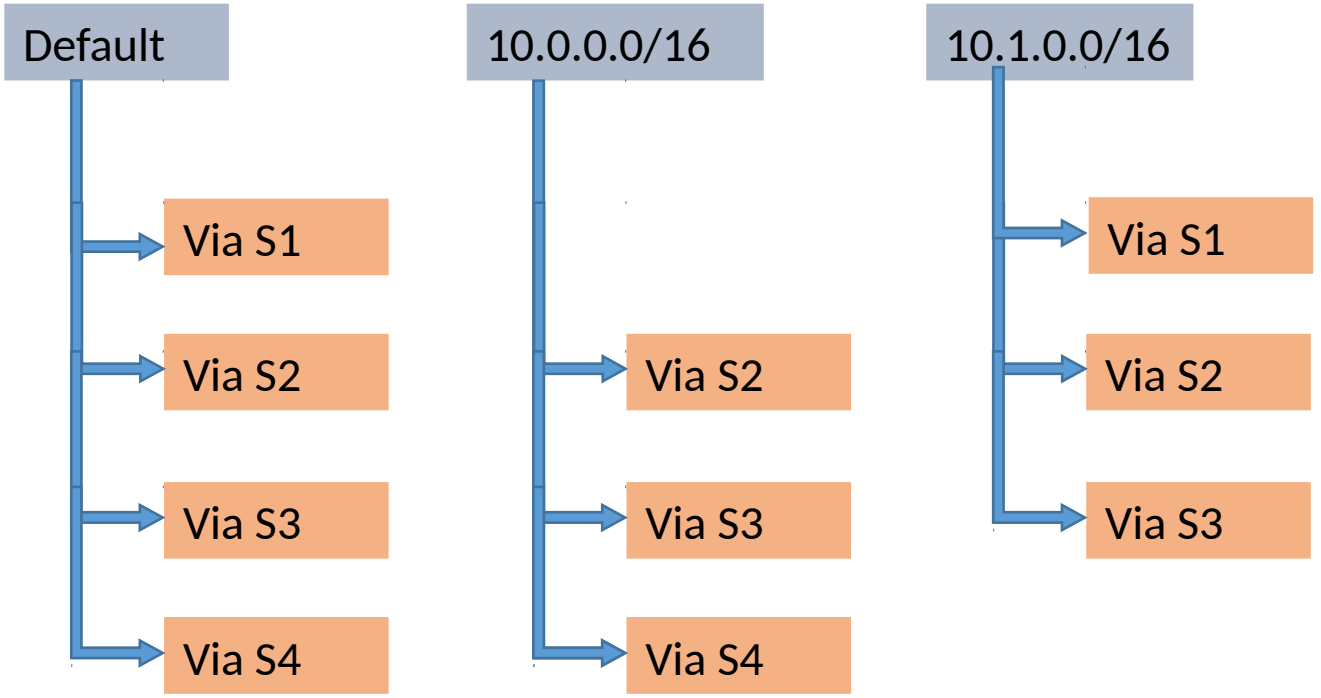


FIB

Getting a negative for 10.1.0.0, installing matching routes in FIB

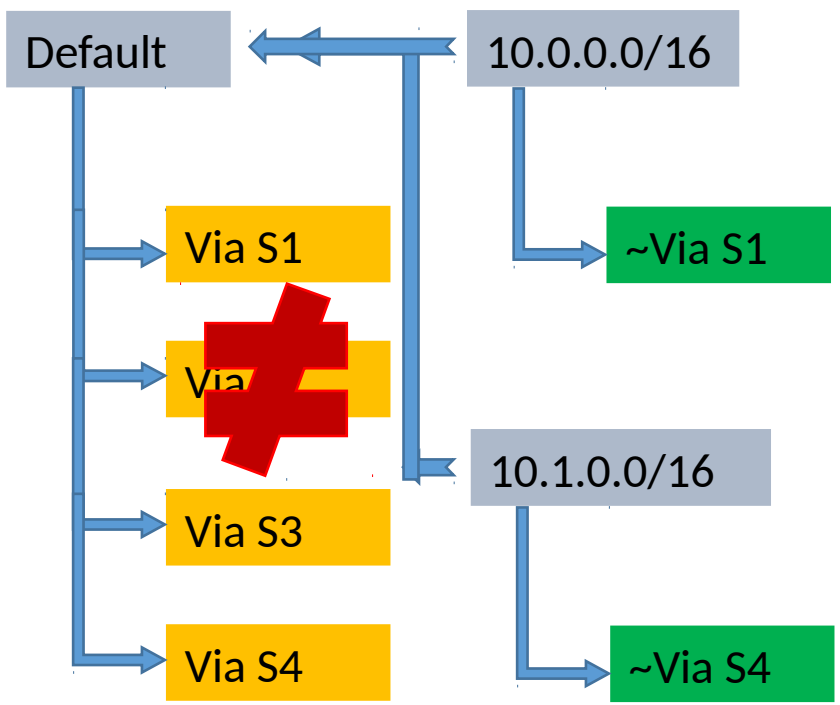


RIB

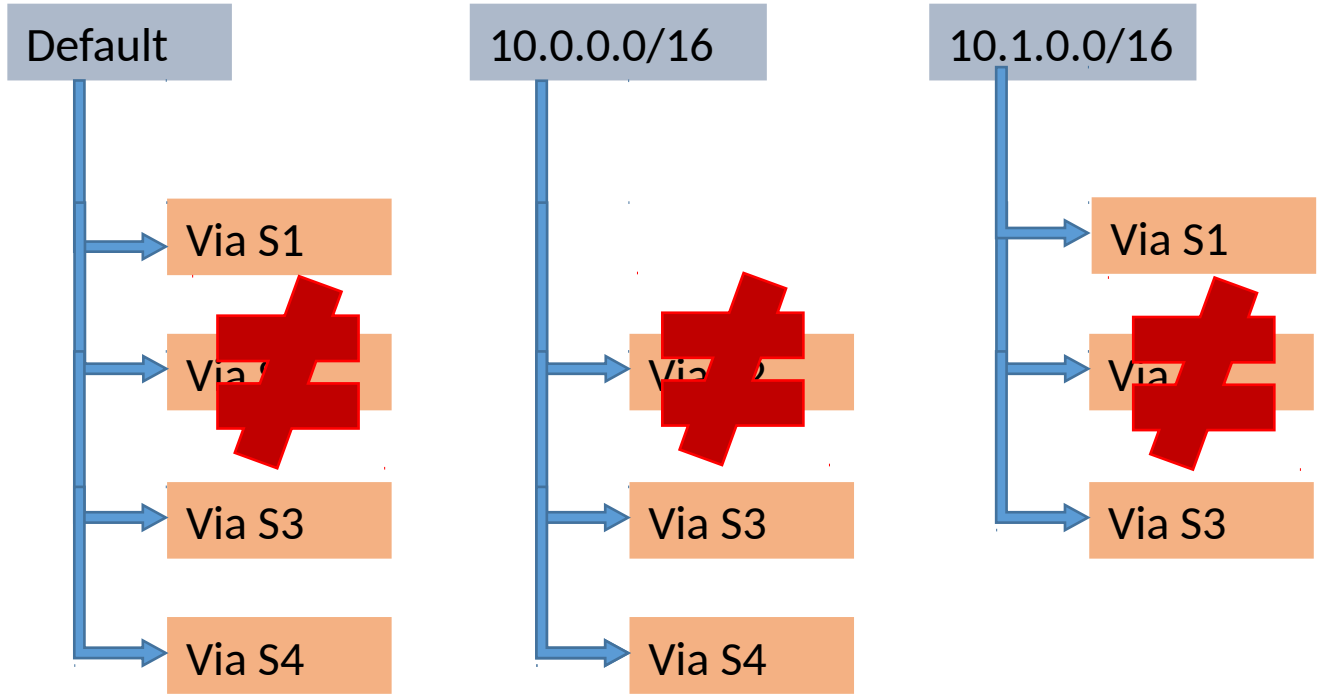


FIB

