ECE 333: Introduction to Communication Networks
Fall 2001

Lecture 19: Switching and Multiplexing

- More on transparent bridges.
- Introduction to Switching and Multiplexing

Transparent bridges continued:

Last time we began talking about bridges. Recall, bridges are devices for interconnecting 2 or more LANs at the MAC layer. One type of bridge is a transparent bridge - transparent bridges use a forwarding table that is updated from observing the source address in all transmitted packets.

One problem that transparent bridges need to address is how to avoid sending packets in loops; this is addressed by forming a ***spanning tree***. The bridges execute a distributed algorithm to decide on a specific spanning tree for a network. Bridges then only forward packets along this spanning tree. Today we will look the algorithm used to construct this spanning tree. This is a basic example of a routing algorithm; we will see more examples of this type of algorithm at the network layer.

Spanning tree algorithm

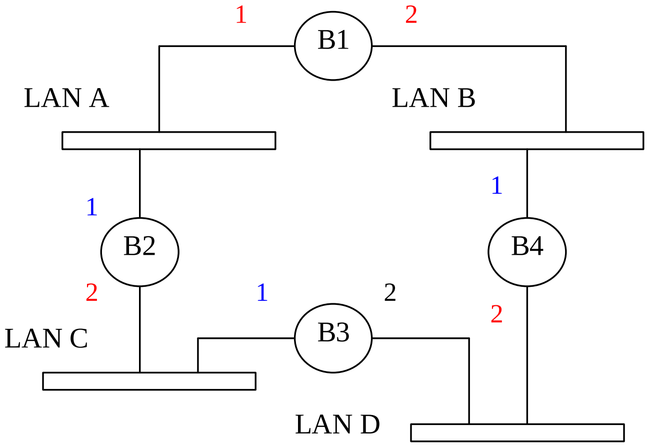
For the spanning tree algorithm, each bridge is assigned a unique ID number. Each bridge also gives each of its ports an ID that is unique for that bridge. The spanning tree algorithm executed by the bridges, attempts to do the following:

1. Elect a ***root bridge***, this will be the bridge with the smallest ID.
2. Determine a ***root port*** for each bridge (except the root), this is the port that is on the ***shortest path*** to the root bridge. Distance is measured as the number of LANs a packet would have to cross to reach the root bridge. Ties are broken by choosing the port with the smallest ID.
3. Select a ***designated bridge*** for each LAN. This is the bridge which provides the shortest path to root bridge from the LAN. Ties are broken by choosing the bridge with the smallest ID. The port that connects a LAN to its designated bridge is called the ***designated port***.

Each bridge then blocks any port that is not a root or designated port, *i.e.* it does not forward any packets to a LAN attached to such a port. This removes in any loops in the topology and forms a spanning tree.

Example:

The figure below shows a collection of LANs and bridges, the bridges are numbered 1,2,3,4, the port numbers for each bridge are also indicated. In this case B1 would be the root bridge, since it has the smallest ID. The designated ports are shown in red and the root port are shown in blue. For example, on LAN C, bridge B2 is the designated bridge, since it is only one LAN away from the root, while B3 is 2 LANs away. Notice that port 2 on bridge B3 is neither a root bridge nor a designated bridge. Thus bridge B3 will not forward any packets, creating a tree.



Spanning Tree Algorithm

To construct and maintain the above spanning tree, the bridges exchange a sequence of control messages called *Bridge Protocol Data Units (BPDU's)*.

At any time, each bridge stores the following two values in memory:

- **Root_ID** - the bridge's estimate of the root.
- **Root_distance** - the bridge's estimate of its distance to the presumed root.

The bridges then transmit BPDU with the format:

[Sender_ID|Root_ID|Root_distance]

Where **Sender_ID** is the bridge's ID own ID number.

Initially all bridges think they are the root (**Root_ID=Sender_ID**). A bridge updates its values of **Root_ID** and **Root_distance** based on the messages received to date.

