

Performance Evaluation of Adaptive RLC PDU size in HSPA+ Networks

1 Introduction

Release 7 of the High Speed Packet Access (HSPA) specifications introduced support for increased data rates on the downlink through features such as Multiple-Input Multiple-Output (MIMO) and Higher Order Modulation (HOM). To support such higher data rates, changes to Radio Link Control (RLC) and Medium Access Control (MAC) layers, described below, were also introduced.

Prior to Release 7, the RLC layer supported only fixed size RLC Protocol Data Units (PDUs), whose size was configured through Radio Resource Control (RRC) messages. Moreover, the pre-Release 7 MAC layer, MAC-hs, did not have the ability to segment RLC PDUs into smaller payloads. Thus, the fixed size of RLC PDUs was typically chosen to be a small value to allow RLC PDUs encapsulated in the MAC payload to be served to users at the lower geometries with acceptably low Block Error Rate (BLER). With a typical choice of 40 bytes for the RLC PDU size, the maximum downlink throughput was limited to ~4-5 Mbps (assuming an RLC Round-Trip-Time (RTT) of 100 ms and maximum RLC window of 2048).

To support the higher rates introduced in Release 7, the MAC layer was modified to support segmentation of RLC PDUs. This new MAC layer, MAC-ehs, allowed larger RLC PDU sizes to be configured, with the ability to segment RLC PDUs to smaller payloads, if required by lower geometry conditions. Release 7 also introduced changes to the RLC layer to support Flexible size RLC PDUs, which allowed the RLC PDU size to be changed during a data session. However, lack of knowledge of the channel at the Radio Network Controller (RNC), where RLC PDUs are built, meant that there was no good trigger to dynamically adapt the RLC PDU size, and the size of the RLC PDUs was fixed in practice.

With the Release 7 changes, the highest data rate without MIMO is ~21 Mbps. This requires an RLC PDU size of 200 bytes or larger (again assuming an RLC RTT of 100 ms and maximum RLC window of 2048) to support the peak data rate. To support the non-MIMO peak rate for 2/4 carriers (as in Dual-Cell HSDPA, DC-HSDPA or Four Carrier HSDPA, 4C-

HSDPA, standardized in Release 8, 9 and 10) would require an RLC PDU size of 400/800 bytes¹.

However, such large RLC PDU sizes may not be the most efficient for low geometry users. With such large RLC PDU sizes, segmentation of RLC PDUs using MAC-ehs is needed to reach the low geometry users, who usually can support only the small payloads (for example, 40 bytes). With an 800 byte RLC PDU size being sent to such a low geometry user, each RLC PDU could be segmented into 20 payloads, and the loss of any one of these causes the entire RLC PDU to be retransmitted. As an example, if the residual BLER is 1%, this could translate into 20% RLC retransmission rate² (assuming i.i.d. errors) and hence reduced throughput. The situation is a little less severe for 400 byte or smaller RLC PDUs, but the inefficiency still exists. Moreover, the UEs that are receiving the lowest throughputs in the cell, i.e., the low geometry UEs, are impacted the most, making a bad situation worse.

A logical solution to this issue is for the RLC PDU size to be adapted, depending upon the average channel quality seen by the UE.. This paper proposes a simple algorithm for feeding back channel information from the Node B to the RNC, and for the RNC to select an RLC PDU. The performance of this algorithm is then evaluated through simulations, lab and OTA testing.

Results show that a dynamic adaptation of the RLC PDU size overcomes the cell edge performance loss caused by selection of larger RLC PDU sizes. Our intuition is also that other simple algorithms for feeding back channel quality to the RNC may give similar performance as the algorithm described in this paper. In fact, one main contribution of this paper is to show that simple solutions do exist that allow large RLC PDU sizes to be used to reach peak rates, while still not degrading performance at the cell edge.

The paper is organized as follows. Section 2 discusses the interaction of RLC PDU size and RNC queue size. Section 3 explains our implementation of adaptive RLC PDU size. Section 4 presents our simulation setup and results. In Section 5 and Section 6, we present the setup and results for lab and Over-the-Air (OTA) tests respectively. Section 7 concludes the paper.

¹ In Appendix I, we summarize the RLC PDU sizes required to support the peak rates of different features in HSPA+ networks.

² In this paper, the *RLC retransmission rate* is defined as the number of retransmitted RLC PDUs divided by the total number of transmitted RLC PDUs.

2 Interaction of RLC PDU Size and RNC Queue Size

As mentioned in Section 1, the choice of a large RLC PDU size, when the UE is at the edge of the cell, can lead to a higher RLC retransmission rate, and potentially even multiple RLC retransmissions. Since each round of RLC retransmission adds up to 1 RLC RTT (typically ~100 ms) to the transmission time of a packet, multiple RLC retransmissions could easily cause a few hundred msec of “delay jitter” in the transmission time. This would lead to TCP ACKs arriving at the TCP server with a similar “delay jitter”. Depending upon the current value of the Retransmission Timeout (RTO) at the TCP server, this could lead to a TCP timeout, causing degradation in the TCP throughput.

