Computer Networks (CS425)

Instructor: Dr. Dheeraj Sanghi

Prev| Next| Index

CSMA with Collision Avoidance

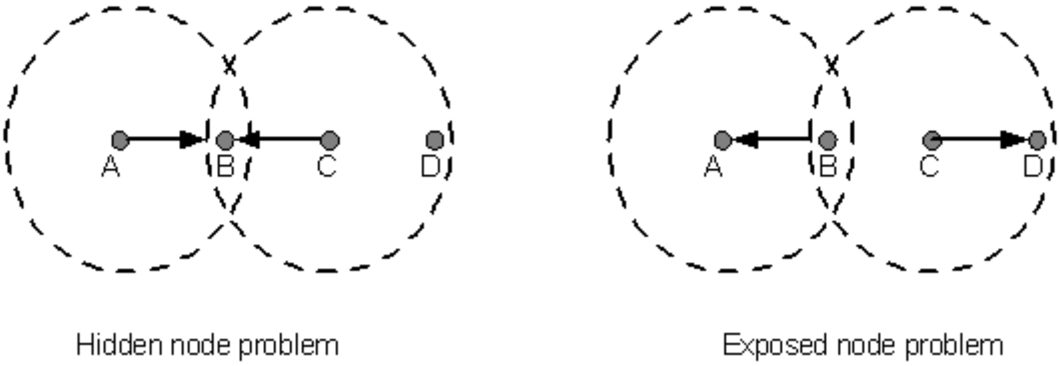
We have observed that CSMA/CD would break down in wireless networks because of hidden node and exposed nodes problems. We will have a quick recap of these two problems through examples.

Hidden Node Problem

In the case of wireless network it is possible that A is sending a message to B, but C is out of its range and hence while "listening" on the network it will find the network to be free and might try to send packets to B at the same time as A. So, there will be a collision at B. The problem can be looked upon as if A and C are hidden from each other. Hence it is called the "hidden node problem".

Exposed Node Problem

If C is transmitting a message to D and B wants to transmit a message to A, B will find the network to be busy as B hears C trnasmitting. Even if B would have transmitted to A, it would not have been a problem at A or D. CSMA/CD would not allow it to transmit message to A, while the two transmissions could have gone in parallel.



Addressing hidden node problem (CSMA/CA)

Consider the figure above.Suppose A wants to send a packet to B. Then it will first send a small packet to B called **"Request to Send" (RTS)**. In response, B sends a small packet to A called **"Clear to Send" (CTS)**. Only after A receives a CTS, it transmits the actual data. Now, any of the nodes which can hear either CTS or RTS assume the network to be busy. Hence even if some other node which is out of range of both A and B sends an RTS to C (which can hear at least one of the RTS or CTS between A and B), C would not send a CTS to it and hence the communication would not be established between C and D.

One issue that needs to be addressed is how long the rest of the nodes should wait before they can transmit data over the network. The answer is that the RTS and CTS would carry some information about the size of the data that B intends to transfer. So, they can calculate time that would be required for the transmission to be over and assume the network to be free after that.Another interesting issue is what a node should do if it hears RTS but not a corresponding CTS. One possibility is that it assumes the recipient node has not responded and hence no transmission is going on, but there is a catch in this. It is possible that the node hearing RTS is just on the boundary of the node sending CTS. Hence, it does hear CTS but the signal is so deteriorated that it fails to recognize it as a CTS. Hence to be on the safer side, a node will not start transmission if it hears either of an RTS or a CTS.

The assumption made in this whole discussion is that if a node X can send packets to a node Y, it can also receive a packet from Y, which is a fair enough assumption given the fact that we are talking of a local network where standard instruments would be used. If that is not the case additional complexities would get introduced in the system.

Does CSMA/CD work universally in the wired networks ?

The problem of range is there in wired networks as well in the form of deterioration of signals. Normally to counter this, we use repeaters, which can regenerate the original signal from a deteriorated one. But does that mean that we can build as long networks as we want with repeaters. The answer, unfortunately, is NO! The reason is the beyond a certain length CSMA/CD will break down.

The mechanism of collision detection which CSMA/CD follows is through listening while talking. What this means is so long as a node is transmitting the packet, it is listening on the cable. If the data it listens to is different from the data it is transmitting it assumes a collision. Once it has stopped transmitting the packet, and has not detected collision while transmission was going on, it assumes that the transmission was successful. The problem arises when the distance between the two nodes is too large. Suppose A wants to transmit some packet to B which is at a very large distance from B. Data can travel on cable only at a finite speed (usually 2/3c, c being the speed of light). So, it is possible that the packet has been transmitted by A onto the cable but the first bit of the packet has not yet reached B. In that case, if a collision occurs, A would be unaware of it occurring. Therefore there is problem in too long a network.

Let us try to parametrize the above problem. Suppose "t" is the time taken for the node A to transmit the packet on the cable and "T" is the time , the packet takes to reach from A to B. Suppose transmission at A starts at time t0. In the worst case the collision takes place just when the first packet is to reach B. Say it is at t0+T-e (e being very small). Then the collision information will take T-e time to propagate back to A. So, at t0+2(T-e) A should still be transmitting. Hence, for the correct detection of collision (ignoring e)

$$t > 2T$$

t increases with the number of bits to be transferred and decreases with the rate of transfer (bits per second). T increases with the distance between the nodes and decreases with the speed of the signal (usually 2/3c). We need to either keep t large enough or T as small. We do not want to live with lower rate of bit transfer and hence slow networks. We can not do anything about the speed of the signal. So what we can rely on is the minimum size of the packet and the distance between the two nodes. Therefore, we fix some minimum size of the packet and if the size is smaller than that, we put in some extra bits to make it reach the minimum size. Accordingly we fix the maximum distance between the nodes. Here too, there is a tradeoff to be made. We do not want the minimum size of the packets to be too large since that wastes lots of resources on cable. At the same time we do not want the distance between the nodes to be too small. Typical minimum packet size is 64 bytes and the corresponding distance is 2-5 kilometers.

