① MAC ①

The protocols used to determine who goes next on a multiaccess channel belong to MAC sublayer. (bottom of data link layer) This sublayer is specially important for LAN as users use a multiaccess channel in broadcast fashion as the basis for communication. WAN normally use point-to-point network.

Allocating single broadcast channel among multiple competing users — in LAN/MAN —

1. Static channel allocation

Traditional way of allocating a single channel among multiple competing users is Frequency Division Multiplexing (FDM). If there are N users, the bandwidth is divided into N equal-sized portions, each user being assigned one portion. Since each user has a private frequency band, there is no interference between users. When number of users is small and each has a heavy load of traffic, FDM is an efficient allocation mechanism. However, when number of user is large and users have varying traffic, FDM is not efficient. This is because, when some users are quiescent, their bandwidth is simply wasted. They are not using it and no one else is also allocated to use it. At the same time, some amount of bandwidth is not in use whereas some users are denied for lack of bandwidth.

Let T is the mean delay, C is the channel capacity, λ frames/sec is the arrival rate; let each frame having a length drawn from an exponential distribution with mean $\frac{1}{\mu}$ bits/frame.

with these parameter, service rate is μC frames/sec. (2)

Hence from queuing theory,

(2)
$$T = \frac{1}{\mu C - \lambda}$$

Ex: $C = 100$ Mbps, $\frac{1}{\mu} = 10{,}000$ bits, $\lambda = 5000$ frames/sec.

$\Rightarrow T = 200$ μsec

If we ignore the queuing delay and asked how long it takes to send a frame of 10000 bits on a 100-Mbps network, the answer is 100 μsec. this holds only if there is no contention for the channel.

If we devide single channel of capacity C into N independent sub channels each with capacity C/N bps, the delay in each sub channel is
$$T_{FDM} = \frac{1}{\mu(C/N) - (\lambda/N)} = NT.$$

of the frame arrival on each sub channel is λ/N.

Thus, FDM is N times worse than if all frames where queued in a single long queue. In other words, if we replace the 100-Mbps network with 10 network of 10 Mbps each and statically allocate each uses to one of them, the mean delay would jump from 200 μs to 2000 μs (2 Msec).

Dynamic channel allocation in LAN/MAN –

channel will be assigned to user based on their demand. Key assumptions on all dynamic channel allocation –

1. Station model – there are N independent users /stations that generates frames for transmission. once a frame is generated, the station is blocked and does nothing until the frame has been successfully transmitted.

2. Single channel assumption — A single channel is available for all communication. All stations will ~~t~~ transmit on it and receive from it.

3. Collision Assumption — If two or more frames are transmitted simultaneously, they collide. All stations can detect ^either instantly or through ack^ collision. A collided frame must be retransmitted. There are no ~~e~~ errors other than collision.

4a — Continuous time — frame transmission begin at any time. There is no master clock.

4b — Slotted time — Time is divided into discrete slots. frame transmission always begin at the start of a slot. Thus, a slot may be idle, successfull or collision. clocking mechanism is required.

5a Carrier sense — station can sense if the channel is in use before trying to ~~use~~ use it. If it is busy, no station will attempt to use it until it becomes idle.

5b — No Carrier sense — stations can not sense the channel. they go ahead and transmit. only later they can determine whether the transmission was successful or not.

Multiple access protocol —

ALOHA — pure ALOHA and slotted ALOHA. They differ with respect to whether time is divided into discrete slots into which all frames must fit. pure ALOHA does not require global synchronization, slotted ALOHA does.

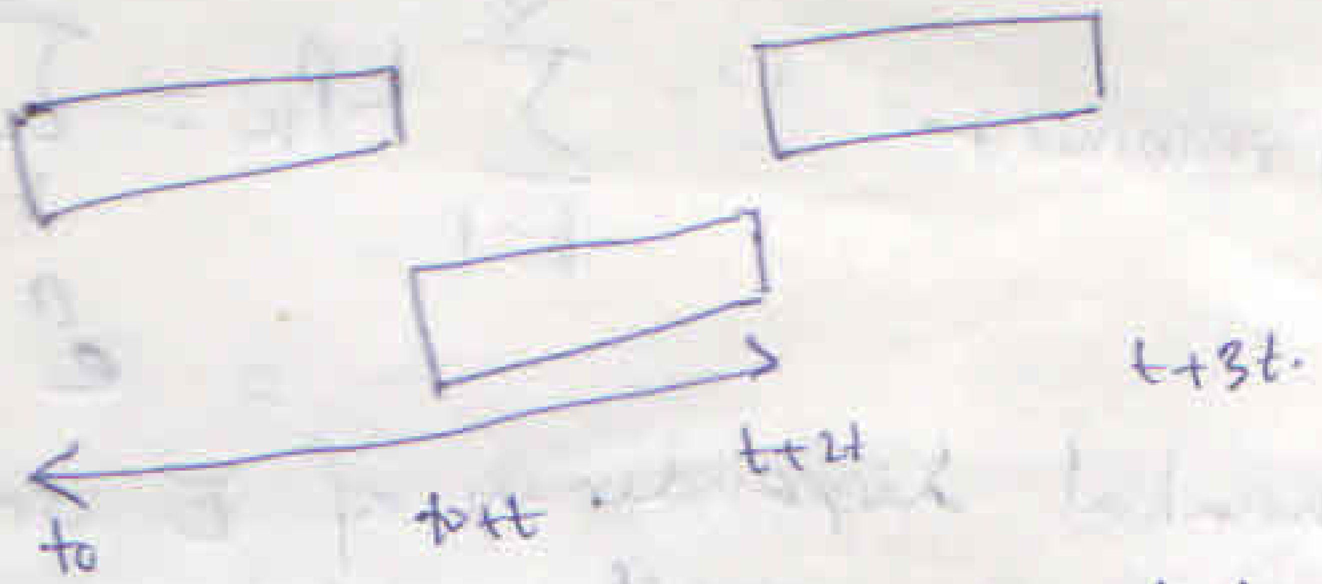
pure ALOHA — stations for transmit whenever they have data to be sent. there may be collision and collided frames will be ~~sorted~~ damaged. Due to feedback property of broadcast, sender can always find out whether the frame was Colided or not. If Colided, the sender wait for a random amount of time and send again. The waiting time is random to avoid the same frame collide over and over. If the first bit of a new frame overlaps with the last bit of a frame almost finished, both frames will be destroyed and requires retransmission. This is because checksum can not distinguish between total loss and near miss. Bad is bad.

