BROADCAST & COLLISION DOMANS

Objective:

To learn how to identify broadcast and collision domains in a network topology.

Background:

In any networking design selection of networking devices can depend upon isolation of traffic using knowledge of broadcast domains and collision domains.

A *broadcast domain* is an area in which any "network broadcast" is sent to every device in the broadcast domain. For example, if a workstation is set up to get its IP address from a DHCP server it uses a "broadcast address" that is sent over the network to retrieve the IP address from the DHCP server. So, in a way, a broadcast address is like a maintenance channel. It exists so individual devices can broadcast messages to one or every device within the broadcast domain. By keeping the broadcast domains smaller we are reducing the overall network traffic. We use routers to create separate broadcast domains. Each interface port on a router is a completely separate broadcast domain. Therefore broadcasts within one network on an interface will not pass to the network on another-interface (unless we program the router to do so which is not likely).

A *collision domain* is an area where collisions can occur in a network. Using Layer 1 devices, such as hubs, creates one large collision domain. Each port on a Layer 2 device, such as a switch, is its own collision domain reducing the possibility of collisions and errors down to nothing.

So let's jump into defining and identifying collision and broadcast domains. Along the way you will also learn more about how networking devices function.

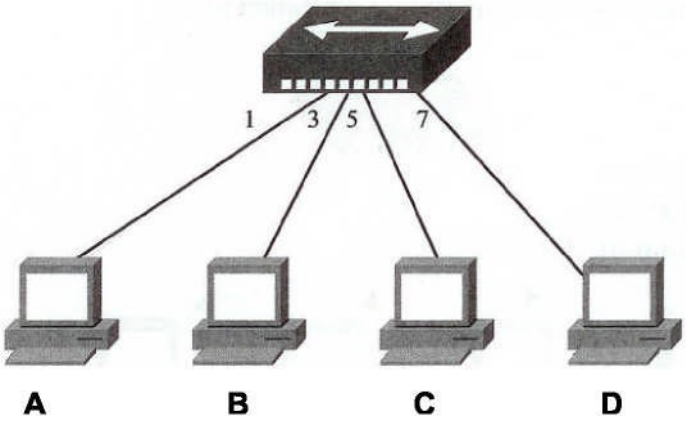


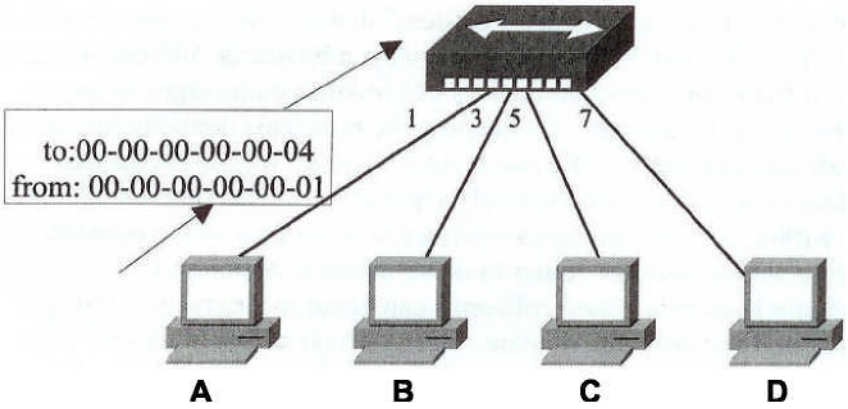
Figure 1. A small network using a hub.

