

Transmission Rate Control through MAC Layer Acknowledgements Feedback

Mohamed Arabi

Center for Security, Communications and Network
Research
University of Plymouth
Plymouth, UK
mohamed.arabi@plymouth.ac.uk

Bogdan Ghita

Center for Security, Communications and Network
Research
University of Plymouth
Plymouth, UK
bogdan.ghita@plymouth.ac.uk

Abstract—Ad hoc networks suffer from unfairness when multiple flows exist in the same interference region. The unfair channel distribution among competing flows is due to the random exponential back off algorithm used by the IEEE 802.11 Medium Access Control (MAC) protocol. The protocol can lead to unfair channel utilization among different flows, as some of the nodes may capture the channel for a long period leaving the competing nodes to starve. This paper introduces a novel algorithm named Transmission Rate Control through Acknowledgements Feedback (TRCAF), which controls the channel resources distribution through feedback, included in MAC layer acknowledgements, about the status of the network. The signaling determines whether the intermediate nodes carry on transmitting or stop in order to help the starving nodes to gain access to the channel. In addition, the TRCAF algorithm integrates a scheme that improves fairness in situations where there is no information about the status of the network through monitoring the severity of packet collisions that nodes experience. Simulation results show that the TRCAF algorithm achieves 99% fairness, in contrast to the standard MAC protocol that achieves only 33% under comparable overall utilization of the channel resources.

Keywords: MAC 802.11; fairness; channel utilization; exponential back off algorithm.

I. INTRODUCTION

Improving quality of service in wireless ad hoc networks has been a topic of extensive research for many years; despite their advantages in terms of infrastructure setup, ad hoc networks suffer from unfair bandwidth distribution among competing flows due to the single channel access, interference, hidden nodes and the absence of a central station to manage the network. To solve the unfairness problem, the IEEE802.11 [1] MAC protocol uses the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) [2] protocol to reduce collisions by sensing the channel before each transmission. If the channel is not idle, the node backs off transmission by a period specified by the Exponential Back off Algorithm [3] implemented in the Distributed Coordination Function (DCF) [4] and schedules its transmission by updating its Network Allocation Vector (NAV) with the transmission time required specified in the RTS/CTS messages when the competing nodes are within the same transmission range of

each other, hence, the nodes can hear each other's transmission and schedules them accordingly. On the other hand, if the nodes are in the interference range of each other, updating NAV is not possible as the RTS/CTS message cannot be read and the nodes may transmit at the same time, leading to collisions which exponentially increase the back off time. This leads to unfair channel distribution as the MAC protocol would always favor the last successful transmission, while the other nodes are backing off.

Many researchers [5-10] have examined the unfair bandwidth allocation and proposed schemes to improve fairness in ad hoc networks. Researchers in [5] investigated the limitations of the exponential back off algorithm in the presence of high contention and proposed to increase the contention window size (CW) linearly as opposed to exponentially based on the node's transmission and reception rate over a time interval. However, researchers in [6] calculated the successive back off time as the product of the previous back off time, its log and a time slot in order to prevent increasing the contention window size exponentially. Researchers in [7] presented a fair bandwidth distribution scheme among TCP flows by varying the queue output rate at the network layer according to the severity of the contention experienced by the network every time a packet is passed from the network layer to the MAC layer; although it provided an improvement in terms of fairness, the proposed scheme experienced more than 11% of throughput loss. The concept of authority and ordinary nodes is introduced in [8] in addition to a Contention Window Based Fairness Back off algorithm, which favors the authority nodes to access the channel while limiting the ordinary nodes from accessing the channel. The ordinary nodes only gain access to the channel after a number of successive transmission failures. The proposed scheme also ensures that no node would be left to starve. In [9], a neighborhood Random Early Detection (RED) algorithm was implemented at the network layer, requiring no modification to the MAC protocol, in order to detect contention and improve fairness. The approach required each node to monitor its queue size and broadcast it through network congestion notification (NCN) control packets to the nodes within its transmission range to decide whether to drop packets from the queue and ease contention. In [10], the researchers proposed a

fair share estimation algorithm of the channel resources between nodes sharing the same channel. Each node in the network estimates the amount of channel resources that is being assigned to other nodes based on how many packets they have transmitted and then modifies their contention window size according to a predefined fairness metrics in order to achieve the desired fairness. The researchers also noticed that the proposed algorithm does sacrifice some throughput in order to achieve an acceptable level of fairness.