Specifically, assume that a bridge has **Root_ID** = ***R*** and **Root_distance** = ***D***. If it receives a BPDU containing [***S***'|***R***'|***D***'] on a port connected to LAN *i*. Then it does the following:

IF ***R***' < ***R*** THEN set **Root_ID** = ***R***', and **Root_distance** = ***D***' +1, and assume not designated bridge on LAN *i*.

IF ***R'*** = ***R*** and ***D'*** < ***D*** THEN set **Root_distance** = ***D'***+1, and assume not designated bridge on LAN *i*.

IF ***R'***= ***R***, and ***D'***=***D*** and ***S'*** < **Sender_ID**, THEN assume not designated bridge on LAN *i*

ELSE assume it is designated bridge on LAN *i*.

Bridge also update their designation on other LAN's to which they are connected, based on previous messages received on those LANs, *e.g.,* if it had previously received [***S"***|***R"***|***D"***] on LAN *j*, and now **Root_ID** < ***R"***, then the bridge will assume it is now designated on LAN *j*.

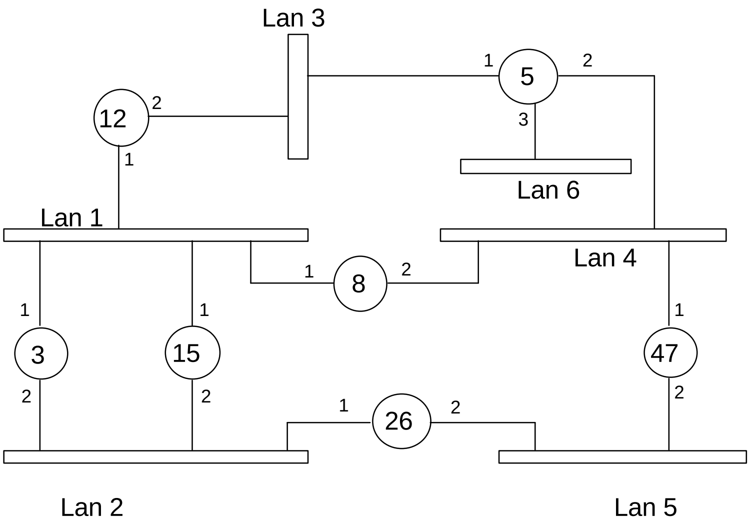
Bridges transmit BPDU's according to the following rules:

1. When a bridge thinks it's the root (**Root_ID** = **Sender_ID**), it periodically transmits BPDU.
2. When a bridge learns it is not root, it only forwards BPDU's (after adding one to the distance).
3. Bridges only forward BPDU on LANs where it thinks it is the designated bridge.

But note even if a bridge is not designated it still receives all BPDU's transmitted on all LANs it is attached to.

An example of this algorithm is presented next. To simplify this example, we assume that all the bridges are synchronized and send out their BPDUs at the same time. In practice this algorithm would be implemented asynchronously.

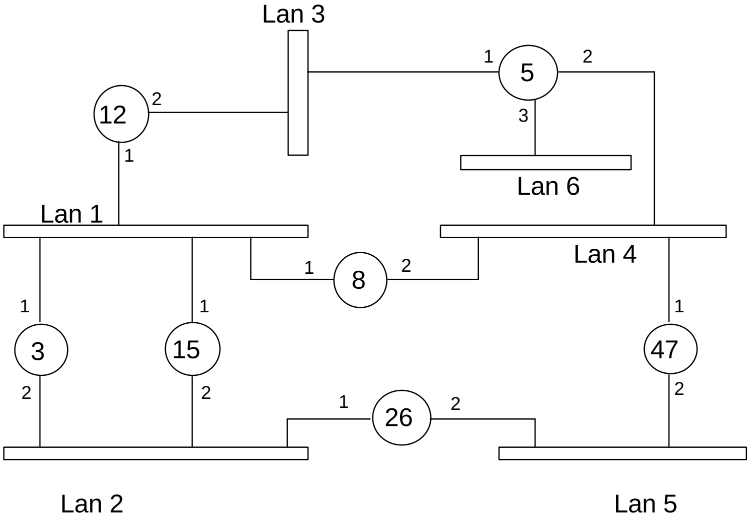
Spanning Tree Example



Initially:

- All bridges assume they are the root
- All bridges assume they are the designated bridge on all LAN's to which they are connected
- All bridges send a BPDU to that effect.

