2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)

An AQM based Congestion Control for eNB RLC in 4G/LTE Network

Anup Kumar Paul, Hidehiko Kawakami, Atsuo Tachibana and Teruyuki Hasegawa
KDDI R&D Laboratories Inc.
Saitama, 356–8502 JAPAN
Email: anup@kddilabs.jp, hi-kawakami@kddilabs.jp, tachi@kddilabs.jp, teru@kddilabs.jp

***Abstract*—In the context of a heterogeneous mobile environment, we analyze the eNodeB (eNB) Radio Link Control (RLC) buffer overflow problem for the 4G/LTE network, and show the benefits of deploying Active Queue Management (AQM) strategies at wireless access links. In order to satisfy the increasing demands of wireless mobile communications, LTE has been proposed as an all-IP-based network to bring higher peak data rates and better spectral efficiency. One of the LTE system performance optimization points is the RLC buffer management mainly at the downlink. RLC buffers at the eNB hold user data before it is selected by a MAC scheduler for transmission over the radio interface. A buffer management scheme is required to keep the RLC buffer occupancy to a minimal level without a noticeable degradation in user application performance. This is a challenging issue for the mobile network operator and has to be thoroughly investigated. When the traffic load is low, the available bandwidth is underutilized and when the traffic load is high, the end-to-end delay becomes large. This paper presents a minimal adjustment to RED called Smart RED (SmRED) in which the packet dropping probability is adjusted based on the traffic load to achieve optimal end-to-end performance. Additionally, the migration from RED to SmRED in a real network needs very little work because of its simplicity. Finally, for a delay-sensitive data stream, a cross layer approach based on SmRED is proposed to make it more practical.**

Keywords*—*Congestion Control; eNB RLC; SmRED; LTE; AQM

I. Introduction

The road-map of a next generation mobile network is to provide mobile broadband services to the end-users. To this end, Long Term Evolution (LTE) and System Architecture Evolution (SAE) were formulated by the $3rd$ Generation Partnership Project (3GPP) in 2004 [1]. LTE introduces a new air interface and radio access network, which provides much higher throughput and low latency, greatly improved system capacity and coverage than those of the WCDMA systems. These improvements lead to increased expectations on the end-user quality of experience (QoE) over LTE as compared to existing $2G/3G$ systems. The LTE radio interface, the interface between user equipment (UE) and eNB, consists of four main protocol layers to transfer the data between eNB and UE securely and with certain reliability. They are the Packet Data Convergence Protocol (PDCP) layer, Radio Link Control (RLC) layer, Medium Access Control (MAC) layer and Physical (PHY) layer. All these four protocol layers are part of the protocol stack in UE and eNB.

To date, there have been many researches on LTE, not only on the physical layer but also on higher layers, e.g., the PDCP and RLC layer. On the eNB side, each radio bearer has one PDCP entity that processes the IP packets in the user plane. The PDCP layer mainly performs ciphering/deciphering, header compression/decompression, security including data integrity/verification and also reordering during handovers. The RLC layer is responsible for segmenting or concatenating the RLC Service Data Units (SDU) [2][3][4]. RLC entities provide three modes of operation that are Transparent Mode (TM), Unacknowledged Mode (UM) and Acknowledged Mode (AM). Among the three modes, AM RLC and UM RLC are provided for non real-time and real-time applications, respectively. Real-time applications have strict requirements on end-to-end packet delay that must be followed to achieve acceptable user QoE. For example, conversation VoIP specifies a mouth-to-ear delay of less than 150 ms to achieve transparent interactivity [5]. On the other hand, non real-time applications do not impose strict packet end-to-end delay but still user QoE for such applications is closely related to packet end-to-end delay. The main data buffers of the user interface are located at the RLC layer where data is held before its transmission to the destination UE. However, if there is congestion or overflow in buffers of RLC entities, low latency will not be guaranteed.

