TESTING PROTOCOL FOR OVERALL NETWORK FUNCTIONALITY

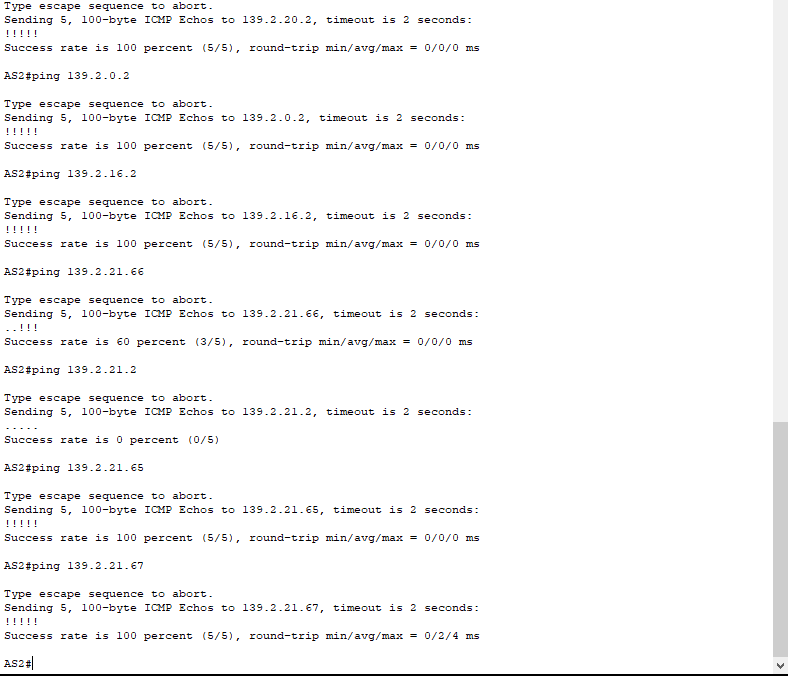
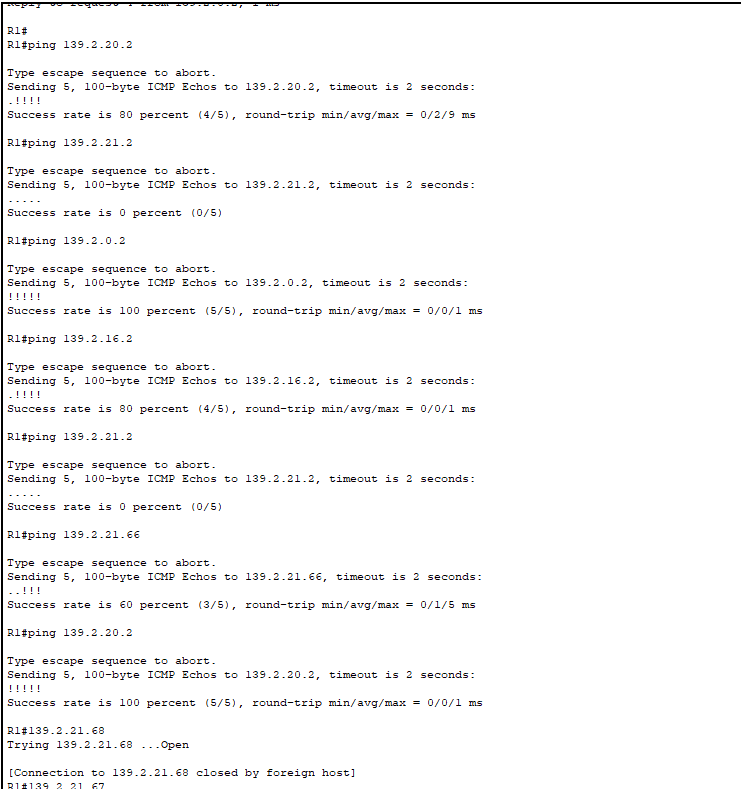
Upon the completion of building the network we used the **ping** command to test connectivity to all hosts in the network. Starting with the router, every end device was checked for connection with the intent of proving a properly configured network. We used a table to systematically ping every address from each device which is provided below