What is the efficiency of pure ALOHA? Let frame time denote the time required to send a frame. Here it is assumed that all frames are of equal size. In fact throughput of ALOHA is maximized when all frames are same size rather than variable length frames.

Let new frames are coming according to poison distribution with mean N frames per frame time. If N>1, naturally user community is generating frames at a higher rate than the channel can handle, and nearly every frame will suffer a Collision. For reasonable throughput 0<N<1. In addition to the new frames generated, stations also generates retransmission of frames that were previously Collided. Let the probability of k transmission attempts per frame time (old and new) is also poisson with mean G frames per frame time. Clearly, G≥N. At low load, G≈N as N≈0. At high load G>N. Under all cases, the throughput S is the offered load G times the probability P_0 of a transmission is successfull. That is, $S = GP_0$. A frame will not suffer a collision if no other frames are sent within a frame time of its start. Let t be the time required to send a frame and assume that at time to a frame transmission started. If first bit of new frame overlaps with the last bit of the previous frame, both will collide.

thus the frame will not suffer collision if no other frames are send during the period t_0 to t_0+2t. This is called the vulnerable period. (5)

(5)



Now, the probability of k frames generated during one frame time is

$$pr(k) = \frac{G^k e^{-G}}{\lfloor k}.$$ Hence $Pr(0) = e^{-G}$.

In an interval of 2 frames times, the mean number of frames generated is 2G. Thus probability that no other traffic generated during the whole vulnerable period is $P_0 = e^{-2G}$.

Thus throughput $S = G.e^{-2G}$. S is maximum when $G=\frac{1}{2}$. and $S = \frac{1}{2e} = 0.184$.

Hence the maximum channel utilization of ALOHA is 18% only.

<u>Slotted ALOHA</u>: Each user will send frames only at the begining of a slot. This synchronization can be achieved by emiting a pip at the start of each slot by a special station. Now the vulnerable time is halfed. Hence $P_0 = e^{-G}$. $S = Ge^{-G}$.

S is maximum when $G=1$. and $S = \frac{1}{e} = .368$.

Thus Slotted ALOHA throughput is 37%. Here 37% slots are empty, 37% slots are successful and 26% collisions. Higher G can reduce empty slot but exponentially increase collisions.

The probability of no collision $= e^{-G}$. Thus, probability of collision $= (1-e^{-G})$. Probability that a transmission requires K attempt (k-1 collision folled by one success) is ⑥