After lst set of BPDU's



Bridge 3

- receives claims of root-hood from Bridges 15, 8, 26, 12
- keeps **Root_ID** = 3, **Root_distance** = 0.
- maintains its belief that it is designated on LAN 1 and LAN 2.

Bridge 5

- received BPDU's from Bridges 12, 8, 47
- keeps **Root_ID** = 5, **Root_distance** = 0.
- maintains its belief that it is designated on LAN's 3, 4, and 6

Bridge 12

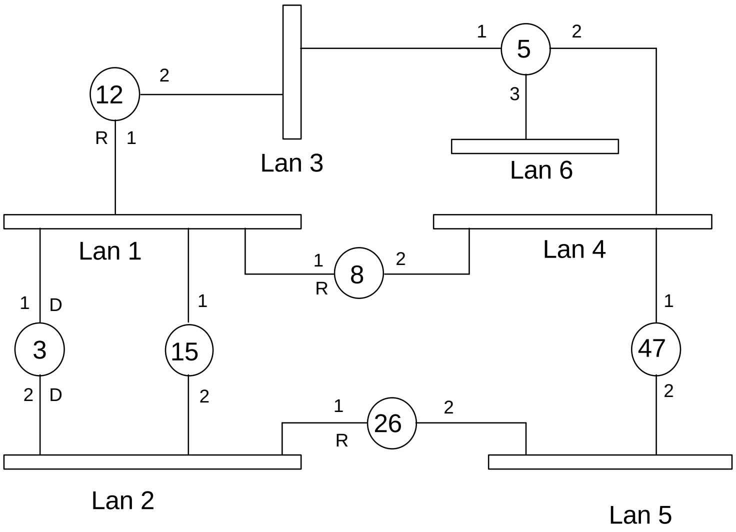
- received claims of root-hood from Bridges 3, 15, 8, 5
- sets **Root_ID** = 3, **Root_distance** = 1.
- sets Bridge 3's as designated on LAN 1
- sets Port 1 as root port
- sets port 2 as designated port on LAN 3

NOTE: Subsequent BPDU will not be sent toward root by Bridge 12.

Bridge 15

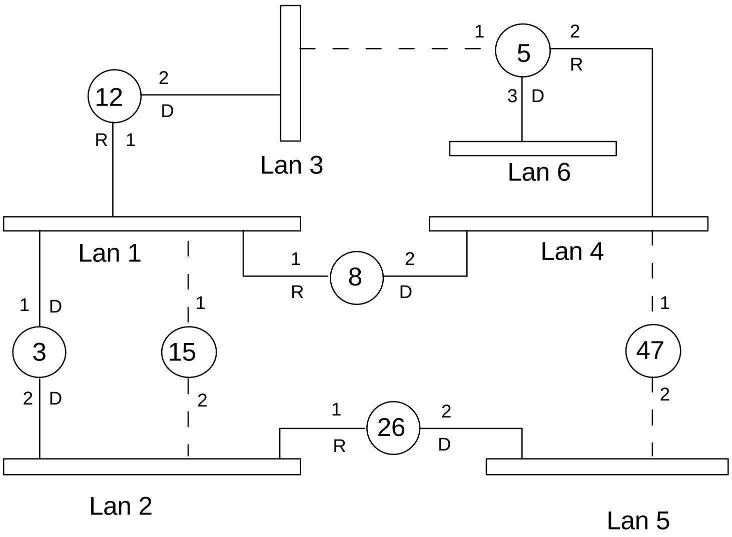
- receives claims of roothood from 3, 12, 8, and 26
- sets **Root_ID** = 3, **Root_distance** = 1.
- sets bridge 3 as designate bridge on LAN 1 and LAN 2
- Since it is connected to no other LAN, it goes into a listen mode.