*Image: table used to test network connectivity*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEST 1 |  | Apartments | Retail | Rental | Housing | R1 | DL1 | AS1 | AS2 |
| Router(R1) |  | ✅ | ✅ | ❌ | ✅ | ✅ | ✅ | ❌ | ❌ |
| Switch (DL1) |  | ✅ | ✅ | ❌ | ✅ | ✅ | ✅ | ✅ | ✅ |
| Switch (AS1) |  | ✅ | ✅ | ❌ | ✅ | ✅ | ✅ | ✅ | ✅ |
| Switch (AS2) |  | ✅ | ✅ | ❌ | ✅ | ✅ | ✅ | ✅ | ✅ |

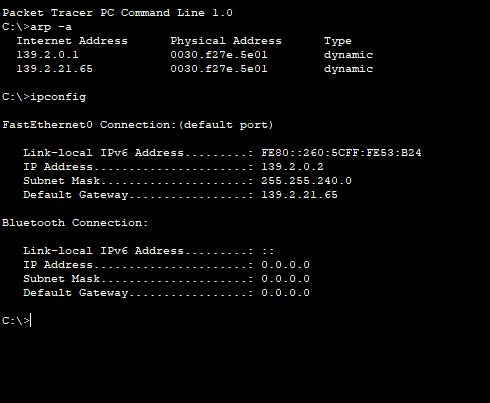
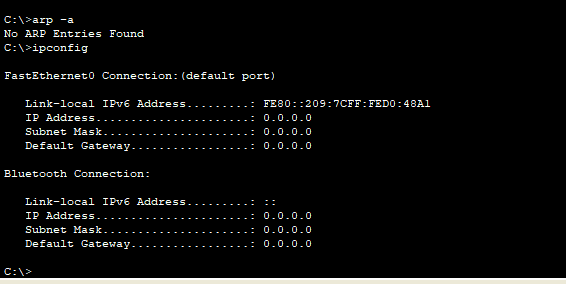
Our initial test revealed a lack of connection with the rental departments end device. This would require further investigation. As the rest of the network was working correctly we decided the end device in question must have a fault.

*Image: example of ping command being ran from R1 and AS2 revealing error with Rental end device.*



By running the **arp – a** command on each of the end devices it was revealed that an IP address had been removed from the Rentals Department, resulting in no connection with that particular host. This was resolved and the table was again executed, this time revealing full network connectivity.

*Image: arp-a command being used on end devices showing a problem with Rentals.*

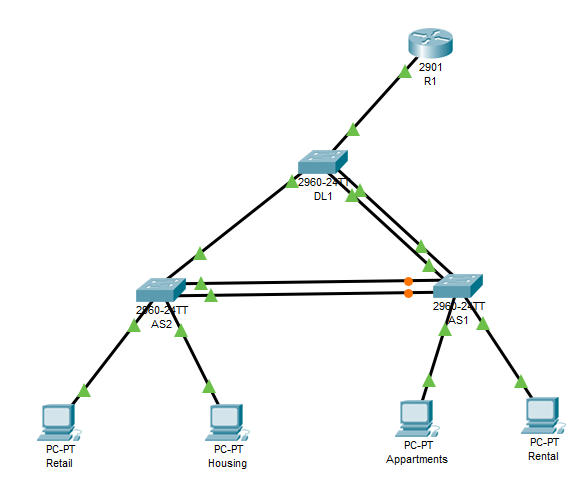
Once the issues was resolved, the ping command was again implemented using the table, revealing a fully operational network and giving us a platform to further test the more intricate parts of the network.

TESTING PROTOCOL FOR REDUNDANT LINKS

Implementing redundancy protects the network from a single point a failure by increasing the availability of devices in the network topology. We chose to create two lines of connection between switches as a way of inputting redundancy in our network.

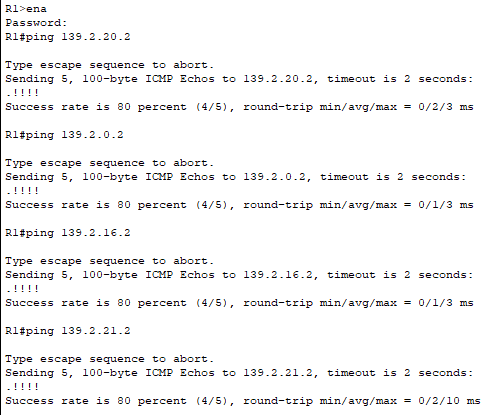
With two lines of connection between each device, our network should always be able to send packets even if one line is damaged or disconnected. To test this theory, once the network was built, we systematically removed points of connection.