As a high speed broadband network, LTE promises an uninterrupted connectivity to the packet data network. Congestion may take place on the radio interface due to various reasons. One such reason for congestion is communication quality in the channel that varies with the distance of the UE from the base station (eNB) and also different communication systems differ from each other in effective data rate that leads eNB to select low bit rate modulation scheme in case of poor communication channel quality and may segment the packets. Therefore, if the channel condition is not good enough to transmit RLC SDUs in time, they will be in buffers of RLC entities waiting for the instruction for segmentation or concatenation from the MAC layer. Hence, there is a high probability of congestion or overflow in RLC buffers due to the large volume of traffic in a short period of time, leading to a high delay that results in poor end-to-end application performance [6]. To guarantee high throughput and low delay when congestion occurs, numerous researches on Active Queue Management (AQM) based congestion control schemes in the network have been proposed in the past, especially for transport-layer protocols. Among the proposed AQM schemes, the Random Early Detection (RED) [7] algorithm is one of the most famous algorithms.

The primary goal of this paper is to design a simple and practical buffer management scheme for eNB RLC buffer that

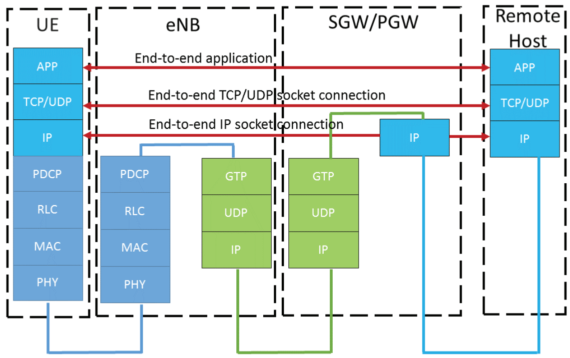

Fig. 1. The LTE protocol stack (user plane).

is able to maintain the queue size to the predictable value under a wide range of network loads. This paper proposes a minimal adjustment to RED, which is called Smart RED (SmRED), in which the packet dropping probability function is divided into two sections to distinguish different network loads. The control law of SmRED can regulate the queue size to the predictable queue size that achieves higher throughput at low traffic congestion and improves end-to-end delays at high traffic congestion without degrading the throughput much. As a result, SmRED dynamically adapts the packet dropping probability to the level of network load. Furthermore, to migrate from RED to SmRED in a real network, very little work needs to be done because only the packet dropping probability profile is adjusted.

The remainder of this paper is organized as follows. LTE protocol structure and AQM are described briefly in Sect. II, and the proposed strategy for congestion control is explained in Sect. III. Sect. IV describes the performance of the proposed scheme by simulation. Finally, Sect. V concludes the paper and discusses the future work.

II. Background

A. LTE Protocol Structure (User Plane)

The user plane protocol structure of LTE is shown in Fig. 1. The RLC entity is specified in the 3GPP technical specification [4], and comprises three different types of RLC modes: TM, UM, and AM. The RLC entities provide the RLC service interface to the upper PDCP layer and the MAC service interface to the lower MAC layer. Among the three modes, TM RLC is configured to the Radio Resource Control (RRC) messages that do not need RLC configuration. Error-sensitive and delay-tolerant non real-time applications are provided by AM RLC. On the other hand, UM RLC is for delay-sensitive and error-tolerant real-time applications such as VoIP [2].

The AM RLC entity manages 3 buffers:

- Transmission Buffer: it is the RLC SDU queue. When the AM RLC entity receives a SDU from the upper PDCP entity, it enqueues it in the transmission buffer.
- Transmitted Protocol Data Units (PDUs) Buffer: it is the queue of transmitted RLC PDUs for which an ACK/NACK has not been received yet. When the AM RLC entity sends a PDU to the MAC entity, it also puts a copy of the transmitted PDU in the transmitted PDUs buffer.
- Retransmission Buffer: it is the queue of RLC PDUs that are considered for retransmission (i.e., they have been NACKed). The AM RLC entity moves this PDU to the retransmission buffer, when it retransmits a PDU from the transmission buffer.