The value of RTO, updated by the TCP server on the reception of TCP ACKs, depends upon the RTT as well as the Variance of the RTT [2]. In general, a larger Receive Window advertised by the UE and/or a larger RNC queue size lead to a larger RTT, thus dampening the effect of the TCP ACK’s “delay jitter” and reducing the probability of a TCP timeout.

In our simulations, lab and OTA testing, we assumed that the Receive Window as well as the RNC queue are provisioned so as to handle peak rates in the system. Specifically, we assumed a Receive Window of 250k bytes and RNC queue size of 300k bytes, which can support a peak rate of 20Mbps (assuming TCP RTT of 100 ms). At cell edge rates, this translates to a fairly large queue size as well as TCP RTT.

It should be noted that such a fixed RNC queue size has some obvious drawbacks: at low rates, the buffer occupied by a TCP flow can take a long time to be drained, thus reducing interactivity for a newer TCP flow, whose packets have to wait in queue behind the existing TCP flow’s packets. Some RNC implementations, thus, may adopt an Adaptive Queue Management (AQM) strategy, which maintains the RNC queue size for a UE as a function of the throughput seen by the UE. Such a strategy would reduce the RNC queue size if the UE were receiving low data rates, as is typical at the cell edge.

It should be noted that theoretically, an RNC queue size equal to the Delay Bandwidth product (where Delay is the RTT seen by the TCP server, and Bandwidth is the throughput seen by the UE) is considered large enough to keep the pipe full, i.e., to avoid any throughput reduction resulting from buffer under-run. A side effect of such an AQM strategy

is its interaction with a large RLC PDU size. Compared to fixed RNC queue size, AQM may lead to reduced RNC queue size when the UE is at the cell edge, leading to a smaller RTT for the TCP connection. As a consequence, a higher probability of multiple RLC retransmissions may be observed for the UEs with large RLC PDU sizes, and a TCP timeout may be triggered.

To summarize, use of AQM strategies may lead to adverse interactions with TCP flows, particularly when large RLC PDU sizes are used. Adapting the RLC PDU size to use smaller sizes for cell-edge UEs is likely to improve performance under such scenarios. Additional study is required to fully understand this impact.

3 Adaptive RLC PDU Size Implementation

Our implementation of Adaptive *RLC PDU size* involves the RNC selecting an RLC PDU size for a UE based on a channel quality metric reported by the NodeB. The procedure works as follows:

- The Channel Quality Indication (CQI) reported by the UE to the NodeB is filtered by the NodeB using an IIR filter, with a configurable time constant. When the UE is configured with more than 1 carrier (i.e., a DC-HSDPA/4C-HSDPA UE), the Node B computes the average of filtered CQIs from all carriers.
- This average filtered CQI is sent periodically to the RNC over the Iub interface. This requires changes to the Iub interface. Note that Iub implementations of most network vendors tend to be proprietary, so this could be introduced as a proprietary change, without making changes to Iub specifications. It may also be possible to use information contained in Iub flow control requests for adapting the RLC PDU size: this may also require no change to the specifications.
- The RNC then chooses the RLC PDU size based on the received filtered CQI value (the mapping is shown in the Appendix II).

4 Simulation Setup and Results

4.1 Simulation Setup

Our simulator implements all the layers of the protocol stack (PHY, MAC, and RLC) per 3GPP Release 7 specifications. Some of the key features are:

- TCP is modeled by connecting our simulator with the public simulator ns-2 [3], which has support for a variety of TCP congestion control protocols.
- The receive window advertised by the UE is 250k bytes. The RNC queue size is 300k bytes. These are large enough to support the maximum peak rate of ~21 Mbps for a non-MIMO single carrier UE.
- The UE supports an LMMSE Equalizer receiver.
- The HS-PDSCH scheduler selects the transport block size based on the filtered CQI value and an outer loop algorithm which targets 10% BLER after the first Hybrid ARQ (HARQ) transmission.
- Channel fading is Pedestrian-A with speed of 3km/h.
- The RLC Status Prohibit Timer is set to 100ms.
- The time constant of the IIR CQI filter is 1 second.
- The averaged filtered CQI is sent to the RNC every 100ms.
- RLC window size is set to 2048 RLC PDUs.
- The MAC layer used on the downlink is MAC-ehs, which supports segmentation of RLC PDUs.

Our simulation consists of a single carrier UE, downloading a large file using TCP NewReno [4] as the transport protocol. The parameters we vary are the UE geometry and the number of receive antennas at the UE. We compare the throughput of adaptive RLC PDU size with fixed size RLC PDU. We choose three fixed RLC PDU sizes, 40 bytes, 400 bytes and 800 bytes, the former chosen for optimizing cell edge performance and the latter two for peak rates. Note that 400 bytes and 800 bytes are the PDU sizes that would be required to achieve peak rates for DC-HSDPA and DC-HSDPA+MIMO respectively.

4.2 Simulation Results

Figure 1 shows the TCP throughput at two extreme values of UE geometry. At high geometry (15dB), we see a throughput loss for a dual Rx UE, when a small RLC PDU size (40 bytes) is used. The main reason for this is RLC window limitation. This loss is not seen for a single Rx UE, since the maximum throughput in this case is relatively smaller.

