Queue Management Mechanism for 802.11 Base Stations

Dzmitry Kliazovich, *Member, IEEE,* Pedro Henrique Gomes, *Student, IEEE,* Fabrizio Granelli, *Member, IEEE,*
and Nelson L. S. da Fonseca, *Senior Member, IEEE*

***Abstract*—This paper presents a novel queue management mechanism for base stations in infrastructure IEEE 802.11 networks. The mechanism is transparent to high-level protocols and is compatible with the standard. Its effectiveness is illustrated by showing that it can ameliorate the RTT unfairness and the MAC anomaly problems in wired-cum-wireless networks.**

***Index Terms*—802.11, Wi-Fi, active queue management, RTT-fairness, MAC anomaly.**

I. INTRODUCTION

IN wired-cum-wireless networks, the wireless link usually becomes the bottleneck of the end-to-end path, making the management of the Base Station (BS) buffer a key point for performance enhancement. Moreover, the operation of 802.11 networks has various deficiencies among them the TCP round trip time (RTT) fairness and the MAC anomaly problems.

Providing throughput fairness among TCP flows in IEEE 802.11 networks (Wi-Fi) networks is a known problem. Flows generated by senders closer to their receivers obtain a greater share of the available bandwidth than do those far away, since the transmission windows of the connections with small RTT values increase significantly faster than those of connections with large RTTs. This discrepancy is known as the RTT fairness problem [1]. Actually, this problem is not specific to TCP over 802.11 networks but it is especially relevant for these networks. The RTT-fairness problem in heterogeneous networks has been addressed by the adoption of TCP variants such as Libra and Hybla [1], [2], as well as active queue management mechanisms developed for wired networks [3].

Another problem is that the nodes transmitting at low data rates limit the overall throughput of a WiFi cell. This is the so called Multi-rate MAC throughput anomaly that is the result of the slower nodes taking longer to transmit frames than do faster ones. Some solutions for this problem [4], [5] include the modification of the link layer frame size [5] as well as that of the contention window, although these require changes in the standard.

These major deficiencies can be addressed by the mechanism for the management of the BS queue proposed here. Locating the mechanism at the BS leads to a scalable solution. Furthermore, the proposed scheme does not require any changes in the IEEE 802.11 standard. The proposed scheme is transparent to the wired part of the network, as well as to high-level protocols, and can, therefore, coexist with legacy devices. The proposed mechanism is a revised version of that introduced by the authors in [6].

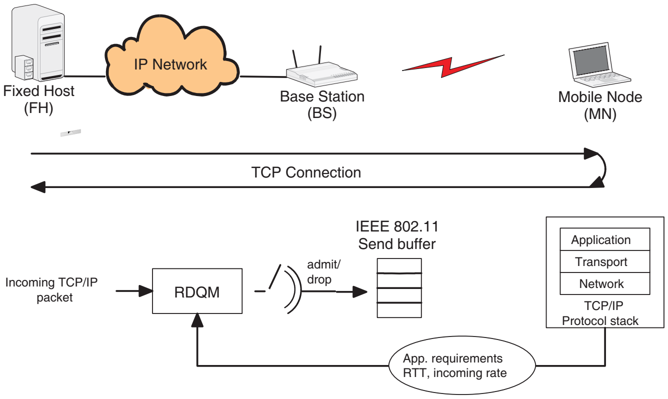


Fig. 1: Receiver-Driven Queue Management architecture.

II. RECEIVER-DRIVEN QUEUE MANAGEMENT

The mechanism proposed here, the Receiver Driven Queue Management (RDQM) adopts a complete partitioning buffer management that do not dropout enqueued data frames. In addition, the sizes of the partition associated to each flow is defined dynamically. Figure 1 shows the main components of the RDQM scheme.

To define the size of each flow partition, information can be provided by the mobile receiver to the BS in the header of the link layer acknowledgement frames, specifically in the Duration/ID field of the IEEE 802.11 link layer header. Thus, no modification of the standardized frame exchange is needed.

For each frame received from the wired network, the BS verifies whether or not the number of enqueued data frames of this flow exceeds the maximum number of frames allowed in the queue for that flow. If it does exceed this number or if the buffer is full, the frame is dropped. Otherwise, it is enqueued. Since there is no dropout of enqueued frames and the flow partition size is dynamic, the number of enqueued frames of a flow can even exceed the flow partition size. In this scenario, an incoming frame can be dropped.