The segmentation and concatenation for the RLC SDU of the AM RLC entity follow the same philosophy as the UM RLC entity but there are new state variables only present in the AM RLC entity. Depending on the channel condition or the distance of the UE from the eNB, the MAC entity can give instruction to the RLC entity to segment or concatenate the RLC SDU. For example, if the channel condition is not good then the RLC entity has to segment the RLC SDU to fit the length decided by the MAC layer to avoid large bit errors. Thus, a long queue in the RLC buffer or an overflow is likely to occur. Either of these will cause a long delay and lower end-to-end throughput, which are undesirable for a delay-sensitive stream service and non real-time application services, respectively.

B. Active Queue Management (AQM)

A congestion control scheme based on AQM has become a research hot spot in the industry, and the AQM mechanism is recommended on Internet routers to achieve the following goals: managing queue lengths to absorb short-term congestion (e.g., bursts), providing a lower interactive delay, and avoiding global synchronization [8]. Among the AQM mechanisms, the most typical scheme is RED [7] proposed by Floyd and Jacobson. The RED algorithm functions by detecting incipient congestion and notifying the Transmission Control Protocol (TCP) by probabilistically dropping packets before the queue fills up. Briefly, the algorithm works by maintaining an average queue size. As the average queue size varies between the minimum and maximum thresholds, the packet dropping probability linearly changes between zero and maximum drop probability P_{max}. Thus, the packet dropping probability function is linear to the change of the average queue size. If the average queue size exceeds the maximum threshold, all arriving packets are dropped. Since the packet dropping mechanism is based on the moving average algorithm, RED can control the transient congestion by absorbing arrival rate fluctuations. Although RED is a significant improvement over simple Droptail [9] that simply drops all incoming packets when a queue is full, RED is particularly sensitive to the traffic load and the parameters of the scheme itself [10].

It has been proved that the performance of the RED algorithm on avoidance congestion is excellent [11]. However, the RED algorithm is employed in TCP/IP layers, which is very different from the RLC layer. RED is used with TCP window control. But delay-sensitive data streams transmit by the User Datagram Protocol (UDP), which does not have the window control mechanism. Therefore, for delay-sensitive data streams, a cross layer approach is necessary to control the congestion in RLC based on AQM mechanisms.

III. Proposed Congestion Control Scheme

The goal of a proper AQM scheme is to maintain the average queue size between minimum threshold and maximum threshold of the queue at low oscillations and in turn help avoid forced drops. It is inappropriate that the original RED packet dropping probability and the average queue size are linearly related. It has been found in [12] that the link bandwidth is not fully utilized with a small average delay in the low-load scenario; thus, a smaller packet dropping probability should be used in order to improve the link utilization. The link bandwidth is fully utilized with a large average delay in the high-load scenario; thus, a larger packet dropping probability should be used in order to reduce the average delay.

When the average queue length is close to the minimum threshold, which indicates that the network congestion is not very serious, the packet dropping probability should be smaller than RED so that there will be fewer numbers of packets to be dropped to improve the utilization of the network. In the same way, the throughput could be enhanced. Meanwhile, the smaller the average queue length, the smaller the change rate of the packet dropping probability should be to ensure all the packet dropping probability is lower than RED in this stage. In other words, the slope of the packet drop probability curve increases slowly with the average queue length in this stage.

Contrarily, when the average queue length is close to the maximum threshold, which indicates that the network congestion is very serious, the packet dropping probability should be larger than RED to ensure that the queue does not overflow. Thus fewer packets will face forced drops and retransmissions, and this is how the average delay can be decreased, that changes with the average queue length. Additionally, the larger the average queue length, the greater the change rate of the packet dropping probability should be in this stage to avoid serious congestion earlier. In other words, the slope of the packet dropping probability curve increases moderately with the average queue length in this stage.

Considering the above requirements, we modified the packet dropping probability of RED, in which the packet dropping probability function is divided into two sections to distinguish between low and high traffic load conditions to achieve a trade-off in the delay and the throughput.

