**Chapter 15: Connecting LANs, Backbone Networks and Virtual LANs**

Table of Contents

[15.1 Connecting Devices 3](#_Toc86834957)

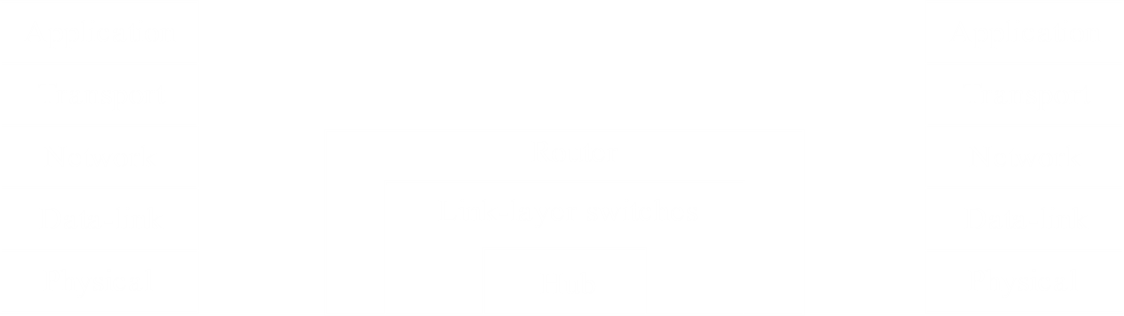
[Repeater 3](#_Toc86834958)

[Bridge 4](#_Toc86834959)

[15.3 Virtual LANs 9](#_Toc86834960)

It is very unlikely that devices are just connected directly via a LAN cable, a setup called a **passive hub**. More commonly, there are many intermediate devices involved that help connect multiple devices to a network.

