

# Channel Quality Aware Active Queue Management in Cellular Networks

**Abstract**—Queue management plays an important role in the performance of IP based networks and cellular networks in particular where large buffers are often deployed in order to absorb bursts resulting from the dynamic nature of the radio channel. Long standing queues building up in large buffers leads to “Bufferbloat”, degrading the performance especially for delay-sensitive applications while small buffers may lower the link utilization. This paper proposes a novel AQM algorithm tailored to cellular networks, mainly by utilizing the Channel Quality Indicator (CQI) periodically reported by user equipment, in order to mitigate Bufferbloat and maintain acceptable levels of performance. Simulation results show that the proposed algorithm reduces the average queuing delay of packets for each user by 40% on average with TCP traffic compared with the CoDel algorithm which has recently become popular.

## I. INTRODUCTION

The emergence of powerful smart devices and their integration in people’s daily lives place huge strains on networks. As forecast by Cisco [1], global mobile traffic will increase eight folds by 2020 and the link speed will increase by three fold. Since 4G, cellular networks have been all IP-based while widely used applications such as video streaming, gaming and online chatting. Large buffers are often deployed in intermediate devices in order to absorb bursty traffic and increase link utilization. Over buffering results in long standing queues which leads to huge queuing delays as well as a reduction in overall network throughput, which is referred to as the “Bufferbloat” phenomenon [2].

Research work [3] has demonstrated the Bufferbloat issue in broadband networks and [2] shows that the issue exists in both uplink and downlink directions. Bufferbloat is also reported in wireless access networks, as shown in [4] (Wi-Fi) and [5] (cellular networks). Wireless devices are often equipped with large buffers due to time-varying wireless channels to absorb bursts, which makes Bufferbloat more severe in cellular networks.

Solutions to alleviate Bufferbloat can be divide into 2 categories. One is modifications to TCP on the realization that over buffering delays the signaling of congestion. Variations of TCP have been proposed which try to sense the congestion as early as possible, such as TCP Vegas [6] which is delay-based; however it faces fairness issues when competing with loss-based TCP. TCP Cubic [7] adjusts how the sending rate is increased to make it less aggressive. Cubic is now the most successful variation and widely adopted as default TCP setting in Linux systems. Modifications to TCP

are not ideal as too many variations co-exist and it is hard to achieve fairness among different versions and the response to congestion is sluggish.

Alternatively, Active Queue Management (AQM) whereby packets are dropped actively before too many packets accumulate in the queue. They indicate the congestion in the network by dropping packets which can give quick responses to congestion. Random Early Detection (RED) [8] which is a classical technique begins to drop the incoming packets when the queue size reaches a threshold and the drop probability is linearly related to the the size of the queue. RED parameters are pre-configured and cannot adapt to various network conditions. Delay-based adaptive AQMs have emerged recently and Controlled Queuing Delay (CoDel) [9] is the most widely know one for its simplicity and effectiveness.

Even though a whole variety of AQMs and variation exist, most of these focus on wired networks such as adaptive RED (ARED) [10], flow-queue CoDel (fq-CoDel) [11]. Evaluations and testings have also been done in wireless networks such as in [4], [12] and [13] but none of them considers the effect of wireless features. Existing AQMs are primarily based on the status of the queue (queue size or the delay each packet suffers in the queue). However, unique features of the underlying cellular networks such as the channel conditions can have a considerate impact on the behavior of the queue. The time-varying channel is affected by the mobility of users, the density of users in one cell and etc. Good channel conditions mean advanced Modulation And Coding Schemes (MCS) are chosen, and so users can achieve faster Internet connections. And with poor channel conditions, a lower rate MCS is chosen which results in slower Internet connections in order to achieve reliable transmission. A faster connection will transmit the packets in the queue faster while a slower connection means extra queuing delays occur. The channel condition can be easily obtained as UEs will measure it and report to the eNodeB (eNB) every Transmission Time Interval (TTI).

In this paper, we propose a CQI-aware AQM which is light weight but can effectively control the delay. Section II gives the background knowledge of the cellular networks structure together with state-of-the-art and section III gives the details of the proposed algorithm. Section IV gives the simulation results and discussion. Section V gives the conclusion and future work.

