CS 425: Computer Networks **Jan-Apr 2020**

Lecture 10: Computer Networks – February 4 , 2020

Lecturer: Swaprava Nath Scribe(s): Chaman Agrawal, Sahil Dhull, Shubham Singh, Siddharth Nohria

Disclaimer: *These notes aggregate content from several texts and have not been subjected to the usual scrutiny deserved by formal publications. If you find errors, please bring to the notice of the Instructor.*

10.1 Multiplexing

Multiplexing is a method by which multiple analog or digital signals are combined into one signal over a shared medium. In computer networks, multiplexing is used to manage transmissions by multiple users over a single link/channel. Multiplexing methods can be fixed (static) or dynamic. Some of the major multiplexing methods are discussed below.

10.1.1 Time Division Multiplexing

In Time Division Multiplexing (TDM), each user periodically gets the entire bandwidth for a little burst of time. Sharing of the channel is accomplished by dividing available transmission time on a medium among users.

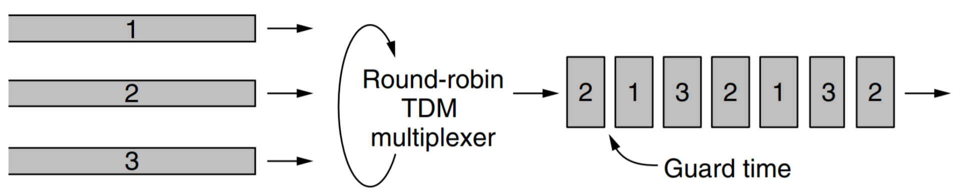

Figure 10.1: Time division Multiplexing

For this method to work, all users must be synchronized in time. Users are allocated the channel in a round-robin manner.

10.1.2 Frequency Division Multiplexing

Frequency Division Multiplexing (FDM), is a networking method of sharing the total available bandwidth of any communication channel by dividing them into many non-overlapping bands of frequency. Each user has an exclusive possession of some band in which to send their signal. These channels are then separated by the strips of unused bandwidth called guard bands to avoid signal interference.

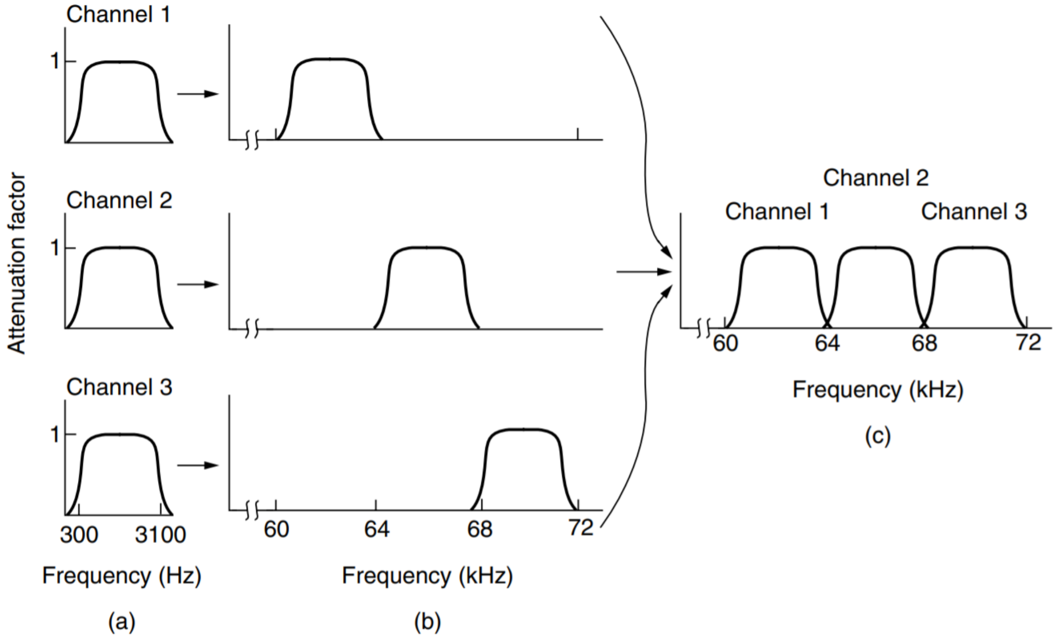

Figure 10.2: Frequency division multiplexing. (a) The original bandwidths. (b) The bandwidths raised in frequency. (c) The multiplexed channel.

All users can operate simultaneously in this model. The most natural example of frequency-division multiplexing is radio and television broadcasting, in which multiple radio signals at different frequencies pass through the air at the same time.

