

An Algorithmic Approach to avoid Starvation in Wireless Mesh Networks

B. Ramesh Babu¹

SK. Mansoor Rahaman²
Venkaiahnaidu. A²

T. Pavan Kumar¹

¹Assoc. Prof., Dept. of CSE

²Student, Dept of CSE
KL University, India

ABSTRACT

Upon the advancement of recent technologies in wireless communications the wireless mesh networks has made people's life easier. The wireless mesh networks provide low cost high speed internet access through the spontaneously deployed WIFI hotspots. The users will experience performance degradation due to starvation between the nodes in wireless mesh networks. The nodes which are nearby to the gateway will have high performance compared to the nodes which are far away from the gateway. The nodes which are far away from the gateway are starved by the nodes which are one hop away from the gateway. So to subjugate starvation in wireless mesh networks the gateway acts as a decision maker to the corresponding network to control the overall activities performed in a network. In this paper we are introducing a new starvation mitigation algorithm which mitigates starvation in wireless mesh networks. The gateway will prevent the network without being starved by sending a warning message to all the nodes when the channel is being utilized by a same node in a network. Then the remaining nodes come to know about the situation in a network and they will increase their contention window by $3/2$ times. By this algorithm the network performance and bandwidth utilization will be increased compared to the existing system.

Key words: wireless mesh network, Wi-Fi, starvation mitigation algorithm, network bandwidth

1. INTRODUCTION

The definition of wireless mesh network varies from type to type. The definition is explained based upon the architecture used. There are 3 different types of architectures in wireless mesh networks:

- 1.) Infrastructure/ backbone WMN's
- 2.) Client WMN's
- 3.) Hybrid WMN's

Infrastructure/ backbone WMN's only mesh routers are used. These are mainly used to form a backbone to the clients which are connected to them. Infrastructure/backbone WMN's are most commonly used in community and neighbourhood networks which are able to build infrastructure meshing. [1]

In this type of meshing the routers are placed on the top of the buildings which provide the internet access to the person inside the building and people on the roads. The client wireless meshing is mainly used to provide peer to peer networks to the clients. In this type of meshing the client nodes establish the actual network to perform routing and Configurations in order to provide user applications to the customers. Compared with the infrastructure/backbone WMN's the end user requirements are increased to perform additional functionalities like self-healing and self-configuring. The combination of infrastructure/ backbone WMN's and client WMN's resulted as hybrid WMN's. this hybrid WMN's is mainly used to provide access to the cellular networks, Wi-Fi, Wi-Max, sensor networks. Compared to the infrastructure and client meshing the routing capabilities are increased in hybrid WMN's which provide the better connectivity and coverage. [1][2][3]

2. STARVATION IN WIRELESS MESH NETWORKS

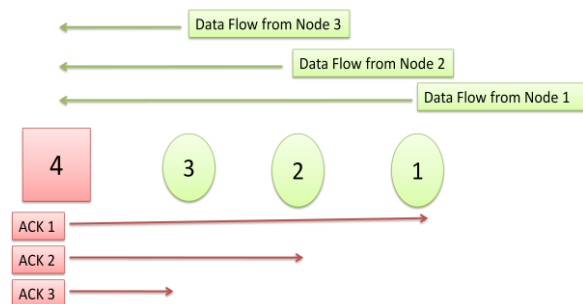


Fig 1: A network containing 3 nodes and a gateway node showing the data flow

To explain how starvation occurs in a network we consider the above example which consists of 3 nodes and a gateway node. As the nodes which are closer to the gateway node i.e., node 4 will first engages the channel for the transmission of packets. After one successful transmission the node will resets its contention window and again ready to start its transmissions. As the remaining nodes in a network mostly node 1 will get starvation. The node 1 will not get a chance of using the network bandwidth as a result the node starves. So to mitigate the problem of starvation we are introducing an algorithm called Starvation Mitigation Algorithm (SMA) for

efficient use of network bandwidth by the all the nodes in a network which results in reducing the starvation as well. [5]