The buffer space is divided between RDQM and non-RDQM flows in a ratio reflecting the number of flows of each type. Each non-RDQM flow receives an equal share of the non-RDQM partition, while each RDQM flow receives a share of the RDQM partition proportional to its bandwidth delay product (BDP), which is estimated by the product of the RTT and the physical data rate of the wireless node. Such an estimation provides only an approximation of the BDP value. Nonetheless, no precise estimation is needed, since the setting

---

Manuscript received March 24, 2011. The associate editor coordinating the review of this letter and approving it for publication was J. Holliday.

D. Kliazovich is with the VTT Technical Research Centre of Finland.

P. H. Gomes and N. L. S. da Fonseca are with the State University of Campinas, Institute of Computing, Campinas SP Brazil (e-mail: nfonseca@ic.unicamp.br).

F. Granelli is with the University of Trento, Italy.

Digital Object Identifier 10.1109/LCOMM.2011.051911.110642

of partition size is based on the ratio between the BDP, of the flows rather on their exact values.

Algorithms 1 to 3 implement the RDQM scheme. Algorithm 1 describes the implementation of the RDQM algorithm at the mobile nodes, where the measured RTT values are inserted into outgoing LL-ACK frame. Algorithm 2, triggered by the arrival of a TCP data frame at the BS, either accepts (line 12) or discards (line 9) the incoming frame. The initial partition size, MAX_{def} and the initial estimation of the BDP value, BDP_{def} are set to the value corresponding to an equal share of the RDQM buffer space.

Algorithm 1 LL-ACK frame transmission at MN

```
1: if TCP data frame received then
2:     Encapsulate Data Rate and RTT information into outgoing LL-ACK
3:     Send LL-ACK
4: end if
```

Algorithm 2 TCP DATA frame arrival at BS

```
 1: if The buffer is full then
 2:     Discard the frame
 3: else
 4:     if Frame belongs to a new flow then
 5:         Create new entry for incoming flow
 6:         Set Each Flow Partition Size to $MAX_{def}$
 7:         Set Each Flow Frame Count equal to zero
 8:         Set Flow BDP equal to $BDP_{def}$
 9:     else
10:         if Flow Frame Count is greater than the Flow Partition Size then
11:             Drop incoming frame
12:         else
13:             Increase Flow Frame Count by one
14:             Accept frame
15:         end if
16:     end if
17: end if
```

Algorithm 3 is activated upon the arrival of each acknowledgement frame for updating the Flow Partition Size. By doing so, the number of active flows is updated. Lines 2 to 5 compute the total estimated BDP value. Line 7 determines the ratio of a flow BDP to the already computed total BDP value. This weighting determines the proportion of the RDQM partition which the RDQM flow will be assigned. Line 8 sets the size of a flow partition. Lines 11 and 12 set the partition of non-RDQM flows to be equal to that of the non-RDQM flows.

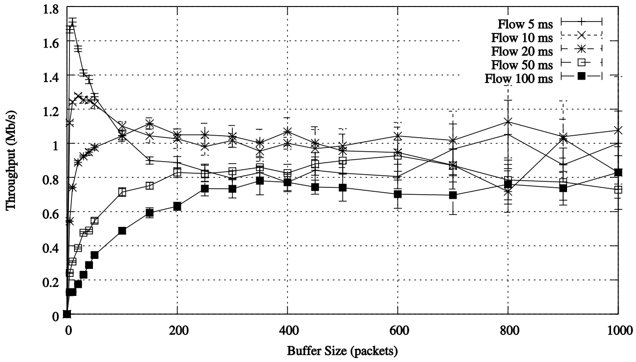
Although RDQM resembles the Snoop-TCP [8] and other TCP proxy solutions [7], it does not require extremely large buffers, nor does it violate end-to-end TCP semantics, as do these solutions.