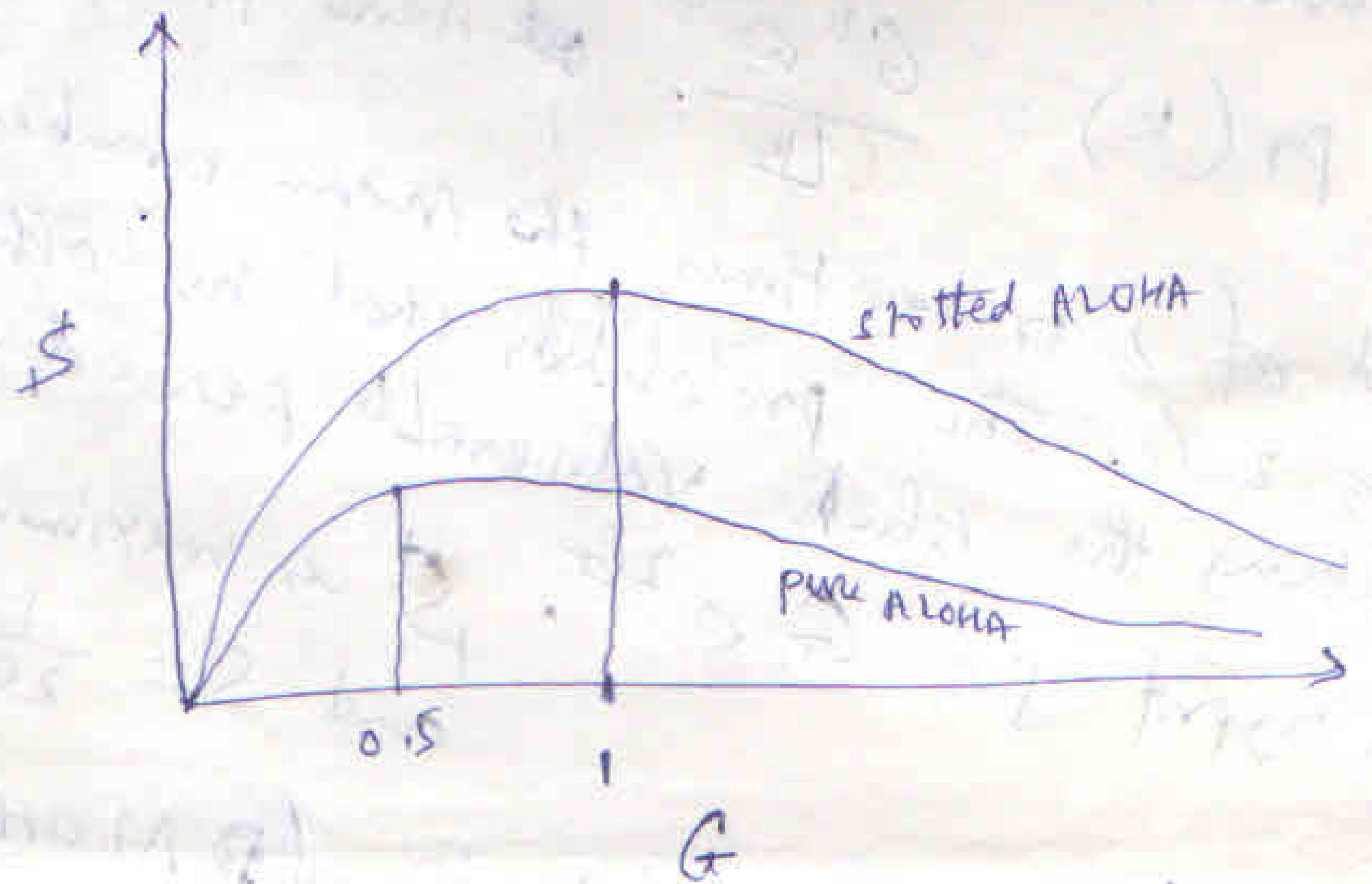
⑥

$$P_k = e^{-G}(1-e^{-G})^{k-1}$$

Expected number of transmission $E = \sum_{k=1}^{\infty} kP_k = \sum_{k=1}^{\infty} k e^{-G}(1-e^{-G})^{k-1}$

$= e^{G}$

As a result of exponential dependency of E on G, small increase in load, drastically reduce performance.



Carrier sense multiple access protocol (CSMA)

With slotted ALOHA channel utilization can be at most 1/e. This is hardly surprising, since stations are transmitting without paying any attension to what the other stations are doing. then, there are bound to be many collision. In LAN it is possible for stations to detect what the other stations are doing, and adapt their behavior according to avoid collision. By this we can achieve much better channel utilization than 1/e.

1-persistent CSMA: when a station has data to send it first sense the channel, to see if anyone else is transmitting at that moment. If it is found to be busy, the station continuously sense the channel until it is found to be idel. Once found idle it transmitts. If collision occurs the station wait for a random amount of time and repeat the process.

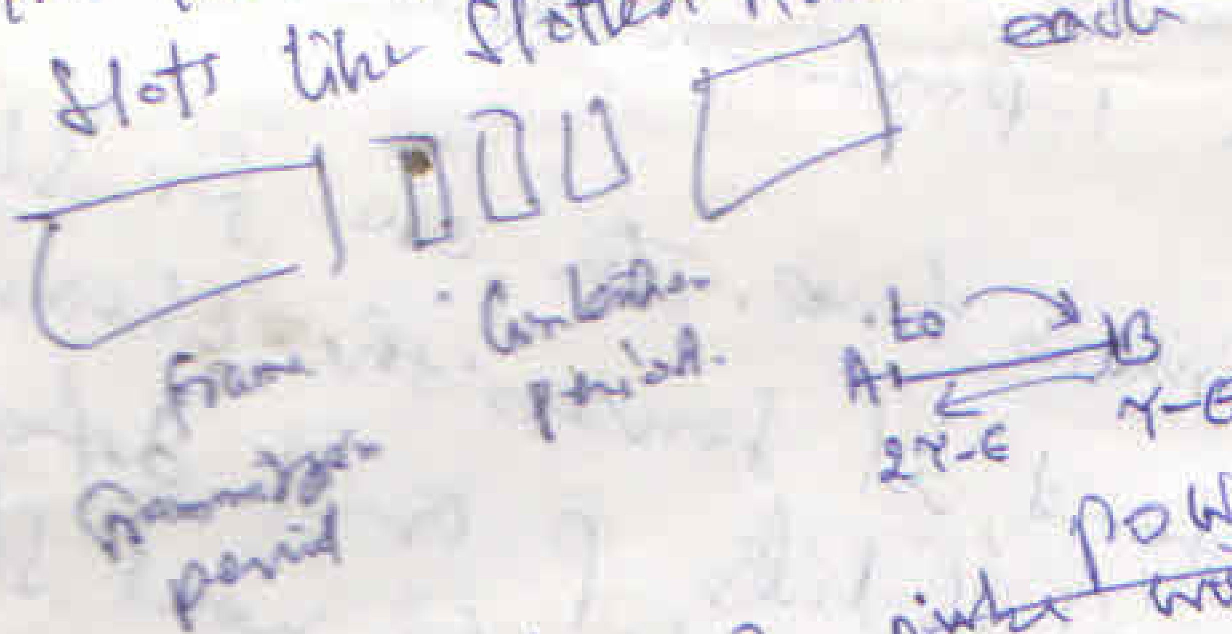
[illegible]