Since no "intelligent functions" can take place with a hub (they only clean-up, amplify and retime signals) we have *one big broadcast domain* and *one big collision domain.* The likelihood of collisions is high. A hub basically allows transmission on only one port at a time. The hub allows port 1 so many seconds to transmit (but it doesn't send a notification to any other ports when

it is their turn) then changes to port 2 if no information is transmitted. It allows port one to finish then changes to port 2. It will allow port two so many seconds to transmit and then it will change

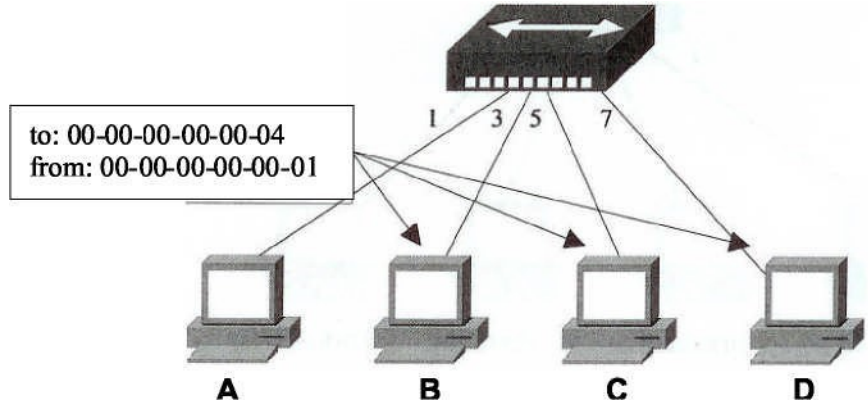
to port 3 if no information is transmitted. The process is repeated on port 3, then 4, then 5 and then to all the ports one at a time. But, hubs are not intelligent. Once the hub finds information being transmitted over a port it does not go to the next port, it starts back over at the port 1! *Therefore you want your more important devices on the first ports.*

In the diagram below, look at an example for workstation "A" to send information to workstation "D." The information from workstation "A" enters the hub on port 1. The hub then makes duplicate copies of that information and sends it to each port (active or not). In this case workstations "B," "C," and "D" will receive the copies.



MAC: 00-00-00-00-00-0 1 MAC: 00-00-00-00-00-04

Figure 2. Workstation A sends a request to workstation D.



MAC: 00-00-00-00-00-01 MAC: 00-00-00-00-00-04

Figure 3. The information is duplicated and sent to every node attached to the hub.

The information is received on the workstations and the de-encapsulation process is started. The frame has the header and footer information removed. First an error checking process will reveal if the information is correct. Next, the destination MAC address is checked to see if it

matches the MAC on the workstation (Is this for me?). If they match then the de-encapsulation process continues (which it does only on computer D). If they do not match (which it does not on computers B and C) then the frame and all its information is discarded and ignored. Therefore only the destination device (computer D), for which it was intended, will process the information.

With a hub making multiple copies of *each* incoming request the chances for a collision are high. What happens during a "collision"? Most resources will tell you workstations will "listen" before transmitting. Each NIC monitors the transmitting ping and receiving pin voltage, for a short period of time. By detecting this voltage the workstation is "listening" to the network for transmissions. When the voltage is detected on both pins the networking devices "sees" this as a collision and grounds the media for a period of time (which stops the collision. This is called a *jam signal.* Then the workstation randomly picks a number of milliseconds to wait to retransmitting its information (called the *back-of algorithm).*

This is why networking devices must be selected carefully: to reduce the possibility of collisions. Today higher-level networking devices, such as switches and routers, are available at lower costs, which make them more accessible for installation. Switches eliminate the possibility of collisions because each port is its own collision domain. With one device on a port we have absolutely no chance of a collision happening. Using a switch also "divides" up the available bandwidth from a backbone line to each port. Unlike a hub, our switch can have many simultaneous transmissions. The switch is therefore a more robust device that performs better in networks. In the past few years the prices have come down on switches so much that it is not even worth buying hubs because switches are only a few pounds more.