At low geometry (-3dB), use of large RLC PDU sizes (400 bytes and 800 bytes) leads to ~20% reduced throughput, compared to 40 bytes Fixed or Adaptive RLC PDU size, for both dual Rx and single Rx UEs. This is due to the significantly higher RLC retransmission rate in the case of the large RLC PDU sizes.

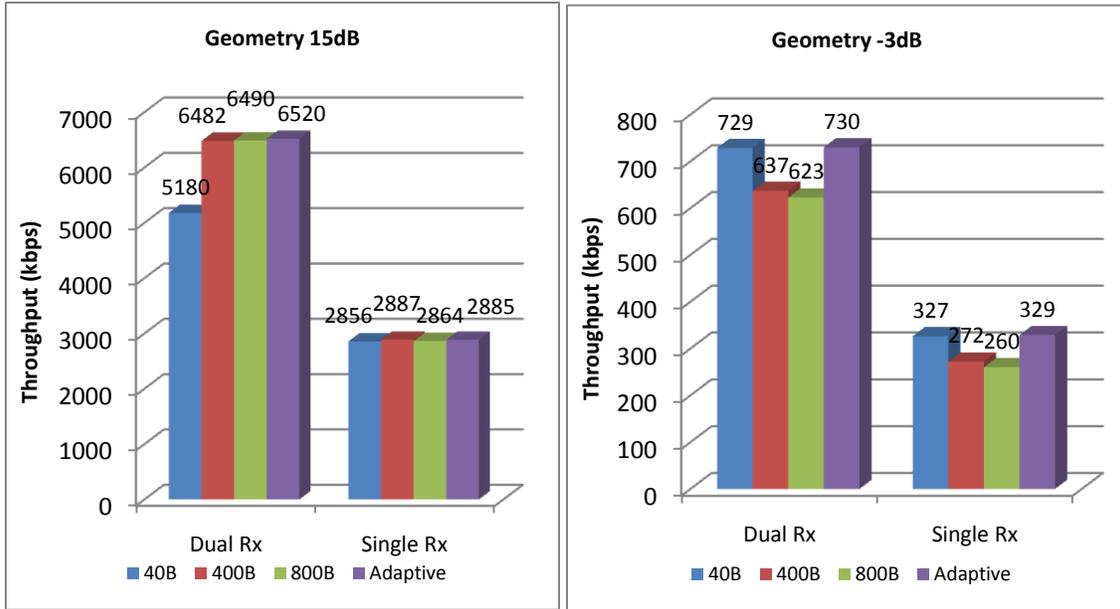


Figure 1: FTP Throughput at Different Geometries

Overall, we see that Adaptive RLC PDU size allows the UE to reach high rates at high geometry without being RLC window-limited, while at the same time, preventing any degradation in throughput at low geometry.

5 Lab Setup and Results

5.1 Lab Setup

Our prototype implements all the layers of the protocol stack per 3GPP Release 8 specifications. Our lab testing consists of a DC-HSDPA UE downloading a file over TCP. The key motivation to use DC-HSDPA UEs is the higher throughput supported, and the consequently larger RLC PDU size required.

In our lab setup, the main parameters we vary are geometry, type of fading channel, the number of receive antennas and the RLC PDU size (40 bytes, 400 bytes, 800 bytes, and Adaptive). Fading is modeled on both the downlink and the uplink using a channel emulator.

Each carrier goes through independent fading. Note that 800 bytes is the PDU size that would be required to achieve peak rates for a dual-carrier MIMO system, while 400 bytes is the PDU size required for a dual-carrier (non-MIMO) system.

5.2 Lab Results

Figure 2 shows the TCP throughput and RLC PDU retransmission rate at high geometry (10dB) with a PA3 channel. As in some of the earlier results, we see a throughput loss due to RLC window-limitation when the 40 bytes RLC PDU size is used.