## II. BACKGROUND AND STATE-OF-THE-ART

### A. Structure of Cellular Networks

A cellular communication system is shown in Figure 1a. Although link speed of the last hop has increased with the advance of technology, the access networks are still considered as the bottleneck as there large number of UEs sharing the bandwidth, while other links within the core network and the link between servers and the core network is fast. Compared with Wi-Fi and wired network, the protocol stack of cellular networks is different to those of wired and Wi-Fi networks. Data transmitted between the base station and the UE is carried by a virtual concept, “bearer” [14], which means each UE has a dedicated buffer for communication in the base station. As shown in Figure.1b, the Radio Protocol Stack has a Radio Link Control (RLC) Layer and Packet Data Convergence Protocol (PDCP) Layer, where the queuing of packets for different UE happens. Each UE, when connected to an eNB, will be allocated a dedicated PDCP and RLC buffer for downlink data transmission. RLC layer has three different transmission modes, Transient Mode (TM), Unacknowledged Mode (UM) and Acknowledged Mode (AM) mode.

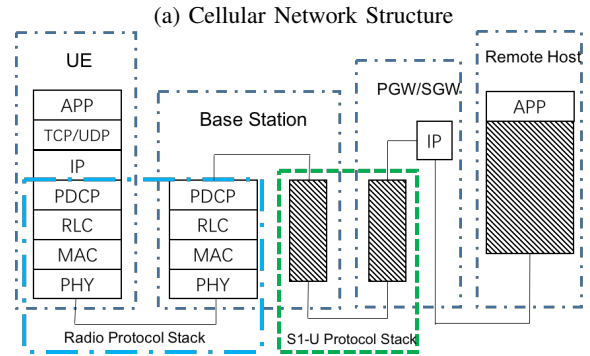
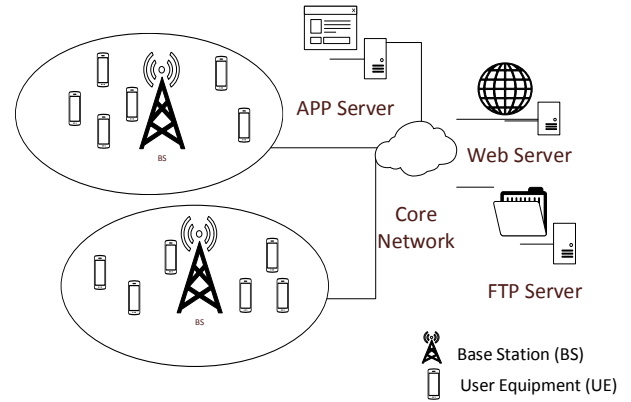
- TM Mode: it does not make any modifications to data, which means no headers are added or removed, no segmentations created and no aggregations. And it does not require any ACK/NACK from the receiver.
- UM Mode: it is similar to TM mode but the difference is that UM mode has its own headers and it can segment or concatenate data.
- AM Mode: it is used to guarantee reliable transmission which requires ACK/NACK from the receiver. It can also segment or concatenate data and has its own headers. Moreover, it will make a copy of the transmission buffer for a possible retransmission.

TM mode and AM mode are used in the control plane and UM mode is used for transmitting data. Our algorithm is deployed in the UM buffers so that it does not affect the control and signaling messages.

### B. State-of-the-art

An AQM tuned for cellular networks is proposed in [15] and this is a variation of RED and it is implemented in the RLC layer. The authors change the control function from linear function to non-linear and simulation results show that it outperforms RED from the aspect of end-to-end average delay. However, it is based on RED and cannot solve the tuning issue as the length of the queue is not directly related to delay.