10.1.3 Observations and Uses of TDM and FDM

- TDM requires a synchronous clock, while FDM doesn't require this
- Roughly, there is no difference in average data rate

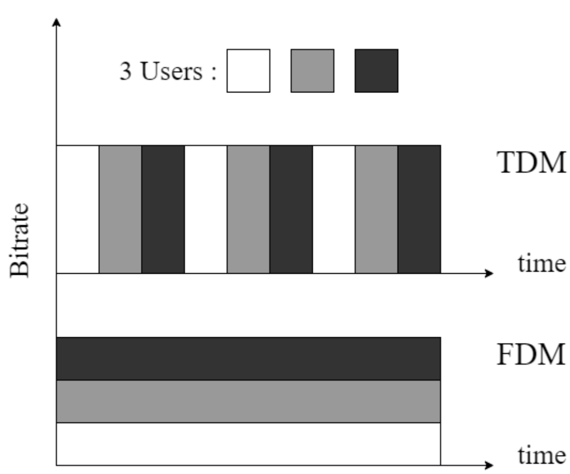

Figure 10.3: Data transfer over a time interval

- Both divide the resource statically.
- Fixed number of users.
- Continuous traffic (not much variation). e.g. voice/video transmission.
- 2G telephone system (GSM) allocates calls using TDM within FDM. Multiples users transmit using different frequencies in each time slot.

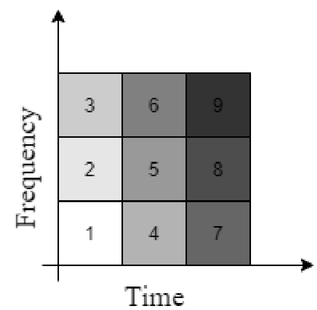

Figure 10.4: Simultaneous division of time and frequency in GSM

10.2 Statistical Multiplexing

So far, what we have seen was static allocation of resources. Typical sources that occur are bursty - there are periods when there is high network traffic, while there are other periods when there is almost no traffic at all. It is very unlikely that all sources have high traffic simultaneously. But in static allocation, each user is given equal access to the channel all the time. Thus, a system with static allocation can be very wasteful.

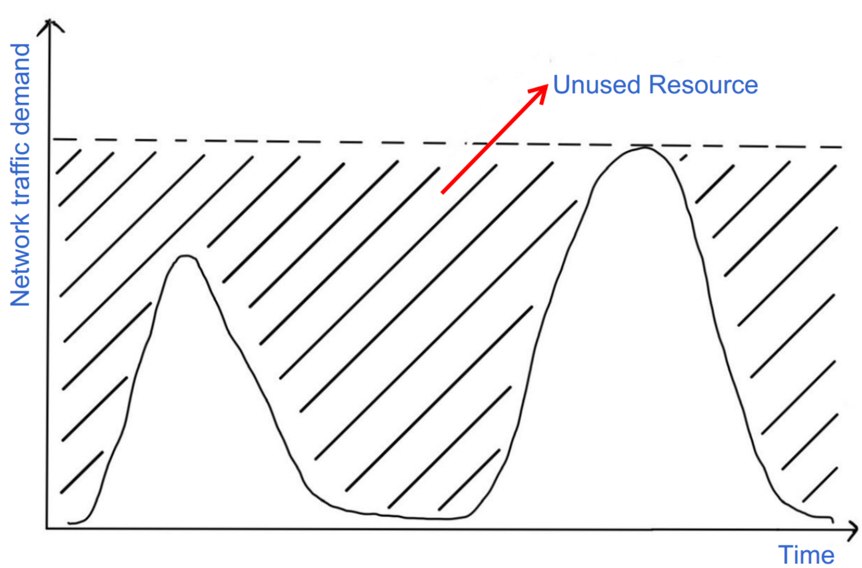

Figure 10.5: Problem with static multiplexing

With statistical multiplexing, the capacity allocated to each user varies with time, depending on its instantaneous data rate: the higher the rate, the larger the capacity allocated to it at that time.

The following are two protocols of statistical multiplexing:

- **Randomized access to the resource:** In this method, any user in the network can transmit data at any time. This method is ideal for the scenario with low load. When the traffic is high, there is very high probability of collisions in this method. Ethernet, 802.11 (WiFi) are some examples.
- **Contention Free access:** In this protocol, nodes order their demand (in a decentralized manner). Nodes communicate with each other and convey their demand, and then decide the order of transmission. This is done in cases where network traffic is high.