*Image: example removal of link from As2 to DL1*



With one line disconnected we once again used the ping command from the Router to each end device to see the to see if there was still connectivity. The results, captured below, show that all packets reached their desired destination therefore the redundancy protocol had been implemented and the second connection had been used as means of delivering the packet.

*Image: ping command being implemented after removal of connection between As1 and Ds1*



This testing procedure was then implemented on all lines to all switches, to check the our redundancy was evident across the network.

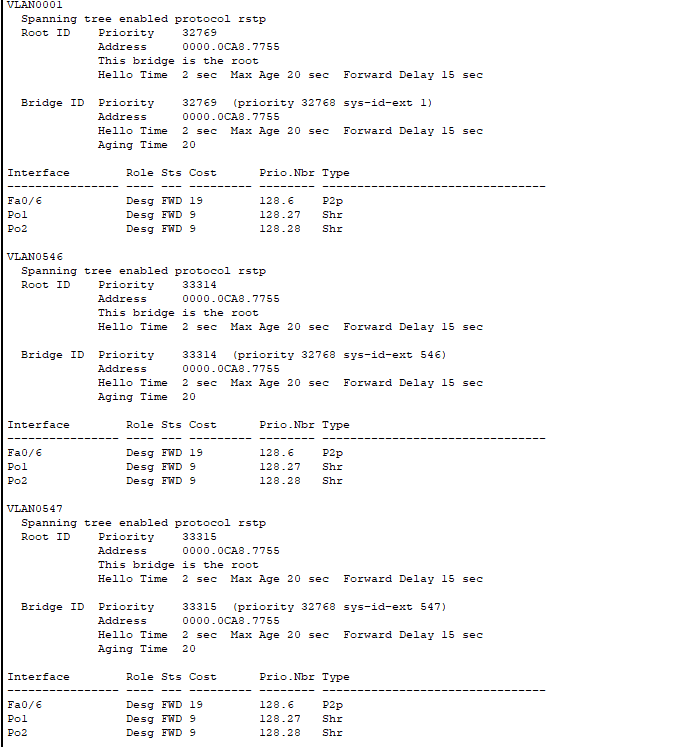
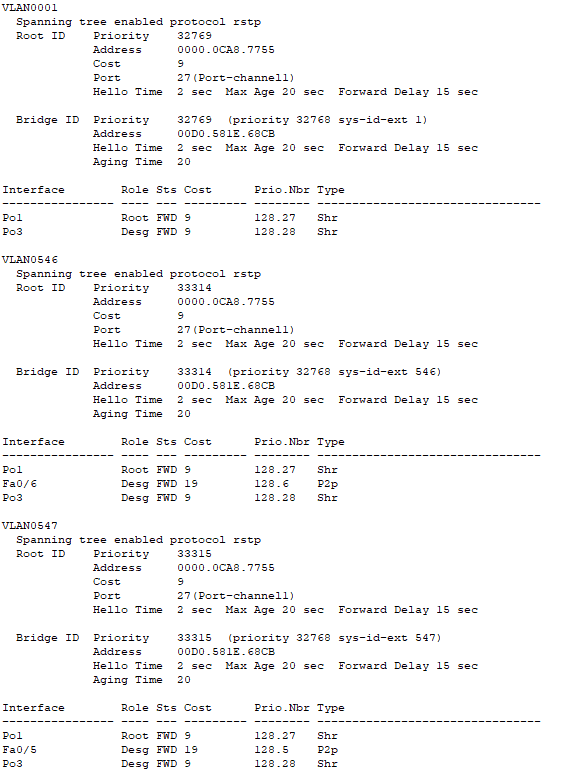
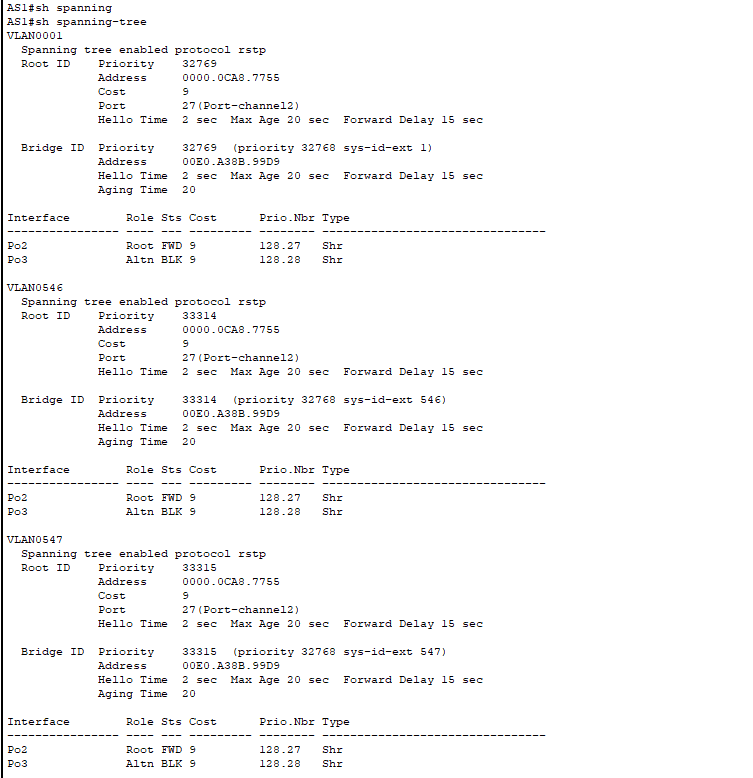
TESTING PROTOCOL FOR SPANNING TREE