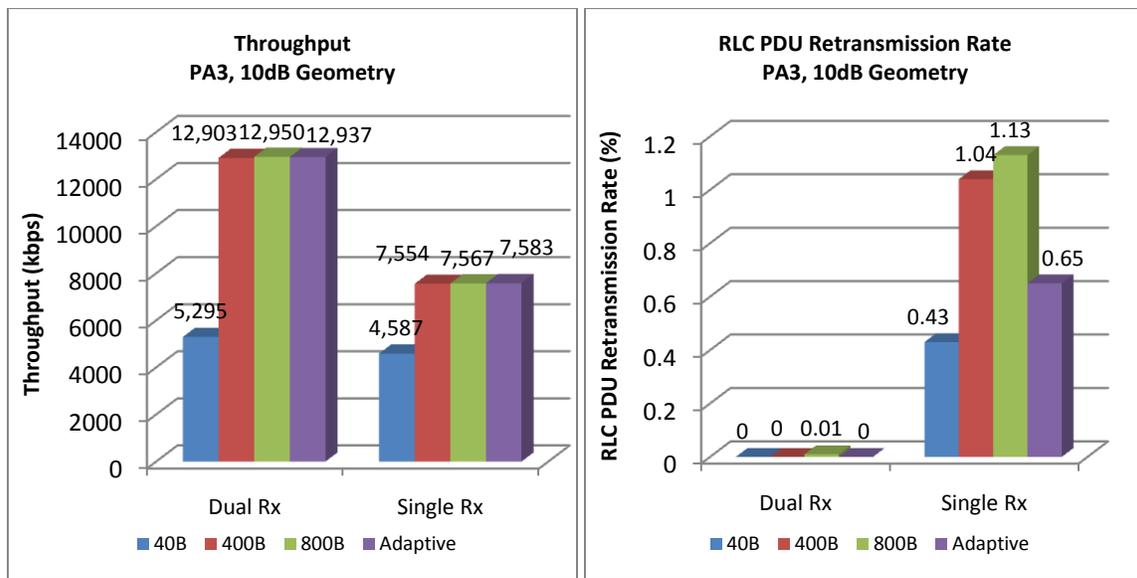


Figure 2: FTP Throughput and RLC PDU Retransmission Rate at 10dB Geometry with PA3 Channel

Figure 3 shows the TCP throughput and RLC PDU retransmission rate at low geometry (-6dB) with a PA3 channel. As in some of the earlier results, we see that using adaptive RLC PDU size can achieve similar throughput as small RLC PDU size, while using large RLC PDU sizes causes 10% to 25% throughput loss due to higher RLC PDU retransmission rate³.

³ The high RLC PDU retransmission rate for single Rx case is partially due to the high percentage of HS-SCCH mis-detection at such a low geometry.

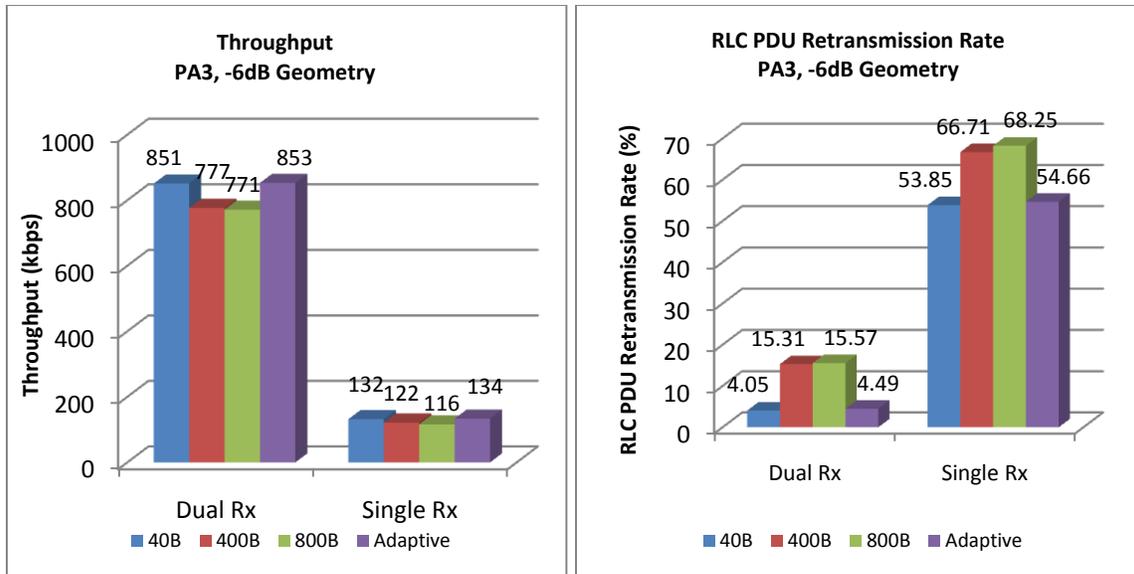


Figure 3: FTP Throughput and RLC PDU Retransmission Rate at -6dB Geometry with PA3 Channel

Figure 4 shows the TCP throughput at 10dB geometry with VA30 channel. We observe the performance loss due to RLC window limitation for the small RLC PDU size. The RLC retransmission rate is 0 for all the 10dB geometry cases.