The rest of the paper is organized as follows: in section II, a detailed description of the Transmission Rate Control through Acknowledgements Feedback (TRCAF) algorithm is provided, followed by an evaluation of the performance of the proposed algorithm through simulations in section III. Finally, a conclusion and future work is provided in section IV.

II. TRANSMISSION RATE CONTROL THROUGH ACKNOWLEDGEMENTS' FEEDBACK (TRCAF) ALGORITHM

The MAC 802.11 does not distribute the channel resources fairly among multiple flows that compete over the same channel resources. Scheduling transmissions is not possible in scenarios where the competing nodes cannot communicate or hear each other's transmissions leading to collision and unfair bandwidth distribution. Fig. 1 shows 9 nodes forming three flows competing with each other over a single shared channel at the receiving nodes (n_6 ; n_7 ; n_8) which are two hops away from the sending nodes (n_0 ; n_1 ; n_2). The source nodes are outside the interference range of each other and the intermediate nodes (n_3 ; n_4 ; n_5) are within interference range but outside the transmission range. Finally, the destination nodes (n_6 ; n_7 ; n_8) within transmission range of each other. Based on the topology design, if (n_0 ; n_1 ; n_2) have data to transmit to their intermediate nodes (n_3 ; n_4 ; n_5) respectively, the channel is always idle and the nodes proceed with the transmission leading to collisions at their intermediate nodes. The intermediate nodes would not be able to update their Network Allocation Vector as the RTS/CTS messages cannot be read by the intermediate nodes as they are outside transmission range. In this case, the nodes that suffer collisions would randomly choose a back off time and exponentially increase it every time a collision occurs. On the other hand, the nodes that did not experience collisions would always have higher chances of accessing the channel than the nodes that backed off their transmissions. This leads to severe unfairness among the competing flows as the standard MAC protocol does not have any mechanism that tackles the unfair bandwidth sharing in scenarios as the one presented in Fig. 1.

In order to alleviate the unfair bandwidth distribution among the competing flows the Transmission Rate Control through Acknowledgements Feedback (TRCAF) algorithm is proposed. The Algorithm is implemented as an enhancement to the MAC protocol and is triggered in case DCF fails to fairly distribute the channel resources among competing flows. Each time a node hears a transmitted packet that is not destined for itself, the source address is extracted and compared to the addresses of its intermediate hops. If the address is not in the list, then it is of a competing node. The competing node's address is stored and every time a packet is

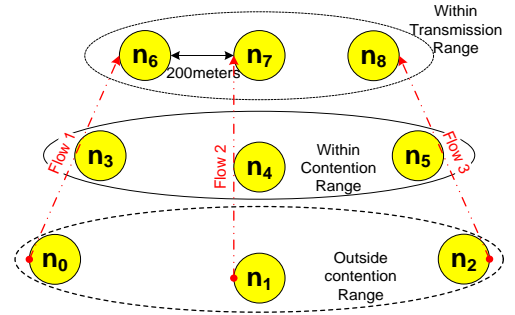


Figure 1. Topology 1

heard from that node a counter is incremented and reset upon the expiry of a time interval. According to the topology shown in Fig. 1, node n_6 would hear the acknowledgement packets sent by nodes n_7 and n_3 but node n_3 is its intermediate node so the address of node n_7 is stored by n_6 and the counter is incremented every time a packet is heard by n_6 from n_7 . On the other hand, node n_7 would be monitoring the acknowledgement packets sent by nodes n_6 and n_8 and would have two counters of how many packets have been heard from each node. Next, a moving average for each node in the list based on the number of packets counted is calculated every time the interval time expires as per equation (1). Initially, the parameters were set to $\alpha = \beta = 0.5$ but following from the preliminary tests, α was set to 0.9 and β was set to 0.1 to avoid sharp changes to the moving average. Also, the waiting time (Δt) was set to 0.2 second.

$$avg_{i+1} = avg_i * \alpha + \frac{count_{i+1}}{waitingTime} * \beta \quad (1)$$