Research papers [16] and [17] try to control the traffic sending rate by making modification on the congestion window of the receiver side (cwnd). Both of the works take advantage of Round Trip Time (RTT) and aim to solve



(b) Protocol Stacks of Cellular Network

Fig. 1: Cellular Network Systems

Bufferbloat in cellular networks. [16] is primarily based on the estimated RTT and the minimum RTT value. The estimated RTT is the average of RTT value from all the samples of RTT and if the estimated RTT is larger than the minimum RTT, the Receiver Window (rwnd) will be reduced. Work [17] controls rwnd by monitoring queue states. The queue state is estimated using the difference between the minimum RTT value and the real RTT value. It assumes that the minimum RTT value is the RTT when there is no queue in the buffer. However, when there is no queue in the buffer, the dynamic nature of wireless channel and the number of UEs competing for the bandwidth will also affect RTT. Additionally, the calculation of rwnd in their work is a function of the dropping function of AQM deployed in the router and different AQMs may behave very differently.

Parameters from the physical layer play an important role in cellular communications, e.g. CQI which indicates the quality of the wireless channel for data transmission. CQI is measured at the UE side and reported to eNB using the Public Uplink Control Channel (PUCCH). The Modulation And coding Scheme is selected at eNB according to the CQI reported by a specific user which reduce the Bit Error Rate (BER) [18]. CQI is also used in the data link layer as a

parameter in scheduling schemes. More resource blocks are allocated as a compensation of bad channel quality so as to achieve fair throughput and low delay. It can also be used in another way such as maximizing the overall throughput by giving priority to UEs with good channel quality.

Queue aware scheduling, such as reported in [19], gives priority to real time (RT) traffic. However, most of the traffic in cellular networks are RT traffic, so giving priority is not easy and it can be conflict with 3GPP specifications.

The idea of our algorithm is straightforward. In cellular networks, each UE has a unique buffer in the eNB and the respective UE can only transmit data when resource blocks are allocated to it. With the increasing number of devices and much more powerful devices, the competition of wireless resource is fierce. Increasing the capacity of served UEs or allocating more resources to UEs will not solve the issue of congestion in the buffer. In our algorithm, UEs with poor channel quality may drop more packets compared with the ones with good channel quality. We also take delay into consideration as UEs with bad channel quality is more likely to suffer large queuing delays.

### C. Implementation of CoDel

CoDel makes the decision about whether to drop a dequeuing packet periodically. If the waiting time of a dequeuing packet goes over the minimum allowed value (target), it starts to count certain time period which is called interval. During this interval, if the waiting time of all the dequeuing packets goes over the target, it will drop one packet when the interval ends.

Drop from head makes CoDel not suitable for cellular networks due to the limitations from MAC layer. The HOL packets might be fragmented and the remaining part will be returned to the queue. If the returned part is dropped, it will influence to entire flow as on retransmission mechanisms can help to recover the missing part.

Hence, we make a slight change to CoDel. Instead of examining the HOL packet in the buffer, our version of CoDel will examine the second packet, every time the HOL packet is about to dequeue. In this way, we tail CoDel into the cellular networks.

## III. CQI-AWARE QUEUE MANAGEMENT

The proposed algorithm keeps tracking the HOL delay of each packet. When a packet arrives at the queue, a timestamp is added to the packet and when it leaves the queue, the waiting time experienced by the packet is calculated. The random dropping decision for next incoming packet is made according to channel quality of the user i.e. CQI, and the delay experienced by the packet about to leave the queue. The dropping probability is controlled by Eq (1), when  $d_{min} \leq$

$d^i_{queuing} \leq d_{max} \cdot d_{queuing}$  is the Head-of-Line (HOL) delay of each packet, and  $d_{max}$  is the maximum queuing delay.

$$P^i_{drop} = -\beta * (d_{max} - d^i_{queuing}) * k_i^3 + 1 \quad (1)$$

$d_{max}$  can be preset and when  $d^i_{queuing}$  reaches  $d_{max}$ , the drop probability equals to 1. When  $d^i_{queuing}$  is below  $d_{min}$ , the drop probability will be 0, as shown in Eq (2).

$$P^i_{drop} = \begin{cases} 0, & \text{if } d^i_{queuing} \leq d_{min} \\ 1, & \text{if } d^i_{queuing} \geq d_{max} \end{cases} \quad (2)$$

$k_i$  is the CQI index reported by the  $i_{th}$  UE to the BS.  $d^i_{queuing}$  can range from tens of million seconds to hundreds of million seconds, hence the cube of CQI is adopted in the formula.  $\beta$  is the index normalization coefficient, as shown in Eq (3) where  $k_{i_{max}}$  is the maximum CQI value can be achieved.

$$\beta = \frac{1}{(d_{max} - d_{min}) * k_{i_{max}}^3} \quad (3)$$

The range of the drop probability is shown in Figure 2. With the increase of the queuing delay, the drop probability increase, which can help reduce the waiting time of each packet. Different UEs may experience different channel qualities and the CQI value determines the transmission rate. For the UEs with low CQI value, the queue will build up quickly as the depletion rate is low. Hence, UEs with low CQI value has a higher drop probability.