3. RELATED WORK

Extensive research have been done in [4] [6] [7] [8] [9] [10] [11] [12] for reduction of starvation in wireless mesh networks as it is a main issue in wireless mesh networks. As per the literature in [4] the author introduced an algorithm called fair binary exponential backoff algorithm (FBEB) which is the extension of binary exponential backoff algorithm (BEB). The binary exponential backoff algorithm is a widely used backoff algorithm, which adjusts the contention window size by indirectly estimating the traffic in the communication medium at individual nodes in effect by counting consecutive collisions involving the same packet. The FBEB algorithm states that a node which successfully transmits a packet will not reset its contention window for the next transmission; it has to wait for a time T . By controlling the node for resetting will make other nodes to access the channel. The time T depends on the previous contention in the network. But this algorithm also states that if a packet collision occurs the node will have to double its contention window and starts the process again, if the contention window reaches its maximum value it is reset and the transmitting packet has been dropped. If the contention window is doubled it becomes problem to other nodes which suffer from starvation. So to overcome these problems from the existing system we are introducing a new algorithm called Starvation Mitigation Algorithm (SMA) in which the gateway acts as a decision maker and broadcasts a message if any node is causing the starvation by accessing the channel continuously with out giving a chance to remaining nodes. Then by receiving this message remaining nodes will increase their contention to 1 or 2 packets and transmits again and the node which is causing the starvation will decrease their contention window to half of its actual size.

4. PROPOSED WORK:

4.1 Starvation mitigation algorithm (SMA):

The main reason for starvation is that, the nodes which are close to the gateway will not give a chance to other nodes for accessing the channel. So the flow gets starved and the bandwidth is not efficiently utilized by the network. So to avoid this problem we introduce a starvation mitigation algorithm (SMA) by which we can provide efficient bandwidth utilization by all the nodes in a network and provide fairness in a network.

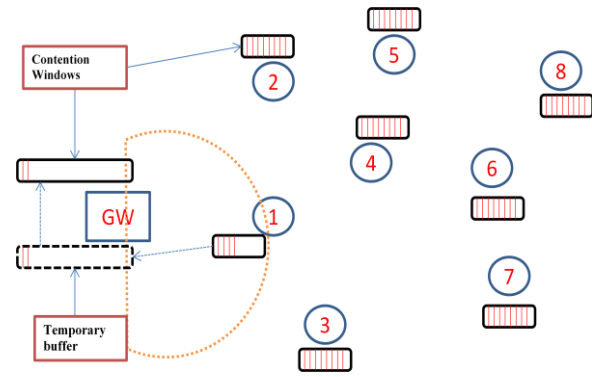


Fig 2: Wireless mesh network in which node 1 is transmitting packets to gateway.

In the above figure node 1 is transmitting packets continuously to the gateway, at the gateway the packets are collected in temporary buffer and those packets are moved to the contention window of the gateway.

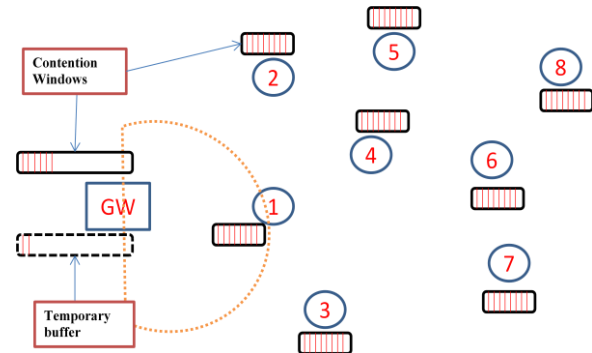


Fig 3: Node 1 after resetting the contention window

In the above figure the packets are successfully transmitted to the gateway by the node 1 and the node 1 resets its contention window for aggressively transmitting packets continuously without giving a chance to other nodes for transmitting the packets by causing the flow to be starved.

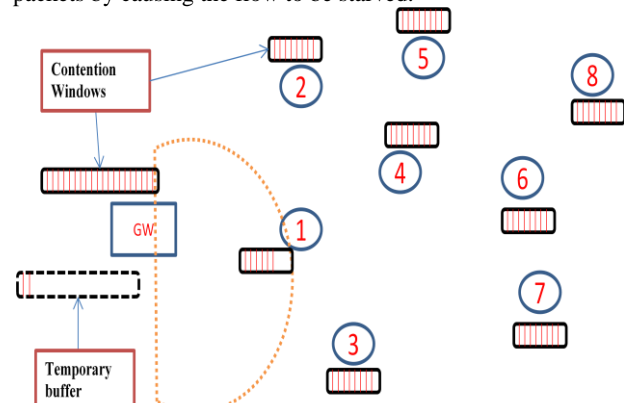


Fig 4: A wireless mesh network when the contention window of gateway is full.

After the contention window of gateway is full the gateway checks whether the packets are from same node or different node.

