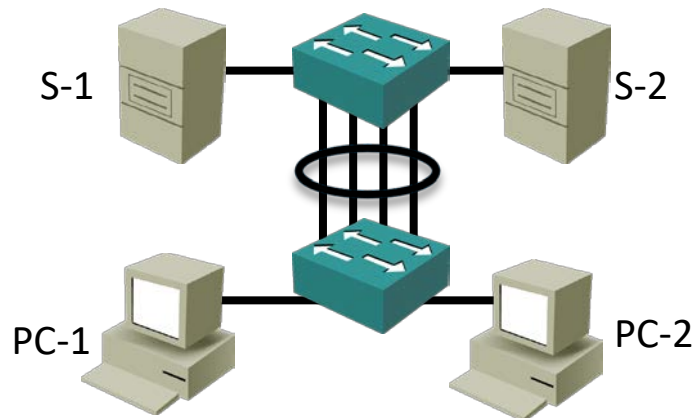


EtherChannel Load Balancing

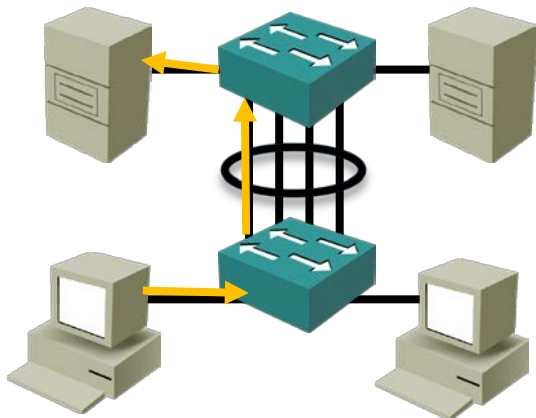


- A flow is a communication from a client to a server
- If PC-1 opens a web session to Server-1, and PC-2 opens an FTP session to Server-2, we have two flows going through the switch
- A single flow is load balanced onto a single port channel interface
- For example all packets in the flow from PC-1 to Server-1 always go over interface G0/1, all packets in the flow from PC-2 to Server-2 always go over interface G0/2



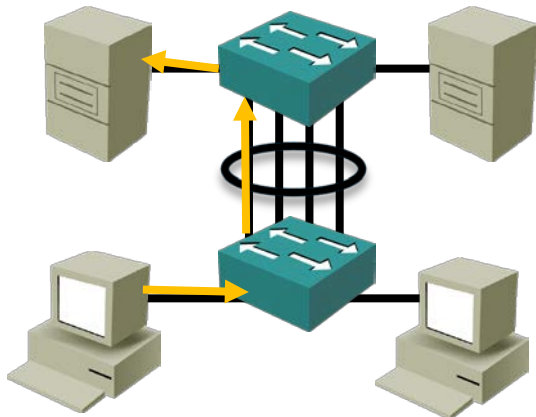
EtherChannel Load Balancing – 1st Packet

- 1st packet in the flow from PC-1 to Server-1



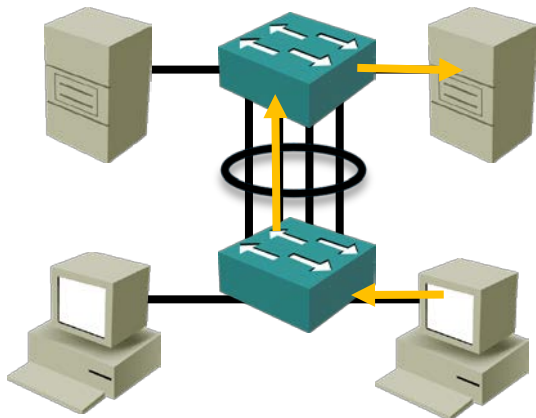
EtherChannel Load Balancing – 2nd Packet

- 2nd Packet from PC-1 to Server-1 goes over the same link



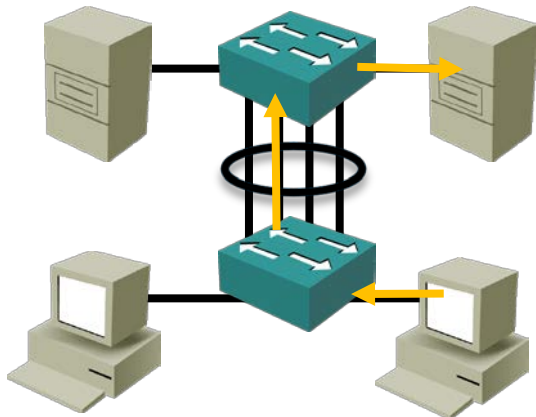
EtherChannel Load Balancing – 1st Packet

- 1st packet in the flow from PC-2 to Server-2



EtherChannel Load Balancing – 2nd Packet

- 2nd Packet from PC-2 to Server-2 goes over the same link



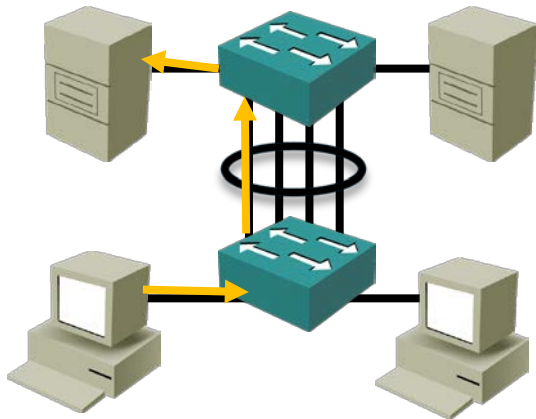
EtherChannel Load Balancing



- Packets from the same flow are not load balanced round robin across all the interfaces in the port channel
- We do not load balance the first packet from PC-1 to Server-1 on to interface G0/1, the second packet onto G0/2
- Round robin load balancing could cause packets to arrive out of order which would break some applications

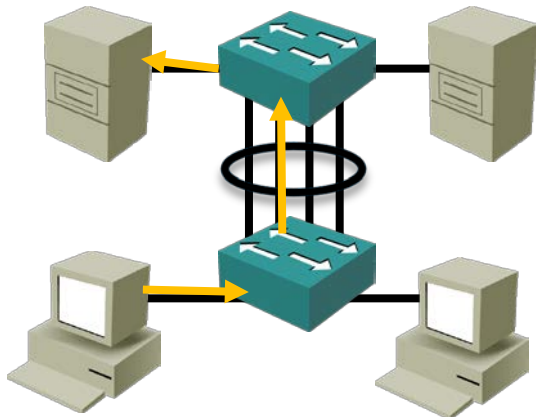
EtherChannel Load Balancing – 1st Packet

- This does NOT happen:



EtherChannel Load Balancing – 2nd Packet

- This does NOT happen:



EtherChannel Load Balancing



- Any single flow receives the bandwidth of a single link in the port channel as a maximum
- That's a maximum of 1 Gbps bandwidth per flow in our example, with the aggregate bandwidth of 4 Gbps available across all flows

EtherChannel Load Balancing



- You can think of a port channel as a multi-lane motorway. The cars stay in a single lane, but because there are multiple lanes the overall traffic gets there quicker
- Etherchannel provides redundancy as well as load balancing. If a link fails the flows will be load balanced to the remaining links