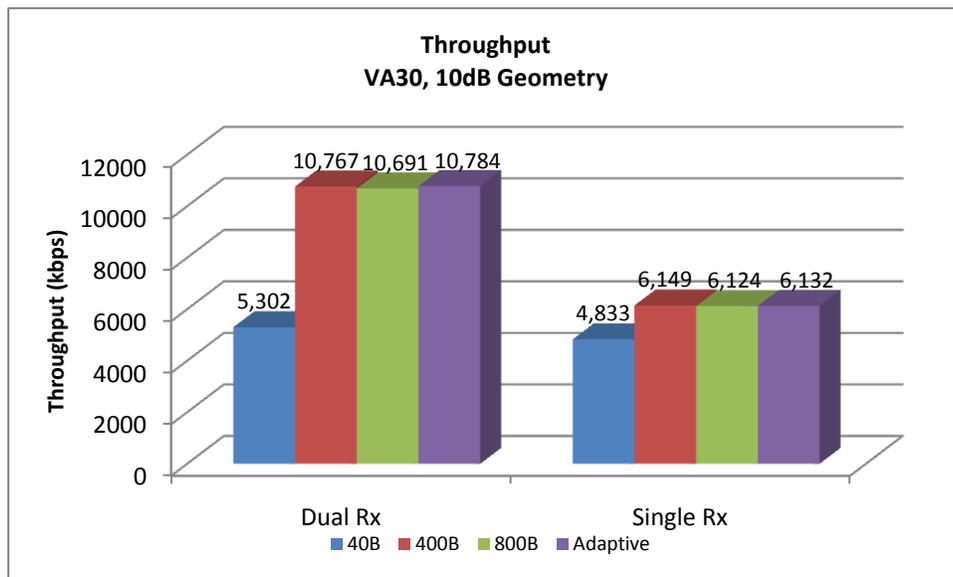


Figure 4: FTP Throughput at 10dB Geometry with VA30 Channel

In Figure 5, we present the TCP throughput and RLC PDU retransmission rate at -3dB geometry with VA30 channel. We see similar trends in VA30 results, as in PA3 results. One notable difference is that, for VA30 at -3dB geometry, a large RLC PDU size degrades the UE throughput (by ~15%) only for single Rx UEs. The main reason is that the VA30 channel does not show significant downfades for a dual Rx UE.

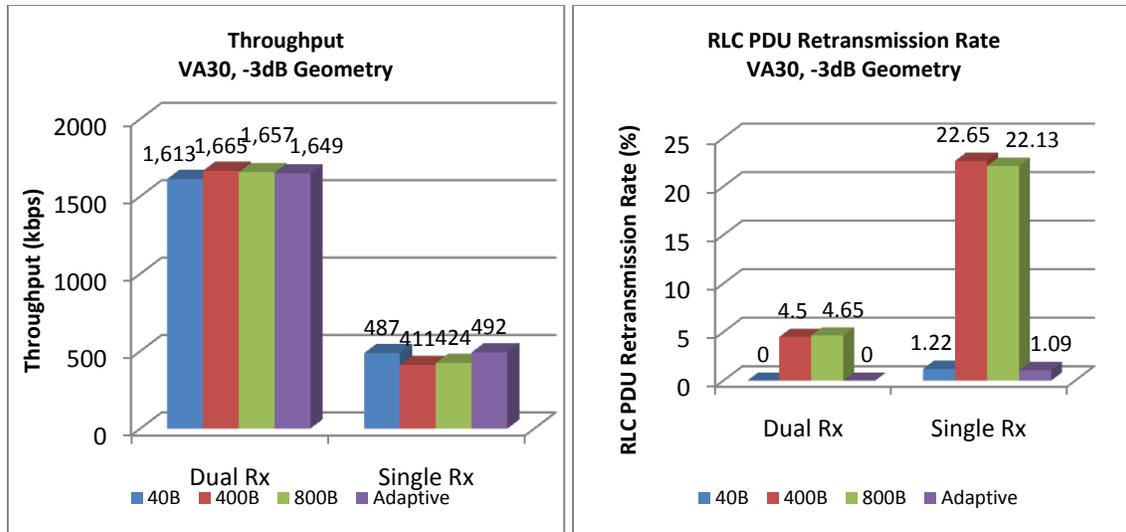


Figure 5: FTP Throughput and RLC PDU Retransmission Rate at -3dB Geometry with VA30 Channel

In summary, lab testing results show similar trends as simulations. For large RLC PDU sizes (400 bytes and 800 bytes), 10-25% throughput degradation is observed for cell-edge UEs, compared to small RLC PDU size (40 bytes) or Adaptive RLC PDU size .

6 OTA Setup and Results

6.1 OTA Setup

Our OTA setup consists of 3 Node Bs, as shown in Figure 6.