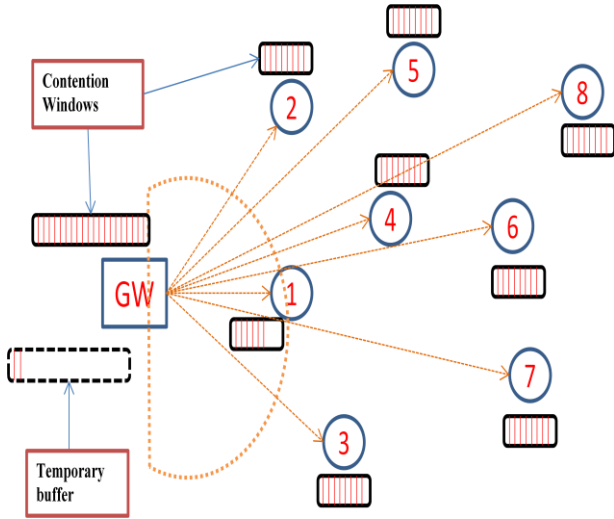


Fig 5: A wireless mesh network where gateway is broadcasting message to all the nodes.

As the gateway comes to know that all the packets are from same node then the gateway broadcasts a message to all the nodes that node 1 is using utilizing the total bandwidth of the network by causing starvation to remaining flows.

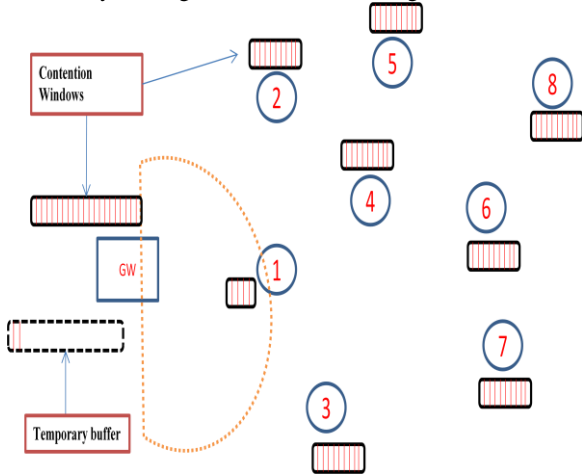


Fig 6: A wireless mesh network after reducing the contention window of node 1.

The gateway sends a message, that node 1 has to decrease its contention window i.e., $CW/2$ so as to slow down the flow at node 1 and to provide sufficient access to gateway by the remaining nodes.

4.2 Starvation Mitigation Algorithm (SMA) executed by the gateway in a network:

1. $GW_{max} = 20$
2. $N_{max} = 8$
3. $N_{min} = 4$
4. $TemBuf = 30$
5. $GW_{size} = 1$
6. *if* ($GW_{size} \leq GW_{max}$)
7. **repeat**

8. accept packets from nodes
9. *if* ((choose a packet P_{id} from $TemBuf$) == 0)
10. **wait for packets to be received from nodes**
11. **else**
12. **while** (P_{id} is a packet of new node)
13. $node[P_{id}] = 1$
14. $flag[P_{id}] = true$
15. **push** the packet to GW_{size}
16. $GW_{size} = GW_{size} + 1$
17. **end if**
18. *if* ($flag[P_{id}] == true$)
19. $node[P_{id}] = node[P_{id}] + 1$
20. **push** the packet to GW_{size}
21. $GW_{size} = GW_{size} + 1$
22. **end if**
23. **until** ($GW_{size} == GW_{max}$)
24. *if* ($GW_{max} == 20$)
25. $ID = node[P_{id}]$
26. **while** ($ID = GW_{max}$)
27. Broadcast message to all nodes that $node[P_{id}]$ is using the channel continuously
28. $limit[P_{id}] = \frac{N_{max}}{2}$
29. **end if**

In this algorithm the gateway will accept up to 20 packets from each node. When its contention window is full then it checks that all the packets are from a single node or not, by maintaining the MAC address of each node in a variable P_{id} . if a node sends more than 20 packets then it generates a message and send that message to all the nodes that $node[P_{id}]$ is using the channel continuously with the MAC address of the particular node in $node[P_{id}]$ variable. Then after receiving this message from the gateway the $node[P_{id}]$ will reduce its contention window to half of its normal size and the remaining nodes come to know that $node[P_{id}]$ is using the channel continuously which is causing starvation to the flow and the remaining nodes will increase the size of contention window depending upon their requirement.