RED's original packet dropping probability can be defined as

$$P_d = P_{max} \times \frac{avg - Min_{th}}{Max_{th} - Min_{th}} \tag{1}$$

where P_d is the packet dropping probability, P_{max} is the maximum packet dropping probability, avg is the average queue length, Min_{th} is the minimum threshold that the average queue length must exceed before any packet marking or dropping is done, and Max_{th} is the maximum threshold that the average queue length must exceed before all packets are marked and dropped.

We first decided one target value below which we treat the traffic volume as low and above the target the traffic volume is high. We define the target value as

$$Target = Min_{th} + \frac{Max_{th} - Min_{th}}{2} \tag{2}$$

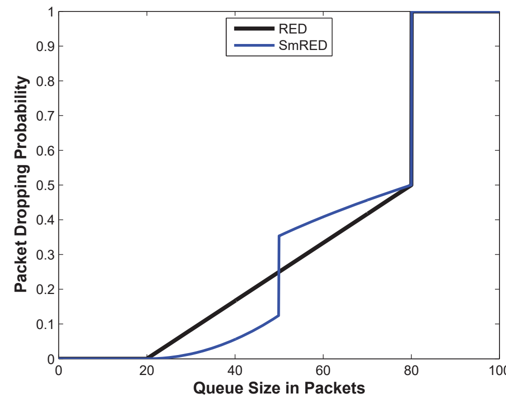

Fig. 2. The packet dropping probability function.

If the avg is below the target value, we set the packet dropping probability as

$$P_d = P_{max} \times \left(\frac{avg - Min_{th}}{Max_{th} - Min_{th}} \right)^2 \tag{3}$$

On the other hand, when the traffic volume starts getting high, i.e., if the avg is between $Target$ and Max_{th}, then the packet dropping probability function is defined as

$$P_d = P_{max} \times \sqrt{\left(\frac{avg - Min_{th}}{Max_{th} - Min_{th}} \right)} \tag{4}$$

Therefore, the expected SmREDs packet dropping probability function is depicted in Fig. 2 based on the aforementioned idea[1], where $Min_{th} = 20$, $Max_{th} = 80$, and $P_{max} = 0.5$. In summary, the expressions of SmREDs packet dropping probability being the function of the average queue length are shown in Eq. 5.

$$P_d = \begin{cases} 0, avg \in [0, Min_{th}) \\ P_{max} \times \left(\frac{avg - Min_{th}}{Max_{th} - Min_{th}} \right)^2, avg \in [Min_{th}, Target) \\ P_{max} \times \sqrt{\frac{avg - Min_{th}}{Max_{th} - Min_{th}}}, avg \in [Target, Max_{th}) \\ 1, avg \in [Max_{th}, +\infty) \end{cases} \tag{5}$$

Essentially, one cannot simultaneously have a high link utilization and low queuing delays. Therefore, a reasonable tradeoff is required between these two performance measures [13].

Moreover, delay-sensitive streams are tolerant of packet loss up to a specific limit. But if packets are discarded continuously, it is intolerable. Thus, when congestion is detected or the average queue size exceeds Max_{th}, the proposed strategy solves the problem by means of discarding packets in specific intervals. We set a counter to prevent packet loss continuously. A counter is used to count the interval from the last discard to present. When we configure the counter greater than zero, continuous discard can be prevented effectively for delay-sensitive streams and for delay-insensitive streams, we set

[1] We chose a step function at the target to distinguish sharply between the low traffic and the high traffic congestion scenario.

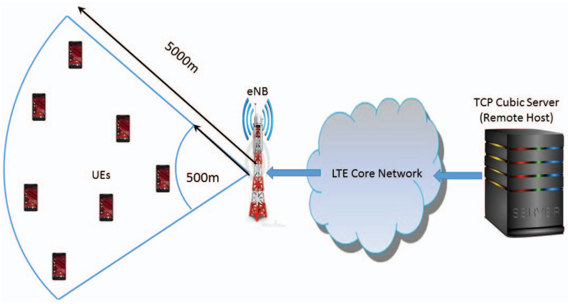

Fig. 3. The LTE network topology.