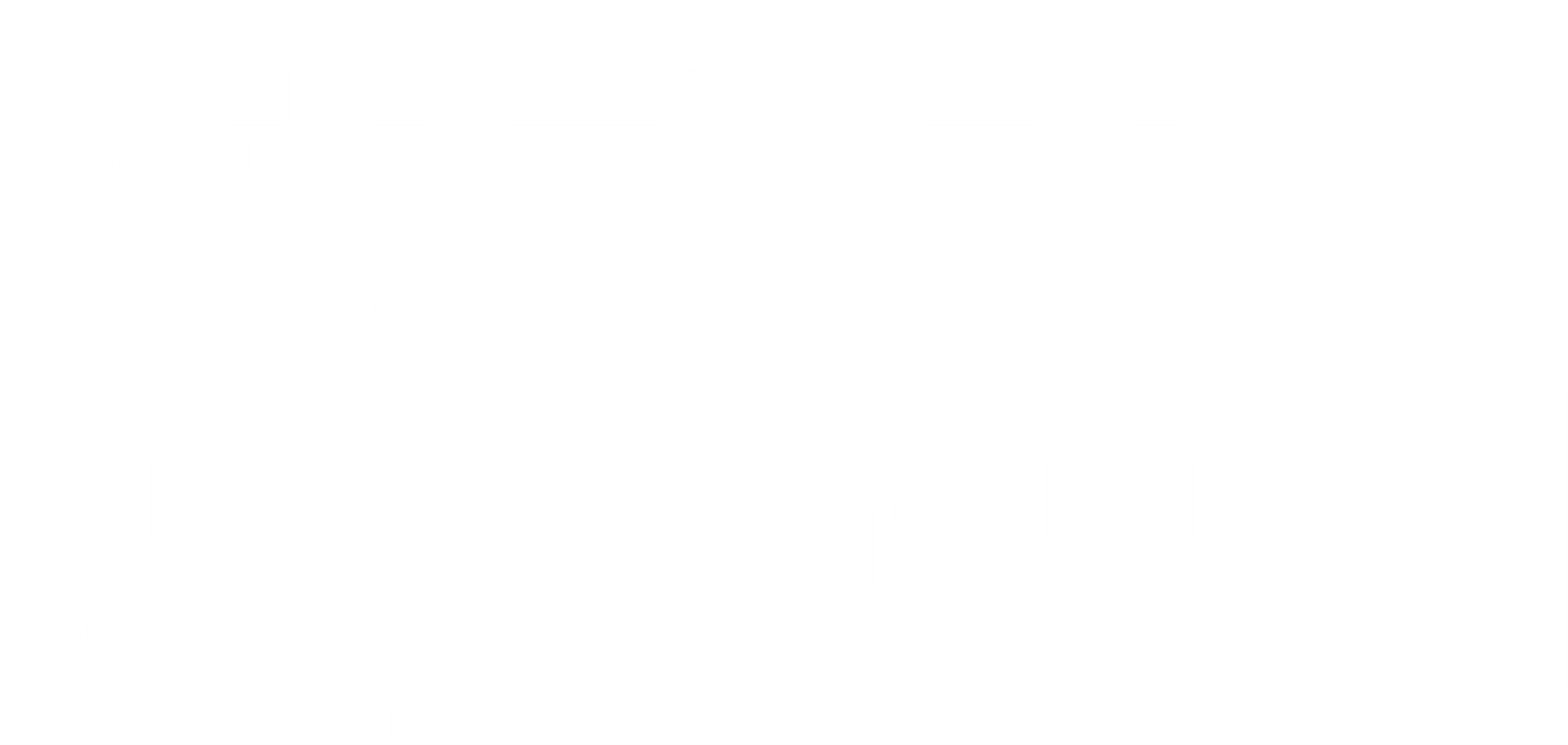
## 15.1 Connecting Devices



The diagram above shows the different categories of **connecting devices** and the **layers** in which they operate. The entirety of this section is based on this diagram.

### Repeater

A **repeater** or a hub simply repeats the incoming data. It forwards every single frame and has **no sophistication** to differentiate between frames. However, note that a repeater is a **regenerator**, not an amplifier. This means that it will regenerate the original signal, not the noise, unlike an amplifier. As such, a repeater needs to be placed at a location where noise has not corrupted the signal yet to an extent that it is unrecoverable.



Since a repeater has no sophistication, it works only at the **physical layer**. It connects two parts of the **same LAN**, not two different LANs. It cannot be used to connect LANs that use different protocols.

### Bridge

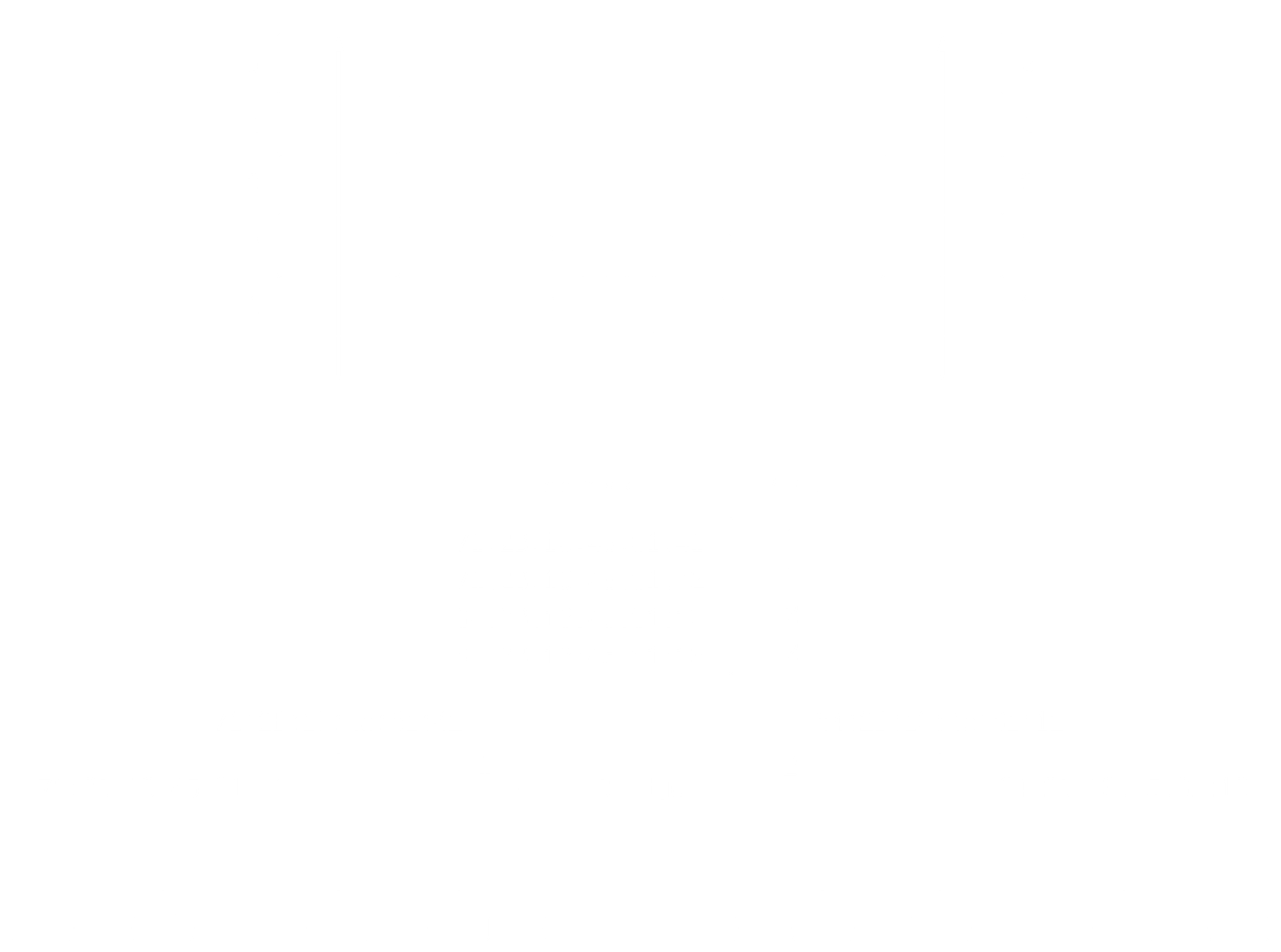
A bridge is a **layer-2 device**, meaning it works up to the data-link layer. At the physical layer, it works like a **repeater**. At the data-link layer, it can check the **physical addresses**. It has a **table** that it uses to make **filtering** decisions.