CSMA with CD

Persistent and non-persistent CSMA protocols are clearly an improvement over ALOHA because they ensure that no station begins transmitting when the channel is busy.

Another improvement is for the station to abort their transmission as soon as a collision is detected. Rather than finish transmitting their frames (garbled anyway). LANs uses CSMA/CD in their MAC layer. After a station detects a collision, it aborts and waits for a random period of time and then tries again.. If two stations sense the channel and find idle and begin transmitting a collision will be detected immediately. It has a contention period and transmission period.

If both station starts transmitting after detecting the channel idle, how long will it take them to realize that there has been a collision? It is 2τ if τ is the time ~~to~~ for a signal to propagate between the two farthest stations. Thus contention interval is chosen as slots like slotted ALOHA with width 2τ. It is just 1 bit that each slot contains.



Min Size 2τ @ datarate
Max Speed 1 bit per 2τ second.

* How to detect collision? ~~pulse width~~ power or pulse width of receive signal comparing to the transmitted signal. ~~It is a analog process~~.
* Half-duplex on the receiving logic is in use for collision detection during every transmission.
* Special encoding is used as two 0-volt signals may be be impossible to detect...

In CSMA/CD, find min. frame length for 1 Mbps data rate and max network span of 10 km with no repeaters. Prop. delay (One way) = 4.5 ns per meter.

~~Any~~ Prop. delay for 10 km = $4.5 \times 10^{-9} \times 10 \times 10^3$ sec $= 4.5 \cdot 10^{-5}$ sec.

$2\tau = 9 \cdot 10^{-5}$ sec.

1 sec ... 10^6 bits

$9 \cdot 10^{-5}$ sec .. $9 \cdot 10^{-5} \cdot 10^6 = 90$ bits $= 11.25$ bytes.

Performance of CSMA/CD

* Collisions can be detected and resolved if ~~2tp~~ time is slotted in $2t_{prop}$ slots during contention periods.

* Assume n bussy stations, each may transmit with probability p in each contention time slot.

* Once the contention period is over (that is, a station successfully acquires the channel), it ~~k~~ takes X seconds for a frame to transmit.

* It takes t_{prop} before the next contention period starts.

How long doe it take to resolve contention?

Ans: Contention is resolved if exactly 1 station transmit in a slot $\Rightarrow P_{succ} = np(1-p)^{n-1}$. Now P_{succ} is maximum when $p = 1/n$. So $P_{succ}^{max} = \left(1-\frac{1}{n}\right)^{n-1} \rightarrow \frac{1}{e}$ as $n \to \infty$.

expected number of slots to resolve contention

$$= \sum_{i=0}^{\infty} i \,\frac{1}{e}\left(1-\frac{1}{e}\right)^{i-1} = \frac{1}{e} + 2\frac{1}{e}\left(1-\frac{1}{e}\right) + 3\frac{1}{e}\left(1-\frac{1}{e}\right)^2 + \cdots$$

$$= \frac{1}{e}\frac{1}{\left[1-\left(1-\frac{1}{e}\right)\right]^2} = \frac{1}{e}\cdot e^2 = e.$$

Average contention period $\Rightarrow$ e slots of length $2t_{prop}$

$\Rightarrow 2t_{prop} \cdot e$. seconds.

What is the maximum throughput?

Ans: At maximum throughput system alternates between contention period and frame transmission.

$$\Rightarrow \text{Throughput}_{max} = \frac{X}{X + t_{prop} + 2t_{prop}e} =$$

10

$$= \frac{1}{1+(1+2e)a} \quad \text{where } a = \frac{t_{prop}}{X}$$

Now X = frame transmission time = L/R second where L is the length of the frame and R is the data rate.

$t_{prop} = d/v$, where v meters/sec is the speed of light in medium and d meters is the distance between two farthest stations or diameter of the system.

$1+2e = 6.44$ as $e = 2.72$

So max throughput $= \dfrac{1}{1+(1+2e)\,Rd/vL}$.

(11) collision-free protocol.