A drawback feature of the implementation of two lines of connection for physical redundancy between each switch is the possibility of packets getting stuck in loops or duplicates frames occuring.

The Spanning Tree protocol is a layer 2 loop avoidance mechanism for redundant links that ensures there is only one logical path between destinations on the network. It intentionally blocks any redundant path that could cause a loop.

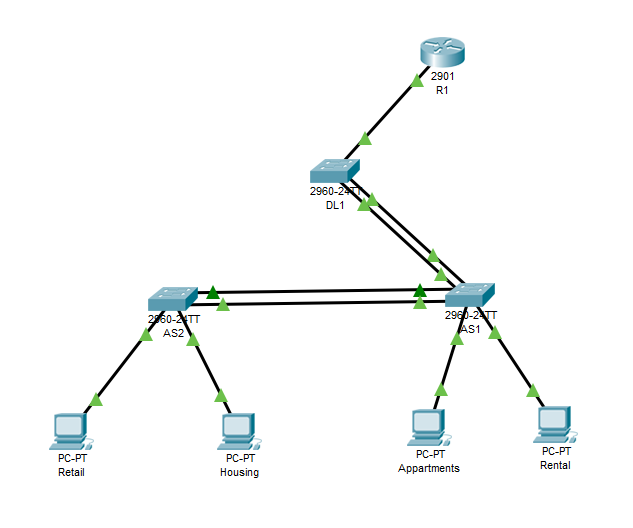
To test this we used the **show spanning-tree** command on all three switches while everything was connected to observe the election process of the root bridge.

*Image: show spanning-tree command on all 3 switches revealing DS1 as allocated root bridge.*