Recursive negative clean up if positive aggregation goes away

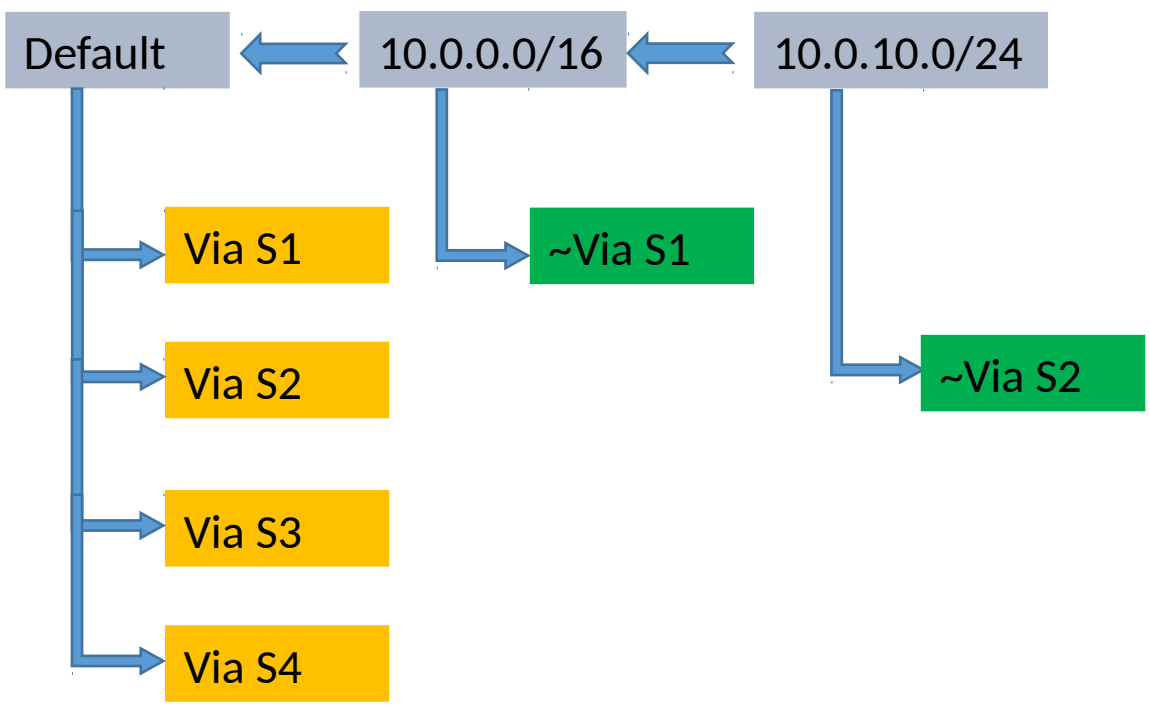


RIB

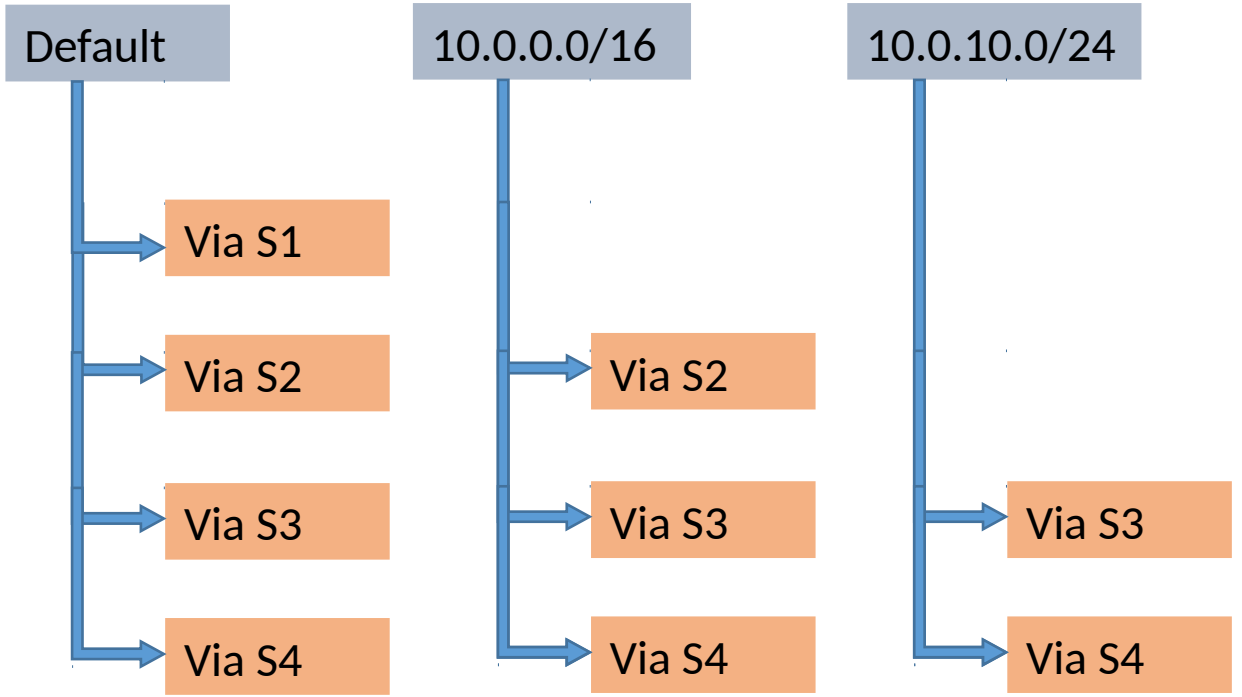


FIB

Getting a negative for 10.0.10.0, installing matching routes in FIB