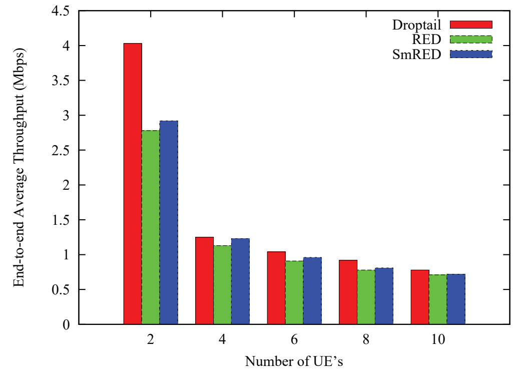

Fig. 4. End-to-end average throughput with increasing traffic load.

the counter to zero. The specific deployment of the counter depends on the requirements of QoS of the RLC SDUs.

IV. Simulation Result and Performance Evaluation

In order to systematically explore the interactions between the mobile network architecture and TCP, we have to delve deeper to consider mobile network elements and protocols. We implemented our proposed strategy using the LTE module of the $ns3$ [14] simulator configured with the topology depicted in Fig. 3. We used the LENA module [15] to create an end-to-end LTE network which has all the major elements of a real LTE system including the air interface Evolved UMTS Terrestrial Radio Access (E-UTRA) and Evolved Packet Core (EPC). The LTE model in $ns3$ provides a detailed implementation of various aspects of the LTE standard such as OFDMA, hybrid ARQ, and adaptive modulation and coding. The $ns3$ implementation follows detailed specification of TCP and 3GPP LTE. Hence, the results provided should be representative of what happens in a real system.

With the topology in Fig. 3, Guaranteed Bit Rate (GBR) video traffic are simulated, the remote host on the right side acts as sources, the UEs on the left side act as sinks. TCP data senders are of the TCP-Cubic type. We used TCP-Cubic as it is the default TCP congestion control algorithm in Linux OS in real networks. The TCP packet sizes are 1000 bytes. We set 15 resource blocks in eNB to serve the UEs. We also used the SISO transmission mode for both UE and the eNB. The remote host is connected to the LTE core network via wired link with the link capacity of 10 Mbps (1250 packets per second) and the propagation delay on this link is 50 ms. The application data rate is 100 Mbps. The thresholds for the packet dropping function are set as $Min_{th} = 20$ packets and $Max_{th} = 3 \times Min_{th}$ packets and the maximum packet dropping probability is set as 0.1. The eNB RLC buffer size is 100 packets. The total simulation time is 100 seconds.

To validate the improvement achieved by SmRED, we compared the performance of SmRED with Droptail and RED with the topology as shown in Fig. 3. We varied the number of UEs from 2 to 10 to simulate different traffic load conditions. The UEs are distributed randomly in the cell within a distance of 500 m to 5000 m from the eNB so that the UEs face different channel conditions. All UEs downloaded GBR Video traffic from a single server. We compared the download performance in terms of end-to-end average throughput and end-to-end average delay. The results are shown in Fig. 4 and Fig. 5. The figures show that the end-to-end average throughput decreases with the increase in traffic load. With low traffic load conditions, such as with 2 UEs, Droptail outperforms RED and SmRED in terms of throughput but at the cost of a significantly higher end-to-end average delay. Because Droptail has no mechanism to inform the TCP source about the possible congestion, the TCP window size grows significantly until the queue in the eNB RLC becomes full and continuous drop occurs. Therefore, it takes more time to resend those lost packets that incurs larger end-to-end average delay. SmRED outperforms RED in terms of throughput and delay.

