Uplink Data Compression for futuristic wireless networks

Shalini Govil, Rohit Kumar and Pavan Kumar Devarayanigari
Samsung Semiconductors India R&D Centre
Samsung Electronics
Bangalore, India

***Abstract*—3GPP Uplink Data Compression (UDC) improves the network resource utilization, reduces transmission time during high data transfer and improves user experience in poor signal condition. The paper discusses the issue of UDC packet loss arising due to UDC checksum failure as well as the limited scope of UDC interworking along-with new 3GPP features like PDCP duplication, RLC Out-of-Order delivery, Split bearer in dual connectivity and PDCP discard timer for effective QOS maintenance. The paper proposes a novel method to enhance UDC support across new 3GPP features. It also proposes two methods; a preventive approach and a recovery approach for handling packet loss issue. In our simulation on LTE test bed, method one saves 50% of network resources and recovery method recovers all UDC packet losses.**

Keywords— Deflate, Uplink Data compression, ultra-reliable low latency (u-RLLC), PDCP duplication

I. Introduction

3rd Generation Partnership Project (3GPP) Long Term Evolution (LTE) Release 15 (Rel-15) introduced Uplink Data Compression (UDC) feature with the motive to enhance uplink (UL) transmission (TX) for power starved/limited devices and at the same time help network scheduler, or evolved NodeB (eNB), to enhance the UL bandwidth usage/capacity. UDC is primarily effective for traffic types which have repetitive blocks. There are many traffic types which have shown great compression result [1]. UDC uses Deflate algorithm [2]. To ensure higher compression efficiency, Deflate updates the compression buffer every single data block compression. Therefore, the success of UDC depends on strict in-sequence delivery and compression buffer synchronization between transmitter and receiver node; failing which receiver may decode the content incorrectly. To ensure this, 3GPP configures UDC in Radio Link Control (RLC) Acknowledged mode (AM) only and deploys it at Packet Data Convergence Protocol (PDCP) level. Fig. 1. depicts the working of the UDC method on UE. SDU P1 is compressed to $P1'$. To check buffer synchronization at receiver, checksum C1 is computed using compression buffer with which data block was compressed. Thus U1 is compressed data containing $\mathrm{C1} + \mathrm{P1}'$ and subsequently compression buffer is updated with P1 content by left shifting the buffer with P1 size. As the compression buffer update is per SDU basis, this seems as a double edged sword for compression efficiency as writing the compression buffer with stale data may reduce subsequent efficiency. To address this, 3GPP provided user equipment (UE) with a choice and left the decision on UE's wisdom to transmit the data uncompressed and thus not alter the compression buffer content [3].

In the current state of art, including UDC over new radio (NR)/ new features is not being treated by 3GPP [4] for now. The 'strict in sequence delivery' requirement has also lead to some apprehensions on further extending existing UDC as data compression scheme [5]. Current paper attempts to target the dual issue of packet loss and narrowness of scope of UDC observed with current implementation.

Rest of the paper is organized in below format. Section II explores the limitations of current UDC method. Section III, presents novel solutions for handling data loss and enabling UDC interworking with new features. Further, Section IV talks about simulation results and Section V about conclusion.