Finally, if the moving average is below the threshold set to 1 established from observing the moving averages of the nodes, then the competing node that the moving average corresponds to is starving and when the node acknowledges the reception of the data frame a MAC layer acknowledgement is sent with a flag in its header to indicate contention. Upon the reception of the acknowledgement packet by the intermediate node, the intermediate node backs off its transmission in order to allow the starving nodes to capture the channel, transmit packets and achieve fairness. In addition, the TRCAF algorithm also employs a mechanism that tackles unfairness in scenarios where there are no competing nodes within the transmission range and the transmitted packets encounter collisions. If the TCP packets encounter RTS messages collision rate is higher than a predefined threshold then this implies that there are nodes competing for the channel even though they cannot be detected as they are outside the transmission range but within interference range of each other. RTS messages were chosen as they are control packets and transmitted with less power, hence, RTS messages gets dropped when collided with data packets. In such case, the TRFAC counts the number of collisions for each packet type such as control packets (RTS/CTS/ARP/AODV) and Data (TCP/ACK) over a time interval and calculates a moving average as per equation (1). If the moving average for data packets is high then the node is

greedy and is penalized by starting its back off timer. Also, if the moving average for control packets is high then it is an indication that the node is starving and its back off timer is disabled.

A. TRCAF Pseudo Code

```

/* Terminology
pkt_rcv = packet received
PktType = DATA || ACK || AODV || RTS || CTS
Δt = 0.2 second
cwnd = Contention Window
BK = Back Off Algorithm
coll = collisions
*/
while(simulation)
do
for node i=1 to N
case received packet:
if(pkt(dst) != node_i(addr))
CompetingArray[j].addr = pkt_rcv(src)
// Source address is my competing node's
// address or my next hop address
end if
case sent:
if(pkt(dst) == CompetingArray[j])
// The address is my next hop address
// so remove from myArray in order not
// to influence on decisions
delete CompetingArray[j]
end if
while (timer <= Δt)
if(relay packet)
CompetingArray[j].counter++
end if
end while
update moving_average
if (moving_average[i] < β1)
// Competing node is starving
MAC_Acknowledgement.flag = 1
else
MAC_Acknowledgement.flag = 0
end if else
if (MAC_Acknowledgement.flag)
BK.start(cwnd)
end if
while(timer <= Δt)
coll[RTS]++
avg[RTS]=(avg[RTS]*α+(coll[RTS]/Δt)*β)
if(competing node !found && avg[RTS]>β2)
coll[PktType]=0
while(timer <= Δt)
coll[PktType]++
end if
avg[PktType]=(avg[PktType]*α+
(coll[PktType]/Δt)*β)
if (avg[(TCP||ACK)]>β3)
// Node is greedy
Drop(packet)
Backoff.start(cwnd)
end if
if (avg[(RTS||CTS||AODV||ARP)]>β4)
//Node is starving
//Reward-cancel exponential backoff
algorithm
if(BK.on)
BK.stop
else

```

```

send(packet)
end if
end if
done

```

In the program, the thresholds β_1 , β_2 and β_3 were set to 0.1 and the contention window was set to 831 as the tests showed that the scheme fairly distribute the channel resources at these values.

III. SIMULATION RESULTS AND TRCAF EVALUATION

The TRCAF algorithm was implemented in the network simulator ns2 as an enhancement to the MAC protocol in order to evaluate its performance. To investigate its efficiency, a radial fixed structure to cover half a circular area (similar to the one shown in Fig. 1) was used as the basic topology, given its potential for the MAC 802.11 protocol to perform extremely unfair. All the simulations have been run for 500 seconds and the competing flows start transmitting at the same time.

Fig. 2 shows that the standard MAC protocol is unfair as the pair ($n_1;n_7$) dominates access to the channel and maintain a transmission speed of about 85000 B/s leaving the other two pairs to completely starve, even though the three flows started transmission at the same time. The source nodes transmitted their packets at the same time assuming that the channel is free as they were outside the contention range of each other. When the packets reached the intermediate nodes, collisions occurred and the intermediate nodes could not synchronize their transmissions as they were outside the transmission range of each other and randomly backed off. Pair ($n_1;n_7$) did not experience collisions, hence, did not back off transmission and gained access to the channel. The other two pairs ($n_0;n_6$) and ($n_2;n_8$) were unable to access the channel as whenever their intermediate nodes have packets to transmit they find that pair ($n_1;n_7$) is occupying the channel.