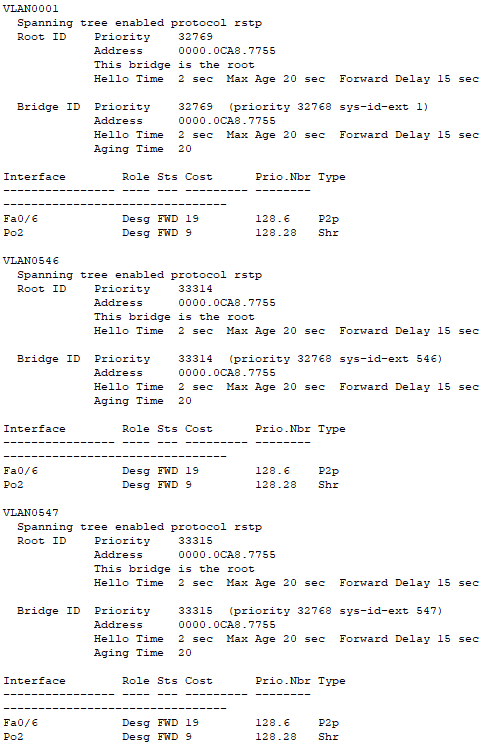
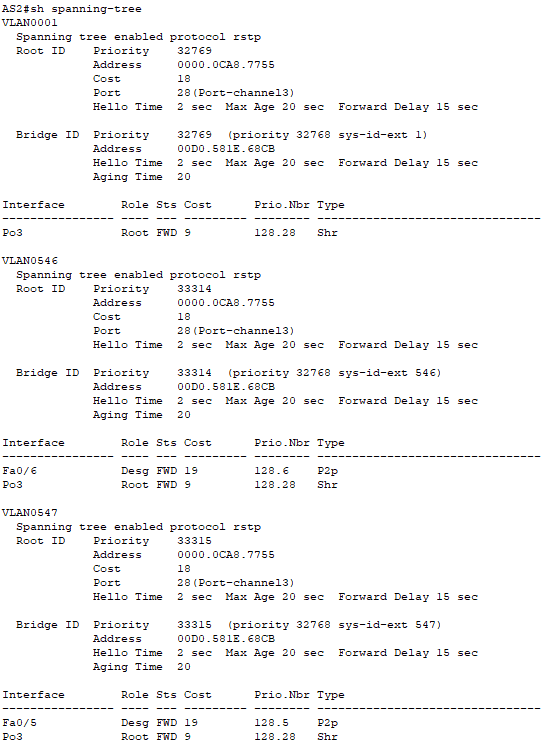
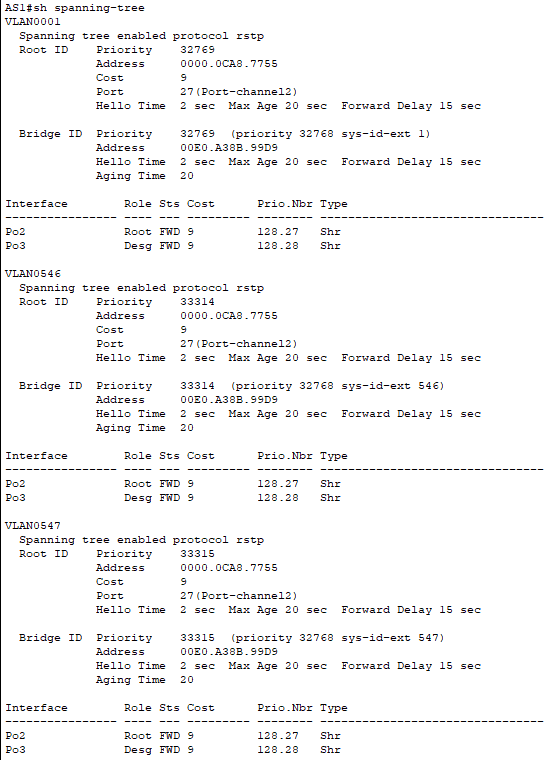


Next we completely disconnected DL1 from AS1 and re ran the spanning tree protocol to see the changes in the output. Our results showed that STP was implemented and the blocked ports were reopened allowing frames to be passed to the router even though there is major line of connection missing.

*Image: shows our test scenario implemented with the disconnection of lines between DL1 an AS1*

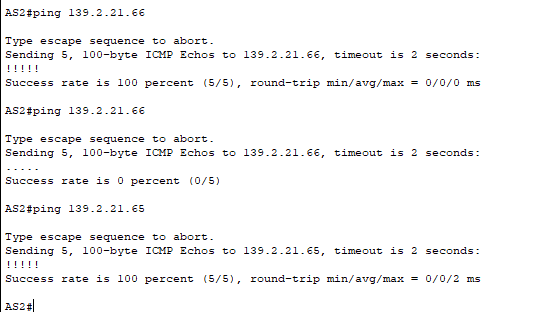


*Image:* ***sh spanning-tree command*** *issued post removal of connection show reorganization of blocked ports.*