When the traffic load is low, SmRED's throughput is higher than RED's throughput but as the traffic load increases, the end-to-end throughput of SmRED becomes almost the same as that of RED. This is because, when the traffic load is low, SmRED's dropping probability is low and thus fewer packets are dropped. So, the TCP window size does not shrink too frequently which helps to achieve higher throughput than that of RED. On the other hand, the end-to-end delay is almost the same as that of RED's end-to-end average delay when the traffic load is low, but as the traffic load increases, SmRED outperforms RED. This is because SmRED has a higher dropping probability than RED when the traffic load is high, which helps to inform the source earlier in advance than RED about the possible congestion. Thus, the TCP source reduces the window size in advance to avoid resending lost packets, which helps to achieve lower end-to-end average delay at the cost of lower throughput. Figure 6 shows the performance of SmRED and RED in terms of forced drops i.e., packets drop due to full queue. The figure shows that, when the traffic load is low, RED's forced drop is lower than SmRED but as the traffic load increases, SmRED significantly reduces the forced drop compared to that of RED, which helps to achieve lower end-to-end average delay.

Lower end-to-end delay is the most important criterion for delay-sensitive streams. But as we know, AQM algorithms are used to cooperate with the TCP source to manage congestion. However, delay-sensitive streams are always with the UDP protocol that does not cooperate with the TCP window control mechanism. Therefore, to effectively utilize our idea with delay-sensitive streams, cross layer information is necessary to control the congestion in the eNB RLC buffer. The LTE protocol stack in Fig. 1 shows that the PDCP layer is upon the RLC layer. The PDCP layer utilizes a discard timer to

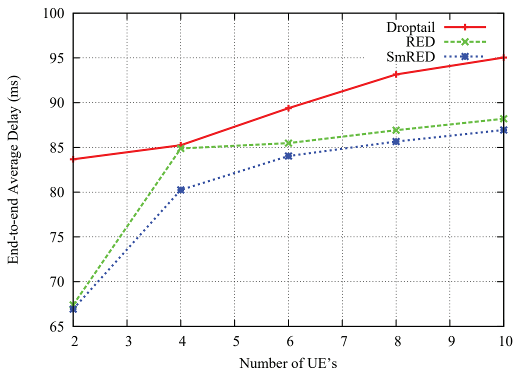

Fig. 5. End-to-end average delay with increasing traffic load.

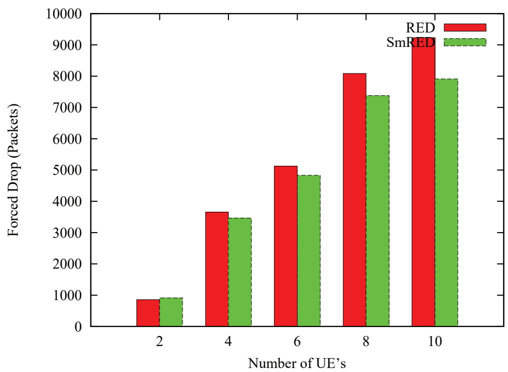

Fig. 6. Packet drop due to queue overflow.

discard packets [16]. When a packet enters from the higher layer, a maximum limit is imposed on the waiting time of a packet inside the queue. Packets are time stamped upon their arrival to the PDCP layer. When the discard timer expires, the packet is discarded. Thus when the RLC entity detects the congestion using SmRED, it will inform the PDCP layer to decrease the value of the discard timer and vice versa when the traffic load is low. But, based on congestion notifications from the RLC layer, how much the discard timer should be decreased depends on the QoS/QoE of particular applications and is a matter for further research. Due to the lack of space, we will address this topic in our future studies.