However, the performance of the TRCAF shown in Fig. 3 fairly distributes the channel resources among the three competing flows. As the destination nodes (n_6 ; n_7 ; n_8) are within transmission range, they can hear each other's packets. According to the TRCAF algorithm whenever a node notices

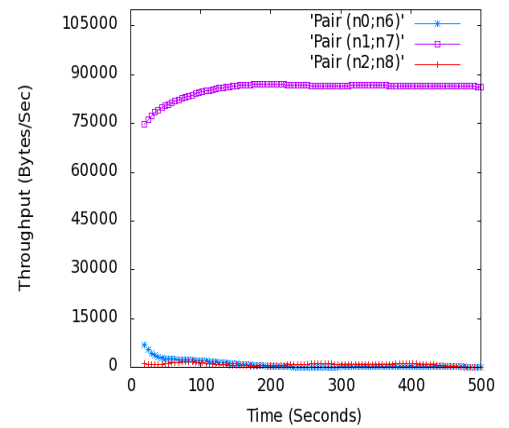
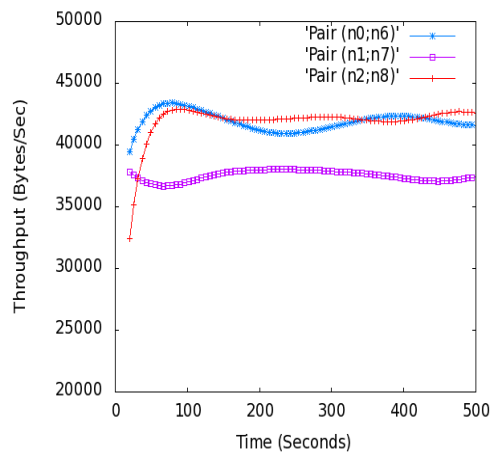
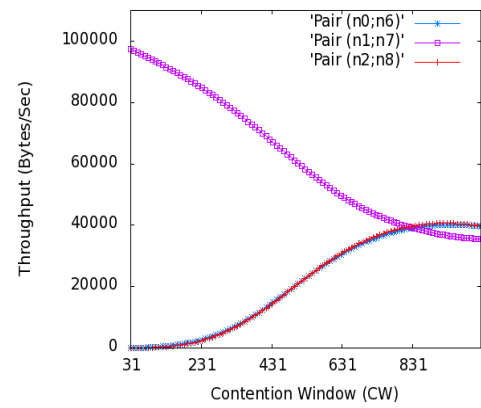


Figure 2. Standard MAC Protocol Performance



that its competing nodes are starving, it sends a MAC acknowledgement with a flag to indicate contention. This flag instructs the intermediate nodes to back off transmission and allow the starving nodes to pick up transmission. The pairs $(n_0; n_6)$ and $(n_2; n_8)$ achieve on average 42000 B/s throughput and the pair $(n_1; n_7)$ achieve around 38000 B/s which is about 10% less than the other two pairs due to node n_7 having two nodes to compete with and therefore having to back off further in order to allow the other two nodes to increase their transmission and achieve fair bandwidth distribution among the competing nodes.



In order to establish the correct back off time, the contention window size (CW) was incremented from 31 to 1024. Fig. 4 shows that as the contention window is increased; the difference between the throughputs achieved by each flow gets narrower and a fair bandwidth distribution is achieved when the contention window size is 831 at which point the three pairs ($n_0;n_6$), ($n_1;n_7$) and ($n_2;n_8$) achieve throughput speeds of 41916 B/s, 37482 B/s and, respectively, 41970 B/s. This contrasts significantly with the standard MAC protocol where the pairs ($n_0;n_6$), ($n_1;n_7$) and ($n_2;n_8$) achieved 834 B/s, 85656 B/s and 834 B/s respectively as obtained from the simulation shown in Fig. 2. Thus, the standard MAC protocol achieves 87324 B/s total throughput, whereas, the TRCAF algorithm achieves a total throughput of 121368 B/s. Therefore, not only the TRCAF algorithm fairly distributes the channel resources among the three flows but also achieves 28% higher aggregate throughput than the standard MAC protocol.