In the previous example was a demonstration of how collisions occur. In this example the hub is replaced by a switch, which eliminates the possibility of collisions. Each port becomes its own collision domain. A switch, unlike a hub, also has the possibility to store information to be sent out later. That way, if workstation A and D were transmitting at the same time, the switch could store information from one workstation while passing on the transmission from the other over the backbone. This is called *store-and-forward.*

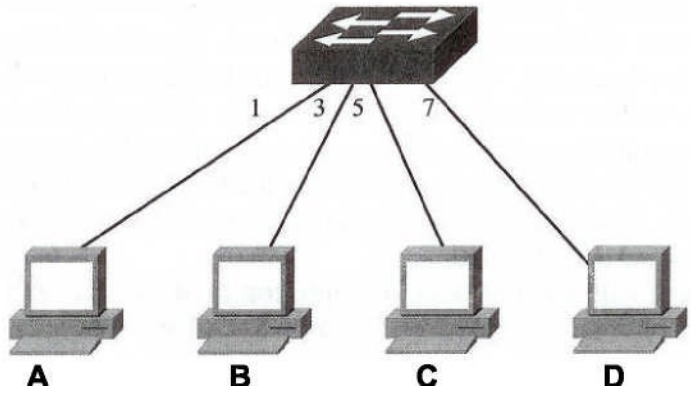


Figure 4. Small switched network.

A switch is an intelligent device. It allows an administrator to change the priorities of the ports to determine who gets to transmit first in the event of tie. The information from the other ports would be stored and transmitted later after the first one is done. Since the possibilities of two workstations transmitting at exactly the same time is remote, setting port priorities is not a major concern. So why go through all of that hassle and expense to replace hubs with switches? First, switches do not cost much anymore. Second, a key word in networking design is *scalability,* the ability to grow without replacing equipment. We get more functionality out of a

switch than out of a hub for about the same money! A switch is more scalable than a hub. And, third, switches are cool. Many believe switching will become more prevalent in networking than routing. We use switches at the core of our networks, not routers. Switches only use Layer 2 information to make decisions. Routers need Layers 2 and 3 information to make decisions, so they tend to be slower (in geek-speak: switches have less *latency* than routers).

So, switches eliminate collision domain problems. Now, with a switch, there are *many collision domains* (one per port) and *one big broadcast domain.* Workstation A and D could communicate almost instantaneously with each other or to other ports and their devices.

But that still leaves one BIG broadcast domain hanging out there. Big broadcast domains aren't necessarily bad, but we would like to keep them as small as possible. As was said earlier, a broadcast domain is used for network "maintenance." One analogy for a broadcast domain may be the public address system in your classroom. The staff can make announcements to the whole school or can communicate with just an individual classroom. By keeping the broadcast domain as small as possible we keep our "overhead" traffic as minimal as possible and, therefore, lessen any possible network traffic.

It's time to expand the simple network to two networks:

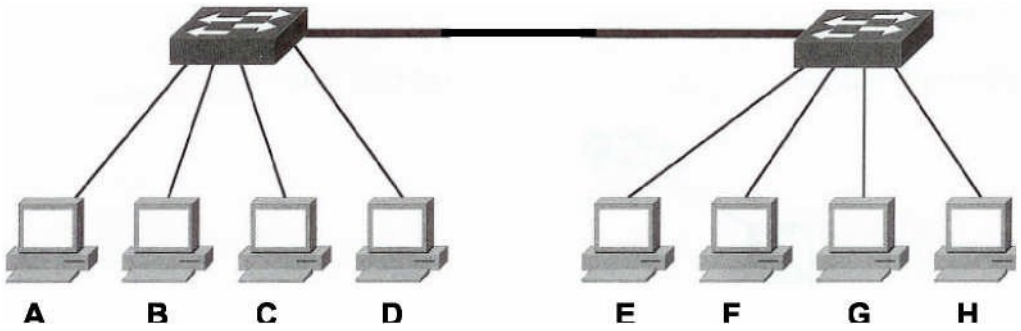


Figure 5. Small multiple-switched network.

Now there are 8 collisions in the one broadcast domain. Wouldn't you think the link between the switches should be considered a collision domain too? No, because switches have the ability to store information and send it off later, remember that little fact? Therefore no collision possibility exists. It's like have one really big switch!

