**Comparison Of Scheduling Approaches In Mac Protocols**

Computer Engineering Dep, the Amirkabir University of Tehran, Iran

Author: Amirhossein Najafizadeh

Date: April 2023

**Abstract**

In Data communications between system in a local area network, medium access control[[1]](#footnote-1) protocols are mechanisms for allowing several users to transmit their data into a common medium or channel[[2]](#footnote-2). But when multiple users want to access a same channel at a same time there will be "colliding" and data losses. Therefor we need a mechanism to control the channel so that users can send their data without any colliding. Scheduling is an approach to control the channel and it comes with three major methods, reserve system, polling, and token ring. In this paper we are going to compare these methods to see which one is better to use.

**Medium Access Control**

Switched networks provide interconnection between users by means of transmission lines, multiplexers, and switches. The transfer of information across such networks requires routing tables to direct the information from source to destination. To scale to a very large size, the addressing scheme in switched networks is typically hierarchical to help provide location information that assists the routing protocol in carrying out its task.

Broadcast networks, in contrast, are much simpler. Because all information is received by all users, routing is not necessary. A nonhierarchical addressing scheme is sufficient to indicate which user the information is destined to. However, broadcast networks require a medium access control protocol to orchestrate the transmissions from the various users. Local area networks (LANs), with their emphasis on low cost and simplicity, have been traditionally based on the broadcast approach. In this chapter we consider medium access control protocols and local area networks. We also consider the access methods used in cellular radio networks.

In broadcast networks a single transmission medium is shared by a community of users. For this reason, we also refer to these networks as multiple access networks. Typically, the information from a user is broadcast into the medium, and all the stations attached to the medium listen to all the transmissions. There is potential for user transmissions interfering or “colliding” with each other, and so a protocol has to be in place to prevent or minimize such interference.

Medium access control protocols are mechanisms that allow several users or transmitters to access a common medium or channel. They play an important role in the development of both wired and wireless networks. The role of the medium access control (MAC) protocols is to coordinate the access to the channel so that information gets through from a source to a destination in the same broadcast network.

The basic role of both protocol classes is the same: to transfer blocks of user information despite transmission impairments. In the case of peer-to-peer protocols, the main concern is loss, delay, and resequencing of PDUs during transmis- sion. In the case of MAC protocols, the main concern is interference from other users.

Figure 1 shows a generic multiple access communications situation in which a number of user stations share a transmission medium. *M* denotes the number of stations. The transmission medium is broadcast in nature, and so all the other stations that are attached to the medium can hear the transmission from any given station. When two or more stations transmit simultaneously, their signals will collide and interfere with each other.

There are two broad categories of schemes for sharing a transmission medium. The first category involves a static and collision-free sharing of the medium. We refer to these as channelization schemes because they involve the partitioning of the medium into separate channels that are then dedicated to particular users. Channelization tech- niques are suitable when stations generate a steady stream of information that makes efficient use of the dedicated channel. The second category involves a dynamic sharing of the medium on a per frame basis that is better matched to situations in which the user traffic is bursty. We refer to this category as MAC schemes. The primary func- tion of medium access control is to minimize or eliminate the incidence of collisions to achieve a reasonable utilization of the medium. The two basic approaches to medium access control are random access and scheduling.

In Figure 2 you can see the sharing approaches in MAC.



Figure 1. Multiple access communication



Figure 2. Approaches to sharing a transmission medium[[3]](#footnote-3)

**Delay-Bandwidth Product and MAC Performance**

Consider the situation shown in Figure 3 in which two stations are trying to share a common medium. Let’s develop an access protocol for this system. Suppose that when a station has a frame to send, the station first listens to the channel to see whether it is busy with a transmission from the other station. If it is not busy, then the station begins to transmit, but it continues observing the signal in the channel to make sure that its signal is not corrupted by a signal from the other station. The signal from station A does not reach station B until time . If station B has not begun a transmission by that time, then station A is assured that station B will refrain from transmitting thereafter and so station A has captured the channel and its entire message will get through.

