CSMA/CD with Priority Queue

Akshita Saini

Computer Science and Engineering, Vellore Institute of Technology

DOI: 10.29322/IJSRP.9.11.2019.p95116
http://dx.doi.org/10.29322/IJSRP.9.11.2019.p95116

ABSTRACT

CSMA/CD suffers from drawbacks like packet starvation and wastage of bandwidth due to unutilised periods of the channel. Its performance can be enhanced greatly by modifying the protocol after taking all these factors into consideration.

The proposed protocol makes 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 audio and video transmission assuming that this feature is not required often.

INDEX TERMS

CSMA/CD, Ethernet, network congestion, packet starvation

1. INTRODUCTION

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 eventuality.

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. Through 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 for another attempt at transmission [1].

The node that detects a collision sends out a short jamming signal usually 48 bits 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, T_b. 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 $R.T_{FR}$, 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 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 a collision in a congested network . To overcome this the highest value that K can take is limited to 15, allowing a total of 16 retransmissions[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.

2. RESEARCH PROBLEM

The probability that a particular node manages to get access 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 its chance of success to 25%. This value will go on decreasing with increase in the value of 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.

It causes other nodes to timeout by manipulating the network. Thus preventing other nodes from transmitting. It is especially common when a node is transmitting high volume data such as video packets [3].

Also, since the entire process is randomised 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 [3].

CSMA/CD protocol does not provide a provision for real time audio and video transmissions. This is a major drawback because many up-coming technologies and applications require this feature.

I have modified the CSMA/CD protocol considering all these factors and proposed an alternative to increase its efficiency in case of heavy traffic and provide accommodation for real-time audio and video transmissions.

3. MODIFIED PROTOCOL FOR HEAVY TRAFFIC

3.1 Determining Network Congestion using Priority Counter

The problem of packet starvation and ethernet capture will only arise when there is heavy traffic or congestion on the network. To solve this it is essential to determine whether the channel is congested or not.

Suppose it is, there will be many nodes with data frames to send out and they will all be competing against each other to access the channel. This contention will mean that most of them will collide again and again and with every collision the value of their collision counter K will increase. Until it eventually exceeds 15 and is reset to 0. Instead of this, we can have another counter P that decides the priority of the frame by incrementing it after every 5 transmission attempts. I have chosen 5 here because the minimum value of the counter can be 0 and maximum can be 15. So we will increment the P counter when K becomes 5, 10 or 15 and again when it becomes 5. This is necessary to ensure that a node which has re-entered the algorithm is given the highest priority.

Ethernet can only have a maximum of 16 nodes on a single shared media channel and each node can have at most one frame ready for transmission into the channel. Assuming that 6 nodes competing for the channel is considered heavy traffic, when more than 6 nodes have priority counter P >=2. We assume can that the channel is congested and activate the alternative protocol for heavy traffic.