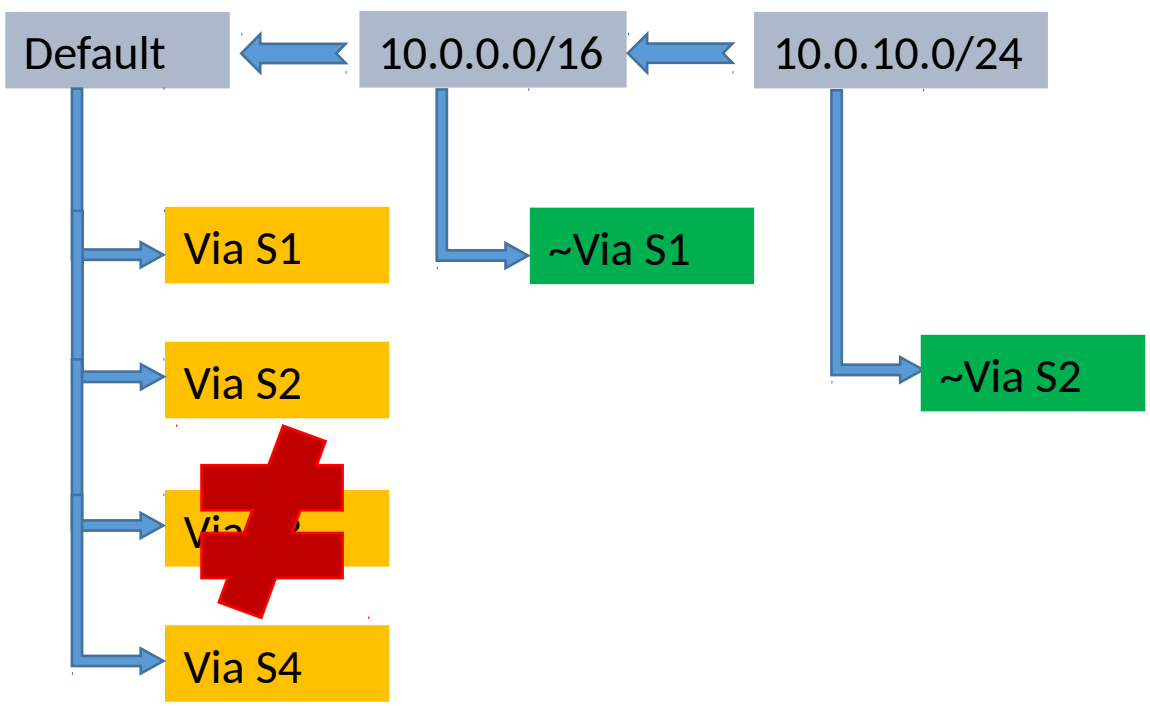


RIB

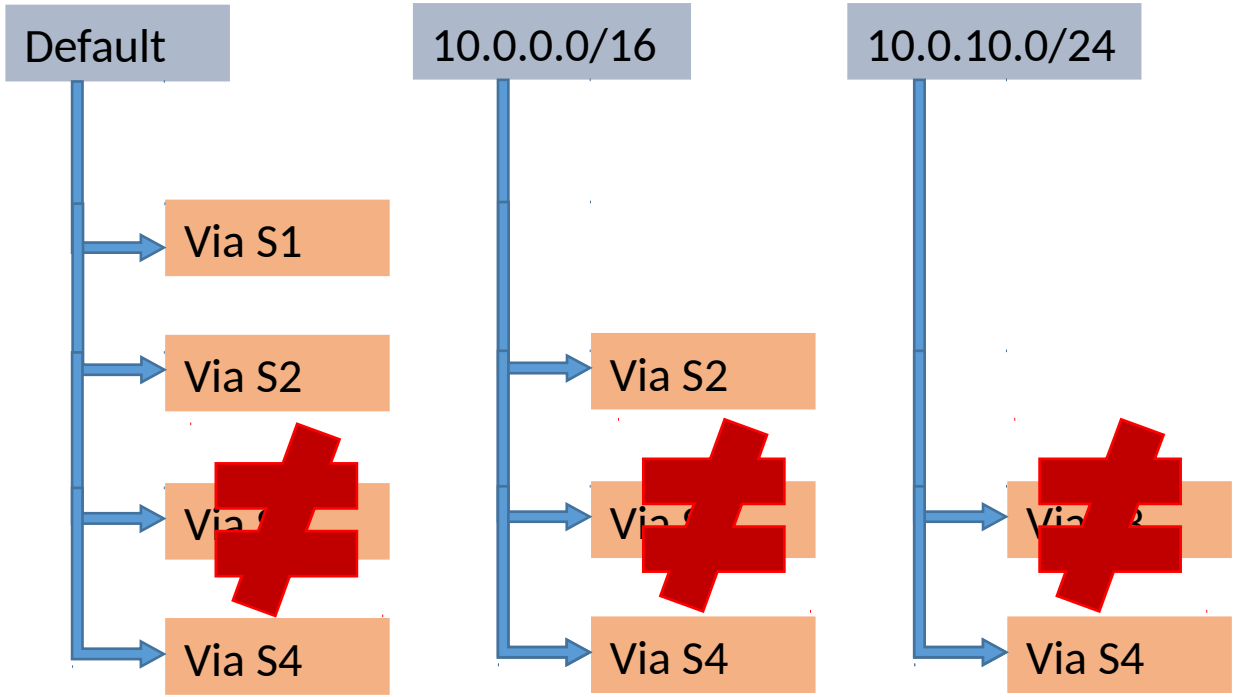


FIB

Recursive negative clean up if positive aggregation goes away



RIB



FIB

The resulting routing is as follows;

Default routing applies to all parents north

More specific routes to A:: exist in L4's North shadow cone

A new more specific route is installed on L3 via M4 so as to avoid M3, and in L1 and L2 to avoid M1

(should they have other sources of packets which is not the case is the topology below) M1 and M3 would keep forwarding packets for prefix A via S1 and S2 though in fact it has no solution, and the packets will be dropped or forwarded along another default route .