III. PERFORMANCE EVALUATION

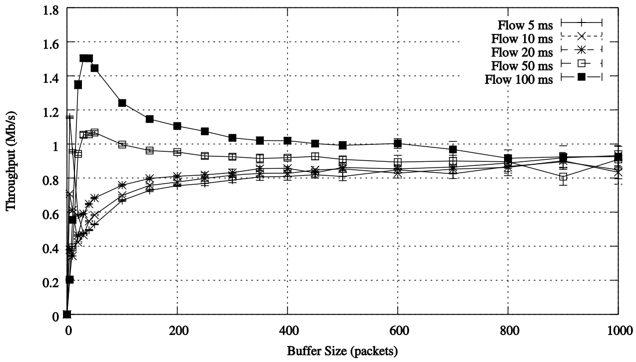
To validate the proposed scheme, extensive simulation experiments using the Network Simulator 2 (NS2) were conducted. The simulated topology consists of wired and wireless sections. The Fixed Nodes (FNs) were connected to the router by links of 100 Mb/s with a propagation delay ranging from 5 to 150 ms. The Mobile Nodes (MNs) were connected to an IEEE 802.11b BS operating at standard bitrates (1, 2, 5.5, and 11 Mb/s). The duration of each simulation runs was 600 seconds, which was sufficient to reach the steady-state phase of TCP window growth. Confidence intervals with 95% confidence level were derived.

Algorithm 3 LL-ACK frame arrival at BS

```
 1: if Flow is RDQM enabled then
 2:     Set BDP Sum equal to zero
 3:     for Each Flow do
 4:         Increase BDP Sum by Flow BDP
 5:     end for
 6:     for Each RDQM Flow do
 7:         Set Flow Weight to Flow BDP / BDP Sum
 8:         Set Flow Partition Size to Flow Weight x Buffer size x (RDQM flows / total number of flows)
 9:     end for
10: else
11:     for Each non-RDQM Flow do
12:         Set Flow Partition Size to Buffer size x (1 / total number of flows)
13:     end for
14: end if
```



(a) Drop Tail (default)



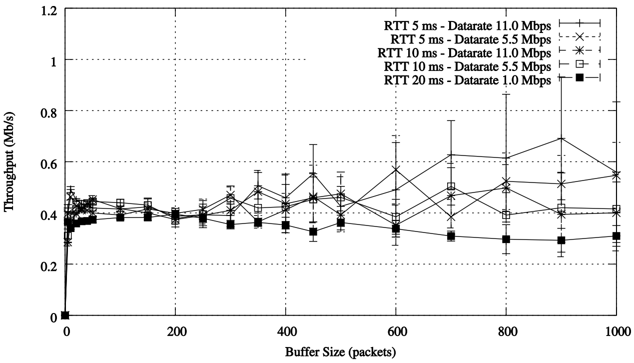
(b) RDQM (proposed)

Fig. 2: Throughput of six TCP flows with different RTTs.

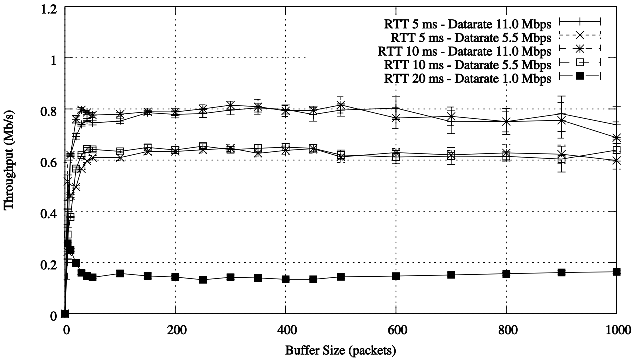
A. Scenario 1: RTT fairness

The first experiments evaluate the improvement in throughput fairness among TCP flows with different RTT values for a network with a physical data rate of 11 Mbps.

Figure 2 shows the throughput obtained by both the droptail (e.g. FIFO) queue (Fig. 2a) and by the RDQM queue (Fig. 2b) for a scenario with 6 flows and propagation delays ranging from 5 to 120 ms. It can be seen that flows with lower



(a) Drop Tail (default)



(b) RDQM (proposed)

Fig. 3: Throughput of six TCP flows with different BDPs.

propagation delays are prioritized by the drop-tail mechanism, although this does not happen using the RDQM mechanism (Fig. 2b). Since flows with 100 ms and 120 ms of propagation delays received only a tiny portion of the bottleneck buffer under the drop-tail scheme, they were transmitted at rates slower than 700 Kbps, even for buffer sizes above 400 frames. On the other hand, these flows, when prioritized under the RDQM scheme, achieved rates of 800 Kb/s regardless of the buffer size.

