**CAPITULO 4**

**4. The medium access control sublayer**

In any broadcast network, the key issue is how to determine who gets to use the channel when there is competition for it. When only a single channel is available, it is much harder to determine who should go next. Many protocols for solving the problem are known. In the literature, broadcast channels are sometimes referred to as **multiaccess channels** or **random access channels**.

The protocols used to determine who goes next on a multiaccess channel belong to a sublayer of the data link layer called the **MAC (Medium Access Control)** sublayer.

**4.1 The Channel Allocation Problem**

A channel connects each user to all other users and any user who makes full use of the channel interferes with other users who also wish to use the channel.

**4.2. Multiple access protocols**

4.2.1 ALOHA

This protocol uses short-range radios, with each user terminal sharing the same upstream frequency to send frames to the central computer. It included a simple and elegant method to solve the channel allocation problem. Although the ALOHA system, used groundbased radio broadcasting, the basic idea is applicable to any system in which uncoordinated users are competing for the use of a single shared channel.

**------ Pure ALOHA -----**

The basic idea of an ALOHA system is simple: let users transmit whenever they have data to be sent. There will be collisions, of course, and the colliding frames will be damaged. Senders need some way to find out if this is the case. In the ALOHA system, after each station has sent its frame to the central computer, this computer rebroadcasts the frame to all of the stations.

A sending station can thus listen for the broadcast from the hub to see if its frame has gotten through. In other systems, such as wired LANs, the sender might be able to listen for collisions while transmitting.

A picture containing diagram

Description automatically generatedIf the frame was destroyed, the sender just waits a random amount of time and sends it again. The waiting time must be random or the same frames will collide over and over, in lockstep.

Systems in which multiple users share a common channel in a way that can lead to conflicts are known as **contention systems**.

If the first bit of a new frame overlaps with just the last bit of a frame that has almost finished, both frames will be totally destroyed and both will have to be retransmitted later. The checksum does not (and should not) distinguish between a total loss and a near miss. Bad is bad.

A user is always in one of two states: **typing** or **waiting**. Initially, all users are in the typing state. When a line is finished, the user stops typing, waiting for a response. The station then transmits a frame containing the line over the shared channel to the central computer and checks the channel to see if it was successful. If so, the user sees the reply and goes back to typing. If not, the user continues to wait while the station retransmits the frame over and over until it has been successfully sent.

A picture containing graphical user interface

Description automatically generatedThe probability that k frames are generated during a given frame time, during which G frames are expected, is given by the Poisson distribution:



Using S = G\*Pr[0], S being the throughput (amount of data transferred),we get:

**----- Slotted ALOHA -----**

Soon after ALOHA came onto the scene a method for doubling the capacity of an ALOHA system was published. The new method divided time into discrete intervals called slots, each interval corresponding to one frame. This approach requires the users to agree on slot boundaries.

With slotted ALOHA, a station is not permitted to send whenever the user types a line. Instead, it is required to wait for the beginning of the next slot. The probability of no other traffic during the same slot as our test frame is then *e ^ (−G)*, which leads to:



4.2.2 Carrier Sense Multiple Access protocols

With slotted ALOHA, the best channel utilization that can be achieved is 1/e. In LANs, however, it is often possible for stations to detect what other stations are doing, and thus adapt their behavior accordingly. These networks can achieve a much better utilization than 1/e.

Protocols in which stations listen for a carrier and act accordingly are called **carrier sense protocols**.

**------------- Persistent CSMA -------------**

The first carrier sense protocol that we will study here is called **1-persistent CSMA (Carrier Sense Multiple Access)**.

When a station has data to send, it first listens to the channel to see if anyone else is transmitting at that moment. If the channel is idle, the stations sends its data. Otherwise, if the channel is busy, the station just waits until it becomes idle. Then the station transmits a frame. If a collision occurs, the station waits a random amount of time and starts all over again. The protocol is called 1-persistent because the station transmits with a probability of 1 when it finds the channel idle.

Unfortunately, this protocol does not completely avoid collisions. If two stations become ready in the middle of a third station’s transmission, both will wait politely until the transmission ends, and then both will begin transmitting exactly simultaneously, resulting in a collision.