3.2 Priority Queue

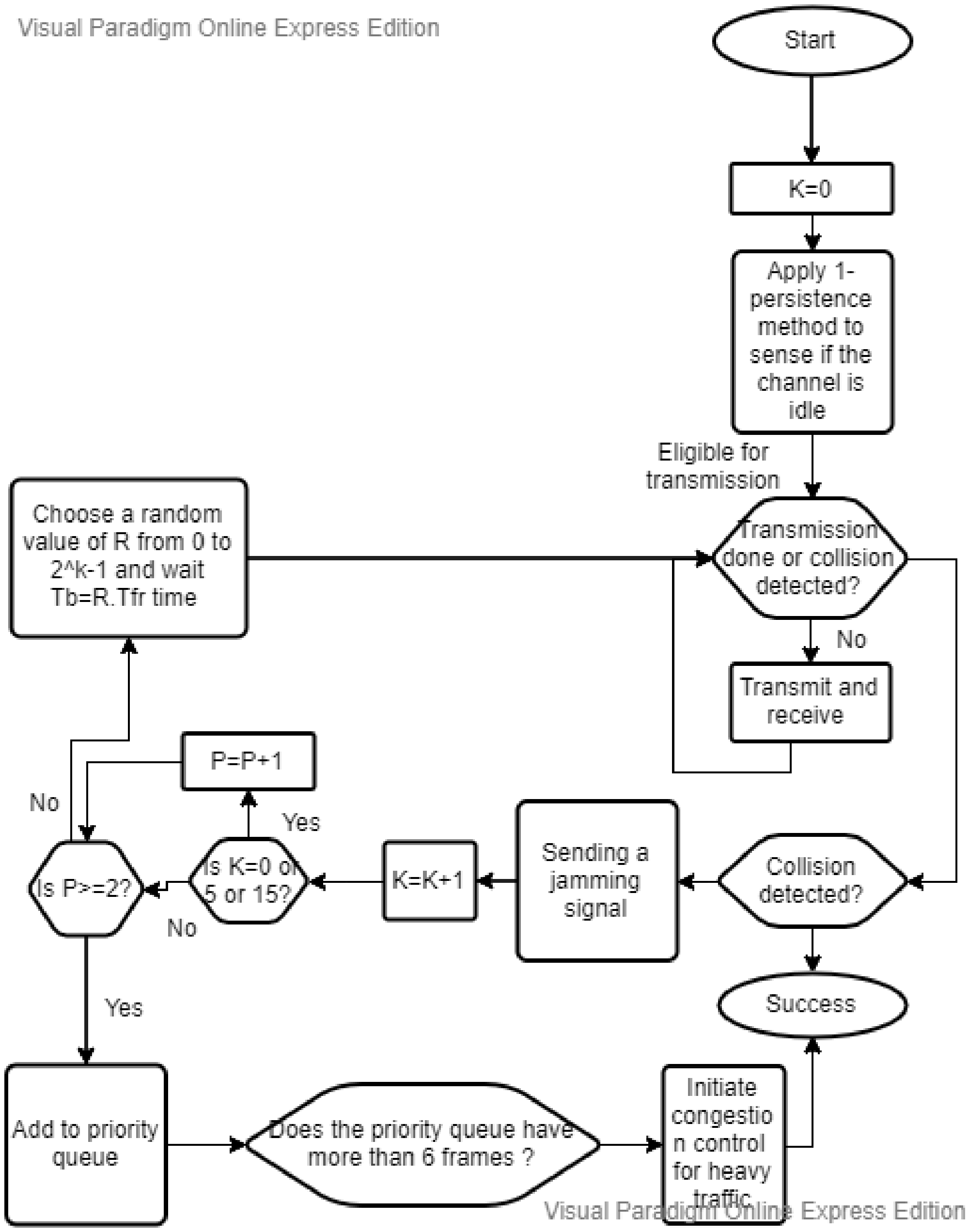
As the P value for a node exceeds the decided value which in this case is 5, it will be added to the priority queue. The frame will be assigned a position in the queue according to the number of transmissions it has already attempted. The node with a higher value of P will be before a node with a lower value of P. In case two nodes have the same value of P then they will follow the First Come First Serve algorithm to decide priority implying that the node whose P value exceeded 2 first will get priority. Again choosing the value of P as 2 for employing the following procedure is just conjecture at this point. Further analysis can be done to determine the optimal value at which Priority Counter, P should be incremented and the minimum value of P at which a frame is qualified to enter into the priority queue.

If there are more than 6 nodes in the priority queue the protocol for congested network is automatically activated and the priority queue sends a jamming signal to let all the nodes in the network know that it is about to start transmitting.

In essence, the priority queue is just a node with a 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 the nodes to assume normal behaviour and revert back to the original CSMA/CD protocol being followed earlier.

The use of the priority queue will reduce network traffic.

Refer to the following link for a clearer view of the algorithm: https://diagrams.visual-paradigm.com/#proj=0&type=Flowchart



4. ACCOMMODATION FOR REAL-TIME TRAFFIC

4.1 Flags for Real-Time Data Transmission

Each node will have a flag whose value can be either 1 or 0. When it is 0, the protocol will behave normally. When it is 1, the procedure for real-time transmission will be initiated. It will freeze the state of the transmission counter, the priority counter and the priority queue. The node which wants to send real-time data will set it's flag to 1 and then set out a jamming signal to ensure that all the other nodes have set their flag to 1 as well.

When all the flags are 1, the channel will be utilised by the first node which initiated the procedure for transmission of real time video or audio data.

This is necessary because it requires continuous streaming of data which is otherwise not possible with the existing protocol since that would require monopolising the network entirely by one node.

At the same time, there has to be a restriction on how long a node can hold the network. This will be long but not so long that it blocks all the other nodes from using the channel.

4.2 Collision Resolution during real-time transmission

When one node has just finished transmitting real-time data the other node cannot be immediately allowed to hold the network, it will be unfair to all the other nodes which have ready data to send out. As a result, we place a limit on how many times a node can utilise this feature and also how long it has to wait before making another valid request, this is done using a time out feature. Every node after will have to wait a minimum amount of time after it just made a real-time transmission. Every node will also have to wait after the channel has just been used for a real-time transmission. That is, in either case a time out will be started and the channel becomes available for real-time transmission.

Now, there can still be more than one node which wants to acquire the channel for this, in this case again there will be a collision which will be resolved using the standard Binary Exponential Back-off algorithm. This is done keeping in mind that the assumption that not many nodes will want to transmit real-time data at the same time which will decrease the probability of a collision enough to ensure that this does not create a problem.

5. DRAWBACK OF THE PROPOSED MODIFICATIONS

The solution proposed above is a combination of a centralised and decentralised method. This has improved its efficiency and allowed for real-time transmission at the same time compromising it's maintainability and dependability. This is a result of including a centralised feature which can practically bring down the entire system by acting as the single point of failure.