Figure 3 shows what happens when a collision of frame transmissions takes place. In this case station B must have begun its transmission sometime between times *t* = 0 and *t* = . By time *t* = 2, at the latest, station A will find out about the collision. At this point both stations are aware that they are competing for the channel. Some mechanism for resolving this contention is required. We will suppose for simplicity that both stations know the value of the propagation delay *tprop* and that they measure the time from when they began transmitting to when a collision occurs. The station that began transmitting earlier (which measures the aforementioned time to be greater than /2) is declared to be the “winner” and proceeds to retransmit its frame as soon as the channel goes quiet. The “losing” station defers and remains quiet until the frame transmission from the other station is complete. For the sake of fairness, we suppose that the winning station is compelled to remain quiet for time 2after it has completed its frame transmission. This interval gives the losing station the opportunity to capture the channel and transmit its frame.

Thus we see for this example that a time approximately equal to 2is required to coordinate the access for each frame transmitted. Let *L* be the number of bits in a frame. The sending station then requires *X* = *L*/*R* seconds to transmit the frame, where *R* is the transmission bit rate. Therefore a frame transmission requires *L* / *R* + 2seconds. The throughput of a system is defined as the actual rate at which information is sent over the channel, and is measured in bits/second or frames/second. The *maximum throughput* in this example in bps is:



Figure 3.

and the normalized *maximum throughput* or efficiency is given by

where the normalized delay-bandwidth product *a* is defined as the ratio of the one way delay bandwidth product to the average frame length

When *a* is much smaller than 1, the medium can be used very efficiently by using the above protocol. For example, if *a* = 0.01, then the efficiency is 1/1.02 = 0.98. As *a* becomes larger, the channel becomes more inefficient. For example if *a* = 0.5, then the efficiency is 1/2 = 0.50.

The selection of a MAC protocol for a given situation depends on delay-bandwidth product and throughput efficiency, as well as other factors. The *transfer delay* experi- enced by frames is an important performance measure. The frame transfer delay *T* is defined as the time that elapses from when the first bit of the frame arrives at the source MAC to when the last bit of the frame is delivered to the destination MAC.

These approaches attempt to produce an orderly access to the transmission medium.

**Reservation Systems**

Figure 4 shows a basic reservation system. The stations take turns transmitting a single frame at the full rate *R* bps, and the transmissions from the stations are organized into cycles that can be variable in length. Each cycle begins with a reservation interval. In the simplest case the reservation interval consists of *M* minislots, one minislot per station. Stations use their corresponding minislot to indicate that they have a frame to transmit in a corresponding cycle. The stations announce their intention to transmit a frame by broadcasting their reservation bit during the appropriate minislot. By listening to the reservation interval, the stations can determine the order of frame transmissions in the corresponding cycle. The length of the cycle will then correspond to the number of stations that have a frame to transmit. Note that variable-length frames can be handled if the reservation message includes frame-length information.

The basic reservation system described above generalizes and improves on a time- division multiplexing scheme by taking slots that would have gone idle and making them available to other stations. Figure 6.20a shows an example of the operation of the basic reservation system. In the initial portion only stations 3 and 5 have frames to transmit. In the middle portion of the example, station 8 becomes active, and the cycle is expanded from two slots to three slots.

Let us consider the maximum attainable throughput for this system. Assume that the propagation delay is negligible, that frame transmission times are *X* = 1 time unit, and that a reservation minislot requires v time units where v < 1. Assume also that one minislot is required per frame reservation. Each frame transmission then requires



Figure 4

1 + v time units. The maximum throughput occurs when all stations are busy, and hence the maximum throughput is

for one frame reservation/minislot

It can be seen that very high throughputs are achievable when v is very small in comparison to 1. Thus, for example, if v = 5%, then ρ*max* = 95%.

Suppose that the propagation delay is not negligible. As shown in Figure 5, the stations transmit their reservations in the same way as before, but the reservations do not take effect until some fixed number of cycles later. For example, if the cycle length is constrained to have some minimum duration that is greater than the round-trip propagation delay, then the reservations would take effect in the second following cycle.