More subtly, the propagation delay has an important effect on collisions. There is a chance that just after a station begins sending, another station will become ready to send and sense the channel. If the first station’s signal has not yet reached the second one, the latter will sense an idle channel and will also begin sending, resulting in a collision. This chance depends on the number of frames that fit on the channel, or the **bandwidth-delay product** of the channel.

If only a tiny fraction of a frame fits on the channel the chance of a collision happening is small. The larger the bandwidth-delay product, the more important this effect becomes, and the worse the performance of the protocol.

**------------- Nonpersistent CSMA -------------**

In this protocol, a conscious attempt is made to be less greedy than in the previous one. If the channel is already in use, the station does not continually sense it for the purpose of seizing it immediately upon detecting the end of the previous transmission. Instead, it waits a random period of time and then repeats the algorithm. Consequently, this algorithm leads to better channel utilization but longer delays than 1-persistent CSMA.

**------------- p-persistent CSMA -------------**

It applies to slotted channels. When a station becomes ready to send, it senses the channel. If it is idle, it transmits with a probability p. With a probability q = 1 − p, it defers until the next slot. If that slot is also idle, it either transmits or defers again, with probabilities p and q. This process is repeated until either the frame has been transmitted or another station has begun transmitting. In the latter case, the unlucky station acts as if there had been a collision (i.e., it waits a random time and starts again). If the station initially senses that the channel is busy, it waits until the next slot.

**------------- CSMA with Collision Detection -------------**

Another improvement is for the stations to quickly detect the collision and abruptly stop transmitting, (rather than finishing them) since they are irretrievably garbled anyway. This strategy saves time and bandwidth.

Diagram

Description automatically generatedThis protocol, known **as CSMA/CD (CSMA with Collision Detection)**, is the basis of the classic Ethernet LAN. The station’s hardware must listen to the channel while it is transmitting. If the signal it reads back is different from the signal it is putting out, it knows that a collision is occurring. The model for CSMA/CD consists of alternating contention and transmission periods, with idle periods occurring when all stations are quiet.

CSMA/CD can be in contention, transmission, or idle state.

The difference for CSMA/CD compared to slotted ALOHA is that slots in which only one station transmits (i.e., in which the channel is seized) are followed by the rest of a frame. This difference will greatly improve performance if the frame time is much longer than the propagation time.

4.2.3 Collision Free Protocols

Although collisions do not occur with CSMA/CD once a station has unambiguously captured the channel, they can still occur during the contention period.

**------------- Token Passing -------------**

In this protocol, a small message, called a token, passes from one station to the next in the same predefined order. The token represents permission to send. If a station has a frame queued for transmission when it receives the token, it can send that frame before it passes the token to the next station. If it has no queued frame, it simply passes the token.

A picture containing necklet, accessory

Description automatically generatedIn a **token ring** protocol, the topology of the network is used to define the order in which stations send. The stations are connected one to the next in a single ring. Passing the token to the next station then simply consists of receiving the token in from one direction and transmitting it out in the other direction. Frames are also transmitted in the direction of the token.

Note that we do not need a physical ring to implement token passing. The channel connecting the stations might instead be a single long bus. Each station then uses the bus to send the token to the next station in the predefined sequence. This protocol is called **token bus**.

4.2.5 Wireless LAN protocols

Wireless systems cannot normally detect a collision while it is occurring. The received signal at a station may be tiny, perhaps a million times fainter than the signal that is being transmitted. Finding it is like looking for a ripple on the ocean. Instead, acknowledgements are used to discover collisions and other errors after the fact.

There is an even more important difference between wireless LANs and wired LANs. A station on a wireless LAN may not be able to transmit frames to or receive frames from all other stations because of the limited radio range of the stations. In wired LANs, when one station sends a frame, all other stations receive it. The absence of this property in wireless LANs causes a variety of complications.

An early and influential protocol that tackles these problems for wireless LANs is **MACA (Multiple Access with Collision Avoidance)**. The basic idea behind it is for the sender to stimulate the receiver into outputting a short frame, so stations nearby can detect this transmission and avoid transmitting for the duration of the upcoming (large) data frame. This technique is used instead of carrier sense.

**4.3 Ethernet**

4.3.1 Classical Ethernet Physical Layer

Classic Ethernet snaked around the building as a single long cable to which all the computers were attached. The first variety, popularly called **thick Ethernet**, resembled a yellow garden hose. It was succeeded by **thin Ethernet**, which bent more easily and made connections using industry-standard BNC connectors.