#### Learning Bridge

The specific type of bridge that we are looking into is called a **transparent** or a **learning bridge**. This is because this bridge does not change the frames in any way, unlike a router, which changes the addresses. This means two end devices have no way of knowing that there was a bridge in between them. But more importantly, it is called a learning bridge because it is **self-learning**.

The reason the bridge looks at the physical addresses at all is to determine which **port** the frame should leave from. A bridge has LAN cables connected to multiple different ports and it uses a **table** to decide which frame should go out which port.

Consider the diagram below:



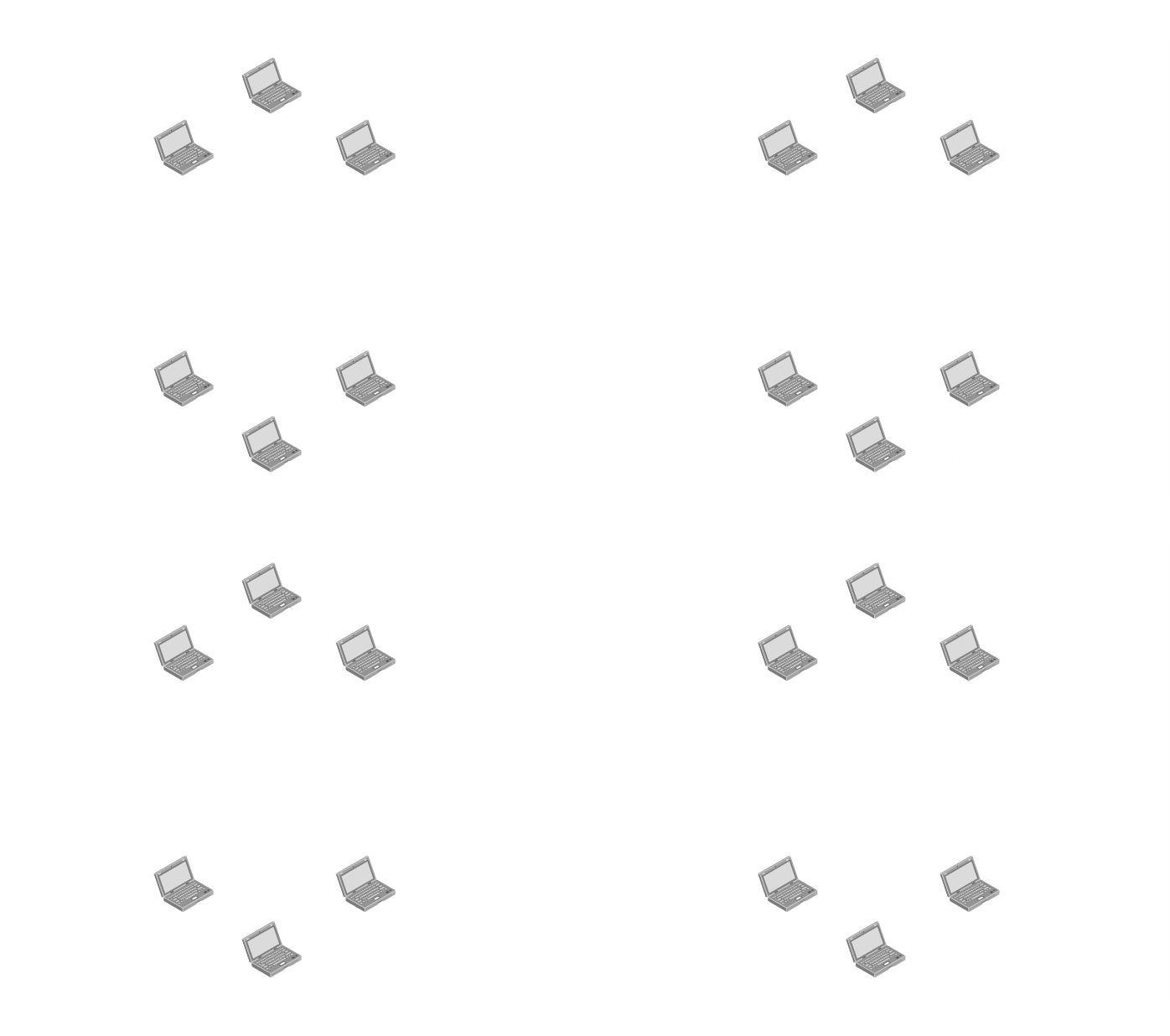
Say a packet comes from A through port 1 and is meant to go to D, through port 2. The bridge has an empty table to begin with, but it just learnt that A is connected to port 1, so it adds this information to its table.