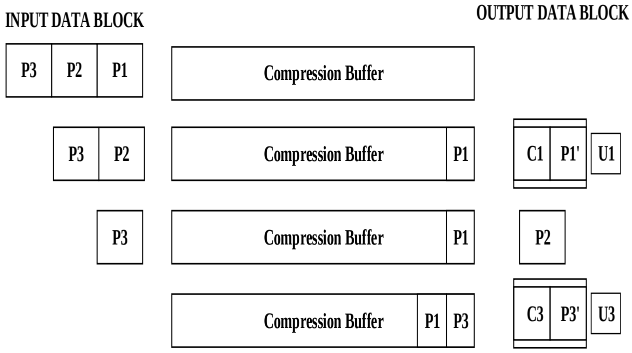

Fig. 1. 3GPP proposed UDC model

II. Limitations of UDC Method

A. Packet Loss Issue

In current system, on encountering checksum failure, receiving entity drops all subsequent compressed data PDUs (compressed with ongoing compression buffer), until it receives freshly compressed PDUs (indicated by PDCP header). The magnitude of dropped PDUs is proportional to delay (RTT) incurred in UDC packet transmission, processing at the receiver and/or transmitter and amount of uplink resource allocation. In other words, RTT time is a period starting from the (corrupted) compressed data block transmission time at UE to the time it learns about checksum failure. Fig.2. depicts the current UE and eNB behaviour in case of checksum failure. UE has 40 compressed PDUs ($\mathrm{U1} - \mathrm{U40}$; compressed size of 200Byte each) ready for transmission. PDUs $U1 - U20$ are transmitted on receiving grant from network. The receiving MAC entity (eNB-MAC), forwards the received PDUs to higher layers for processing and also sends grant to UE for further UL data transmission, over which compressed PDUs $U21 - U40$ are transmitted. Meanwhile, eNB PDCP encounters checksum error in $U20$, generates PDCP status report indicating the failure and discards all the PDUs $(U21 - U40)$ till it receives freshly compressed PDU.

B. Limited UDC Application

3GPP LTE Rel-15 introduced plethora of other features like (1) Split bearer, in which PDCP UL data can be transmitted via Radio Link Control (RLC) entities mapped to master cell group (MCG) and secondary cell group (SCG); (2) PDCP duplication, in which same PDCP data can be sent through multiple RLC entities and; (3) RLC-out-of-order delivery for acknowledged and un-acknowledged mode both, where receiving RLC entity need not wait for the data to arrive in sequence before submitting to higher layer. These features target higher throughput, at the same time achieving balanced data processing load where earlier system failed to achieve balanced processing load due to in-sequence delivery. Applying existing UDC over new features either results in (a) imbalance in packet processing by ensuring in sequence data processing or (b) leads to frequent checksum failure due to out of sequence data processing. Though, approach (a) is preferable. Either way, the system performance will be degraded. Exclusion of UDC from new feature limits the compression scope to very small traffic scenarios. Overall, these features seem to have done injustice for traffic mapped on UDC or vice-versa as they cannot be configured simultaneously and therefore head to limited application or curb network scheduling flexibility.

