CSMA/CD using Network Priority Queue and Scheduling/Aging Techniques

Roopikha Ganesh

SCSE, Vellore Institute of Technology, Chennai

DOI: 10.29322/IJSRP.9.11.2019.p9576
http://dx.doi.org/10.29322/IJSRP.9.11.2019.p9576

Abstract- CSMA/CD suffers from drawbacks like packet starvation and wastage of bandwidth due to un-utilized periods of the channel. Its performance can be enhanced greatly by modifying the protocol after taking all these factors into consideration.
The proposed protocol make use of a network queue in addition with a priority counter to overcome the problems of the existing MAC protocol. It ensures that whenever there is congestion and high probability of collision and therefore also wastage of bandwidth, another alternative protocol is initiated. This increases its efficiency and also provides the regular protocol as a back-up option in case of queue failure. It also provides a mechanism for real-time video transmission assuming that this feature is not required often.

Index Terms- Aging, CSMA/CD, Collision, Scheduling

I. Introduction to CSMA/CD and its Drawbacks

Ethernet currently employs the CSMA/CD protocol also known as the IEEE Standard 802.3 for media access resolution in the data link layer. CSMA/CD or Carrier sense multiple access with collision detection adds on to the CSMA algorithm to define the procedure in case there is a collision.
CSMA was based on the idea that if the channel can be sensed before it is tried to accessed, the number of collisions could be drastically reduced. However, it takes a while for the nodes to receive the bits already being transmitted by another node as a result of propagation delay. This implies that a node might determine the channel is idle and start transmitting before it learns that the channel was already in use. This leads to a collision. CSMA/CD determines how the stations should proceed in this eventually.

In CSMA/CD, the transmission and collision detection is a continuous and simultaneous process that means that the node transmits and receives using two different ports. Though one port, it transmits while through the other it monitors the channel. If the transmission is completed the process is terminated. If a collision is detected, the process is still terminated and the collision procedure is initiated another attempt at transmission.
The node that detects a collision sends out a short jamming signal through usually 48bits long to let the other nodes know about it and inform them that all the frames involved have either been destroyed or modified so that they can be discarded. The nodes involved in a collision then have to wait a random amount of time before they can try to transmit again, the waiting time period is decided using the binary exponential back-off algorithm(BEB) and referred to as the back-off time. If the frame to be transmitted has already suffered K collisions then the node will pick a value R from among {0,1,2,2^k -1} randomly and wait R.TFR , where T FR is average transmission time for a frame before incrementing the counter K and trying again. Eventually, it will pick a value of R small enough to start transmitting before all the other nodes who want to transmit. Now, if there were no restrictions it could mean that the value of R becomes unreasonably large so we will limit the value of R at 1023, i.e., at the most a data frame would have to wait 1023. T FR time. There is also a possibility that the node tries to access the channel but is deterred every time by a collision. This only increases the possibility of collision in a congested network. To overcome this the highest value that K can take is limited to 15,allowing a total of 16 retransmission[2].K max is set at 15.If the value of K exceeds this it is automatically reset to 0 before it enters the process again.
This protocol is efficient when there is low to medium network traffic. However when the network is busy it creates problems.

II. Analysing the Problem More and the Appoximate Solution for It

The probability that a particular manages to get to the channel depends on its waiting period.
The first time a node tries to transmit the value of the counter 'n' is set to zero. In the scenario that it finds the channel to be busy it increments the counter to 1 and picks a value of R from among{0,1} to decide its waiting time. This implies that the first time a node tries to transmit after encountering a collision it has a 50% chance of success. The second time it will have to choose a value of R from among a larger set {0,1,2,3} which will decrease the chance of success to 25%. This value will go on decreasing with increase in the value f K or the number of attempted transmissions for a frame.
This leads to the conclusion that a node with a fresh frame to send has a higher chance of acquiring the channel than a node that has been waiting longer.
Also, some node that has just sent its frame and now has a new frame to send out can access the channel more easily than a node which has been waiting its allotted time because the transmission

probability for a new frame if it senses the channel to be idle is 100%. It can do this again and again and unfairly hold the network. This is often referred to as ethernet capture.
Also, since the entire process is randomized there is a possibility that even though many nodes have data to send they are all waiting. This leaves the channel unoccupied and wastes resources while also increasing the probability of collision.

Approximate Solution

If we see our through are CSMA/CD ,if a collision occurs between 2 nodes or frames they are K value is incremented and then they are made to wait for a random amount of time. Now again during the 2nd transmission if a collision occurs with the same frame ,then again its K value is incremented and its made to wait for a longer period. This can go on till a maximum value of 15 or 16 times in total .But now instead of this I would prefer that we can have a priority counter and a priority queue during such situations. So the counter will decide the priority for the for the frames to be transmitted. So for each 3 transmission the value of k will be incremented. Based on that the values will be incremented at 3,6,9,12,15 and again 3.This process will continue.