The basic reservation system can be modified so that stations can reserve more than one slot per frame transmission per minislot. Suppose that a minislot can reserve up to *k* frames. The maximum cycle size occurs when all stations are busy and is given by *M*v + *Mk* time units. One such cycle transmits *Mk* frames, and so we see that the maximum achievable throughput is now

for *k* frame reservations/minislot

Now let us consider the impact of the number of stations on the performance of the system. The effect of the reservation intervals is to introduce overhead that is proportional to *M*, that is, the reservation interval is *M*v. If *M* becomes very large, this overhead can become significant. This situation becomes a serious problem when a very large number of stations transmit frames infrequently. The reservation minislots are incurred in every cycle, even though most stations do not transmit. The problem can be addressed by *not* allocating a minislot to each station and instead making stations contend for a reservation minislot by using a random access technique such as ALOHA or slotted ALOHA. If slotted ALOHA is used, then each successful reservation will require 1/0.368 = 2.71 minislots on average. Therefore, the maximum achievable throughput for a reservation ALOHA system is



Figure 5

If the propagation delay is not negligible, then it is possible for slots to go unused because reservations cannot take effect quickly enough. This situation results in a reduction in the maximum achievable throughput. For this reason reservation systems are sometimes modified so that frames that arrive during a cycle can attempt to “cut ahead of the line” by being transmitted during periods that all stations know have not been reserved. If a frame is successfully transmitted this way, its reservation in a following cycle is canceled.

**Polling**

The reservation systems in the previous section required that stations make explicit reservations to gain access to the transmission medium. We now consider polling systems in which stations *take turns* accessing the medium. At any given time only one of the stations has the right to transmit into the medium. When a station is done transmitting, some mechanism is used to pass the right to transmit to another station.

There are different ways for passing the right to transmit from station to station. Figure 6 shows the situation in which *M* stations communicate with a host com- puter. The system consists of an outbound line in which information is transmitted from the host computer to the stations and an inbound line that must be shared with the *M* stations. The inbound line is a shared medium that requires a medium access control to coordinate the transmissions from the stations to the host computer. The technique developed for this system involves the host computer acting as a central



Figure 6

controller that issues control messages to coordinate the transmissions from the sta- tions. The central controller sends a *polling message* to a particular station. When polled, the station sends its inbound frames and indicates the completion of its trans- mission through a *go-ahead message*. The central controller might poll the stations in round-robin fashion, or according to some other pre-determined order.

The total walk time τ′ is the sum of the walk times in one cycle and represents the minimum time for one round of polling of all the stations. The walk time between consecutive stations *t*′ is determined by several factors. The first factor is the propa- gation time required for a signal to propagate from one station to another. This time is clearly a function of distance. Another factor is the time required for a station to begin transmitting after it has been polled. This time is an implementation issue. A third factor is the time required to transmit the polling message. These three factors combine to determine the total walk time of the system.

The cycle time *Tc* is the total time that elapses between the start of two consecutive polls of the same station. The cycle time is the sum of the *M* walk times and the *M* station transmission times. The average cycle time *E*[*Tc*] can be found as follows. Let λ/*M* frames/second be the average arrival rate of frames for transmission from a station, and let *E*[*Nc*] be the average number of message arrivals to a station in one cycle time. If we assume that all messages that arrive in a cycle time are transmitted the next time the station is polled, then *E*[*Nc*] = (λ/*M*)*E*[*Tc*]. Assume that all stations have the same frame transmission time *X*. Therefore, the time spent at each station is *E*[*Nc*]*X* + *t*′, where *t*′ is the walk time. The average cycle time is then *M* times the average time spent at each station:

The preceding equation can be solved for *E*[*Tc*]:

Note the behavior of the mean cycle time as a function of load ρ = λ*X*. Under light load the cycle time is simply required to poll the full set of stations, and the mean cycle time is approximately τ′, since most stations do not have messages to transmit. However, as the load approaches 1, the cycle time can increase without bound.

The walk times required to pass control of the access right to the medium can be viewed as a form of overhead. The normalized overhead per cycle is then given by the ratio of the total walk time to the cycle time. Note that as the load approaches 1, τ′ remains constant while the cycle time increases without bound. In effect, the overhead due to polling, that is, the walk time, becomes negligible. This implies that polling can achieve a maximum normalized throughput of 100% when stations are allowed to send all of the frames in their buffers.

It is interesting to consider the cases where the walk time approaches zero. This yields a system in which frames from different queues are transmitted in turn according to a polling list. The zero walk time implies that the system can switch between queues very quickly, with negligible overhead. In this sense, the system operates much like a statistical multiplexer that serves user queues in some order.