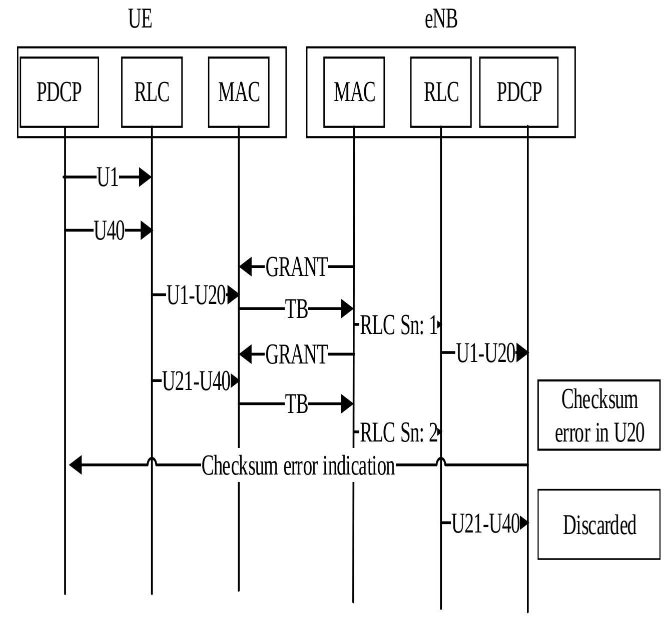

Fig. 2. Packet loss incurred in UDC method

III. PROPOSED SOLUTION

A. Preventive method to decrease packet loss

The method proposes to define a packet formation rate (PFR) for UDC enabled bearers to control the number of compressed PDUs available for transmission. PFR governs the amount of PDUs to be compressed at PDCP in a given time and is decided on the basis of grant rate, prioritized bit rate and uplink error rate observed. On encountering poor channel conditions or when network is not meeting the PBR/QCI allocation, PFR values decreases thus decreasing the number of generated compressed PDUs available for transmission. By decreasing the compressed PDUs available, it decrease the amount of packet lost due to RTT delay in case of checksum failure encounter but, it also leads to decreased compression efficiency in poor channel conditions.

B. Recovery method to reduce packet loss

The method proposes to maintain the successfully acknowledged RLC data PDU for additional $T2$ time. This $T2$ time is proportional to the Block Error Rate and Signal to Noise Interference ratio (SINR), RTT etc. On receiving UDC checksum failure before $T2$ timer expiry: (1) PDCP allocates successive new Sequence Number (SN), starting from first not transmitted SN, (2) Re-transmit $T2$ alive PDCP PDUs for which application layer ACK has not been received or PDCP SN for which Network faced checksum failure as indicated by the PDCP UDC control PDU. This $T2$ timer can be stand-alone UE implementation or can be configured by Network. On expiry of $T2$, all the acknowledged PDUs are freed. As shown in Fig. 4, by using $T2$ timer UE incurs 0% packet loss with few re-transmissions overhead due to UE computed corrupted PDCP SN. By using network indicated PDCP SN solution, the number of re-transmissions are 0%.

C. Method for interworking UDC with NR/New features

The method proposes to synchronize the compression buffer between transmitter and receiver periodically every T time and/or every N data block as defined by network configuration or based on aperiodic request. During the time, the same compression buffer, as $B1$, is used for (de)compressing the data block at receiver and transmitter side respectively. This will ensure immediate decompression at the receiver side (without checksum failure) even though data PDUs are delivered/processed out of sequence and thus ensuring balanced processing load. Fig. 3 depicts usage of semi-static compression buffer $B1$ for data compression for data block $P1$ to Pn and in parallel maintains a running compression buffer $B2$ which is updated every data block. $B1$ is updated with content of $B2$ once $B2$ is synched between transmitter and receiver. This synchronization can be achieved by exchanging a checksum computed solely based on $B2$ (thus ensuring reduced network resource consumption) or $B2$ itself.