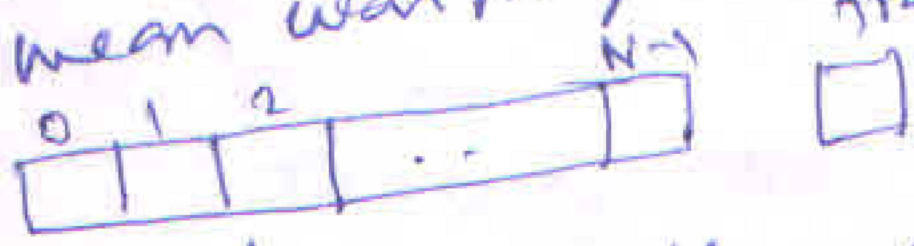
Collisions do not occur with CSMA/CD once a station unambiguously captured the channel but collisions can still occur during Contention period. Collision free protocols are those where collisions will never occur even during the contention period. These Protocols are not currently used in major system because of longer delay.

A Bit-map protocol: Assume that there are exactly N stations each with a unique address from 0 to N-1. The Contention period consists of exactly N slots each of width 1 bit. Station j announces that it has a frame to send by inserting a 1 bit into slot j. After all N slots have passed by, each station has Complete knowledge of which stations wish to Communicate. At that point, they begin transmitting in numerical order.

Since everyone agrees who goes next, no collision will occur. If a station becomes ready just after its bit slot has passed by, it must remain silent until next bit map has come around again. We measure time in units of the contention bit slot. Let each frame consists of d time units. At low load bit map will repeat over and over for lack of data frames.

Low-numbered stations at low load: When station 0 is ready the current slot will be somewhere middle of the bit map. So on average the station has to wait N/2 slots for the current scan and another N slots for the next scan. So Total 1.5N slots is average waiting for low numbered station.

For high numbered station (station N-1) the average waiting is only N/2 slots. Thus the mean for all station is N slots. So efficiency is d/(d+N) since only one frame is transmitted at low load and N slots is mean waiting.



low load → when a station is ready, no other station is ready.

(12)

At high load: every station is read at all the time. So N frames will be transmitted after N-bit contention period. So overhead is 1 bit per frame. Thus efficiency = $(Nd/(Nd+N)) = d/(d+1)$. The mean delay for a frame is equal to the sum of the time it queues inside its station plus an additional $N(d+1)/2$ once it gets to the head of its internal queue.

$\frac{N(d+1)}{2} = \frac{N}{2}.d + \frac{N}{2}.1$ = on average one station has to wait for $1/2$ of the stations data transmission + $N/2$ slots — at the contention window.

Binary Countdown: In basic bit map protocol, the overhead is 1 bit per station. So it does not scale well for a network of thousands of stations. In binary Countdown, the addresses of the stations are represended as binary string. If N stations are there, $n = \log_2 N$ bits is required for these addresses.

A station wishes to transmit, broadcast its address with the high order bit. The bits in each address position from different stations are BOOLEAN ORed together. As soon as a station sees that a high-order bit position that is 0 in its address has been overwritten with a 1, it gives up. For example, if stations 0010, 0100, 1001, 1010 are all trying to get the channel, in the first bit time the stations transmit 0, 0, 1, and 1 respectively. These are ORed to form 1. Stations 0010 and 0100 see the 1 and give up for the current round. Station 1001 and 1010 continue. The next bit is 0 for both station so they both continue. For next bit 1, so station 1001 gives up. The winner is 1010. After winning the station 1001 transmit a frame and then next bidding cycle starts. The channel efficiency is $(d/(d+\log_2 N))$

Mok and Ward (1979) variation of binary Countdown: In basic binary Countdown high order stations has higher priority. So low order stations may fall into starvation. Mok and Ward suggested virtual station number and provides dynamic priority.

It gives higher priority to stations that have been silent for long. For example, if C, H, D, A, G, B, E, F have priorities 7, 6, 5, 4, 3, 2, 1, and 0 respectively then a successful transmission by D puts it at the end, giving a priority order C, H, A, G, B, E, F, and D. Thus, C remains virtual station 7, but A moves up from 4 to 5, and D drops from 5 to 0. So station D will now have least priority.

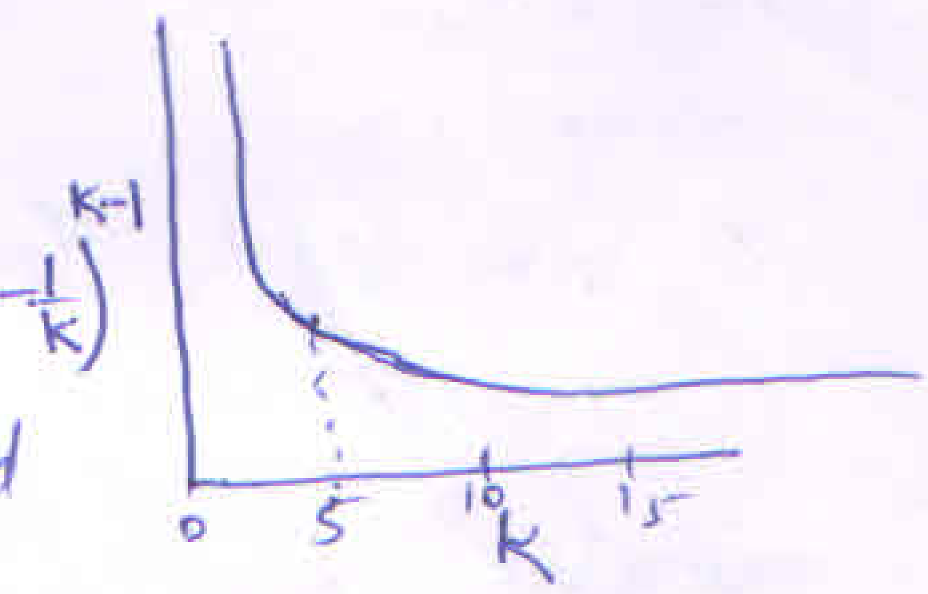
Limited Contention Protocols.