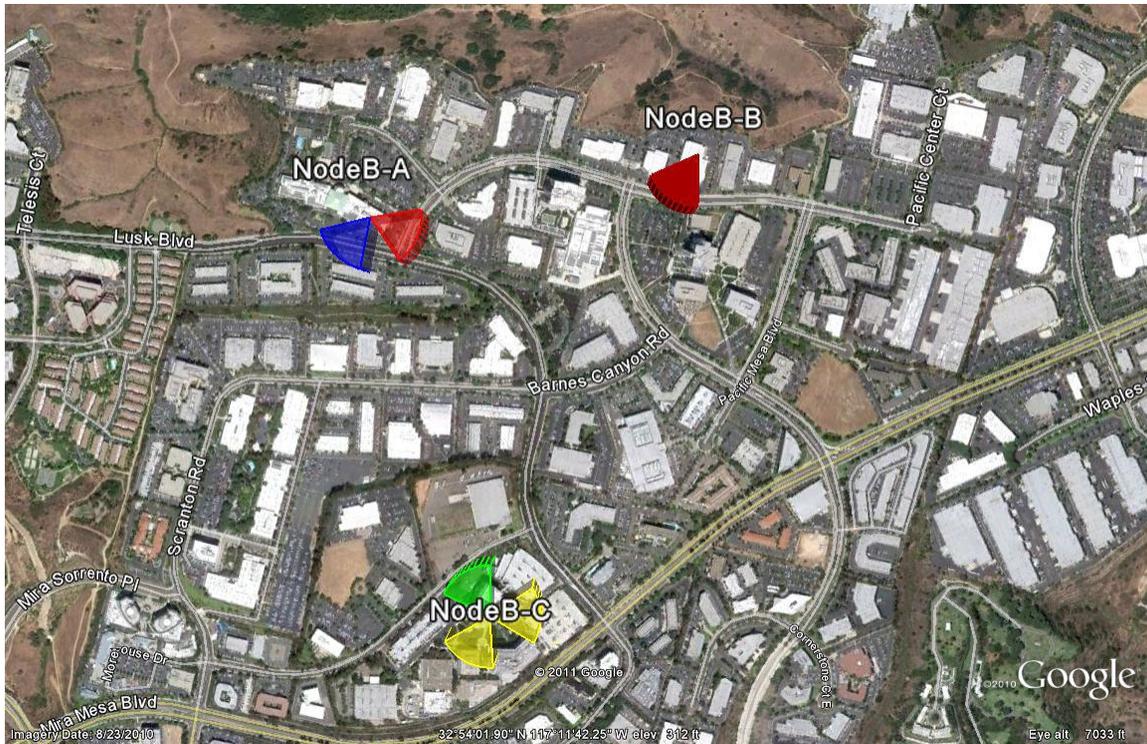


Figure 6: NodeB Locations of our OTA Setup (Image generated using Google Earth™)

In the OTA tests, we have a DC-HSDPA UE downloading a file over TCP with different RLC PDU sizes (40 bytes, 800 bytes, and adaptive). The following test scenarios are chosen for the OTA test:

- **High CQI High Speed Route:** The average CQI seen by a UE with a single receive antenna is 23 (corresponding to 9dB geometry). The map of the drive route is shown in Figure 7.
- **Low CQI Stationary Test:** During this test, the UE is stationary in a parking lot. We use a fixed location, since in our prototype network a vehicular route may not be able to provide a consistent low CQI route. The average CQI seen by a UE with a single receive antenna is 8 (corresponding to -6dB geometry). The stationary test location is shown in Figure 8.