In situations where stations carry delay-sensitive traffic it is desirable to have a cycle time that has a strict upper bound. The stations will then receive polling messages at some minimum frequency. The cycle time can be bounded by limiting the time or amount of information that a station is allowed to send per poll. For example, if a station is limited to one frame transmission per poll then the maximum cycle time is given by *MX* + τ′. Note however that the maximum normalized throughput will now be given by *MX*/(*MX* + τ′) = (1 + τ′/*MX*)−1 which is less than 100 percent.

**Token-Passing Rings**

Polling can be implemented in a distributed fashion on networks with a ring topology. As shown in Figure 7, such ring networks consist of station interfaces that are connected by point-to-point digital transmission lines. Each interface acts like a repeater in a digital transmission line but has some additional functions. An interface in the listen mode reproduces each bit that is received from its input to its output after some constant delay, ideally in the order of one bit time. This delay allows the interface to monitor the passing bit stream for certain patterns. For example, the interface will be looking for the address of the attached station. When such an address is observed, the associated frame of information is copied bit by bit to the attached station. The interface also monitors the passing bit stream for the pattern corresponding to a “free token.”

When a free token is received and the attached station has information to send, the interface changes the passing token to busy by changing a particular bit in the passing stream. In effect, receiving a free token corresponds to receiving a polling message. The station interface then changes to the transmit mode where it proceeds to transmit frames of information from the attached station. These frames circulate around the ring and are copied at the destination station interfaces.

While the station is transmitting its information, it is also receiving information at the input of the interface. If the time to circulate around the ring is less than the time to transmit a frame, then this arriving information corresponds to bits of the same frame that the station is transmitting. When the ring circulation time is greater than a frame transmission time, more than one frame may be present in the ring at any given time. In such cases the arriving information could correspond to bits of a frame from a different station, so the station must buffer these bits for later transmission.



Figure 7



Figure 8

A frame that is inserted into the ring must be removed. One approach to frame removal is to have the destination station remove the frame from the ring. Another approach is to allow the frame to travel back to the transmitting station. This approach is usually preferred because the transmitting station interface can then forward the arriving frame to its attached station, thus providing a form of acknowledgment.

Token rings can also differ according to the method used to reinsert the token after transmission has been completed. There are three approaches to token reinsertion, as shown in Figure 8. The main differences between the methods arise when the ring latency is larger than the frame length. The ring latency is defined as the number of bits that can be simultaneously in transit around the ring. In the *multitoken operation*, the free token is transmitted immediately after the last bit of the data frame. This approach minimizes the time required to pass a free token to the next station. It also allows several frames to be in transit in different parts of the ring.

The second approach, the *single-token operation*, involves inserting the free token after the last bit of the busy token is received back and the last bit of the frame is transmitted. If the frame is longer than the ring latency, then the free token will be inserted immediately after the last bit of the frame is transmitted, so the operation is equivalent to multitoken operation. However, if the ring latency is greater than the frame length, then a gap will occur between the time of the last bit transmission and the reinsertion of the free token as shown in Figure 8. The recovery from errors in the token is simplified by allowing only one token to be present in the ring at any given time.

In the third approach, a *single-frame operation*, the free token is inserted after the transmitting station has received the last bit of its frame. This approach allows the transmitting station to check the return frame for errors before relinquishing control of the token. Note that this approach corresponds to multitoken operation if the frame length is augmented by the ring latency.

The token-ring operation usually also specifies a limit on the time that a station can transmit. One approach is to allow a station to transmit an unlimited number of frames each time a token is received. This approach minimizes the delay experienced by frames but allows the time that can elapse between consecutive arrivals of a free token to a station to be unbounded. For this reason, a limit is usually placed either on the number of frames that can be transmitted each time a token is received or on the total time that a station may transmit information into the ring. These limits have the effect of placing a bound on the time that elapses between consecutive arrivals of a free token at a given station.