At low load, Contention based protocol (slotted ALOHA) is preferable due to its low delay. But as load increases Contention becomes less attractive because of high collisions. Just the reverse is true for Collision-free protocols. At low load, they have high delay, but as load increases, channel efficiency improves rather than gets worse as it does for Contention protocols. Limited Contention Combines the best properties of contention and Collision free protocol.

The Contention protocols we have studied so far is symmetric, i.e., each station attempts to acquire the channel with some probability p with all the stations using the same p. Interestingly, the overall system performance can sometimes be improved by using asymmetric protocol which assigns different probabilities to different stations.

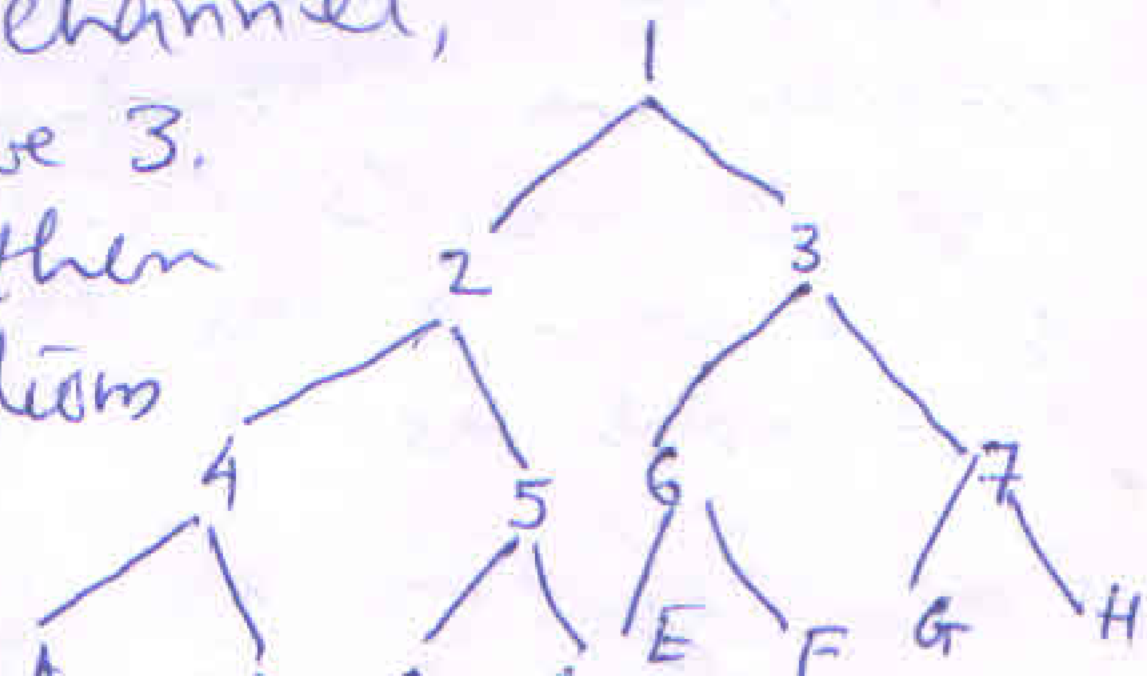
Before looking at the ~~asym~~ asymmetric protocol, we first review the performance of symmetric case.

let k stations are contending for a channel where each station has probability p of transmitting during each slot. So the probability that some station will successfull acquire the channel during a given slot is $^kC_1\, p(1-p)^{k-1} = kp(1-p)^{k-1}$.
this is optimal when $k = 1/p$ and the optimal value is $(1-\frac{1}{k})^{k-1}$. The asymptotic value of this is $1/e$. When k is close to 5, it reaches close to $1/e$. But performance is very good if k is less than 5.



the probability of successfuly acquiring a channel can be increased by decreasing the amount of competition. The limited-contention protocol do precisely that. It divides the stations into groups. Only the members of group 0 are permitted to compete for slot 0. If one of them succeeds, it acquires the channel and transmits. If the slot remain empty or a collision occurs, the members of group 1 will contend for slot 1 and so on. By making appropriate division of members into groups, the amount of contention could be control. In one extreame, if each group has one member (bit map / binary countdown) or a single group contains all the members (slotted ALOHA). We need a way to assign stations to slots dynamically with many stations per slot when low load, and few stations per slot at high load.

The ~~the~~ adaptive Tree walk protocol:- In this protocol, stations are organized on the leaves of a binary tree. In slot 0 all stations are permitted to try to acquire the channel. If one of them successfully acquires, fine. If there is a collision, then during slot 1 only those stations falling under node 2 in the tree may compete. If one of them acquires the channel, slot 2 will be reserve for stations under node 3.