Spanning Tree Example After 1st Iteration



- Bridges 3, 15, 8, 26, 12 know correct root
- Bridges 5 and 47 think bridge 5 is the root.
- Next, bridges 3 and 5 will send out BPDU's; the other bridges will forward these on ports that are not towards the root or blocked.
- In future rounds, Bridge 15 will not send out or forward any BPDU's

Spanning Tree Example



Final Tree

Recovery from Loss of Bridge

Bridges can also detect and recover from the loss of a bridge. To accomplish this, the root node is required to send out BPDU's at regular intervals. The other bridges forward these according to the above rules. Each bridge has timer that times out if it doesn't get an update from root. When a bridge times-out, it assumes it is the root bridge again, and starts the algorithm over.

Summary of Medium Access Control:

- Static approaches - TDMA, FDMA
- Dynamic approaches -
  - ♦ Contention based -
    - o Aloha, CSMA, CSMA/CD (Ethernet)
    - o Wireless - MACA, MACAW, (802.11)
  - ♦ Non-contention based -
    - o Reservations
    - o Token rings - (802.4, 802.5, FDDI)
- Interconnection of LANs - bridges/LAN switches
  - ♦ Transparent bridges
  - ♦ Source routing bridges

Switching and Multiplexing

Above approaches are for shared media or broadcast networks. These are most commonly used in LANs. Next, we begin to discuss point-to-point networks, which are more common in WAN's.

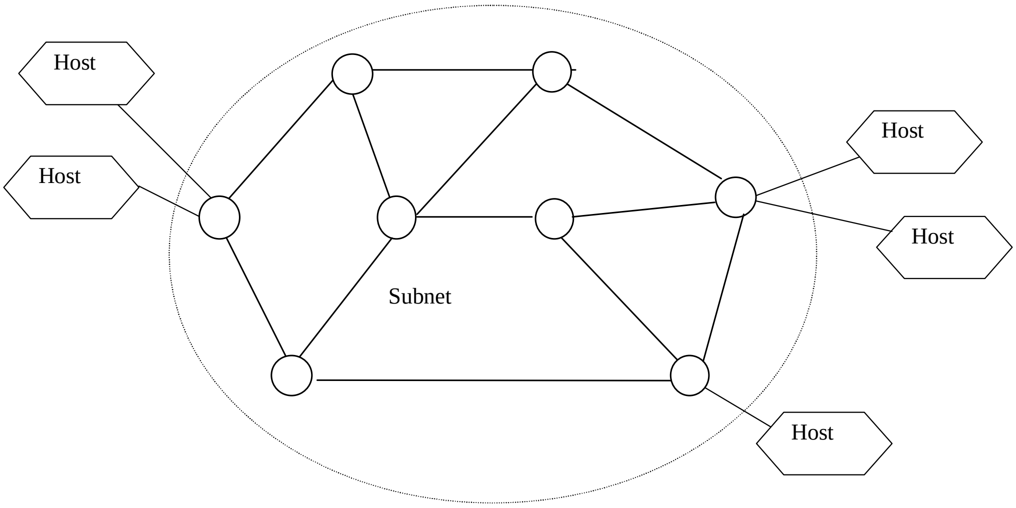
Why the difference?

One reason, as we have seen is that with most MAC techniques, performance goes down as the geographical size and the number of users increases.

One solution to this is to use bridges to interconnect LANs. This still has limited use as size of network grows. Usually at most on the order of 10 LANs are interconnected this way.

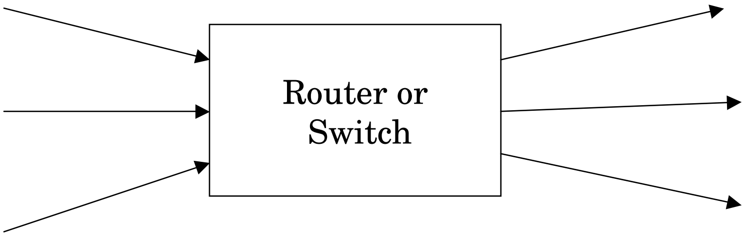
Another reason is differences in economics and traffic characteristics between WANs and LANs. These considerations favor using a point-to-point technology in a WAN.