B. Throughput anomaly

Figure 3 shows the throughput results obtained for 6 TCP flows in a multi-rate scenario. The throughput anomaly under the drop tail can be easily observed in Fig. 3a with per-flow throughput of around 350 Kb/s, corresponding to an aggregated throughput of about 2.1 Mb/s, which is much lower than the average 4.5 Mb/s available in an IEEE 802.11b networks. Moreover, there is no significant difference of individual rates when the anomaly occurs. However, the RDQM scheme mitigates the MAC anomaly effect by keeping the throughput at: 700 Kb/s for the nodes with transmission rates of 11 Mb/s, 500 Kb/s for the nodes with rates of 5.5 Mb/s and 100 Kb/s for nodes with rates of 1 Mb/s. The aggregated throughput is almost 3.2 Mb/s, which represents a 50% improvement on the overall throughput when compared to that achieved using a drop tail queue.

C. RDQM and legacy nodes interoperability

Figure 4 shows the results obtained in a network with both RDQM and non-RDQM nodes for the same scenario portrayed in Fig. 3. Half of the flows are of the RDQM type. It is possible to see that RDQM flows obtain almost the same throughput as in Fig. 3, except for flow number 6 (RTT of 30 ms and datarate of 11.0 Mbps), which achieved a greater throughput for small buffers. The non-RDQM nodes obtained an average throughput of about 600 Kbps and they did not steal bandwidth from the RDQM nodes. It can be concluded that the RDQM scheme can cooperate with legacy nodes to distribute the bandwidth fairly among all flows.

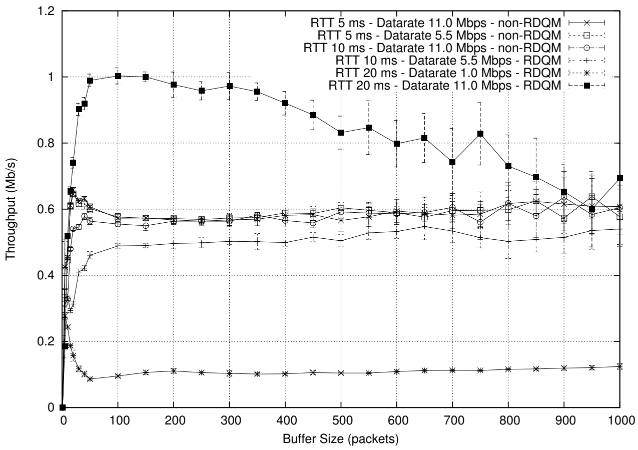


Fig. 4: Throughput of six TCP flows with different BDPs, three of them from RDQM nodes, and three from non-RDQM nodes.

IV. CONCLUSIONS

This paper has introduced a novel management scheme for the buffer at the base station. It has been demonstrated that this scheme can ameliorate common anomalies in Wi-Fi networks. Moreover, it outperforms the traditional drop tail.

Future work will involve the development of a Linux-based implementation as well as a comprehensive performance evaluation based on testbed experiments.

REFERENCES

[1] G. Marfia, C. E. Palazzi, G. Pau, M. Gerla, and M. Roccetti, “TCP Libra: derivation, analysis and comparison with other RTT-fair TCPs,” *Computer Networks*, vol. 54, no. 14, pp. 2327–2344, 2010.

[2] C. Caini and R. Firrincieli, “TCP Hybla: a TCP enhancement for heterogeneous networks,” *Int'l J. Satell. Commun. and Netw.*, vol. 22, no. 5, pp 547–566, 2004.

[3] S. Yi, M. Kappe, S. Garg, X. Deng, G. Kesidis, and C. R. Das “Proxy-RED: an AQM scheme for wireless local area networks,” *Wireless Commun. & Mobile Computing*, vol. 8, no. 4, pp. 421–434, 2008.

[4] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, “Performance anomaly of 802.11b,” in *Proc. IEEE INFOCOM*, vol. 2, pp. 836–843, Apr. 2003.

[5] S. Yoo, J. Choi, J. Hwang, and C. Yoo, “Eliminating the performance anomaly of 802.11b,” in *Networking - ICN 2005 - Lecture Notes in Computer Science*. Springer, pp. 1055–1062, 2005.

[6] D. Kliazovich, P. H. Gomes, F. Granelli, and N. L. S. Fonseca, “Receiver-driven queue management for achieving RTT-fairness in Wi-Fi networks,” in *Proc. IEEE GLOBECOM 2009*, pp. 1–6.

[7] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, “A comparison of mechanisms for improving TCP performance over wireless links, *IEEE/ACM Trans. Netw.*, vol. 5, no. 6, pp. 756–769, 1997.

[8] M. Ivanovich, P. Bickerdike, and J. Li, “On TCP performance enhancing proxies in a wireless environment,” *IEEE Commun. Mag.*, vol. 46, no. 9, pp. 76–83, 2008.