**

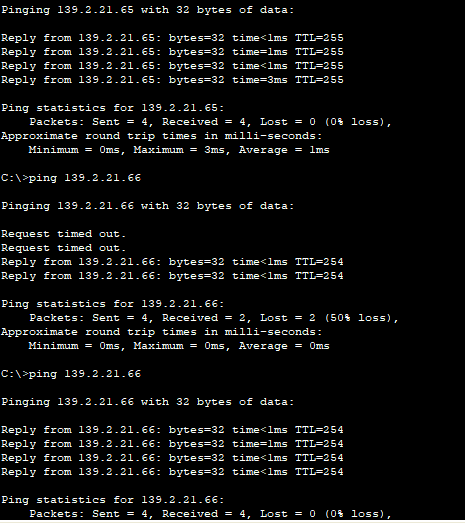
Finally we again issued the **ping command** from from each of the end switches to the router to prove that packets would still be delivered and that the spanning tree protocol was being implemented. The results showed that there was still a connection although the packets where taking a while longer to be delivered.

*Image: ping command issued post removal of connections from AS2 to R1*



Finally we issued the ping command from the end devices connected to AS2 just to confirm there is connectivity right throughout the network.

*Image: ping command issued from end devices retail to R1*

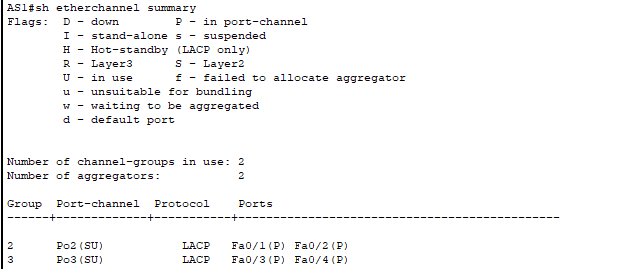
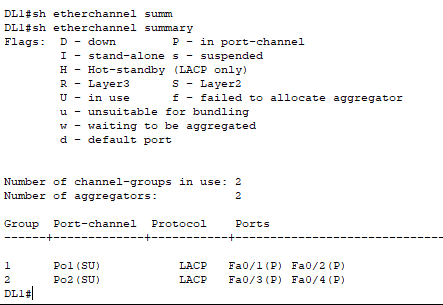
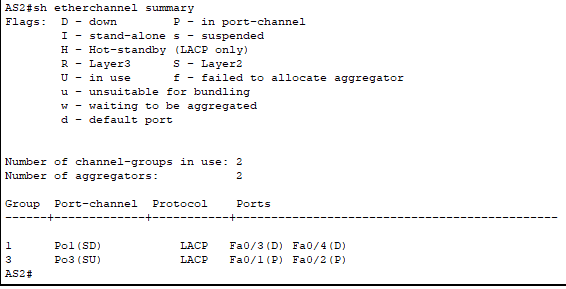


TESTING PLAN FOR ETHERCHANNEL BUNDLING

As we had previously connected two lines between switched to allow for redundancy in our network it made sense to also implement etherchannel bundling where the two physical links will act as one logical link and provide twice the bandwidth two each connection in the network. It also provides us with extra redundancy as you will still have connection from layer to layer if a line is damaged just with a diminshed bandwidth.

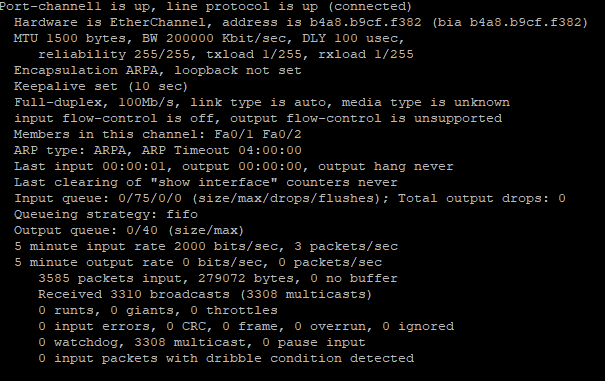
To test the etherchannel bundling is implemented we ran the etherchannel summary command on each switch to firstly display the summary information about etherchannels.

*image: etherchannel summary command on each switch*



Then we ran the **sh interface port number** command on all switches to confirm the line were running at their expected bandwidth.

*image:* ***sh int p0/1*** *on AS1*



Finally we again ran sh interfaces range on the switches again but this time with port lines disconnected to view the decrease in bandwidth to half of the original output. This proving the etherchannel bundling has been implemented.

*Image:* ***sh int p0/1*** *on AS1 with connection removed*