Diagram

Description automatically generatedEach version of Ethernet has a maximum cable length per segment. To allow larger networks, multiple cables can be connected by **repeaters**. A repeater is a physical layer device that receives, amplifies and retransmits signals in both directions.

4.3.2 Classic Ethernet MAC sublayer protocol

Table

Description automatically generatedThe format used to send frames is as described: first comes a Preamble of 8 bytes, each containing the bit pattern 10101010 (except for the last byte, in which the last 2 bits are set to 11). This last byte is called the Start of Frame delimiter. The last two 1 bits tell the receiver that the rest of the frame is about to start.

Next come two addresses, one for the destination and one for the source. They are each 6 bytes long. The first transmitted bit of the destination address is a 0 for ordinary addresses and a 1 for group addresses.

When a frame is sent to a group address, all the stations in the group receive it. Sending to a group of stations is called **multicasting**. The special address consisting of all 1 bits is reserved for **broadcasting**. A frame containing all 1s in the destination field is accepted by all stations on the network.

Conversely, broadcasting does not differentiate between stations at all, so it does not require any group management. Source addresses are globally unique, assigned centrally by IEEE to ensure that no two stations anywhere in the world have the same address.

Next comes the *Type* or *Length* field, depending on whether the frame is Ethernet or IEEE 802.3. Ethernet uses a *Type* field to tell the receiver what to do with the frame. Multiple network-layer protocols may be in use at the same time on the same machine, so when an Ethernet frame arrives, the operating system has to know which one to hand the frame to. The *Type* field specifies which process to give the frame to.

IEEE 802.3 decided that this field would carry the length of the frame, since the Ethernet length was determined by looking inside the data.

This meant there was no way for the receiver to figure out what to do with an incoming frame. That problem was handled by the addition of another header for the LLC (Logical Link Control) protocol within the data. It uses 8 bytes to convey the 2 bytes of protocol type information.

Now the rule is that any number there less than or equal to 0x600 (1536) can be interpreted as *Length*, and any number greater than 0x600 can be interpreted as *Type*.

Diagram

Description automatically generatedIn addition to there being a maximum frame length, there is also a minimum frame length. While a data field of 0 bytes is sometimes useful, it causes a problem. When a transceiver detects a collision, it truncates the current frame, which means that stray bits and pieces of frames appear on the cable all the time. To make it easier to distinguish valid frames from garbage, Ethernet requires that valid frames must be at least 64 bytes long, from destination address to checksum, including both. If the data portion of a frame is less than 46 bytes, the Pad field is used to fill out the frame to the minimum size.

Another (and more important) reason for having a minimum length frame is to prevent a station from completing the transmission of a short frame before the first bit has even reached the far end of the cable, where it may collide with another frame.

4.3.4 Switched Ethernet

Ethernet soon began to evolve away from the single long cable architecture of classic Ethernet, toward a different kind of wiring pattern, in which each station has a dedicated cable running to a central **hub**. A hub simply connects all the attached wires electrically, as if they were soldered together.

However, hubs do not increase capacity because they are logically equivalent to the single long cable of classic Ethernet. As more and more stations are added, each station gets a decreasing share of the fixed capacity.

Fortunately, there is another way to deal with increased load: switched Ethernet. The heart of this system is a **switch** containing a high-speed backplane that connects all of the ports.

Switches only output frames to the ports for which those frames are destined. When a switch port receives an Ethernet frame from a station, the switch checks the Ethernet addresses to see which port the frame is destined for. This step requires the switch to be able to work out which ports correspond to which addresses.

In a hub, all stations are in the same **collision domain**. They must use the CSMA/CD algorithm to schedule their transmissions. In a switch, each port is its own independent collision domain. In the common case that the cable is full duplex, both the station and the port can send a frame on the cable at the same time, without worrying about other ports and stations. However, if the cable is half duplex, the station and the port must contend for transmission with CSMA/CD in the usual way.