The plot in Fig. 5 shows a fairness index based on Jain's fairness equation (2) [11].

$$\text{Fairness} = \frac{(\sum x_i)^2}{n \cdot \sum x_i^2} \quad (2)$$

Where n is the number of flows and x_i is the average throughput achieved by each flow.

According to Jain's equation the worst case where the distribution of channel resources is unfair is 0.33 and the best

consecutive nodes in each flow is 200 meters as well as nodes n_6 , n_7 and n_8 where the network is contention. The source and destination nodes of the three flows are outside the interference range and the intermediate nodes from the senders and receivers are within the interference range but outside the transmission range of each other.

In order to evaluate the performance of the TRCAF algorithm in comparison to the standard MAC, simulations have been run for 500 sec in ns2. The results of the simulation are presented in Fig. 7 and 8. Fig. 7 shows that the MAC protocol does not fairly share the bandwidth between the three competing flows as the pairs $(n_0;n_{12})$, $(n_1;n_{13})$ and $(n_2;n_{14})$ achieved 138 B/s, 39606 B/s and 738 B/s respectively. However, Fig. 8 illustrates that the TRCAF algorithm fairly shares the bandwidth between the three competing flows and the pairs $(n_0;n_{12})$, $(n_1;n_{13})$ and $(n_2;n_{14})$ achieve 13554 B/s, 16056 B/s and 12048 B/s respectively. Also, applying these results to Jain's fairness equation (2), the fairness index is 0.34 for the MAC protocol and 0.98 for the TRCAF algorithm. This demonstrates that the TRCAF algorithm does achieve fairness.

The TRCAF algorithm is designed to be adaptable to a variety of network topologies. In the next set of simulations, the TRCAF algorithm is tested on a further two topologies which have been selected in order to demonstrate the performance of the TRCAF algorithm in comparison to the standard MAC 802.11 protocol. The competing flows in

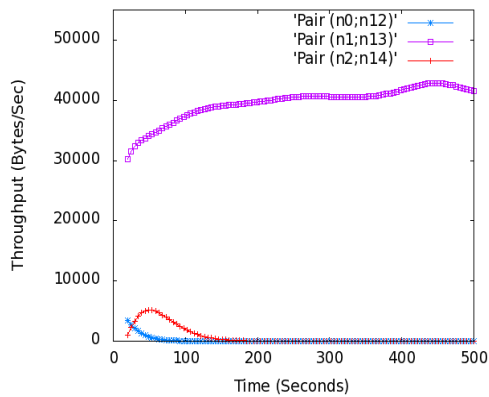


Figure 7. Standard MAC Protocol Performance

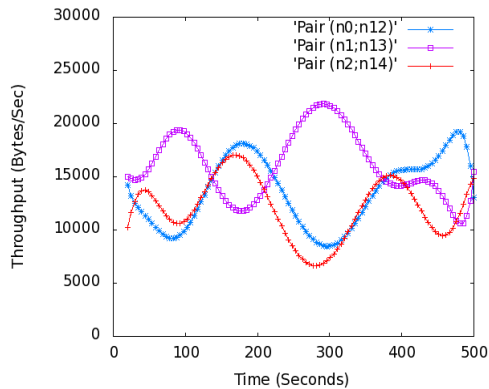


Figure 8. TRCAF Algorithm Performance

topology 3 are outside the transmission range but inside the interference range of each other. Topology 4 includes competing flows consisting of four hops each but with both possibilities included; in other words some of the nodes in the competing flows can hear each other and monitor the packets they have transmitted and some of the nodes are within interference range, hence, they have to rely on monitoring their collisions rate in order to establish if they are greedy and halt transmission in order to alleviate contention and achieve fair bandwidth distribution and at the same time the starving nodes would be more aggressive by not backing off after a collision as would the MAC 802.11 protocol when acquiring access to the channel.

Fig. 9 presents three competing flows where there is a single hop to destination. The distance between the three flows is 400 meters; hence, the flows are within contention range of each other and the distance between communicating nodes is 200 meters.