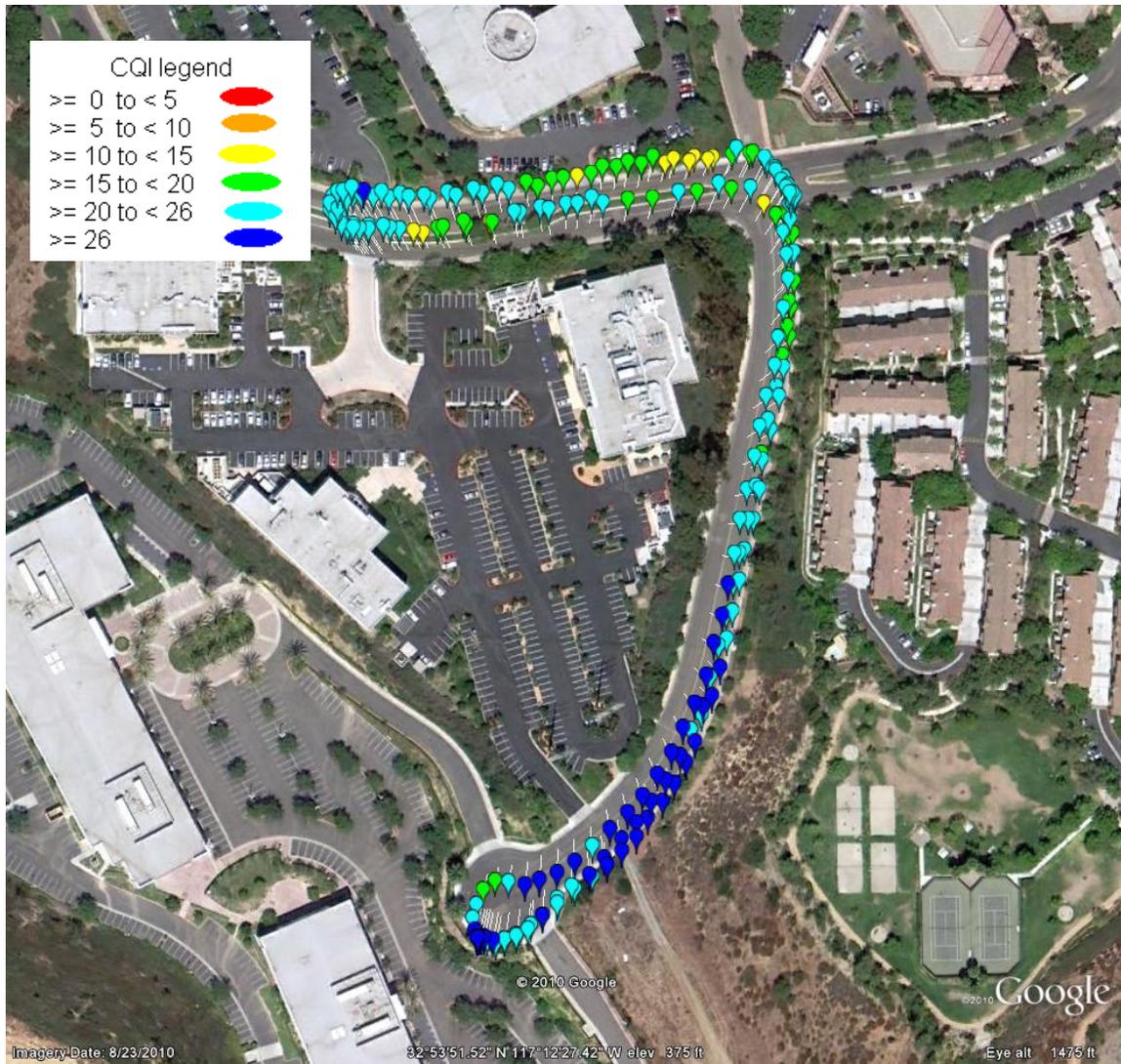


Figure 7: High CQI High Speed Drive Route (Image generated using Google Earth™)



Figure 8: Low CQI Stationary Test Location (Image generated using Google Earth™)

Figure 9 shows the cumulative distribution functions (cdfs) of the CQI for the two test scenarios.

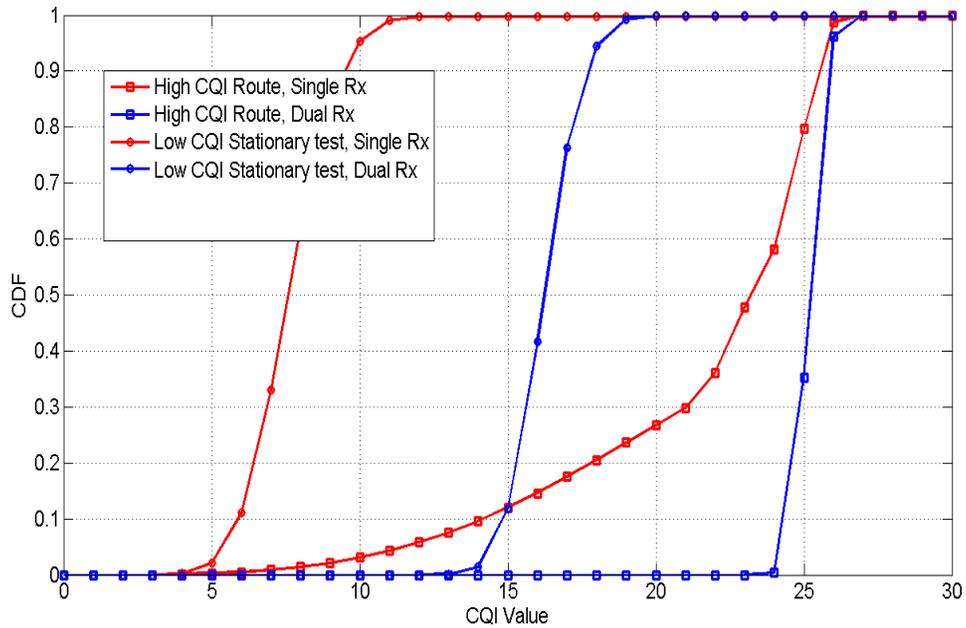


Figure 9: CQI Distribution of OTA Tests

6.2 OTA Results

Figure 10 shows the TCP throughput in the high CQI route. As expected, adaptive RLC PDU size achieves the same throughput as the large RLC PDU size, while the throughput of the small RLC PDU size suffers from RLC window limitation.

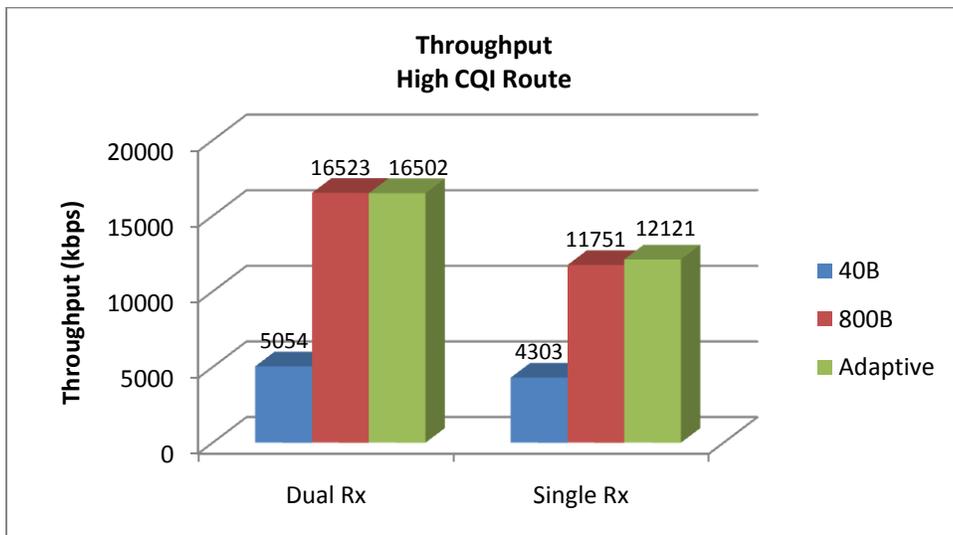


Figure 10: FTP Throughput in high CQI route

Figure 11 shows the TCP throughput and RLC PDU retransmission rate for the low CQI stationary test. A gain of 15-35% is seen for Adaptive RLC PDU size compared to a large RLC PDU size. An RLC retransmission rate of ~15% is seen for the large RLC PDU size.