To minimise the damage this might cause to the network I have provided a provision for back-up. All the nodes will listen in to the channel and if they detect its idle for longer than a certain period of time they will assume that the priority queue has gone down and will use the standard CSMA/CD protocol in case of both heavy or medium to low traffic. This will ensure that even in the eventuality that the priority queue mechanism fails, the nodes can still transmit data across the channel.

6. ANALYSIS

It's been inspired from the idea of a network queue proposed in [3]. However, certain appropriate modifications have been made to it to enable real-time transmission along with the introduction of the priority concept to make the process more fair. This method also ensures that a node is not locked out or is made to wait inevitably because the longer it tries for transmission the higher priority it is assigned.

The idea proposed gives a solution to ethernet capture and packet starvation, thus making the system more fair and increasing its overall efficiency.

7. RELATED RESEARCH

7.1 Network Queue

CSMA/CD is modified to include a network queue with a send variable as given in [3].

The queue constantly listens to the channel and when it senses activity it listens for interruption or completion. The queue becomes active when it senses a collision in the network or a 48 bit jam signal. The queue basically sends out a frame that collects the MAC address of every node involved in the collision. When this frame comes back, the network queue places all these MAC addresses in its queue and network queue takeover begins[3].

Each node has a variable which is changed to true when it is its turn to transmit according to the MAC addresses in the queue. This behaviour continues until there are no more addresses in the queue[3].

When it is finished it sends out a broadcast signal changing all the nodes' variables to true so that the network can function normally.

7.2 The Packet Starvation Effect(PSE)

PSE causes some packets to experience extremely long delays and some to starve out due to 16 collisions. According to [4] stimulation shows that the effect usually becomes significant at an offered load of about 60% and 70% and gets only worse as the load increases even further. The reason for packet starvation is that when two packets compete for access over the other is approximately proportional to the ratio of their maximum backoff values. When two packets become ready at approximately the same time, the two controllers will both wait until they see that the network is free and then attempt to resend, colliding with each other. When this happens they back off a random amount of time based on the number of collisions that the packet has already suffered. The probability that an older packet selects a smaller back-off value than a newer packet with fewer collisions is less than the ratio of the newer packet's maximum backoff divided by the older packet's maximum backoff. Because this value increases exponentially, unless a packet comes ready when no other host is ready to send, it will usually either get access to the bus very quickly or it will experience 16 collisions and starve out. Under high load, there is usually another packet to send and so long delays and packet starvation occurs to a significant percentage of packets [5].

8. SUMMARY

CSMA/CD with Priority Queue is a hybrid protocol because it is a cross between two of the three multiple access protocols, random access and controlled access protocols. It aims to use the channel resources to their maximum capacity by initiating a separate algorithm in cases where the efficiency of the regular CSMA/CD protocol fails such as during congestion. To make this possible in the simplest of ways it makes use of a priority queue which starts transmitting when the channel has been idle for too long or when there is congestion. Since, it is a priority queue it has to assign each node a priority in a fair manner this is done using the already existing collision counter, K to introduce another counter called the priority counter, P. When the P value is high enough the frame will make it to the queue, which will be activated when the required minimum number of frames are already waiting in the priority queue. The application of the protocol in a stimulated environment can allow us to find the most optimum values at which to increment the priority counter, initiate the queue and also when a node is qualified to be added to the priority queue.

It also allows the nodes to transmit real-time data assuming that not many nodes will require to use this feature at the same time. This is done by using a time out feature and flags which indicates that a node wants to request the channel for a real-time transmission. The flags when set to true suspend the procedure otherwise going on and allocate all the resources solely to the node that made the request while ensuring that enough time has passed since the last time the request was made.

However, it is only of use when only a few nodes want to transmit real-time data. Otherwise it fails. This will require further research to overcome the faults of the proposed idea.

REFERENCES

1. Behrouz A. Forouzan Data Communications and Networking 4th edition
2. Computer Networking A Top-Down Approach Kurose and Ross 6th edition
3. Modifying the CSMA/CD (IEEE 802.3) Protocol to Involve a Network Queue (CSMA/CDNQ) by Abbey Wineland George Mason University awinelan@gmu.edu
4. Adaptive CSMA/CD: An Improved MAC Protocol Dr. S. Silakari, Piyush Kumar Shukla and Ujjwal Nigam and Anuj Garg https://www.academia.edu/2304478/Adaptive_CSMA_CD_an_improved_MAC_protocol
5. The Packet Starvation Effect in CSMA/CD LANs and a Solution Brian Whetten, Stephen Steinberg, Domenico Ferrari University of California at Bekerley http://www.ethermanage.com/ethernet/pdf/FDDQ_LCN_1.pdf

AUTHORS

Akshita Saini, Vellore Institute of Technology

Email address: akshitasaini007@gmail.com