The introduction of limits on the number of frames that can be transmitted per token affects the maximum achievable throughput. Suppose that a maximum of one frame can be transmitted per token. Let τ′ be the ring latency (in seconds) and *a*′ be the ring latency normalized to the frame transmission time. We then have

where τ is the total propagation delay around the ring, *b* is the number of bit delays in an interface, *M b* is the total delay introduced by the *M* station interfaces, and *R* is the speed of the transmission lines. The maximum throughput occurs when all stations transmit a frame. If the system uses multitoken operation, the total time taken to transmit the frames from the *M* stations is *MX* + τ ′ . Because *M X* of this time is spent transmitting information, the maximum normalized throughput is then

Now suppose that the ring uses single-token operation. We can see that the effective frame duration is the maximum of *X* and τ′. Therefore, the maximum normalized throughput is then

When the frame transmission time is greater than the ring latency, we see that the single- token operation has the same maximum throughput as multitoken operation. However, when the ring latency is larger than the frame transmission time, that is, *a*′ > 1, then the maximum throughput is less than that of multitoken operation.

Finally, in the case of single-frame operation the effective frame transmission time is always *X* + τ′. Therefore, the maximum throughput is given by

We see that the maximum throughput for single-frame operation is the lowest of the three approaches. Note that when the ring latency is much bigger than the frame transmission time, the maximum throughput of both the single-token and single-frame approaches is approximately 1/*a*′. Recall from Figure 8 that this situation occurs when the distance of the ring becomes large or the transmission speed becomes very high. Figure 9 shows the maximum throughput for the three approaches for different values of *a*′. It is clear that single-frame operation has the lowest maximum throughput for all values of *a*′. Multitoken operation, on the other hand, has the highest maximum throughput for all values of *a*′. In fact, multitoken operation is sensitive to the per hop

latency *a*′/*M*, not the overall ring latency *a*′. The figure also shows how single-token operation approaches single-frame operation as *a*′ becomes large.



Figure 9

**Comparison**

We have discussed two basic scheduling approaches to medium access control: reser- vations and polling. Token-passing rings are essentially an extension of the polling concepts to ring-topology networks. The principal strength of these approaches is that they provide a relatively fine degree of control in accessing the medium. These ap- proaches can be viewed as an attempt to make time-division multiplexing more efficient by making idle slots available to other users.

Reservation systems are the most direct in obtaining the coordination in medium access that is inherent in a multiplexer. Reservation systems can be modified to im- plement the various scheduling techniques that have been developed to provide quality- of-service guarantees in conventional multiplexers. However, unlike centralized multiplexers, reservation systems must deal with the overheads inherent in multiple ac- cess communications, for example, time gaps between transmissions and reaction-time limitations. In addition, the decentralized nature of the system requires the reservation protocols to be robust with respect to errors and to be conducive to simple error-recovery procedures.

Polling systems and token-ring systems in their most basic form can be viewed as dynamic forms of time-division multiplexing where users transmit in round-robin fashion, but only when they have information to send. In polling systems the over- head is spread out in time in the form of walk times. The limitations on transmission time/token can lead to different variations. At one extreme, allowing unlimited trans- mission time/token minimizes delay but also makes it difficult to accommodate frames with stringent delay requirements. At the other extreme, a limit of one frame/token leads to a more efficient form of time-division multiplexing. Polling systems however can be modified so that the polling order changes dynamically. When we reach the ex- treme where the polling order is determined by the instantaneous states of the different stations, we obtain what amounts to a reservation system. From this viewpoint polling systems can be viewed as an important special case of reservation systems.

All the scheduling approaches were seen to be sensitive to the reaction time as measured by the propagation delay and the network latency normalized by the average frame transmission time. The reaction time of a scheme is an unavoidable limitation of scheduling approaches. Thus in reservation systems with long propagation delays, there is no way for the reservations to take effect until after a full propagation delay time. However, in some cases there is some flexibility in what constitutes the minimum reaction time. For example, in token-passing rings with single-frame operation the reaction time is a full ring latency, whereas in multitoken operation the reaction time is the latency of a single hop.

# References

[1] Group, Engineering. 2022. *sciencedirect.* Jan 1. https://www.sciencedirect.com/topics/engineering/medium-access-control#:~:text=Medium%20access%20control%20protocols%20are,both%20wired%20and%20wireless%20networks

[2] Garcia, Leon. 2006. *Communication Networks Fundamental.* Silicon Vally.

[3] Stallings, William. 2014. *Data and Computer Communications.* Pearson.

1. Also known as MAC [↑](#footnote-ref-1)
2. Channels will be discussed in Physical layer of Network [↑](#footnote-ref-2)
3. Random access methods are hard to discussed because they are related to conditions. [↑](#footnote-ref-3)