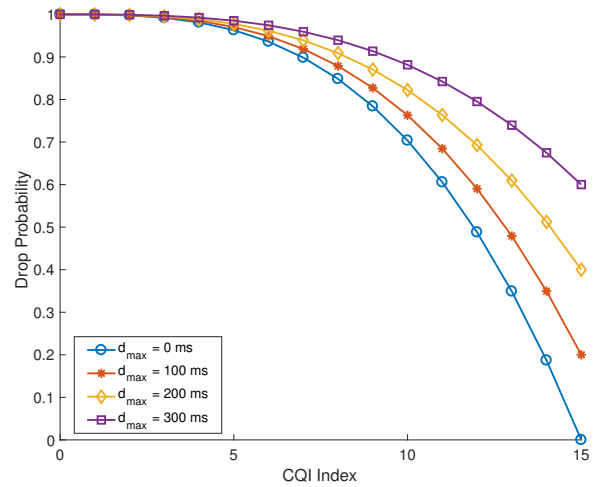


Fig. 2: Dropping probability function of the proposed method.

UEs with different CQI can suffer different delay, so we have to input these parameters into the algorithm that calculates the drop probability so we can give a quick response to congestion. The algorithm examines the waiting time for each packet, and hence can identify congestion rapidly. When congestion happens, UEs with poor CQI drops more packets, and even with good channel quality, UEs suffering larger delay

can drop more packets. In both cases, deploying our algorithm will prevent the queue from further growth.

---

**Algorithm 1** CQI Aware Active Queue Management

---

```

1:
2: Calculate the HOL queuing delay of the  $i_{th}$  UE
3:
4: for each pkt arrives at the RLC of the  $i_{th}$  UE do
5:   Add a timestamp  $T_{in}$ 
6: end for
7: for each pkt leaves RLC do
8:   Record the current time  $T_{current}$ 
9:    $d^i_{queuing} = T_{current} - T_{in}$ 
10: end for
11:
12: Packet dropping decision of the  $i_{th}$  UE
13:
14: if  $d_{min} < d^i_{queuing} < d_{max}$  then
15:    $P^i_{drop} = -\beta * (d_{max} - d^i_{queuing}) * k_i^3 + 1$ 
16: else if  $d^i_{queuing} \leq d_{min}$  then
17:    $P^i_{drop} = 0$ 
18: else
19:    $P^i_{drop} = 1$ 
20:
21: end if

```

---

IV. RESULTS AND DISCUSSION

The proposed algorithm is implemented in Network Simulator (NS) 3. The topology used in the simulation is shown as Figure 3. The UEs are randomly distributed within 2500 meters to 5000 meters. The number of UEs varies from 2 to 10. The Buffer at the RLC layer is set to 100 packets and the TCP Cubic is used. The propagation delay is set to 50 ms and link rate is 10 Mbps for the connection between the core network and the server. Each Base Station (BS) has 15 resource blocks. According to the quality settings, Guaranteed Bit Rate (GBR) video traffic is generated from the server and send to each UE.

The channel quality of a user can change rapidly. For example, a moving obstacle, such as a big truck stops in the way between the UE and the BS for several seconds. To make the simulation more realistic, a random movement model is applied to each UE. When the simulation starts, each UE will randomly choose a direction and a speed uniformly distributed from 50m/s to 100m/s. When they arrive at the edge of the cell, they will stop and choose a new direction and speed. In this way, the CQI of the UE changes with the movement of UE.

The sending rate of TCP traffic changes according to the congestion level of the networks. It will increase when a ACK is received and decreased when the ACK is lost or delayed (depends on different algorithms).