It is worth noting that we did not see TCP timeouts in any of the simulation, lab or OTA cases presented in this paper. We expect to see occurrences of such TCP events, if queueing mechanisms such as AQM are used at the RNC.

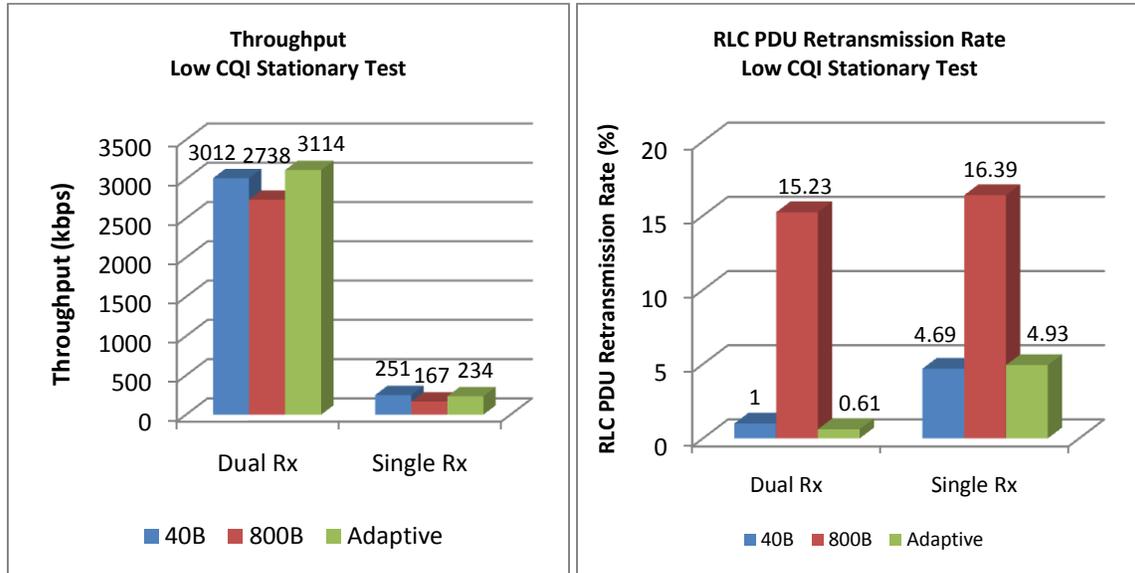


Figure 11: FTP Throughput and RLC PDU Retransmission Rate in Low CQI Stationary Test

7 Conclusions

In this paper, we compared the throughput of adaptive RLC PDU size with fixed RLC PDU size through simulations, lab and OTA tests. Results showed that use of adaptive RLC PDU size helps in achieving the conflicting goals of not being RLC window-limited at high geometries, while avoiding throughput degradation (due to RLC PDU retransmissions) at low geometries.

References

- [1] 3GPP TS25.322, Radio Link Control (RLC) protocol specification, v7.10.0, June. 2009
- [2] RFC 2988 - Computing TCP's Retransmission Timer

[3] ns2 simulator, available online at <http://www.isi.edu/nsnam/ns/>.

[4] RFC 3782 - The NewReno Modification to TCP's Fast Recovery Algorithm.

Appendix I: Minimum RLC PDU Size Required for Peak Rate for Different Combinations of Features

Features	Release	Peak Rate (Mbps)	Minimum RLC PDU size (bytes)
64-QAM	Release 7	21	200
DL MIMO	Release 7	28	300
64-QAM + MIMO	Release 8	42	400
DC-HSDPA	Release 8	42	400
DC-HSDPA +MIMO	Release 9	84	800

Table 1: Minimum RLC PDU size required for different combinations of features

Appendix II: Filtered CQI to RLC PDU size Mapping Table

Filtered CQI value	RLC PDU size (bytes)		Filtered CQI value	RLC PDU size (bytes)	
	Single-Carrier	Dual-Carrier		Single-Carrier	Dual-Carrier
1	40	40	16	100	100
2	40	40	17	100	100
3	40	40	18	100	200
4	40	40	19	100	200
5	40	40	20	100	200
6	40	40	21	100	200
7	40	40	22	100	200
8	40	40	23	200	300
9	40	40	24	200	300
10	40	40	25	200	300
11	40	40	26	200	500
12	40	40	27	300	500
13	40	40	28	300	500
14	40	100	29	300	500
15	40	100	30	300	500

Table 2: Filtered CQI to RLC PDU size Mapping Table