10.3 Randomized Multiple Access Protocol

Design Decision for this protocol:

1. How the link should be shared?
2. Who sends the message at a time?

Examples of this protocol include LAN, Wifi.
Assumptions: There is no centralized control or authority. The users have to decide in a decentralized manner.
The main features of this protocol are:

- There is no fixed time for sending data
- There is no fixed sequence of users sending data

Multiple Access Control, which is a sublayer, is a channel used for sharing between multiple mobile nodes whether it is Wired LAN or Wireless Communication. This sublayer ensures that packet is delivered from one node to another.

One of the main subcategory of Randomized MA Protocols is ALOHA. ALOHA was introduced in 1970s. The ALOHA protocol has different versions. We will discuss two versions of ALOHA here: pure and slotted. They differ with respect to whether time is continuous, as in the pure version, or divided into discrete slots into which all frames must fit.

10.3.1 Pure ALOHA

The basic idea of an ALOHA system is simple:

- Let the transmitters transmit whenever they have data to be sent and expect an acknowledgement from the receiver if the data is correctly received.
- If the received data is corrupted, transmitters re-transmits the data after a random timeout.

The first point implies that Pure ALOHA does not check whether the channel is busy before transmitting, because of which there can be collisions and hence corrupted data frame.

If a frame is corrupted, the sender just waits for a random amount of time and sends it again. This step is repeated until the frame is successfully transmitted. Note that the waiting time must be random or the same frames will collide over and over.

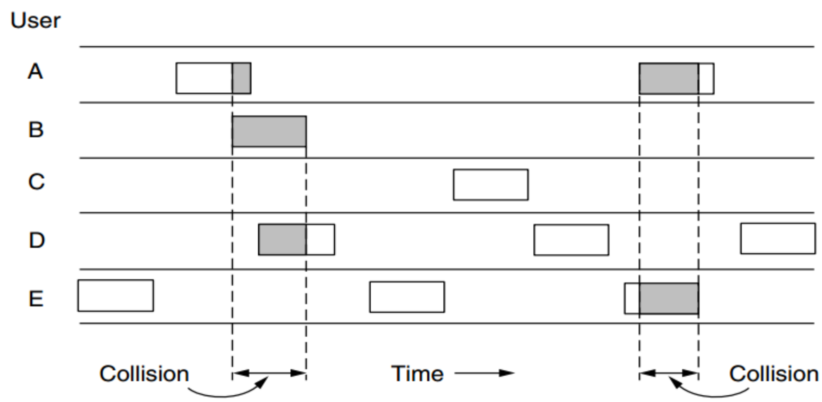

Figure 10.6: Frames transmitted by different users following pure ALOHA protocol.

The figure above illustrates an ALOHA system. Note that whenever two frames try to occupy the channel at the same time, there will be a collision (as shown in the figure). Both the colliding frames will be corrupted and will need to be re-transmitted.

From the figure, it can also be seen that as the number of users increases the probability of collision also increases. Due to this, the pure ALOHA protocol is not suited for high traffic network.

10.3.1.1 Throughput analysis of Pure ALOHA

For the analysis, we will make the following assumptions:

- All the frames are of same length
- The bit rate of the network is constant
- The generation of new frames follow a Poisson distribution with a mean of λT_f frames per unit frame time T_f, which is the time required to transmit a frame.

Aside: Why is Poisson arrival a good assumption?

- Consider n discrete equal time intervals in time t.
- We know that the probability of k arrivals follows Binomial distribution.
  P(k arrivals) $\sim$ Binomial(n,p), where p is the probability of arrival in any interval.
- Mean of Bin(n,p) = np = λt $\implies$ p = $\frac{\lambda t}{n}$, where λ is the frame generation rate
- P(arrivals) $\sim$ Binomial(n,$\frac{\lambda t}{n}$)

- And as n tends to ∞, this is same as Poisson(λt)