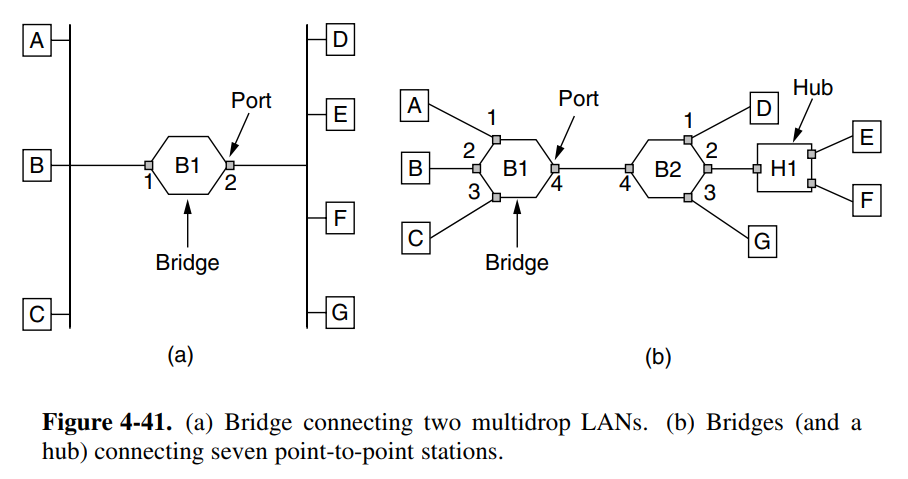
**4.8 Data Link Layer Switching**

Many organizations have multiple LANs and wish to connect them. We can do this when the connections are made with devices called **bridges**. The Ethernet switches previously described are a modern name for bridges; they provide functionality that goes beyond classic Ethernet and Ethernet hubs to make it easy to join multiple LANs into a larger and faster network.

Bridges operate in the data link layer, so they examine the data link layer addresses to forward frames.

4.8.2 Learning Bridges

On the left-hand side, two multidrop LANs, such as classic Ethernets, are joined by a special station—the bridge—that sits on both LANs. On the right-hand side, LANs with point-to-point cables, including one hub, are joined together.



There is a difference, however, in how the bridged LANs are built. To bridge multidrop LANs, a bridge is added as a new station on each of the multidrop LANs. To bridge point-to-point LANs, the hubs are either connected to a bridge or, preferably, replaced with a bridge to increase performance.

Now let us consider what happens inside the bridges. Each bridge operates in promiscuous mode, that is, it accepts every frame transmitted by the stations attached to each of its ports. The bridge must decide whether to forward or discard each frame, and, if the former, on which port to output the frame. This decision is made by using the destination address.

A simple way to implement this scheme is to have a big (hash) table inside the bridge. The table can list each possible destination and which output port it belongs on.

When the bridges are first plugged in, all the hash tables are empty. None of the bridges know where any of the destinations are, so they use a flooding algorithm: every incoming frame for an unknown destination is output on all the ports to which the bridge is connected except the one it arrived on. As time goes on, the bridges learn where destinations are. Once a destination is known, frames destined for it are put only on the proper port; they are not flooded.

The algorithm used by the bridges is **backward learning**. By looking at the source addresses, they can tell which machines are accessible on which ports.

To handle dynamic topologies, whenever a hash table entry is made, the arrival time of the frame is noted in the entry. Whenever a frame whose source is already in the table arrives, its entry is updated with the current time. Thus, the time associated with every entry tells the last time a frame from that machine was seen.

Periodically, a process in the bridge scans the hash table and purges all entries more than a few minutes old. In this way, if a computer is unplugged from its LAN, moved around the building, and plugged in again somewhere else, within a few minutes it will be back in normal operation, without any manual intervention. This algorithm also means that if a machine is quiet for a few minutes, any traffic sent to it will have to be flooded until it next sends a frame itself.

The routing procedure for an incoming frame depends on the port it arrives on (the source port) and the address to which it is destined (the destination address):

1. If the port for the destination address is the same as the source port, discard the frame.
2. If the port for the destination address and the source port are different, forward the frame on to the destination port.
3. If the destination port is unknown, use flooding and send the frame on all ports except the source port.

4.8.4. Repeaters, Hubs, Bridges, Switches, Routers, and Gateways

Graphical user interface, application

Description automatically generated

The key to understanding these devices is to realize that they operate in different layers. The layer matters because different devices use different pieces of information to decide how to switch.

The packet goes to the data link layer, which adds its own header and checksum (CRC) and gives the resulting frame to the physical layer for transmission, for example, over a LAN.

Now let us look at the switching devices and see how they relate to the packets and frames.