The graph in Fig. 10 shows how the MAC 802.11 protocol performs for the scenario presented in Fig. 9. The pair $(n_1;n_4)$ is completely dead and only achieve 467 B/s over the 500 seconds of the simulation due to having to compete with two flows at each side. On the other hand, as the pair $(n_1;n_4)$ is starving, the pairs $(n_0;n_3)$ and $(n_2;n_5)$ are outside the interference range of each and do not compete against each other. Therefore, the pairs $(n_0;n_3)$ and $(n_2;n_5)$ make the most of the channel resources and achieve 186086 B/s and 185943 B/s respectively. This is an extreme scenario where the MAC 802.11 protocol cannot distribute the channel resources fairly.

The graph presented in Fig.11 demonstrates the performance of the TRCAF algorithm on topology 3. Since, the nodes cannot hear each other's packets and count them, they start calculating their collisions rate in order to determine if there are competing nodes within interference range and in parallel the starving nodes stop backing off after collisions and

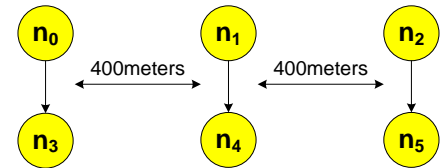


Figure 9. Topology 3

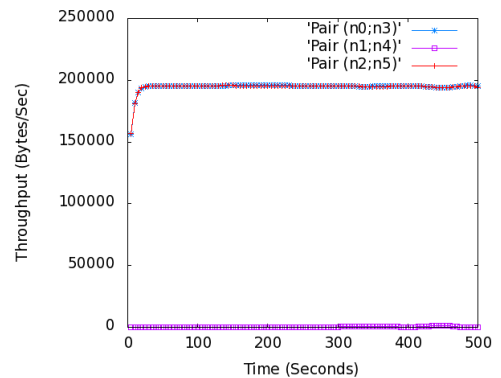


Figure 10. Standard MAC Protocol for Topology 3

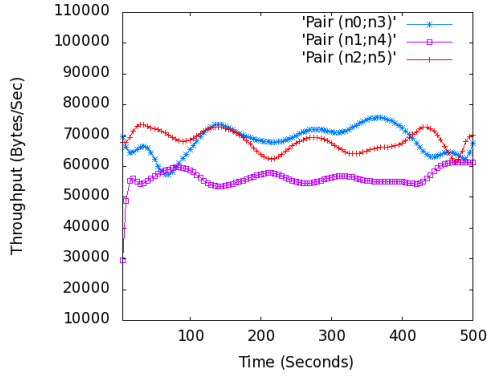


Figure 11. TRCAF Algorithm Performance for Topology 3

become more aggressive upon their collision's moving average crosses a preset threshold. In contrast to the standard MAC protocol, the TRCAF fairly shares the bandwidth among the three flows and the pairs $(n_0;n_3)$, $(n_1;n_3)$ and $(n_2;n_5)$ achieve 68592 B/s, 56242 B/s and 68262 B/s respectively.

The difference between topology 4 illustrated in Fig. 12 and previous topologies is that it includes the two situations in the same scenario i.e. the nodes can hear each others transmissions as in topology 1 and where nodes have to rely on their collisions rate as in topology 3. There are three competing flows where each flow consists of four hops from the source to destination. The senders are within interference distance of each other and the receivers are within transmission range.

Fig. 13 demonstrates that the MAC 802.11 is unfair as the pairs $(n_{12};n_6)$ and $(n_{14};n_8)$ obtain 37962 and 39012 B/s respectively and the pair $(n_{13};n_7)$ obtain 1998 B/s. These results are similar to that of topology 3 as the pair $(n_{13};n_7)$ suffering from having to compete with the other two pairs at the same time.