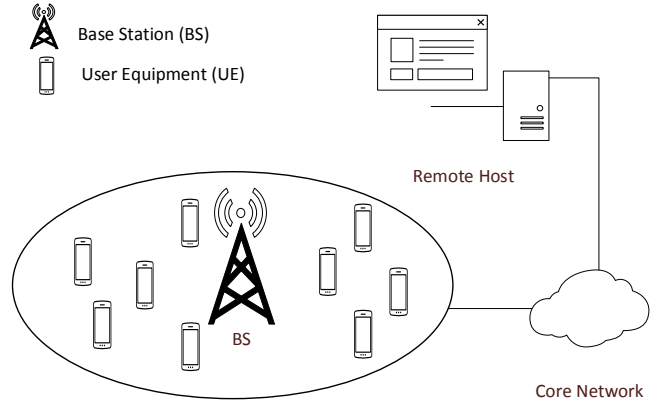


Fig. 3: Simulation Topology.

The performance of the proposed algorithm is evaluated in terms of queuing delay, average goodput and the loss probability given by the algorithm. The drop probability is determined by both queuing delay and the channel conditions i.e. CQI. When the queuing delay exceeds lower threshold, i.e.  $d_{min}$ , the drop probability is calculated by Eq1. The next enqueue packet will be randomly dropped accordingly. Average end-to-end delay is shown in Figure 4. It can be seen from the figure that the average end-to-end delay increases with the increasing number of UEs. It is because the resource at MAC layer is shared by all the users. Deploying AQM at the RLC layer helps reduce the average end-to-end delay, especially when there are more users in the system. Compared with CoDel, the proposed algorithm has better performance from the aspect of average end-to-end delay.

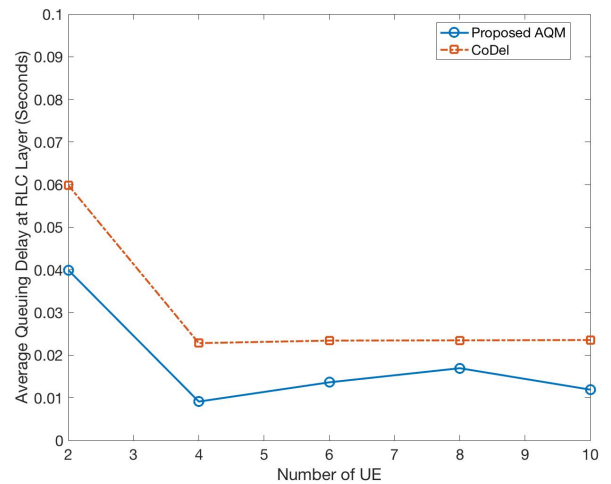


Fig. 4: Average queuing delay at RLC layer with increasing number of UEs.

Lower average end-to-end delay is achieved by dropping

more packets reasonably. The drop probability is shown in Figure 5. As traditional cellular networks do not actively drop packets during the transmission, only two curves are shown. Compared with CoDel, the drop probability of proposed algorithm is a bit higher.

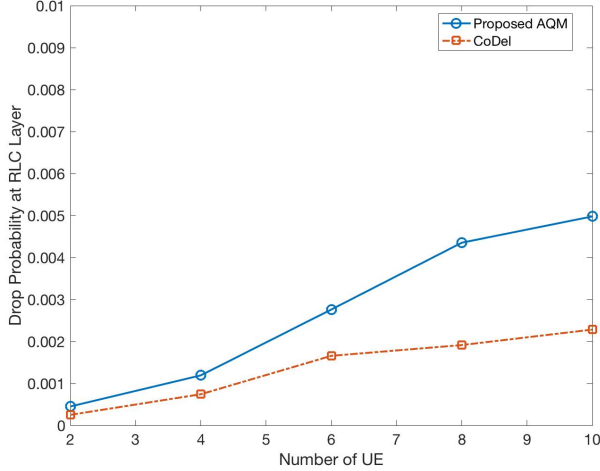


Fig. 5: Drop Probability

Deploying AQM helps to increase the average goodput. As shown in Figure 6, both the proposed algorithm and CoDel have higher goodput compared with the scenario without AQM deployed. Actively dropping a packet gives a signal to the sender that congestion happens. The sender will stop increasing the sending window. In this way, it alleviate the congestion of the whole system.