Recall, from Lecture 1, that the topology of a WAN can be divided into ***hosts*** or ***users*** and the ***subnet.*** The subnet consists of ***nodes***, which are also called ***switches*** or ***routers.***



Routers/switches

Routers or switches are multi-input/multi-output devices. Each input/output is called a ***port*** and is connected to a transmission link. Switches receive data on input link and transmit the data on an output link. The details of how this is done is referred to as ***switching.***



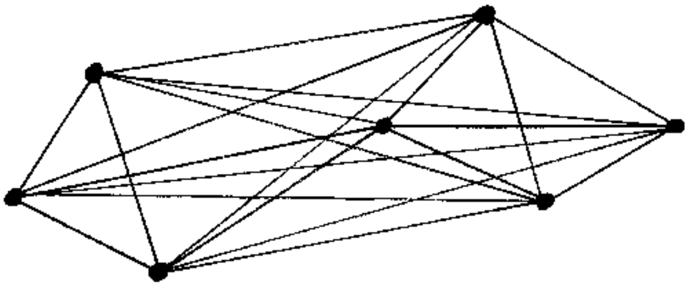
Router or Switch

The transmission links in subnet usually carry traffic from several users. How this traffic is combined onto a link is referred to as ***Multiplexing.***

In the next couple of lectures we discussing switching and multiplexing in greater detail.

The topology of WAN is determined primarily by economic considerations, i.e., the desire to efficiently share resources and take advantage of economies of scale.

Consider a group of ***N*** users. They can all be connected together in a ***fully connected mesh topology.***



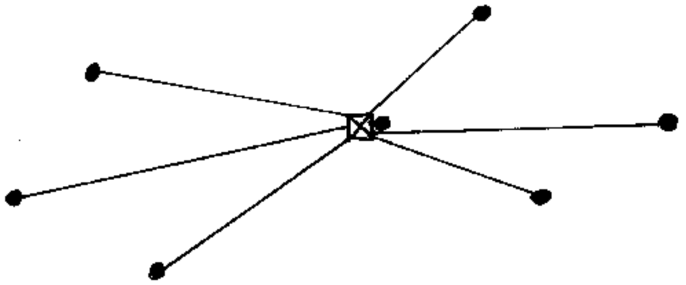
Requires $\frac{N(N \square 1)}{2}$ links.

Here the subnet only consists of point-to-point links and a switch (selector) at each input. Number of switches and switch size grows linearly with ***N.***

No multiplexing is needed.

Suppose at a given time each user sends traffic to at most one other user, in this case at most 1/(N-1) of the links are used at any time. Most WAN's do not have a fully connected topology. Instead some combination of switching/multiplexing is used.

E.g., consider telephone network with one central switch (***star topology***).

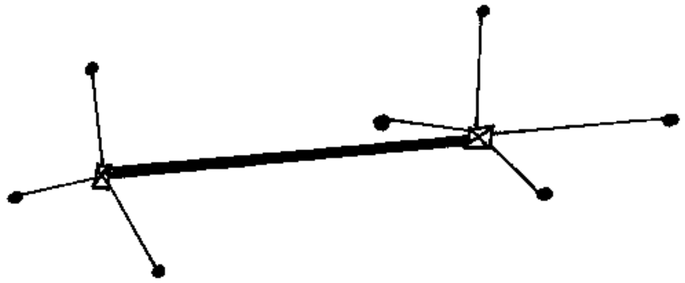


This now requires only N links and one switch with 2N ports. Now, if each user only sends traffic to one other user at any given time, then multiplexing is needed on the links outgoing from the switch (or some users may be **blocked**)

If a user can only send/receive one "call" at a given time (as in the telephone network), the star can handle as much traffic as the fully-connected topology.

Now suppose we want to connect together two groups of N users that are geographically separate.

We can use two star topologies connected together with a multiplexed ***trunk.***



To accommodate all calls, trunk needs to be able to handle N times as much traffic as access lines.

Economies of scale - A transmission line with capacity C is generally much cheaper than N lines with capacity C/N.

Using trunks we can also take advantage of statistical properties of traffic. For example, in the above case, assume that with very high probability at most $N/2$ calls are made, then may be acceptable to use trunk with capacity of less than N.