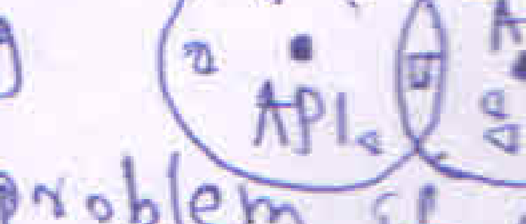


If collision occurs at slot 1, then slot 2 will be reserve for stations under node 4. In general, if collision occurs the search will go in a depth first maner. If success, it will go in a breadth first manner.

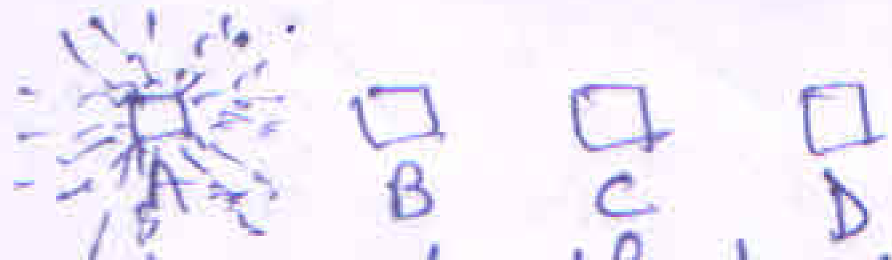
~~Left~~ If q ready stations are uniformly distributed, then the expected number of ready stations under a node at level i is $2^{-i}q$. Intuively we would expect to begin the search at the level where mean number of contending stations per slot is 1. So we would start at level $i = \log_2 q$.

Wireless Local Area Network (LAN):

It is a wireless network that links multiple devices using wireless communication to form a local area network within a limited area such as a home, campus or office building. This gives users the ability to move around within the area and remain connected to the network. Most wireless LANs are based on IEEE 802.11 standards and popularly known as Wi-Fi Network. Typically, it consists of a number of stations (such as laptop, mobile phone) and a number of fixed Access Points (APs). Stations access the network through these APs. An AP serves all the stations which falls within its transmission range. However, when a receiver is within the range of two active transmitters, the resulting



signal will be garbled. We will now see the problem if CSMA is used as a MAC mechanism for wireless LAN. To understand the nature of the problem, consider the following figure where 4 wireless devices are communicating. For this purpose, it does not matter which are access points and which are stations. We assume that each device has a fixed transmission range. The radio



range is such that A and B are within each other's range and can potentially interfere with one another. C can interfere with both B and D, but not with A. Now consider what happens when A is transmitting to B. If C senses the medium, it will not hear A because A is out of range of C and thus falsely conclude that it can transmit to B. If C does start transmitting, it will interfere at B. The problem of a station not being able to detect a potential competitor for the medium because the competitor is too far away is called the hidden terminal problem. No consider the reverse situation as shown below, where B is transmitting to A. Now if C

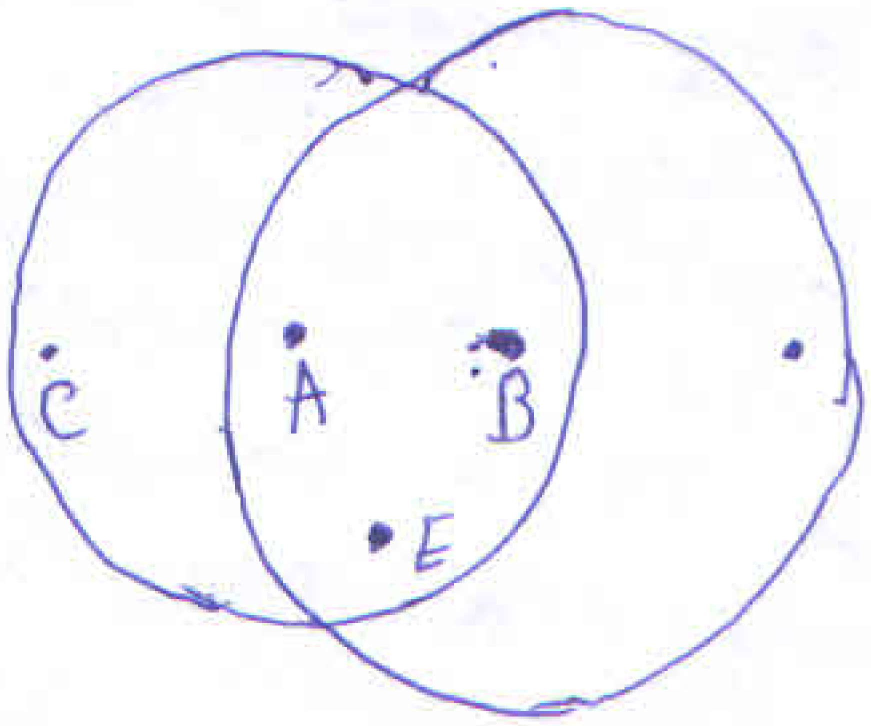
senses the medium, it will hear an ongoing transmission and falsely conclude that it may not send to D, when in fact such a transmission would cause bad reception only in the zone between B and C, where neither of the intended receivers is located. This is called exposed terminal problem.

From the above discussion it is clear that CSMA is not appropriate for wireless LAN.