Collision Free Protocols

Although collisions do not occur with CSMA/CD once a station has unambigously seized the channel, they can still occur during the contention period. These collisions adversely affect the efficiency of transmission. Hence some protocols have been developed which are contention free.

Bit-Map Method

In this method, there N slots. If node 0 has a frame to send, it transmit a 1 bit during the first slot. No other node is allowed to transmit during this period. Next node 1 gets a chance to transmit 1 bit if it has something to send, regardless of what node 0 had transmitted. This is done for all the nodes. In general node j may declare the fact that it has a frsme to send by inserting a 1 into slot j. Hence after all nodes have passed, each node has complete knowledge of who wants to send a frame. Now they begin transmitting in numerical order. Since everyone knows who is transmitting and when, there could never be any collision.

The basic problem with this protocol is its inefficiency during low load. If a node has to transmit and no other node needs to do so, even then it has to wait for the bitmap to finish. Hence the bitmap will be repeated over and over again if very few nodes want to send wasting valuable bandwidth.

Binary Countdown

In this protocol, a node which wants to signal that it has a frame to send does so by writing its address into the header as a binary number. The arbitration is such that as soon as a node sees that a higher bit position that is 0 in its address has been overwritten with a 1, it gives up. The final result is the address of the node which is allowed to send. After the node has transmitted the whole process is repeated all over again. Given below is an example situation.

Nodes	**Addresses**
A	0010
B	0101
C	1010
D	1001

	1010

Node C having higher priority gets to transmit. The problem with this protocol is that the nodes with higher address always wins. Hence this creates a priority which is highly unfair and hence undesirable.

Limited Contention Protocols

Both the type of protocols described above - Contention based and Contention - free has their own problems. Under conditions of light load, contention is preferable due to its low delay. As the load increases, contention becomes increasingly less attractive, because the overload associated with channel arbitration becomes greater. Just the reverse is true for contention - free protocols. At low load, they have high delay, but as the load increases , the channel efficiency improves rather than getting worse as it does for contention protocols.

Obviously it would be better if one could combine the best properties of the contention and contention - free protocols, that is, protocol which used contention at low loads to provide low delay, but used a cotention-free technique at high load to provide good channel efficiency. Such protocols do exist and are called Limited contention protocols.

It is obvious that the probablity of some station aquiring the channel could only be increased by decreasing the amount of competition. The limited contention protocols do exactly that. They first divide the stations up into (not necessarily disjoint) groups. Only the members of group 0 are permitted to compete for slot 0. The competition for aquiring the slot within a group is contention based. If one of the members of that group succeeds, it aquires the channel and transmits a frame. If there is collision or no node of a particular group wants to send then the members of the next group compete for the next slot. The probablity of a particular node is set to a particular value (optimum).

Adaptive Tree Walk Protocol

The following is the method of adaptive tree protocol. Initially all the nodes are allowed to try to aquire the channel. If it is able to aquire the channel, it sends its frame. If there is collision then the nodes are divided into two equal groups and only one of these groups compete for slot 1. If one of its member aquires the channel then the next slot is reserved for the other group. On the other hand, if there is a collision then that group is again subdivided and the same process is followed. This can be better understood if the nodes are thought of as being organised in a binary tree as shown in the following figure.

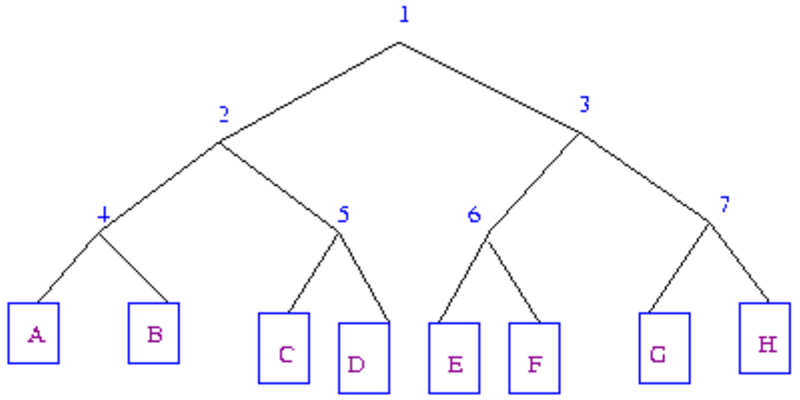


Fig. Adaptive Tree Walk abstraction of nodes in binary tree.

Many improvements could be made to the algorithm. For example, consider the case of nodes G and H being the only ones wanting to transmit. At slot 1 a collision will be detected and so 2 will be tried and it will be found to be idle. Hence it is pointless to probe 3 and one should directly go to 6,7.

Image References:

- http://www.ccs.neu.edu/course/csg150/Solutions-III_files/image002.gif

---

back to top
Prev| Next | Index