At the bottom, in the physical layer, we find the **repeaters**. A signal appearing on one cable is cleaned up, amplified, and put out on another cable. Repeaters do not understand frames, packets, or headers. They understand the symbols that encode bits as volts.

Next, we come to the **hubs**. A hub has a number of input lines that it joins electrically. Frames arriving on any of the lines are sent out on all the others. If two frames arrive at the same time, they will collide, just as on a coaxial cable. All the lines coming into a hub must operate at the same speed.

Now let us move up to the data link layer, where we find **bridges** and **switches**. A bridge connects two or more LANs. Like a hub, a modern bridge has multiple ports, usually enough for 4 to 48 input lines of a certain type. Unlike in a hub, each port is isolated to be its own collision domain; if the port has a full-duplex point-to-point line, the CSMA/CD algorithm is not needed. When a frame arrives, the bridge extracts the destination address from the frame header and looks it up in a table to see where to send the frame. For Ethernet, this address is the 48-bit destination address. The bridge only outputs the frame on the port where it is needed and can forward multiple frames at the same time. If frames come in faster than they can be retransmitted, the bridge may run out of buffer space and have to start discarding frames.

Switches are modern bridges by another name. Bridges were developed when classic Ethernet was in use, so they tend to join relatively few LANs and thus have relatively few ports.

4.8.5 Virtual LANs

Related to broadcasts is the problem that once in a while a network interface will break down or be misconfigured and begin generating an endless stream of broadcast frames. The result of this **broadcast storm** is that: (1) the entire LAN capacity is occupied by these frames, and (2) all the machines on all the interconnected LANs are crippled just processing and discarding all the frames being broadcast.

In response to customer requests for more flexibility, network vendors began working on a way to rewire buildings entirely in software. The resulting concept is called a **VLAN (Virtual LAN)**.

VLANs are based on VLAN-aware switches. To set up a VLAN-based network, the network administrator decides how many VLANs there will be, which computers will be on which VLAN, and what the VLANs will be called. Often the VLANs are (informally) named by colors, since it is then possible to print color diagrams showing the physical layout of the machines.

Diagram

Description automatically generated

To make the VLANs function correctly, configuration tables have to be set up in the bridges. These tables tell which VLANs are accessible via which ports.

**-------------- The IEEE 802.1Q Standard --------------**

To implement this scheme, bridges need to know to which VLAN an incoming frame belongs. Without this information when bridge B2 gets a frame from bridge B1, it cannot know whether to forward the frame on the gray or white VLAN. If we were designing a new type of LAN, it would be easy enough to just add a VLAN field in the header. But what to do about Ethernet, which is the dominant LAN, and did not have any spare fields lying around for the VLAN identifier?

After much discussion, it did the unthinkable and changed the Ethernet header. The new format was published in IEEE standard **802.1Q**.The new format contains a VLAN tag. The key to the solution is to realize that the VLAN fields are only actually used by the bridges and switches and not by the user machines. Thus, it is not really essential that they are present on the lines going out to the end stations as long as they are on the line between the bridges.

Because there can be computers (and switches) that are not VLAN aware, the first VLAN-aware bridge to touch a frame adds VLAN fields and the last one down the road removes them. With 802.1Q, frames are colored depending on the port on which they are received. For this method to work, all machines on a port must belong to the same VLAN, which reduces flexibility.

Now let us take a look at the 802.1Q frame format. The only change is the addition of a pair of 2-byte fields. The first one is the *VLAN protocol ID*. It always has the value 0x8100. Since this number is greater than 1500, all Ethernet cards interpret it as a type rather than a length.

Diagram

Description automatically generated

The second 2-byte field contains three subfields. The main one is the VLAN identifier, occupying the low-order 12 bits. This is what the whole thing is about—the color of the VLAN to which the frame belongs. The 3-bit *Priority field* makes it possible to distinguish hard real-time traffic from soft real-time traffic from time insensitive traffic in order to provide better quality of service over Ethernet.

The last field, *CFI (Canonical format indicator)*. It was originally intended to indicate the order of the bits in the MAC addresses (little-endian versus big-endian), but that use got lost in other controversies. Its presence now indicates that the payload contains a freeze-dried 802.5 frame that is hoping to find another 802.5 LAN at the destination while being carried by Ethernet in between.