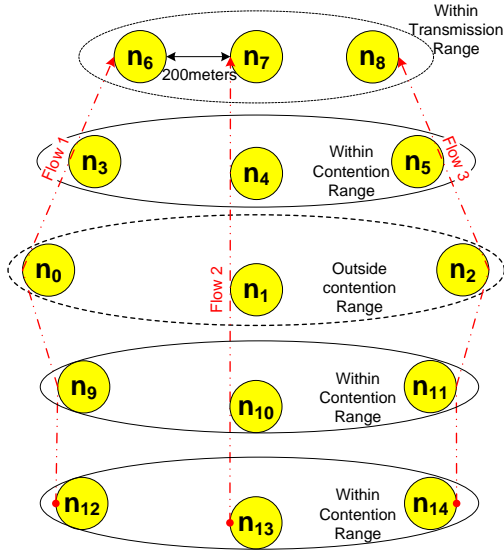


Figure 12. Topology 4

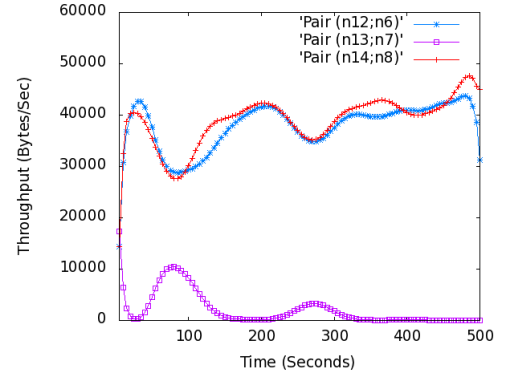


Figure 13. Standard MAC Performance for Topology 4

Fig. 14 shows a graph of the packet that the competing receivers $(n_6;n_7;n_8)$ hear about each other in the case of standard MAC protocol where moving average of n_6 and n_8 shows the packets that n_7 hears and that are for nodes n_6 and n_8 and moving average of n_7 is the average of packets heard by nodes n_6 and n_8 . The graph also demonstrates that nodes n_6 and n_8 can tell from the moving average of n_7 that it is starving as Fig. 13 demonstrates.

The performance of the TRCAF algorithm for topology 4 is shown in Fig. 15. The pairs $(n_{12};n_6)$, $(n_{14};n_8)$ and $(n_{13};n_7)$ achieve 19620 B/s, 18966 B/s and 19308 B/s respectively.

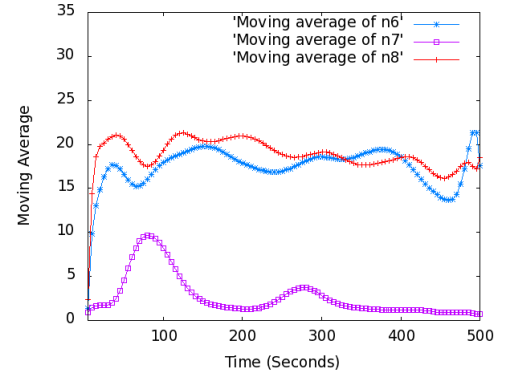


Figure 14. Packets Counted by Competing Nodes for Standard MAC Protocol for Topology 4

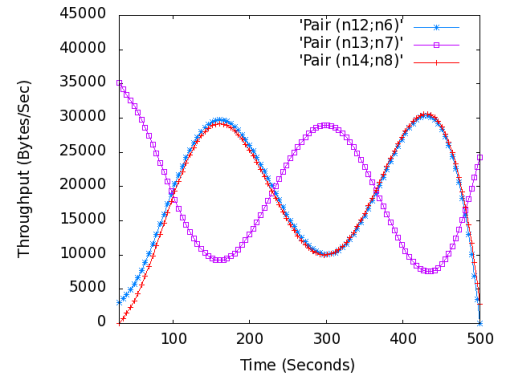


Figure 15. TRCAF Algorithm Performance for Topology 4

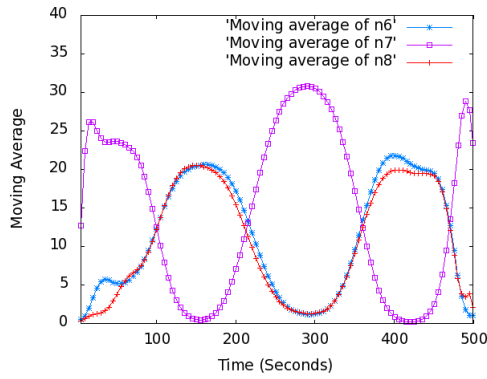


Figure 16. Packets Counted by Competing Nodes for TRCAF Algorithm for Topology 4

Therefore, as opposed to the standard MAC protocol, the TRCAF algorithm distributes the channel resources fairly among the competing flows even in complex scenarios. However, the TRCAF algorithm slowly responds to the changes and decision being made due to the decision having to propagate four hops along the chain in order to reach the senders.