III. PRIORITY QUEUE

If any network congestion occurs then we activate the heavy traffic protocol for rescue.
So, firstly our priority queue is empty. It will accept input when a collision has occurred. Once a collision has occurred the frame will be added to the priority queue. The frame will be assigned a position according to the number of transmissions it has already attempted. The node with the higher value will be before the node with a lower value. In case, both the nodes have the same value then they will follow the first come first serve algorithm to decide priority implying that the node whose value exceeded to first will get a priority. Again choosing a value as 2 for employing the following procedure is just a conjecture at this point. Further analysis can be done to determine the optimal value. For a frame 2 qualify to enter in to the priority queue. If there are more than 6 nodes in the priority queue the protocol for the congested network automatically activated and the priority queue starts producing a jamming signal to tell all nodes in the network that it is going to start transmitting.
In essence, the priority queue is just a node with the transmitter and a queue with network responsibility. It will go on transmitting until the queue is empty. After which it will send another jamming signal to notify all nodes to assume normal behavior and revert back to the original CSMA/CD protocol being followed earlier.
The use of priority queue will reduce the traffic network.

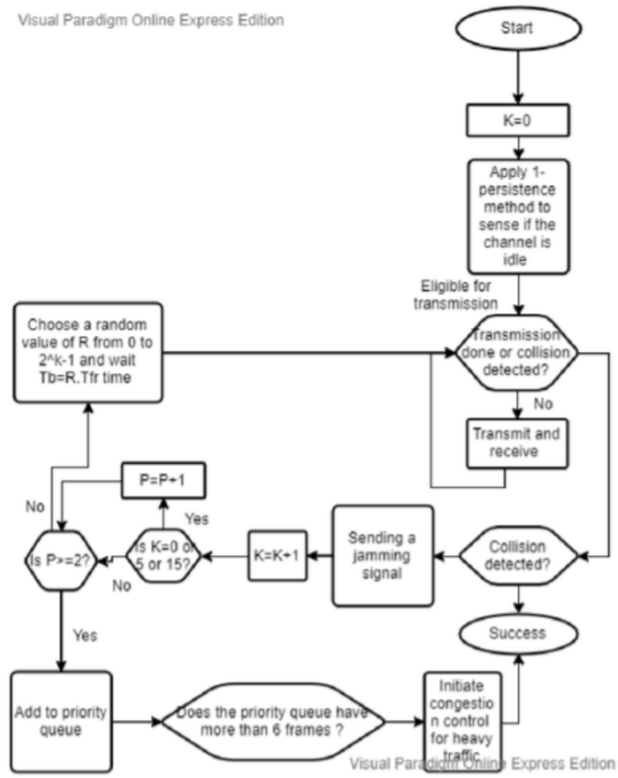

Figure 1: modified CSMA/CD Flowchart with Network Priority Queue

IV. UNDERSTANDING USING AN EXAMPLE

One of the main reasons or disadvantage of packet starvation is that many a times frames with higher priorities just because the low priority once get their chances quicker.
FOR EXAMPLE:
Consider a frame F1 and a frame F2. Here F1 has higher priority since it has encountered maximum number of collisions that F2.Suppose at time t both the frames are executed and a collision is encountered. Thus we need frame F1 to be next. But then we can't decide on which frame among F1 and F2 will executed as they would be waiting for a random amount of time and then be executed or sent. Thus, there is an equal chance for both frame F2 and F1 to be sent next.
So for resolving the above situation we use a priority queue and priority counter so that based on the frames with higher priorities they will be executed accordingly first. Thus the above case can be resolved.
A detailed representation is given in Figure 1.
Otherwise the above method works similar to the normal queue. This method is similar to technique of ageing or scheduling which is popularly used world wide.

V. CONCLUSION

The above proposed solution may be found to complicated, but then it's basically a slight change and modification of the regular CSMA/CD.

The only modification is that we are introducing a priority queue in this which helps us to sort the packet starvation problem which is often encountered in the regular CSMA/CD. Thus the priority queue will set priorities for the frames which are more important during a collision .Hence the one with higher priority may be sent first and then the lower priorities one's subsequently increasing the fairness among transmissions.

AUTHORS

First Author – Roopikha Ganesh, student, SCSE, Vellore Institute of Technology, Chennai .Email address:roopikhap@yahoo.com.