Description

Consider gateway's maximum contention window to be as 20 and minimum to 1, nodes minimum packets in contention window is 4 and maximum is 8 and the capacity of temporary buffer at the gateway is 30 packets. If the contention window at gateway is less than or equal to maximum i.e., 20 then continuously accept the packets from the nodes in a network. Then choose a packet from temporary buffer ($TemBuf$) and save that packet's MAC address into variable P_{id} . if that packet is from new node assign that packet as 1 and enable the flag to true and push the packet into gateway's contention window and increase the contention window of gateway by 1. If the next packet is also from the same node then push the packet to the gateway's contention window and increase the node count by 1, as the node count increases then gateway's contention window size also increases by 1. Accept the packets from all the nodes until the contention window at the

gateway is full i.e., 20 packets. Then check the packets MAC addresses are same or not, if all packets are from same node then broadcast a message that the node x is accessing the channel continuously by causing starvation to remaining flows and take a decision of reducing its contention window by half of its actual size. Then remaining nodes will automatically get a chance of accessing the channel because the packets produced by the node x are reduced. So this algorithm provides the efficient use of bandwidth by all the nodes in a network.

4.3. Algorithm Performed by a node in a

Network:

```

if((message from GW for limit[Pid]) == CWsize/2)
{
    CWsize = CWsize/2
Else if (limit[Pid] ≠ node[Pid])
then CWsize = CWsize *  $\frac{3}{2}$  upto 5msec
}
end if

```

Description

When a node receives a message from the gateway, it limits the contention window with the specific P_{id} then the contention window will be decreased by half of its original size. If the $limit[P_{id}]$ is not equal to the nodes sequence number then contention window of a node will be increased by a multiplication factor of $3/2$.

5. SIMULATION RESULTS

The below bar graph shows the number of packets requested by the nodes in a network and the number of data packets which are transmitted by the gateway in a basic scenario of FIG 2. The green bar indicates the data packets sent by gateway and red bar indicate the request packets. As shown in the below graph the highest data packets are received by node1, which is the main reason for causing starvation. The request packets of node 6 and node 1 are same, but the data packets sent by gateway are less compared to node1. Due to the utilization of total bandwidth by node 1 the overall network gets starved. The data packets are double at the node 1 compared to all the nodes in a network. Table 1 provides network specifications.

Routing Protocol	AODV
Network Size	600×600
Simulation Time	200 sec
Network Connections	TCP
Data Rates	512,1024 etc
Number of Nodes	10 (1 Gateway)
Number of Connections	10, 20, 30, etc

Table 1: Network Parameters

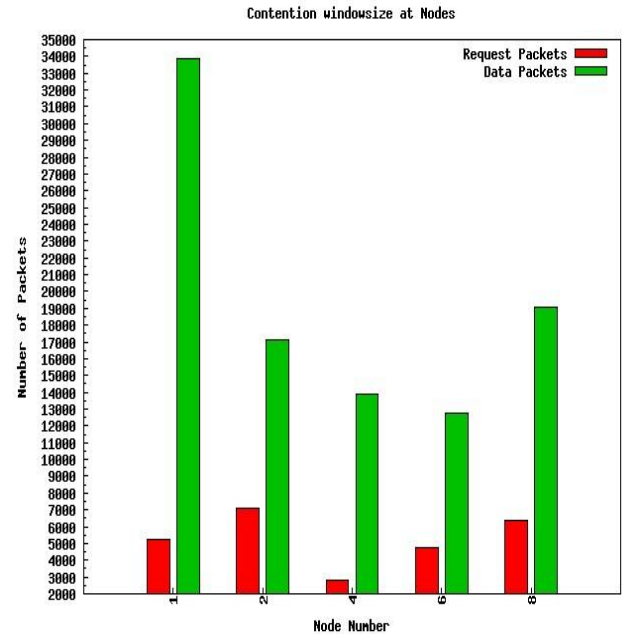


Fig 7: Number of Requested Packets and Transmitted Data Packets in a Basic Scenario of a Wireless Mesh Network.

5.1 Throughput of the overall network:

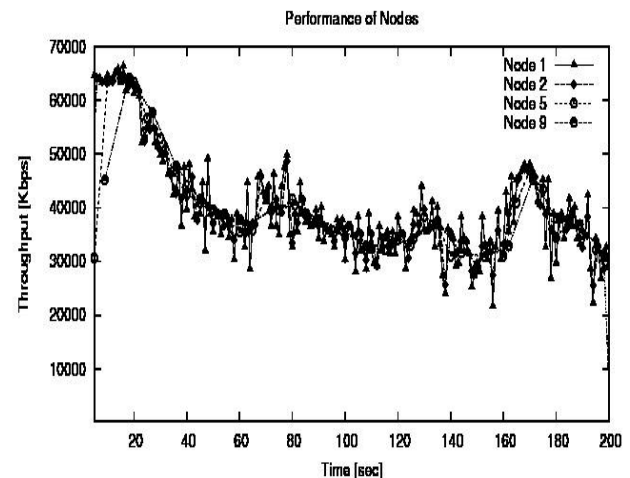


Fig 8: Throughput of the nodes 1,2,5 and 8 of a wireless mesh network.