With multiple switches there exists the possibility for excessive broadcasts that could slow our network down. A router could be used to reduce the broadcast domain size. Recall, each interface on a router, in fact, is its own broadcast domain. So by adding a router into the network there are now *eight collision domains* and *two broadcast domains* as the diagram below shows:

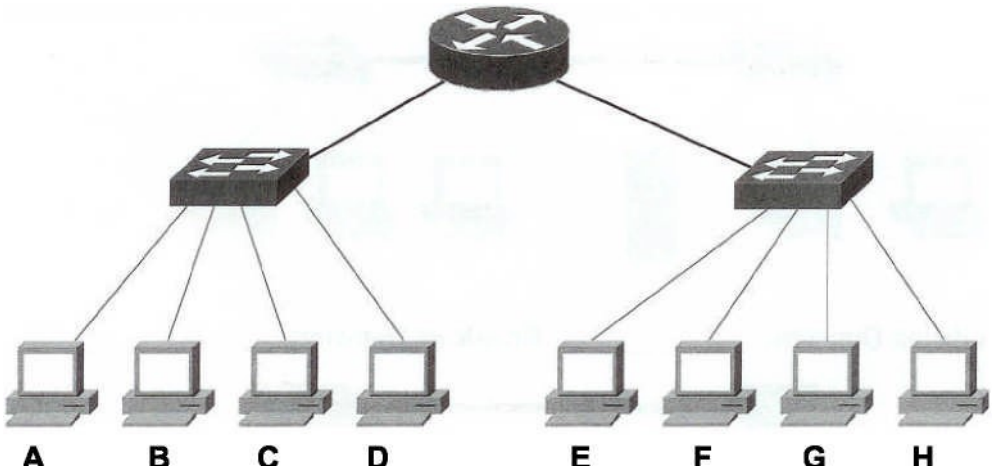
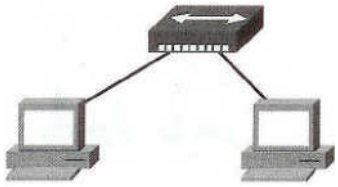


Figure 6. Two small networks connected by a router.

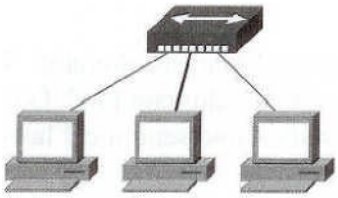
Now it's time to apply what you have learned:

For each networking arrangement, count the number of collision and broadcast domains.

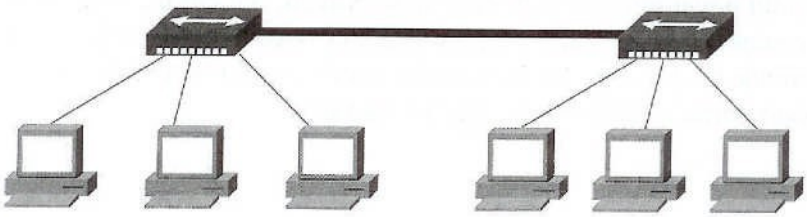
1. Collision Domains: Broadcast Domains:



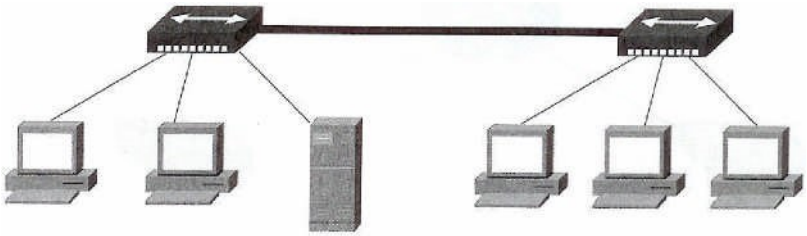
2. Collision Domains: Broadcast Domains:



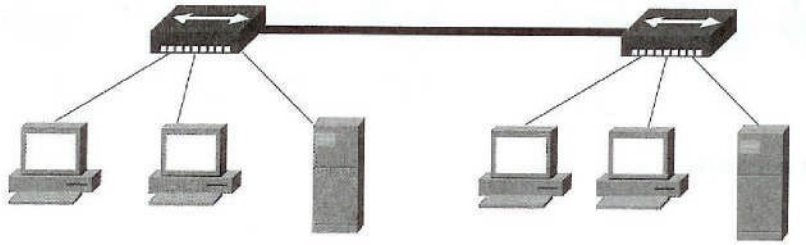
3. Collision Domains: Broadcast Domains:



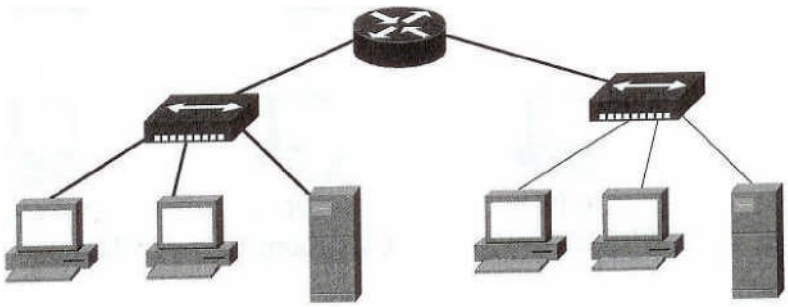
4. Collision Domains: Broadcast Domains:



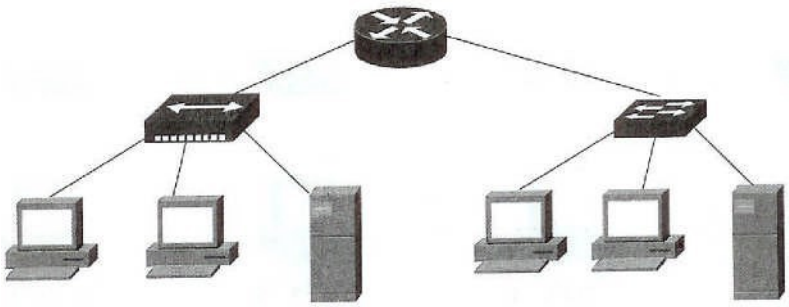
5 Collision Domains: Broadcast Domains:



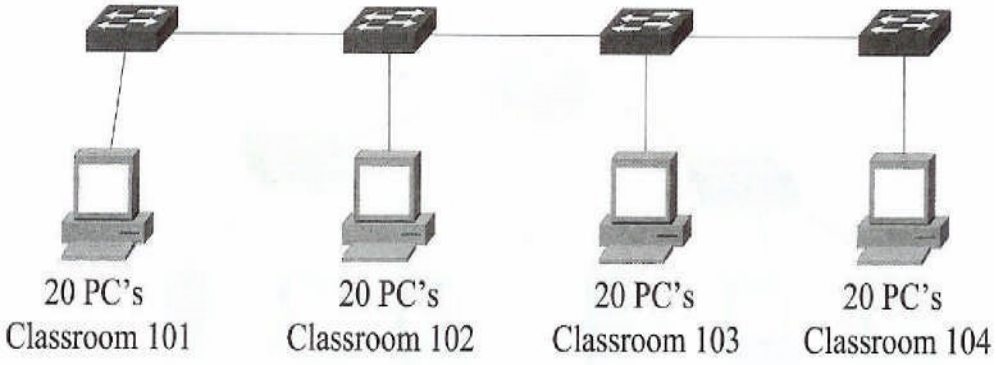
6. Collision Domains: Broadcast Domains:



7. Collision Domains: Broadcast Domains:



8. Collision Domains: Broadcast Domains:



9. Collision Domains: Broadcast Domains: