Evaluating and Improving upon the Fairness of PSMAC with PSMAC-1a Protocol

Matthew Leung and Kalpit Dixit

***Abstract*–The PSMAC protocol in [1] was proposed for purposes of limiting control overhead and maximizing throughput via gated service. In this paper, we demonstrate the shortcomings of PSMAC with the author's skewed fairness scheme, present a new fairness metric that is more relevant to gated service, and propose a way to implement a revised PSMAC-1a protocol that will allow for true fairness in terms of throughput served over all nodes. The suggested PSMAC-1a protocol will aim to improve fairness in heavy, asymmetric traffic while maintaining cognizance of network load such that bandwidth can be maximized when the traffic is light. We conclude with a summary of the road map of on-going research that will prove the above items in NS-2 simulations.**

I. PSMAC Protocol

The PSMAC protocol is defined in [1] as an 802.11 MAC protocol with two key changes. These changes are firstly, gated service (with no limit on packet size) and secondly, p-Persistent backoff contention resolution. Gated service policy is where the sending node will generate an RTS for sending its entire transmit buffer. Note that this does not include any frames that arrive at the sending node between the CTS acknowledgment from the receiving node and the associated successful RTS that was last sent. In lieu of the binary exponential backoff that is standard for 802.11, PSMAC uses a strictly p-Persistent routine for its contention resolution, where if the sending node senses that the medium is idle, it will transmit an RTS with probability p, repeatedly, until it receives a CTS from the receiving node. After it receives the CTS, the sending node will transmit its entire TX buffer queue in back-to-back frames, up to the last RTS – that effectively empties the queue that arrived in the TX buffer before the last RTS. The PSMAC authors generated a timeline illustration in [1] which succinctly summarizes the protocol and is shown in Figure 1.

II. PSMAC Fairness Evaluation

A. PSMAC Fairness based on Average Node Delay

There is a positive aspect to the PSMAC which bears discussion; for a given parameter p made equivalent in both p-Persistent CSMA and PSMAC, PSMAC performs with equal or better average node delay. This is clearly because when we assume that a given node is backlogged with more than 1 frame's worth of data to be transmitted, the node's delay is by default lesser or equal with PSMAC because it allows the TX buffer queue to be emptied all at once. Let us consider a fairness definition as follows and outlined in [1]:

$$f(D_1, D_2, \cdots, D_N) = \frac{(\sum_{i=1}^{N} D_i)^2}{N \sum_{i=1}^{N} D_i^2}. \quad (1)$$

D_i is the average delay for node i. Note that since the average node delay is no longer dependent on the number of frames in the transmit buffer (since they are all transmitted at once upon CTS), then a fairness definition (1) would show up as unity across all loads as we see in the authors' findings in Figure 2. (Note PSMAC-2 and PSMAC-3 are low power equivalents of PSMAC-1 that feature sleep modes.)

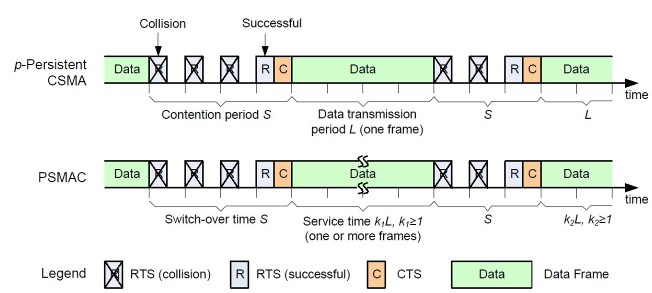

Fig. 1. PSMAC in comparison to p-Persistent CSMA from [1]

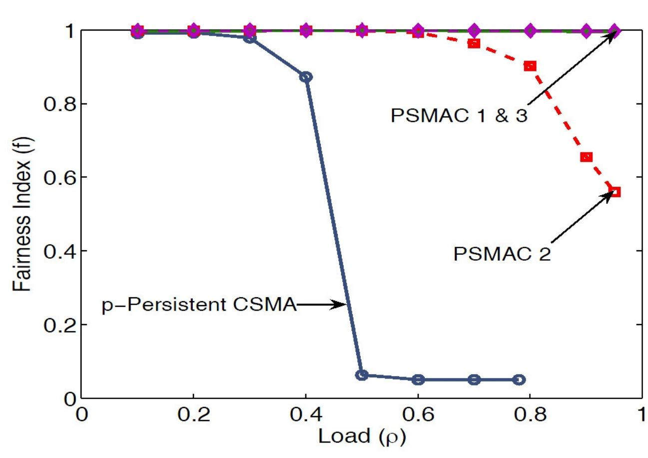

Fig. 2. PSMAC fairness index with high, bursty traffic on Node 1 from [1]

However, as we will see in our simulations, the general benefits of this lowered and highly uniform average node delay do not come without repercussions. In fact, the PSMAC protocol only appears to be fair because the PSMAC authors in [1] use a fairness equation that reflects fairness of winning a CTS when backlogged, *regardless* of the node's request-to-send packet size. Fairness of average delay begets unfairness in terms of served throughput per node when the network is presented with asymmetric loads across the nodes.

Intuitively the PSMAC unfairness scenario can be thought of this way: Operating on a heavy traffic assumption where nodes are always backlogged, Node 1 has on average 10000 backlogged frames to send, and Node 2 is still always backlogged but only has 1 frame to send, on average. Average node delay is 'fair' and equal between Node 1 and Node 2 due to p-Persistence, but this protocol will also give Node 1 10,000 times more throughput than Node 2, even though they both always have something to send in their TX buffer. Throughput-wise, this is unfair.

B. PSMAC Fairness based on Node Flow Throughput

To evaluate fairness of PSMAC based on throughput, we constructed two MATLAB simulations. The first simulation consisted of a network with uniformly distributed traffic across 20 Nodes, such that the expected value of frame arrivals in the TX buffer for each node was equal. In the second simulation, we simulated an asymmetric load where 50% of the packet arrivals were queued in the Node 1 TX buffer while the rest of the 19 other nodes equally shared the remaining 50% of the packet arrivals. With a p-Persistent contention parameter of 1/20, derived from 1/N where N is the number of nodes in the network, we ran the simulation with the PSMAC protocol from various offered loads from 0.2 to 0.9. We summed up the total throughput associated with each Node 1 to Node 20, and made a throughput-based fairness evaluation using the Jain's Fairness Index as defined in [2],

$$f(x_1, x_2 x_3, \ldots, x_n) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \sum_{i=1}^{n} x_i^2}. \quad (2)$$

The x_i term in Equation (2) represents the throughput of node i and n is the total number of nodes. Therefore in Equation (2), the closer each node's throughput is to one another, the closer to unity the Jain's Fairness Index result will be.

C. Implementing the PSMAC Simulation

Please note that our MATLAB simulations model the contention and subsequent throughput reward exactly like how the authors describe it. We also make the same assumption that the authors had made in [1] that the service time *per frame* takes a 'service period' L. Furthermore, we also make the assumption of perfect channel transmissions in our simulation such that upon CTS, the data is always assumed to have been sent successfully and without corruption (and that an ACK was received by the sending node). Also in order to verify replicating PSMAC results in [1] with our simulation, we left out the fixed overhead incurred by ACK and Inter-Frame-Spaces as done by the authors in their PSMAC paper [1]. Testing for their definition of fairness in Equation (1), we were able to duplicate the results in [1], where PSMAC shows an average node delay-based fairness close to unity, as discussed previously in Figure 2.

D. Analysis of Throughput Fairness Results

Since we are contending that the throughput distribution is highly unfair in fully backlogged, asymmetric-loaded networks, we ran our simulation while tabulating each nodes throughput, and calculated fairness from the Jain's Fairness Index in Equation (2). Our preliminary results are shown in Figure 3 below, and we can easily see that PSMAC creates a great disparity in node throughput, even when operating well below the maximum possible offered load.

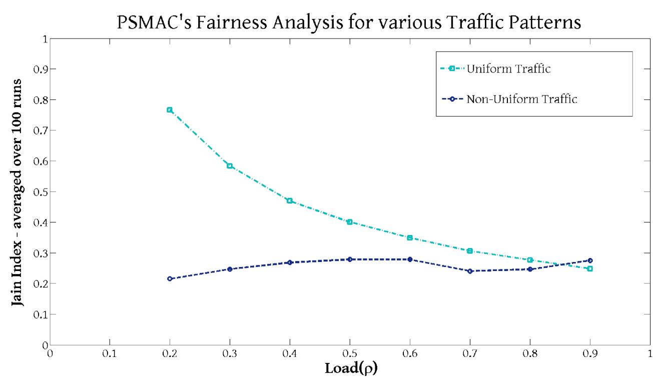

Fig. 3. PSMAC Fairness Analysis over two Traffic Models, Averaged over 100 Runs

1. Uniform Traffic Simulation Discussion

As expected with uniform traffic, the Jain fairness index remained high for light loads closer to the minimum simulated offered load of 0.2. As the offered load increased, nodes began to accumulate increasingly larger transmit buffers (due reduced ability to send data), and the network's Jain index was reduced rapidly in an inverse exponential trend. It was concluded that even in uniform traffic, that the sending of these large packets and the collecting of more nodes with less opportunity to empty their transmit queues caused a feedback scenario that accelerated throughput unfairness. The degree to which the accelerated effect of this congestion took hold with moderate offered load was surprising and unexpected.

2. Non-uniform Traffic Simulation Discussion

Analysis of the throughput data for the non-uniform traffic model showed a poor Jain fairness index for all network loads. This is not surprising because on light offered loads, Node 1 sends many times the throughput (since it was allocated half of the total packet arrivals) of any of the other 19 nodes that share the other half of the total packet arrivals. It is important to note that the Jain's fairness index consistently penalizes the network's fairness for allowing Node 1 to send under very light offered loads, even though Node 2 to Node 20 have few and small packets to send and thus experience rare RTS collisions. Therefore it was seen that for a light traffic assumption, the Jain's index was contradicting and not reflecting our target of maximizing network bandwidth usage when traffic is light and collisions are rare; for instance if we have a light offered load of 0.2, moderating Node 1's behavior to send less than its gated service would be, in essence, wasting bandwidth.

Furthermore, it was observed from Figure 3 that on a heavy traffic assumption with offered loads close to unity, the Jain's Fairness index remained poor for highly asymmetric loads, but it did not continue to decrease below it's starting value that was obtained at lighter loads. This was unexpected. However as predicted under heavy asymmetric load, the Jain's fairness index is low at ~0.3, which correct reflects the throughput unfairness.

The implication from the above results is that the Jain index does not discriminate 'hogging' from 'optimizing bandwidth usage', which is merely the case when Node 1 is emptying its frame arrivals, but Node 2 to Node 20 have scant data to sent. Therefore, it was realized that a better throughput-based fairness index is needed for evaluating PSMAC.

III. Conclusion

In summary, we observed from [1] and verified with our simulation that PSMAC has a good characteristic in that it promotes the emptying of transmit buffer queues in nodes that are heavily backlogged. However as we have proven by evaluating the served throughput per node in our simulations, this results in what was shown in Figure 3 to be unfairness in *both* uniform and non-uniform traffic when operating under a heavy traffic assumption.

The new PSMAC-1a protocol that is currently under development will temper some shortcomings of PSMAC. Specifically, the new PSMAC-1a protocol will address throughput fairness by adjusting the p-parameter in a given node's p-Persistent contention behavior based on 1) it's transmit buffer queue size and 2) its perceived network load based on the number of times it has sent an RTS. Furthermore we will implement a limited k-gated service, which caps the number of data frames that can be sent in any given contention win to *k*.

Also, it was determined from the simulation results that the Jain fairness index was not be the best judge for node fairness since it added penalties for unequal throughput distribution even when the network load was very light. However for a heavy traffic assumption where all nodes are always backlogged, it was a useful indicator both conceptually and numerically.

IV. Future Work

The on-going efforts for PSMAC-1a protocol development will consist of the following action items:

1. Define the new PSMAC-1a protocol
2. Employ a new definition of fairness
3. Implement PSMAC-1a in NS-2
4. Compare NS-2 traffic scenarios of PSMAC-1a to original PSMAC and 802.11 standard.

A. Plans for NS-2 Implementation

In regards to NS-2 implementation, it may be possible to quickly implement a modified 802.11 MAC with p-Persistent gated service (PSMAC) by fixing the contention window CW = CW_{min} = $Cw_{max,}$ and adjusting the CW value such that the expected value of the node to transmit is equal to the desired p-Persistence. In addition, the packet size can be varied to simulate the transmission of multiple back-to-back frames across the channel. This type of modification is discussed in [3] where researchers use a p-Persistent contention window and the 802.11 MAC to model a customized version of 802.11e. Lastly it may be possible to modify the NS-2 c++ BackoffTimer class, but first, an attempt will be made to simulate PSMAC and PSMAC-1a in NS-2 without extravagant modifications that may extend beyond our available time.

References

[1] Y. Li, S. Mao, S. Panwar, and S. Midkiff, "On the Performance of Distributed Polling Service-based Medium Access Control," *IEEE Transactions on Wireless Communication,* vol. 7, no. 11, pp. 4635-4644, Nov. 2008.

[2] R. Jain, A. Durresi, and G. Babic, “Throughput Fairness Index: An Explanation,” http://www.cse.wustl.edu/~jain/atmf/ftp/af_fair.pdf

[3] R. MacKenzie, T. O'Farrell, “Throughput Analysis of a p-Persistent CSMA Protocol with QoS Differentiation for Multiple Traffic Types,” ftp://lenst.det.unifi.it/pub/LenLar/proceedings/2008/icc08/DATA/S07S34P06.PDF