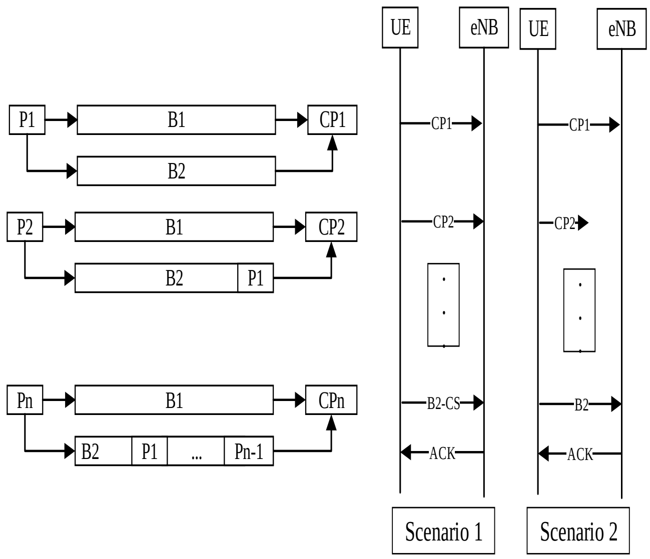

Fig. 3. UDC interworking with new features

IV. SIMULATIONS AND RESULTS

To simulate the UDC packet loss, PDCP, RLC and MAC behaviors were modelled based on 3GPP standard [3][7][6] for UE specifications (with and without proposed solution) and Next Generation NodeB (gNB)/eNB PDCP, RLC and MAC behaviors were modelled with the simulation parameters [3] as shown in Table 1. Network simulation involved modelling of the processing of received UL PDU and generating checksum failure randomly despite acknowledging the UL PDU at RLC level. Checksum failure indication was modeled for different values of RTT, i.e. 6, 7 and 9ms respectively. Fig. 4 depicts UDC packet loss observed for different UL data rate against different values of RTT. Fig. 5 depicts the result of proposed solution 1 for UDC packet recovery using fixed value of $T2$= 7ms (i.e., RTT=6 + 1 ms). In addition, it shows minor overhead of UDC packet re-transmissions due to UE computed PDCP SN of first corrupted UDC packet. In case of network provided PDCP SN for first corrupted UDC packet, the retransmission overhead is zero.

UDC Compression, performed with proposed solution 2 over Internet Protocol (IP) Multimedia Subsystem (IMS) signalling packets, shows average compression efficiency of 50-60% in comparison to 80-85% observed with traditional method [3]. Thus, solution 2 saves 50-60% of the UL resources while extending UDC scope to other features.

TABLE I. SIMULATION PARAMETERS

Poll re-transmit time (ms)	40
Poll PDU	5
Status prohibit time (ms)	30
Compressed packet size (byte)	300
Input data buffer size (byte)	20000

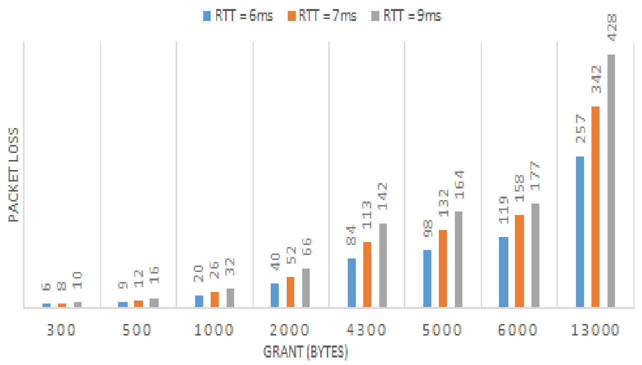

Fig. 4. UDC Packet Loss

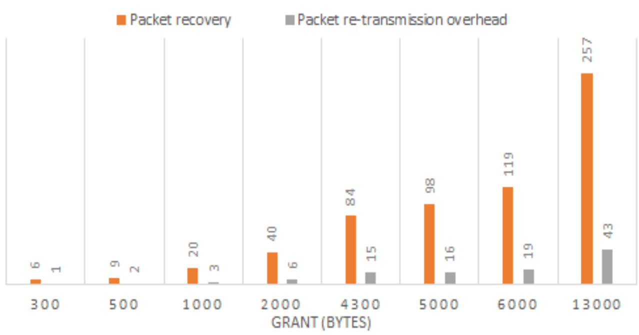

Fig. 5. Results from Packet recovery method

V. CONCLUSION

This paper discusses methods to reduce packet loss in UDC and proposes a novel method to support UDC with new LTE and NR features. By using a combination of static and dynamic buffers it removes the dependency of in sequence data reception with little effect on compression efficiency. By using T2 timer and PDCP level re-transmission, it also shows a high packet recovery. The method can be proposed to 3GPP too as other companies have started submitting their proposals on similar lines as in [4].

REFRENCES

[1] 3GPP, TR 36.754 - Study on Uplink (UL) data compression in LTE
[2] https://www.w3.org/Graphics/PNG/RFC-1951
[3] 3GPP, *TS 36.323* - Packet Data Convergence Protocol (PDCP) protocol specification
[4] 3GPP, R2-1901806 - Comparing RoHC and UDC Header Compression for TSN
[5] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
[6] 3GPP, *TS 36.321* - Medium Access Control (MAC) protocol specification
[7] 3GPP, *TS 36.322* - Radio Link Control (RLC) protocol specification