V. Conclusion

This paper presented a smart congestion control mechanism based on RED aimed at solving link under-utilization and large delay problems in low and high traffic-load scenarios for eNB RLC in the LTE network. An optimal RLC buffer occupancy at eNB can reduce the traffic volume, thereby achieving gains in user application performances. To this end, a buffer management scheme named SmRED is introduced and implemented in $ns3$ based LTE system simulator. The packet dropping probability corresponding to different traffic loads is non-linearly set in order to obtain good performance. Using $ns3$ simulation, SmRED effectively improves the drawbacks of RED, increases the throughput at a low load and reduces the delay at a high load. Lower end-to-end average delay is important to RLC SDUs for delay-sensitive streams. Furthermore, our congestion control strategy has stronger suitability to long time busy traffic than the RED one. This is an attractive performance criterion for voice stream. Using several different simulation scenarios, it is observed that limiting the RLC buffer capacity without buffer management leads to packet drops i.e. the tail drop phenomenon. According to the simulation results, although higher throughput is achieved by the Droptail scheme, it severely degrades user application performance in terms of end-to-end average delay. On the other hand, when the proposed buffer management scheme is used, a finely gained control on the RLC buffering delay can be attained. Our scheme is equally applicable to the delay-sensitive stream and delay-insensitive stream for eNB RLC buffer management. Further simulations will be performed in future work to evaluate SmRED's performance for the delay-sensitive stream as a cross layer approach.

References

[1] Z. Xunwei, L. Hui, and Z. Ming, "3GPP long term evolution: Architecture and specification," in *Post and Telecom Press*, 2010.

[2] S. Sesia, I. Toufik, and M. Baker, "LTE–the UMTS long term evolution:from theory to practice," in *Wiley*, 2011.

[3] J. Lakkakorpi and R. Cuny, "Comparison of different active queue management mechanisms for 3G radio network controllers," in *Wireless Communications and Networking Conference, WCNC 2006. IEEE*, April 2006, vol. 1, pp. 80–85.

[4] "3GPP TS 36.322 v10.0.0, evolved universal terrestrial radio access (E-UTRA)," .

[5] "Recommendation ITU-T G.714, international telephone connections and circuits general recommendations on the transmission quality for an entire international telephone connection," .

[6] "3GPP technical specification TS 23.40, general packet radio service (GPRS) enhancements for evolved universal terrestrial radio access network (E-UTRAN) access," September 2010.

[7] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Netw.*, vol. 1, no. 4, pp. 397–413, Aug. 1993.

[8] J. Wang, L. Rong, and Y. Liu, "A robust proportional controller for aqm based on optimized second-order system model," *Comput. Commun.*, vol. 31, no. 10, pp. 2468–2477, June 2008.

[9] R. Shorten, C. King, F. Wirth, and D. Leith, "Modeling tcp congestion control dynamics in drop-tail environments," *Automatica*, vol. 43, no. 3, pp. 441–449, Mar. 2007.

[10] B. Zheng and M. Atiquzzaman, "Dsred: improving performance of active queue management over heterogeneous networks," in *Communications, 2001. ICC 2001. IEEE International Conference on*, 2001, vol. 8, pp. 2375–2379 vol.8.

[11] W.C. Feng, D.D. Kandlur, D. Saha, and K.G. Shin, "A self-configuring red gateway," in *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, Mar 1999, vol. 3, pp. 1320–1328 vol.3.

[12] E. Plasser, T. Ziegler, and P. Reichl, "On the non-linearity of the red drop function," in *Proceedings of the 15th International Conference on Computer Communication*, Washington, DC, USA, 2002, ICCC '02, pp. 515–534, International Council for Computer Communication.

[13] Chonggang Wang, Bin Li, Y. Thomas Hou, Kazem Sohraby, and Yu Lin, "Lred: a robust active queue management scheme based on packet loss ratio," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, March 2004, vol. 1, p. 12.

[14] "The network simulator ns-3," https://www.nsnam.org.

[15] G. Piro, N. Baldo, and M Miozzo, "An lte module for the ns-3 network simulator," in *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, 2011, SIMUTools '11, pp. 415–422.

[16] U. Toseef, T. Weerawardane, A. Timm-Giel, and C. Gorg, "Lte system performance optimization by discard timer based pdcp buffer management," in *High Capacity Optical Networks and Enabling Technologies (HONET), 2011*, Dec 2011, pp. 116–121.