$$P(\text{k arrivals}) = \frac{e^{-\lambda t}(\lambda t)^k}{k!}$$

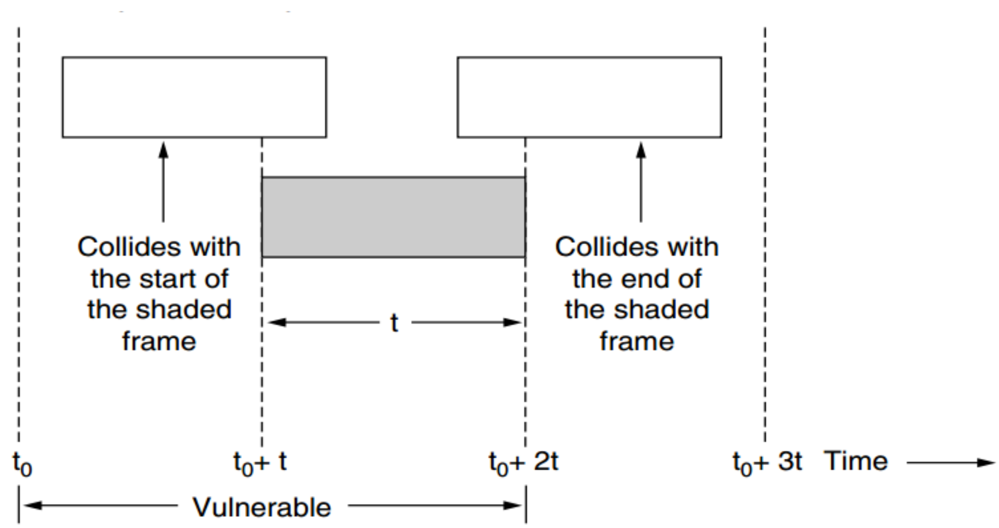

Figure 10.7: Vulnerable period for frames in Pure ALOHA

Let us define a new term called Vulnerable period, which is basically the time interval (w.r.t to the current packet) in which another packet should not be transmitted. Let's suppose our packet (in discussion here) is generated at some time (t_0+t). Then if some packet was started in the channel between t_0 and $t_0 + t$, its end will collide with our packet, and if some packet will be started between time t_0+t and t_0+2t, its start will collide with our packet. So vulnerable time will be $2 \cdot t$ or $2 \cdot T_f$. In this time interval, there should be no other transmission, i.e. 0 number of transmissions in time $2 \cdot T_f$. Hence,

$$\begin{aligned} P(\text{successful transmission}) &= P_{succ} \\ &= P(0 \text{ arrivals}) \\ &= e^{-2\lambda T_f} \cdot \frac{(2\lambda T_f)^0}{0!} \\ &= e^{-2\lambda T_f} \end{aligned} \tag{10.1}$$

Let G be the load or average number of transmission-attempts per T_f time and S be the throughput of the transmission then,

$$G = \lambda \cdot T_f \tag{10.2}$$

$$\begin{aligned} S &= \mathbb{E}(\text{Successful frame transmitted in } T_f \text{ time}) \\ &= (\text{Average transmissions in } T_f \text{ time}) \cdot P_{succ} \\ &= \lambda \cdot T_f \cdot e^{-2\lambda T_f} \\ &= G \cdot e^{-2G} \end{aligned} \tag{10.3}$$

Consider what needs to happen for a frame to be transmitted successfully. Let "t" refer to the time at which it is intended to send a frame. It is preferable to use the channel for one frame-time beginning at t, and all other stations to refrain from transmitting during this time.

10.3.2 Slotted ALOHA

A better version of ALOHA is Slotted ALOHA, wherein the time is divided into discrete intervals called slots, each slot of width 1 time frame (T_f). Each user is required to wait for the beginning of next slot to transmit the data i.e it can't transmit the data as soon as it is received. The same is shown in the following diagram.

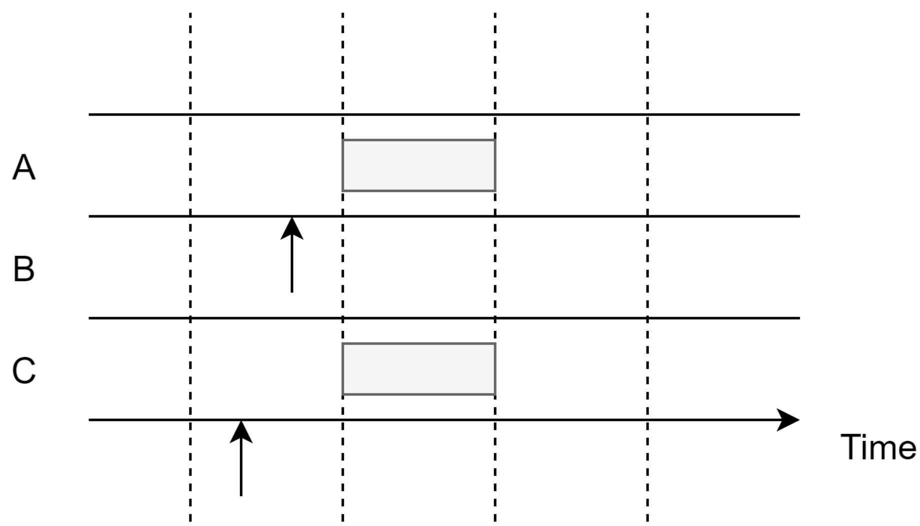

Figure 10.8: Frames transmitted by different users following Slotted ALOHA protocol.