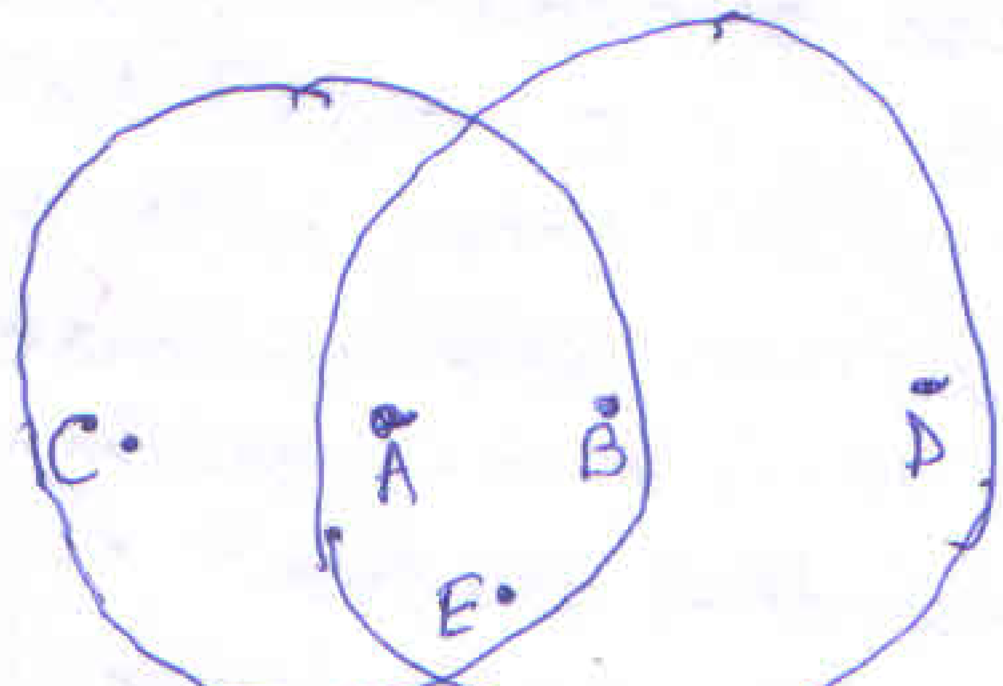
An early protocol design for wireless LAN is MACA (Multiple Access with Collision Avoidance):

MACA

The basic idea behind MACA is for the sender to stimulate the receiver into outputting a short frame & Request to Send (RTS), and clear to Send (CTS), so that stations nearby can detect this transmission and avoid transmitting for the duration of the upcoming data frame. The protocol is illustrated as follows.



A sending an RTS to B
(a)



B responding with a CTS to A
(b)

Consider that A sends a frame to B. A starts by sending an RTS frame to B. This short frame contains the length of the data frame that will eventually follow. Then B responds with a CTS frame. The CTS frame contains the data frame length (copied from the RTS frame). Upon receipt of the CTS frame, A begins transmission.

We now see how the stations overhearing RTS/CTS will react. Any station hearing RTS is clearly close to A and must remain silent long enough for the CTS to be transmitted back to A without conflict. Any station hearing the CTS is clearly close to B and must remain silent during the upcoming data transmission time, whose length it can tell by examining the CTS frame.

From figure (a) above, C is within range of A but not within range of B. So, it hears RTS from A but not the CTS from B. As long as it does not interfere with the CTS, it is free to transmit while the data frame is being sent.

In Contrast, D is within the range of B but not A. It does not hear the RTS but does hear the CTS. Hearing the CTS, it understand that it is close to a station who is about to receive a frame, so it defers sending until that frame is expected to be finished. Station E hears both RTS and CTS and like D must be silent until the data frame is complete.

Despite these precautions, collisions can still occur. For example, B and C could both send RTS frames to A at the same time. These will collide and be lost. In the event of a collision, an unsuccessful station (the one that does not hear a CTS within the expected time interval) waits for a random amount of time and tries again later. These random wait times is determined as per the binary exponential backoff algorithm.

Binary exponential backoff algorithm: After first collision, each station waits either 0 or 1 slot times before trying again. If two stations collide and each one picks the same random number, they will collide again (probability is 1/2). After the second collision, each one picks either 0, 1, 2 or 3 at random and waits that number of slots. If a third collision occurs (probability is 1/4), then the next time the number of slots to wait is chosen at random from the interval 0 to 2^3-1. In general after i-th collision, a random number between 0 and 2^i-1 is chosen and that number of slots is skipped. However, after 10 collisions, the randomization interval is frozen at a maximum of 1023 slots.

MACA is fine tuned to improve its performance and renamed it as MACAW (MACA for wireless).

MACAW

The following modification has been done on MACA to improve its performance:

1. An ACK frame is ~~adde~~ introduced. So, it sends an ACK after each successful data frame transmission.

2. CSMA is added to keep multiple stations from transmitting RTS frames simultaneously to the same destination. Note that CSMA is added only for RTS frame transmission.

3. Binary backoff algorithm is ~~run~~ decided to run saparately for each data frame (source-destination pair) rather than for each station.

4. Added a mechanism for stations to exchange information about Congestion ~~so that loss~~ ~~(...)~~.