Fig. 16 shows how the moving average heard by the competing nodes matches the throughput achieved by each node. When node n_7 is starving the moving average of the competing nodes n_6 and n_8 is high and as the node n_7 picks up transmission the moving average of its competing nodes n_6 and n_8 drops. However, the TRCAF algorithm struggles to maintain the same moving average of the number of packets heard by the competing nodes throughout the duration of the simulation.

IV. CONCLUSION

This paper proposes an algorithm that aims to provide fair bandwidth sharing among the competing flows over the same channel by signaling contention to senders who, in turn, self-police to allow starving nodes to transmit. The algorithm relies on each node being able to dynamically differentiate between its communicating and competing nodes. The TRCAF algorithm does not create any packets overhead as opposed to previous research in the field; each node monitors how many packets its competing nodes have received and, if the competing nodes are starving, the nodes set a flag in the MAC acknowledgement to instruct its communicating node to stop transmitting and allow the starving nodes to receive and send packets. In addition, where it is not possible to hear the packets that the competing nodes send, the TRCAF algorithm switches to monitoring the collision rate that each node experiences. If the collision rate is greater than a predefined threshold; then the node becomes greedy and cancels its exponential back off algorithm. Also, if the collision rate is less than a predefined threshold then the node establishes that

it has to halt transmission and allow the competing node to transmit. Four topologies were chosen to evaluate the performance of the TRCAF algorithm where the standard MAC protocol proves to be inefficient in terms of sharing the channel resources fairly among the competing flows. However, it was noticed that in the second and fourth topology the TRCAF algorithm struggled to maintain fairness for each moment in time and was slow in responding to the decisions being taken. This was due to having long flows each consisting of four hops and leading to the decision having to propagate for four hops to reach the senders. In future work, the TRCAF algorithm would be improved to react quickly to the decisions taken in the presence of flows consisting of more than four hops as well as evaluating the algorithm in the presence of mobile node. In addition, the TRCAF algorithm would also be evaluated in randomly generated topologies.

REFERENCES

- [1] IEEE, Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. IEEE Standard 802.11 (1999)
- [2] Xin, W., Kar, K.: Throughput modeling and fairness issues in CSMA/CA based ad-hoc networks. In: 24th Annual Joint Conference of the IEEE Computer and Communications Societies (2005)
- [3] Vukovic, I.N., Smavatkul, N.: Saturation throughput analysis of different backoff algorithms in IEEE 802.11. In: 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, vol. 3, pp. 1870--1875 (2004).
- [4] Eshghi, F., Elhakeem, A.K., Shayan, Y.R.: performance evaluation of multihop ad hoc WLANs. Communications Magazine, IEEE, vol. 43, issue 3, pp. 107--115 (2005).
- [5] Berqia, A., Angoma, B., Mikou, N., Dehbi, Y.: Fairness and QoS in ad-hoc networks. in Vehicular Technology Conference. VTC Spring 2008. IEEE (2008)
- [6] Manaseer, S.S., Ould-Khaoua, M., Mackenzie, L.M.: Logarithmic based backoff algorithm for MAC protocol in MANETs. In: proceedings of CSIT2006. Jordan (2006)
- [7] Luqing, Y., Winston, K.G.S., Qinghe, Y.: Improving fairness among TCP flows crossing wireless ad hoc and wired networks. In: 4th ACM International Symposium on Mobile Ad Hoc networking and Computing, ACM: Annapolis, Maryland, USA (2003)
- [8] Zhang, D., Zhang, B., Lu, D. : Contention window based fairness backoff algorithm in ad hoc networks. IEEE Int. Conference Neural Networks and Signal Processing, Zhenjiang, China (2008)
- [9] Kaixin, X., Mario, G., Lantao, Q., Yantai, S.: Enhancing TCP fairness in ad hoc wireless networks using neighborhood RED. In: 9th Annual International Conference on Mobile Computing and Networking. 2003, ACM: San Diego, CA, USA (2003)
- [10] Bensaou, B., Yu, W., Chi Chung, K.: Fair medium access in 802.11 based wireless ad-hoc networks. In: Mobile and Ad Hoc Networking and Computing, MobiHOC (2000)
- [11] Jain, R., Chiu, D.M., Hawe, W.: A quantitative measure of fairness and discrimination for resource allocation in shared systems. DEC Research Report TR-301 (1984)