The above graph shows the overall throughput of nodes 1, 2, 5 and 8 in a network shown in FIG 2. The highest performance in a network is at the node 1, which is near to the gateway. Compared to all the nodes in a network the poor performance is at node 8 due to the more number of hops between the gateway and node 8. so the simulation results shows that the nodes which are near to the gateway are utilizing total bandwidth and maintaining high performance compared to remaining nodes in a network.

6. CONCLUSION

We proposed a Starvation mitigation algorithm (SMA) to reduce the extent of starvation in wireless mesh networks. Our proposed algorithm used to reduce the starvation occurred in a network and mesh nodes that have successfully transmitted data packets should not be permitted to transmit more data packets aggressively. By delaying the transmission of excessive packets by the nodes we are able to reduce the degree of starvation. This effect is achieved by adapting the contention window for transmissions in the IEEE 802.11 protocol. Overall Throughput is increased and Delays in the network are mitigated. Bandwidth utilization is more compared to the existing system.

7. ACKNOWLEDGEMENTS

We are greatly delighted to place my most profound appreciation to Mr.K.Satyanarayana Chancellor of K.L.University, Dr.K.Raja Sekhara Rao Principal, S.Venkateswarlu Head of the department, Dr. V.Srikath, Dean SED and Dr.Subramanyam in charge for M.Tech under their guidance and encouragement and kindness in giving us the opportunity to carry out the paper. Their pleasure nature, directions, concerns towards us and their readiness to share ideas encouraged us and rejuvenated our efforts towards our goal. We also thank the anonymous references of this paper for their valuable comments.

8. REFERENCES

- [1] Ian F. Akyildiz, et al, "Wireless Mesh Networks: A survey", Computer Networks, May, 2005
- [2] "Building a rural wireless mesh network" by David Johnson, karel mathee, Dare Sokoya, Lawrence mboweni, Ajay Makan, and Henk Kotze Meraka Institute South Africa 30 Oct 2007.
- [3] "Wireless mesh networks: a survey" Ian F. Akyildiz , Xudong Wang , Weilin Wang Broadband and Wireless Networking (BWN) Lab, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA January 2005.
- [4] "Flow starvation mitigation for wireless mesh networks" Ronasi, K.; Gopalakrishnan, S.; Wong, V. Dept. of Electr. & Comput. Eng., Univ. of British Columbia, Vancouver, BC WCNC 2009.
- [5] "Starvation Modelling and Identification in Dense 802.11 Wireless Mesh Community Networks" Cunqing Hua and Rong Zheng Dept. of Computer Science Eng., Univ. of Houston TX, USA, Aug 2007.
- [6] "A Weighted RED for Alleviating Starvation Problem in Wireless Mesh Networks" Chaegwon Lim; Chong-Ho Choi; Hyuk Lim; Telecommun. R&D Center, Samsung Electron. Co., Ltd., Suwon IEEE conference 2008.
- [7] K. Xu, M. Gerla, L. Qi, and Y. Shu, "Enhancing TCP fairness in ad hoc wireless networks using neighborhood RED," in *Proc. of ACM MobiCom*, San Diego, CA, Sept. 2003.
- [8] A. Raniwala, D. Pradipta, and S. Sharma, "End-to-end flow fairness over IEEE 802.11-based wireless mesh networks," in *Proc. of IEEE Infocom*, Anchorage, AK, May 2007.
- [9] J. Shi, O. Gurewitz, V. Mancuso, J. Camp, and E. W. Knightly, "Measurement and modeling of the origins of starvation in congestion controlled mesh networks," in *Proc. of IEEE Infocom*, Phoenix, AZ, Apr. 2008.
- [10] S. Rangwala, A. Jindal, K. Y. Jang, K. Psounis, and R. Govindan, "Understanding congestion control in multi-hop wireless mesh networks," in *Proc. of ACM MobiCom*, San Francisco, CA, Sept. 2008.
- [11] Wei Wang, Ben Leong, Wei Tsang Ooi, "Understanding and mitigated TCP starvation in 802.11 wireless mesh networks", Network Protocols (ICNP), 19th IEEE International Conference, pp-121-122, 2011
- [12] Rangwala, S., Jindal, A., Ki-Young Jang, Psounis, K., Govindan, R, "Neighborhood-Centric Congestion Control for Multihop Wireless Mesh Networks", Networking, IEEE/ACM Transactions , pp - 1797-1810, Dec' 2011.