This protocol reduces the collision probability since the vulnerable period is halved (as can be seen in the following diagram).

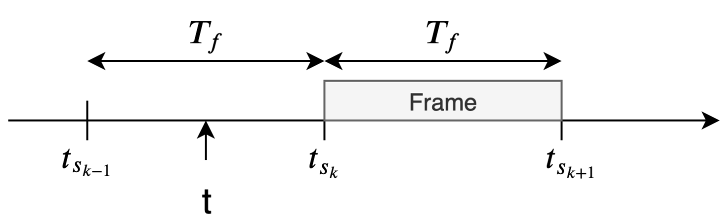

Figure 10.9: Vulnerable period for frames in Slotted ALOHA

The vulnerable period now becomes T_f rather than $2 \cdot T_f$. This is because the slot in which user gets the signal to transmit can't be used for transmission, rather next slot will be used. So the only thing that needs to be ensured is that no other request comes in this time slot. Hence the Probability of success becomes:

$$\begin{aligned} P_{succ} &= P(\text{no arrival in time interval } T_f) \\ &= e^{-\lambda T_f} \cdot \frac{(\lambda T_f)^0}{0!} \\ &= e^{-\lambda T_f} \end{aligned} \tag{10.4}$$

And hence the throughput becomes:

$$S = G \cdot e^{-\lambda T_f} = G \cdot e^{-G} \tag{10.5}$$

As you can see in the following figure, Slotted ALOHA peaks at G=1, with a throughput of 1/e or S=0.36 while Pure ALOHA peaks at G=0.5, with a throughput of 1/2e. Also note that Slotted ALOHA outperforms Pure ALOHA for all values of G.

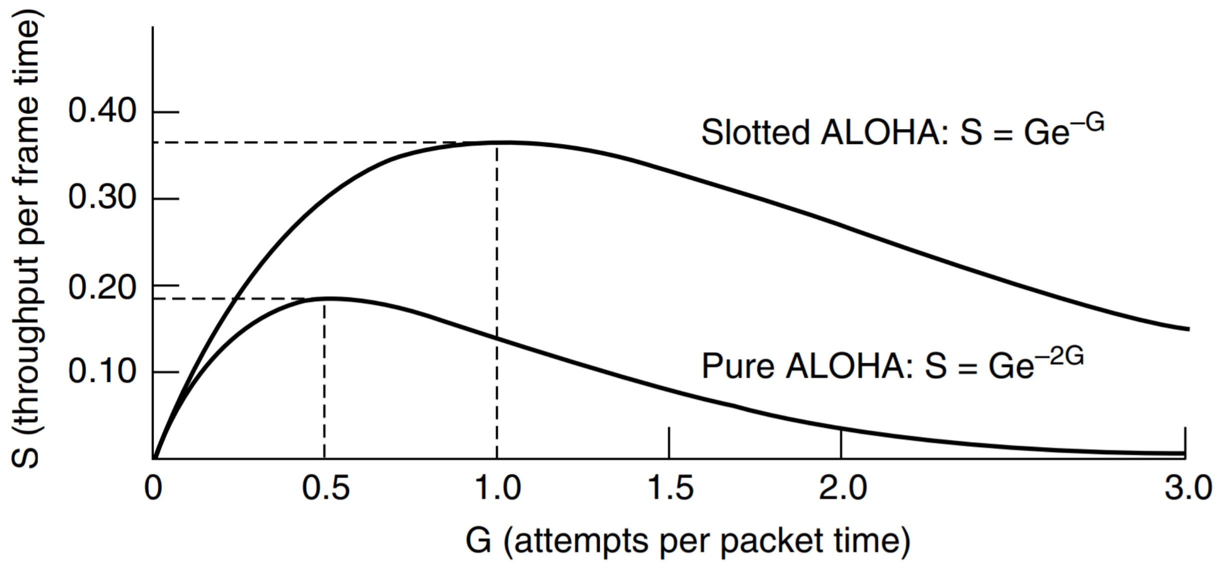

Figure 10.10: Throughput versus offered traffic for ALOHA protocols

The graph peaks at a particular value of G because increasing the value of G decreases the probability of success exponentially beyond certain threshold.

References

[1] Tannenbaum, A. S., and D. J. Wetherall. ”Computer Networks,(5-th edition).” (2010).

[2] https://en.wikipedia.org/wiki/Multiplexing

[3] https://en.wikipedia.org/wiki/ALOHAnet