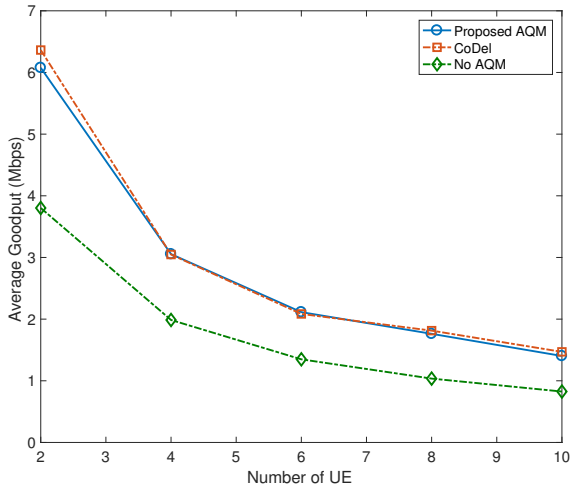


Fig. 6: Goodput with increasing number of UEs.

Jain's fairness index is used to rate the fairness in a network when there are multiple users in the system [20].

The Jain's fairness index is shown in Figure 7. The UEs are randomly distributed, hence when there only 2 in the system and they are suffering different channel quality, the fairness index of the proposed algorithm will be lower compared with CoDel. However, better fairness is achieved with the increasing number of UEs, which shows that the proposed algorithm is suitable for multi user scenario.

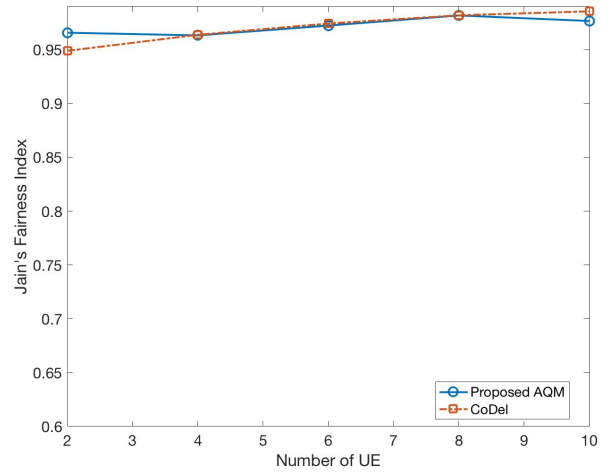


Fig. 7: Jain's Fairness Index

## V. CONCLUSIONS

In this paper, a CQI aware AQM algorithm is proposed for cellular networks with the goal of mitigating Bufferbloat and improving performance. This algorithm is implemented in the base station where all the connected UEs have a dedicated buffer. Our algorithm considers the channel quality of each user and actively drops packets in order to minimize overall delay and maximize goodput. The drop probability is determined by both the HOL queuing delay and the channel quality. Simulation results show that our algorithm is able to control the delay regardless of the number of UEs in the system and does not harm the other metrics, such as fairness. Meanwhile, it improves the average goodput. It is able to achieve this performance due to the consideration of a specific wireless feature, i.e. CQI, unlike existing AQMs which normally only consider network layer parameters. Nowadays, the radio environment is complex because of different types of interferences. These have a considerable effect on the higher layers hence need to be considered for an AQM to be successful in the cellular environment. It can also be used in future generations of cellular networks as the CQI still plays an important role in cellular networks in general.

In addition to CQI, other parameters such as Buffer State Report (BSR) and Discontinuous Reception (DRX) can also influence network delays and these will be considered in our future work in order to further fine-tune our algorithm.