Since the bridge does not know which port D is connected to, it broadcasts the packet through every port other than port 1. Later on, if it receives a packet from D through port 2, it will be able to add that to its table as well. Once that is done, any future packets destined for D will be sent through port 2 alone. This prevents the other LANs from being unnecessarily used.

#### Filtering

In the example we saw above, if A wanted to send a packet to B, the bridge would receive the packet, but would notice that the port for the source and the destination are the same. This means that the destination has already received the packet, since it is connected to the same line as the source. As such, the bridge drops the packet. This process is called **filtering**. It prevents the packet from being passed to every other LAN connected, thus removing unnecessary traffic.

#### Looping Problem



Consider the situation above. A packet from A is trying to get to D. However, both LAN 1 and LAN 2 are connected to two bridges. Both bridges receive the packet from A and record that it enters through port 1. Since it is destined for D, the packet leaves through port 2 from both bridges, entering LAN 2.

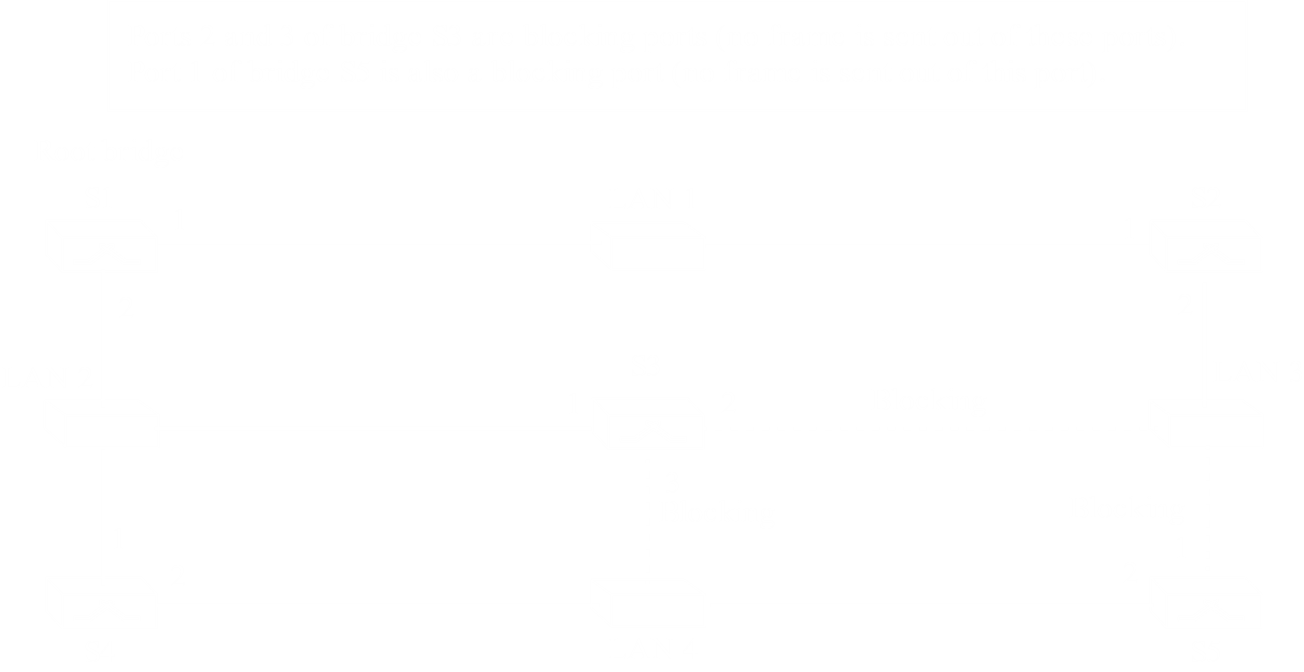
On LAN 2, the packet from Bridge 1 enters Bridge 2 and vice versa. This causes both bridges to now record that a packet that originated at A entered through port 2 and the tables of both bridges are **erroneously updated**. Again, both bridges broadcast the packet through port 1 onto LAN 1 (since they do not yet know where the destination is). Back on LAN 1, the same process repeats. We have entered a **loop**.

