2012

Improving Fairness and Utilisation in Ad Hoc Networks

Arabi, Mohamed

http://hdl.handle.net/10026.1/1167

http://dx.doi.org/10.24382/4537
University of Plymouth

IMPROVING FAIRNESS IN AD HOC NETWORKS BASED ON NEIGHBOURS TRANSMISSION RATE

Mohamed Arabi and Bogdan Ghita

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

ABSTRACT

Mobile Ad Hoc Networks face great challenges due to the absence of an infrastructure to manage the network, an access to a single channel by multiple nodes, interference and the limited transmission range by the wireless media. Therefore, achieving fairness in Mobile Ad Hoc Networks is a difficult task. This paper discusses a specific Mobile Ad Hoc Network topology which leads to unfair bandwidth distribution and proposes a novel Neighbouring Transmission Rate Control (NTRC) algorithm, based on monitoring the transmission speed of the neighbouring nodes, which aims to optimise network fairness through identifying and helping starving nodes to increase their transmission rates. The proposed method is tested in a simulation environment and, according to the results; it provides a balanced distribution of network resources amongst the competing nodes. The proposed algorithm achieves 99% fairness according to Jain's fairness equation as opposed to 33% for the standard MAC protocol.

KEYWORDS

MAC 802.11, Fairness, Channel utilisation, Contention Window, Binary Exponential Back-off Algorithm, Jain's Fairness Index

1. INTRODUCTION

The flexibility and the absence of a base station in Mobile Ad Hoc Networks can lead to severe degradation in the network's performance in terms of collisions, throughput and, implicitly, fairness. From an architectural perspective, the MAC 802.11 protocol, with its Distributed Coordination Function (DCF) [1], aims to eliminate packet collisions and to reduce the hidden node phenomenon [2]. This is achieved by the use of the Binary Exponential Back-off (BEB) Algorithm [3]. When the communicating nodes transmit data to nodes that are outside the transmission range of each other and within interference range, the receiving nodes are not capable of scheduling their transmissions and the BEB algorithm becomes inconsistent due to the use of exponential contention window and leads to unfair channel utilisation among the competing nodes of different flows.

In order to reduce collisions between nodes in the same transmission range, the Distribution Coordination uses the four way handshake RTS/CTS/DATA/ACK before any data transmission. When a node has a packet to transmit it senses the channel for an interval specified in the Distributed Inter Frame Space. If the channel is found to be busy, the node defers its transmission by a duration calculated using the BEB algorithm. If the channel is sensed idle, the node transmits an RTS message to the next hop in the route to destination. The nodes that are in the same transmission range also hear this RTS and update their Network Allocation Vector with the required transmission duration specified in the RTS message in order to prevent transmitting while there are still another transmission taking place. On the other hand, the nodes that are outside the transmission range but within the interference range cannot extract this information and may transmit while the channel is busy which leads to packet collisions. In the event of a collision the BEB algorithm exponentially increases the contention window which exponentially increases the chances of the node not capturing the channel. In contrast, the node that did not experience collisions dominates access to the channel leaving other nodes to starve.

Researchers in recent years have highlighted this problem and improving fairness in Mobile Ad Hoc Networks has increasingly become a hot topic. The authors of [4] illustrate that, in a number of scenarios, the 802.11 protocol does not provide fair access to the channel between the competing flows if the flows are outside the transmission range but within interference range. Therefore, the nodes cannot update their Network Allocation Vector with the correct time required for the transmission as they can not read the information provided in the RTS messages. In [4], an enhanced MAC protocol was proposed to decide whether to stop transmitting or carry on transmitting, based on the number of collisions that the nodes encounter. Other papers such as [5] and [6] also demonstrated that the BEB lead to unfair bandwidth distribution and proposed changes to the back-off algorithm itself. In paper [5], the authors proposed a logarithmic increase of the contention window based on how many bytes the nodes sent and received. However, the authors of [6] increased the contention window by the first back off duration multiplied by its log and by a time slot after the second collision. In addition, researchers in [7] proposed a Contention Window Based Fairness Backoff algorithm that introduces the concept of authority and ordinary nodes where authority nodes are granted access to the channel with more priority in contrast to ordinary nodes that access the channel only after N successive failures. The authors of [8] observed the use of a fixed extended inter-frame space (EIFS) duration between all competing nodes that can lead to packet collisions if the nodes happen to back off by the same duration and proposed to calculate the EIFS based on the length of the frames sensed in the network. On the other hand, rather than looking for a solution at the MAC layer, other researches proposed solution to the unfairness of the 802.11 MAC protocol at the Transport and Network Layers. In [9], the authors proposed a method to control the queue output rate in order to prevent TCP from reaching its maximum congestion window so that the node does not become greedy and prevent its competitors from transmitting. Despite the efforts the researchers have made in order to tackle the unfairness of the MAC protocol, there are limitations and drawbacks to their work in terms of throughput reduction, collisions and delay. This paper tackles the unfairness problem and proposes a novel algorithm that improves the fair bandwidth allocation between the competing nodes through monitoring the transmission speed of the competing nodes then deciding on whether to penalise greedy nodes in order to share the available channel resources fairly.

The rest of the paper is organised as follows: a description of the NTRC algorithm is provided in Section 2. Section 3 presents the simulations carried to achieve the results and an evaluation of the proposed scheme is also provided. Section 4 concludes the paper.

2. PROPOSED SCHEME

The Neighbouring Transmission Rate Control (NTRC) is a novel algorithm that tackles the unfair distribution of channel resources in Mobile Ad Hoc Networks between competing flows by penalising greedy nodes in order to allow starving nodes to transmit and ensuring that the channel resources are distributed fairly among the competing flows in the same network.

The NTRC algorithm is implemented at the MAC layer and allows each node to identify and discover its neighbouring nodes and topology. Every time the node hears a packet not destined for it self, the source address of the packet is extracted and stored in a dynamic array. When the node has a packet to send; the destination address is compared to the addresses stored in the dynamic array. If the address is found in the dynamic array it is then removed as it is the next hop address to destination. The next hop address is removed from the array in order not to include it in the decision making on whether to stop or carry on with transmission. The dynamic array is then left with addresses of only competing nodes. It is important to differentiate between next hop nodes and competing nodes. The next hop nodes are excluded from the neighbouring nodes array because the NTRC algorithm counts the packets that are not destined for itself. If the next hop node was included in the decision making then the intermediate node to destination would always stop transmitting as it assumes that the destination node is starving because the destination node only sends packets back to its intermediate node and which are not counted by the NTRC algorithm. The process of adding nodes to the dynamic array is ongoing as nodes could be mobile and the topology of the network could change at any time and if new nodes are within transmission range and they send packets, they get added to the array. Therefore, the NTRC algorithm takes into consideration the mobility of the nodes in a Mobile Ad Hoc Network and the constant route change. In addition, each time a packet is received a counter of the number of packets heard from the neighbouring nodes is incremented to indicate if they are greedy or starving. Also, nodes that produce no traffic are not assumed to be starving as a node is added to the neighbouring nodes array only when it sends a packet to its next hop in which case it is competing for the channel resources if it is within the same transmission range to other nodes. The NTRC algorithm also implements a moving average which re-evaluates the nodes' behaviour periodically through the use of a timer named waiting time. This moving average is then compared to a threshold to decide whether the node should carry on transmitting or should halt to allow the neighbouring nodes to transmit. The algorithm was designed with assumption that all sending nodes have a packet ready to be transmitted at all time.

As will be shown later, the NTRC algorithm proves to be efficient and improves the networks fairness dramatically by penalising the greedy nodes by backing off transmission when they detect that their competing node's transmission is below a threshold which allows the starving nodes to transmit in fair manner.

2.1. Implementation of NTRC algorithm

The NTRC algorithm uses a moving average consisting of the current new average and the previous average value as per equation (1)

$$avg(i+1) = avg(1) * \alpha + {count(i+1)}/{waitingTime} * \beta \quad (1)$$

Initially, the parameters were set to $\alpha = \beta = 0.5$ but following from the preliminary tests, α was set to 0.9 and β set to 0.1 to avoid sharp changes to the moving average. Also, the waiting time was set to 0.2 second, the threshold was set to 1 and the contention window was set to 1024. The pseudo code of the NTRC algorithm is shown below.

```
while(simulation)
do
if(receive packet not destined for it self)
   extract source address from packet header
       //source address is my neighbour address or my next hop address
       add source address to myArray
end if
if(send packet)
       get packet destination address
       if(destination address == myArray[i])
       // the address is my next hop address so remove from myArray
       // myArray must not include next hop address so that it does not
       // affect on decision making
       Remove next hop address from myArray
       end if
end if
```

```
        while(timer <= waiting time)
              if(relay packet)
                    myArray[i].counter++
              end if
        end while
update moving_average
if (moving_average[i] < threshold)
        // Node is greedy
        //Penalise –start back off algorithm to allow its neighbours to transmit
        Backoff.start(Contention Window)
end if
done
```

In the next section, simulations have been run in order to determine the optimal values of the waiting time, threshold and contention window where the algorithm achieves its best performance in terms of fairness.

3. SIMULATION RESULTS AND PERFORMANCE EVALUATION

The NTRC algorithm has been implemented in the mac-802.11 class of the ns-2 simulator and in order to demonstrate its performance a number of simulations have been run. A radial fixed structure to cover a circular area has been chosen where the performance of the standard MAC 802.11 protocol is extremely unfair. This scenario is presented in Figure 1 below.

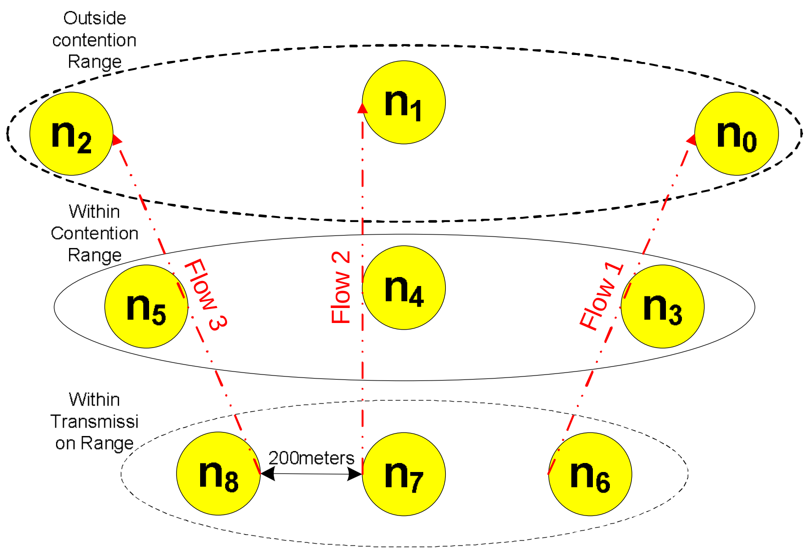

Figure 1. Case Study Illustration

Figure 1 shows three flows competing against each other for the wireless channel. Each flow consists of two hops from source to destination as follows: nodes (n_6, n_3, n_0) make up flow 1; (n_7, n_4 n_1) construct flow 2 and flow 3 consists of nodes (n_8, n_5 n_2). In ns-2, the transmission range is configured to be 250 meters and the interference or contention range is set to be 550 meters. Taking this into consideration, nodes n_6, n_7, n_8 are 200 meters apart hence within transmission range, nodes n_3, n_4 and n_5 are 387 meters apart so outside transmission range of each other but within interference range and finally, nodes n_0, n_1, n_2 are separated by 574 meters which makes them outside the interference range of each other.

In Figure 2 below, the throughput that each flow achieves is plotted in order to show how unfair the 802.11 protocol behaves in such a scenario. The simulation was run for 450 seconds. The pair (n_7;n_1) dominates access to the channel for the duration of the simulation. All transmissions were started at the same time and node n_7 gained access to the channel first because when its packet collided with the packets of the neighbouring nodes n_6 and n_8, it backed off for a shorter period than nodes n_6 and n_8 which allowed node n_7 to request access to the channel while nodes n_6 and n_8 were still waiting for the back off time to expire The other two nodes n_6 and n_8 deferred their transmissions based on the information included in the RTS message which node n_7 sent to its next hop. However, when the packet reached its intermediate node n_4; nodes n_3 and n_5 were unable to schedule their transmissions as they are outside the transmission range of each other. Therefore, they could not schedule when to request access to the channel which led to collisions. For this reason, nodes n_6 and n_8 backed off and kept exponentially increasing their contention window which meant their probability to access the channel reduced too.

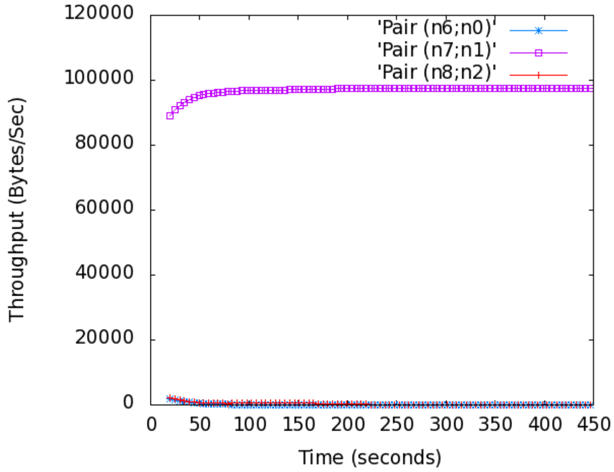

Figure 2. Standard MAC 802.11 performance

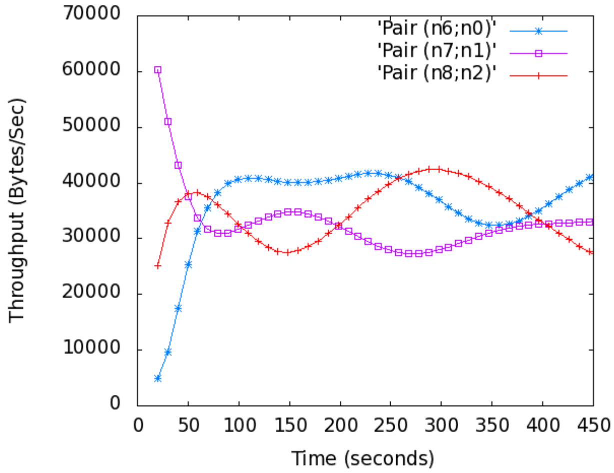

Figure 3. The performance of NTRC Algorithm in terms of Fairness

Figure 3 above demonstrates the performance of the NTRC algorithm in terms of throughput and channel utilisation. The three competing flows do share the bandwidth available to them fairly. At the beginning of the simulation, the nodes in the network are not aware of their competing nodes or neighbours. However, after about 10 seconds of simulation time, they learn dynamically the topology of the network and start to share the channel fairly. The transmission speed of pair (n_7;n_1) starts from about 60000 bytes/second and is gradually reduced to about 32000 bytes/second. The transmission speed of pair (n_6;n_0) at the beginning of the simulation is about 5000 bytes/second and gradually increased to about 40000 bytes/second. Also, pair (n_8;n_2)

does increase its transmission speed from about 25000 bytes/second to about 37000 bytes/second. The graph also shows that it takes approximately 50 seconds to evaluate each other's transmission speed, after which the nodes share the bandwidth fairly between them as required by the NTRC algorithm.

In order to evaluate the fairness of the NTRC algorithm and the standard MAC protocol, Jain's Fairness equation [10] was used and is as per equation (2):

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

Where: $\boldsymbol{n}$ is the number of flows and $\boldsymbol{x_i}$ is the aggregate throughput achieved by each flow.

The aggregate throughput that each flow achieves with the standard MAC protocol is *(144.32; 97011.16; 288.32) bytes/sec* for pair (n_6;n_0), pair (n_7;n_1) and pair (n_8;n_2) respectively. Applying Jain's equation to these values the resulting fairness index is *0.336* which shows that the MAC 802.11 is unfair as the best case is 1 and the worst case is 1/3 as there are three competing flows according to Jain's Fairness index. On the other hand, the aggregate throughput that pair (n_6;n_0), pair (n_7;n_1) and pair (n_8;n_2) achieves with the NTRC algorithm is *(38289.32; 31806.16; 34041.32) bytes/sec* respectively and applying Jain's equation to these values the outcome is 0.994 which shows that the NTRC algorithm does achieve fairness between the competing nodes as the worst case is 0.33 and the best case is 1 according to Jain's Fairness index.

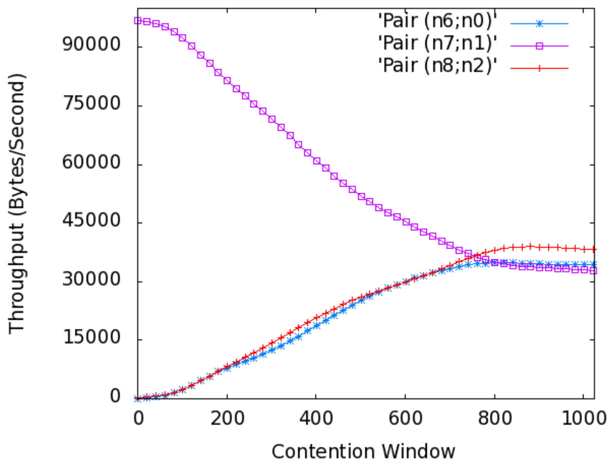

Figure 4. The impact of CW on Fairness

Figure 4 shows the throughput that each flow achieves for each value of the contention window. It illustrates that the three competing flows achieve approximately the same transmission speed of about 37000 bytes/second when the contention window size is greater than 700.

Figure 5 below shows a 3-dimensional plot of Jain's fairness index versus threshold values and contention window values. The results were obtained by setting the waiting time value and varying the contention window and the threshold values.

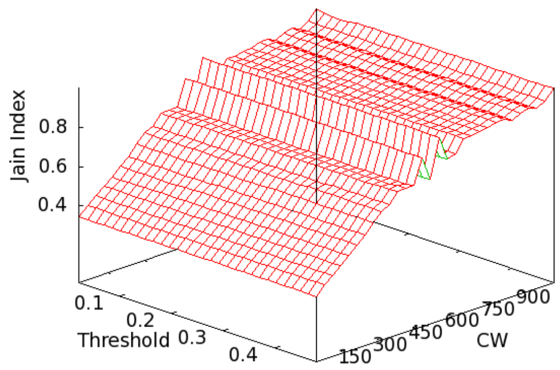

Figure 5. The impact of CW and Threshold on Fairness

Varying the threshold values does not have an impact on fairness because as the channel can only be accessed by one node at a time the moving average is always very close to zero. Therefore, the moving average was always smaller than the threshold values that were chosen hence the chosen threshold values did not have an impact on fairness. However, varying the contention window does influence fairness. The network was fair when the contention window was set to 1024. At this value, the fairness index is very close to 1 which is the best case in Jain's index.

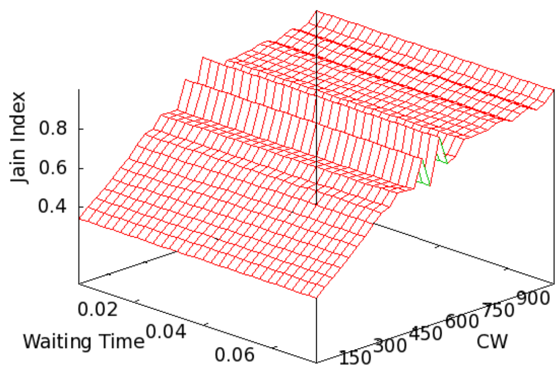

Figure 6. The impact of CW and Waiting Time on Fairness

In Figure 6, the same observations are made as with Figure 5. The results were obtained through fixing the threshold value and varying the contention window and waiting time values in the NTRC algorithm. Then, Jain's Fairness equation was applied to the aggregate throughput achieved from each flow and the results were plotted as shown in Figure 6 above. Again, varying the waiting time values did not have an impact on fairness. In addition, the fairness index is very close to 1 when the contention window is set to its maximum value of 1024.

4. CONCLUSION AND FUTURE WORK

In this paper, a novel algorithm named Neighbouring Transmission Rate Control (NTRC) was proposed to improve fairness. A study case was taken into consideration in order to demonstrate the performance of the NTRC algorithm in comparison to the standard MAC 802.11 protocol.

The NTRC algorithm dynamically detects its neighbours and distinguishes them from the next hop nodes. It also fairly distributes the channel access between the competing nodes by monitoring if a node's neighbours have transmitted any packets during a time interval by calculating a moving average periodically to stay tuned with the state of the network. If the moving average is below a threshold then the node stops transmitting to allow its starving neighbours to transmit. The NTRC algorithm achieves 99.4% fairness based on Jain's fairness equation where the standard MAC protocol only achieves 33.6%. Although, the NTRC algorithm does take into account new neighbours that have come within transmission range when they transmit a packet, it does not take into consideration if the neighbours leave the transmission range of the node. In addition, the algorithm was designed on the basis that there are data to transmit all the time. However, in situations where there is bursty traffic, the NTRC algorithm does not adapt to the changes on time as the moving average slowly reduces by then the node would have stopped transmitting. Also, the NTRC algorithm could be improved to include scenarios where next hop nodes do route traffic for other flows or in other words where next hop nodes are also competing nodes for the channel resources at certain times.

In the future, the unfair channel distribution of the MAC 802.11 in a scenario where the sending nodes are outside the interference range and receivers are within transmission range of each other will be improved. This will be done through feedback from the receiving nodes to the sending nodes in order to learn about the state of the network and whether to stop transmitting in order to alleviate the unfair distribution of channel resources.

REFERENCES

[1] Eshghi, F., Elhakeem, A.K. & Shayan, Y.R. (2005) "Performance evaluation of multihop ad hoc WLANs", *Communications Magazine, IEEE,* Vol. 43, No. 3, pp107-115.

[2] Yoo, J. & Kim, C. (2005) "On the hidden terminal problem in multi-rate ad hoc wireless networks", *in ICOIN'05, Springer,* Heidelberg.

[3] Vukovic, I.N. & Smavatkul, N. (2004) "Saturation Throughput Analysis of Different Backoff Agorithms in IEEE 802.11", *IEEE802.11[A]. 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Vol. 3, pp1870-1875.

[4] Arabi, M., Ghita, B. & Wang, X. (2010) "Improving Fairness In Ad Hoc Networks Through Collision Rate Control", *International Network Conference,* Heidelberg.

[5] Berqia, A., et al. (2008) "Fairness and QoS in Ad-Hoc Networks", *in Vehicular Technology Conference,* VTC Spring 2008, IEEE.

[6] Manaseer, S.S., Ould-Khaoua, M. & Mackenzie, L.M. (2006) "Logarithmic Based Backoff Algorithm for MAC Protocol in MANETs", *in Proceedings of CSIT2006*, Jordan.

[7] Zhang, D., Zhang, B. & Lu, D. (2008) "Contention Window Based Fairness Backoff Algorithm in Ad Hoc Networks", *IEEE Int. Conference Neural Networks & Signal Processing,* Zhenjiang, China.

[8] Li, Z., et al. (2003) "Study of IEEE 802.11 Fairness and its Interaction with Routing Mechanism", *in IFIP MWCN,* Singapore.

[9] Luqing, Y., Winston, K.G.S. & Qinghe, Y. (2003) "Improving Fairness among TCP Flows Crossing Wireless Ad Hoc and Wired Networks", *in Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing,* Annapolis, Maryland, USA.

[10] Jain, R., Chiu, D.M. & Hawe, W. (1984) "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Systems", *DEC Research Report TR-301.*