## REFERENCES

- [1] C. V. Forecast, "Cisco visual networking index: Global mobile data traffic forecast update, 2015-2020 white paper," *Cisco Public Information, February*, vol. 9, 2016. [Online]. Available: <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html>
- [2] J. Gettys, "Bufferbloat: Dark buffers in the internet," *IEEE Internet Computing*, vol. 15, no. 3, pp. 96–96, May 2011.
- [3] M. Dischinger, A. Haeberlen, K. P. Gummadi, and S. Saroiu, "Characterizing residential broadband networks," in *Internet Measurement Conference*, 2007, pp. 43–56.
- [4] T. Høiland-Jørgensen, P. Hurtig, and A. Brunstrom, "The good, the bad and the wifi: Modern {AQMs} in a residential setting," *Computer Networks*, vol. 89, pp. 90 – 106, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1389128615002479>
- [5] H. Jiang, Z. Liu, Y. Wang, K. Lee, and I. Rhee, "Understanding bufferbloat in cellular networks," in *Proceedings of the 2012 ACM SIGCOMM Workshop on Cellular Networks: Operations, Challenges, and Future Design*, ser. CellNet '12. New York, NY, USA: ACM, 2012, pp. 1–6. [Online]. Available: <http://doi.acm.org/10.1145/2342468.2342470>
- [6] L. S. Brakmo and L. L. Peterson, "Tcp vegas: end to end congestion avoidance on a global internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, pp. 1465–1480, Oct 1995.
- [7] S. Ha, I. Rhee, and L. Xu, "Cubic: A new tcp-friendly high-speed tcp variant," *SIGOPS Oper. Syst. Rev.*, vol. 42, no. 5, pp. 64–74, Jul. 2008. [Online]. Available: <http://doi.acm.org/10.1145/1400097.1400105>
- [8] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, Aug 1993.
- [9] K. Nichols and V. Jacobson, "Controlling queue delay," *Queue*, vol. 10, no. 5, pp. 20:20–20:34, May 2012. [Online]. Available: <http://doi.acm.org/10.1145/2208917.2209336>
- [10] A. Showail, K. Jamshaid, and B. Shihada, "Buffer sizing in wireless networks: challenges, solutions, and opportunities," *IEEE Communications Magazine*, vol. 54, no. 4, pp. 130–137, April 2016.
- [11] T. Hoiland-Joergensen, P. McKenney, D. Taht, J. Ghettyts, and E. Dumazet, "Flowqueue-codel: draft-hoiland-joergensen-aqm-fq-codel-00," Internet-draft, IETF, Tech. Rep., 2014.
- [12] T. Jain, A. B., and M. P. Tahiliani, "Performance evaluation of codel for active queue management in wired-cum-wireless networks," in *2014 Fourth International Conference on Advanced Computing Communication Technologies*, Feb 2014, pp. 381–385.
- [13] A. Showail, K. Jamshaid, and B. Shihada, "An empirical evaluation of bufferbloat in ieee 802.11n wireless networks," in *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2014, pp. 3088–3093.
- [14] F. Capozzi, G. Piro, L. A. Grieco, G. Boggia, and P. Camarda, "Downlink packet scheduling in lte cellular networks: Key design issues and a survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 2, pp. 678–700, Second 2013.
- [15] A. K. Paul, H. Kawakami, A. Tachibana, and T. Hasegawa, "An aqm based congestion control for enb rlc in 4g/lte network," in *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, May 2016, pp. 1–5.
- [16] H. Jiang, Y. Wang, K. Lee, and I. Rhee, "Drwa: A receiver-centric solution to bufferbloat in cellular networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 11, pp. 2719–2734, Nov 2016.
- [17] H. Im, C. Joo, T. Lee, and S. Bahk, "Receiver-side tcp countermeasure to bufferbloat in wireless access networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 2080–2093, Aug 2016.
- [18] M. T. Kawser, N. I. B. Hamid, M. N. Hasan, M. S. Alam, and M. M. Rahman, "Downlink snr to cqi mapping for different multipleantenna techniques in lte," *International Journal of Information and Electronics Engineering*, vol. 2, no. 5, p. 757, 2012.
- [19] M. A. Ali, A. Esmailpour, and N. Nasser, "Performance evaluation for lte applications with buffer awareness consideration," in *2016 IEEE Wireless Communications and Networking Conference*, April 2016, pp. 1–7.
- [20] R. Jain, D.-M. Chiu, and W. R. Hawe, *A quantitative measure of fairness and discrimination for resource allocation in shared computer system*.