One may question why we would have two bridges connected like this in the first place. This may be to keep a backup bridge in case the first one fails.

#### Spanning Trees

To solve the **looping problem**, we need to convert the network from a graph into a **spanning tree**. To do this, we can use one of the famous algorithms, either Prim’s Algorithm or Kruskal’s Algorithm.

Once a spanning tree algorithm has been applied, we will know which **edges** need to be broken. Instead of actually breaking the edges, they are just blocked off and not used.

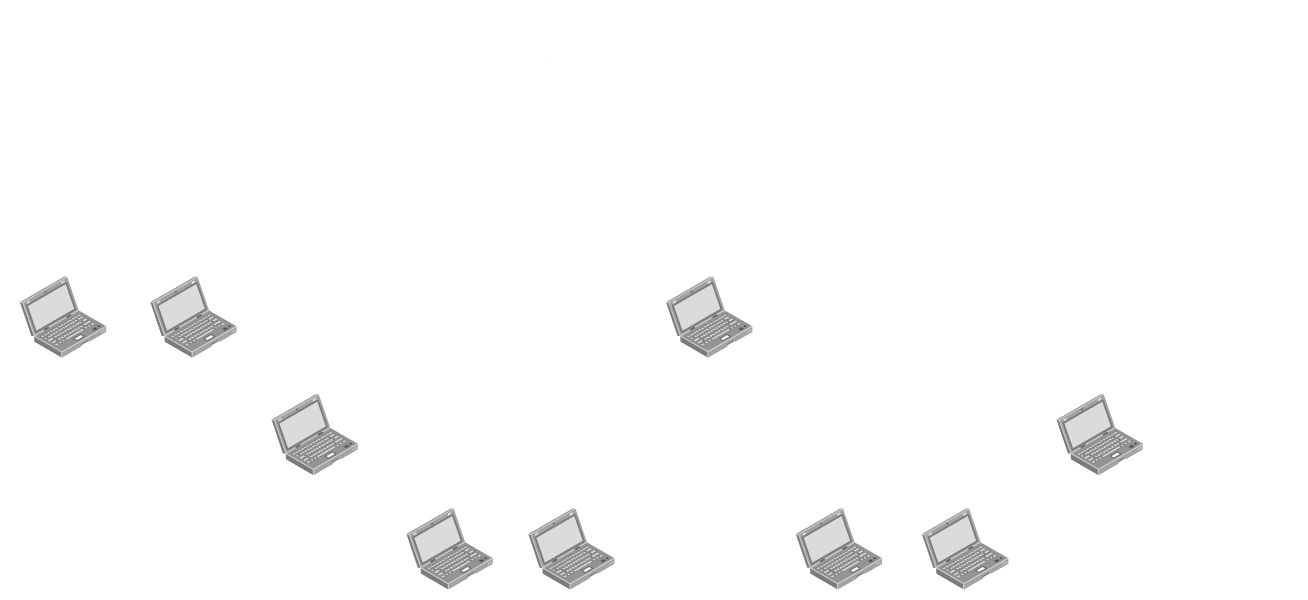


## 15.3 Virtual LANs

A **virtual local area network** (VLAN) is a LAN configured by **software** instead of by physical wiring. This is necessary because there are situations where we may need a new **broadcast domain**.

For example, if there are devices that are on different physical LANs that need to be connected together, we can take those devices and put them on a private VLAN. In this way, any packets sent out by one of the devices will only be deliverable to other devices on the VLAN. The broadcast domain of the VLAN is restricted to these devices. This provides increased security.

This is different from the situation where we just connect all the different LANs together, since in that situation, the broadcast domain would include every device on the LANs, even ones we did not want. Any packets sent out would actually be delivered to every device and the devices that are not the destination would just reject it. This is how LANs always work, but it is less secure.



Some form of **membership tag** is provided to devices on the VLAN, and only devices that have that tag can receive the packets. Devices that are on one of the LANs but not on the VLAN cannot.