

## Publication P2

Jani Lakkakorpi and Renaud Cuny. 2006. Comparison of different Active Queue Management mechanisms for 3G Radio Network Controllers. In: Proceedings of the 2006 IEEE Wireless Communications and Networking Conference (WCNC 2006). Las Vegas, Nevada, USA. 3-6 April 2006, volume 1, pages 80-85.

© 2006 IEEE

Reprinted with permission.

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of Helsinki University of Technology's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to [pubs-permissions@ieee.org](mailto:pubs-permissions@ieee.org).

By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

# Comparison of Different Active Queue Management Mechanisms for 3G Radio Network Controllers

Jani Lakkakorpi  
Nokia Technology Platforms  
Espoo, Finland  
jani.lakkakorpi@nokia.com

Renaud Cuny  
Nokia Networks  
Espoo, Finland  
renaud.cuny@nokia.com

**Abstract**—In WCDMA networks, Radio Network Controller (RNC) is a potential bottleneck for TCP connections – due to difference in available bandwidth between the Core Network (CN) and the Radio Access Network (RAN). This may result in Packet Data Convergence Protocol (PDCP) buffer overflows or excessive delays at the RNC, as PDCP buffers are flow-specific. In order to avoid PDCP buffer overflows, different Active Queue Management (AQM) methods may be applied at the RNC. This paper is a comparison of four AQM schemes: three well-known ones and one new – TTLRED.

## I. INTRODUCTION

The scope of this paper is to study the end-to-end performance of TCP-based Radio Access Bearers (RAB) in a WCDMA network with or without Active Queue Management (AQM) in the Radio Network Controller (RNC). We shall take High Speed Downlink Packet Access (HSDPA) into account, since the effective air interface bandwidth that a user gets may in that case vary more dynamically than in the case of dedicated channels (DCH). However, it should be noted that HSDPA flows have dedicated buffers in the RNC, too.

DCH flows usually have more constant data rates than HSDPA flows. Nevertheless, these (TCP-based) flows may experience long buffering delays in the RNC, as the downlink direction is more likely to be a bottleneck than the uplink direction. Thus, techniques that aim at keeping the buffer size small are useful in both DCH and HSDPA cases.

The rest of this paper is organized as follows: section 2 presents the WCDMA network (with special emphasis on the RNC), section 3 presents a few well-known AQM schemes for the RNC, section 4 proposes a new AQM scheme (TTLRED), section 5 validates the proposed scheme and compares its performance against the other schemes through simulations, while section 6 concludes the paper.

## II. WCDMA NETWORK AND RNC

RNC is responsible of RAB admission control and radio resource management, i.e., allocating bit rates to RABs – in the case of both dedicated and shared radio channels [1].

Gateway GPRS Support Node (GGSN) and Serving GPRS Support Node (SGSN) are consulted in RNC's admission control decisions. Moreover, Random Early Detection (RED) [2, 3] and different scheduling weights are applied for different traffic classes. However, we consider neither SGSN nor GGSN as a system bottleneck in this paper. Fig. 1 illustrates the WCDMA packet data user plane protocol stacks that are used between the different network elements.

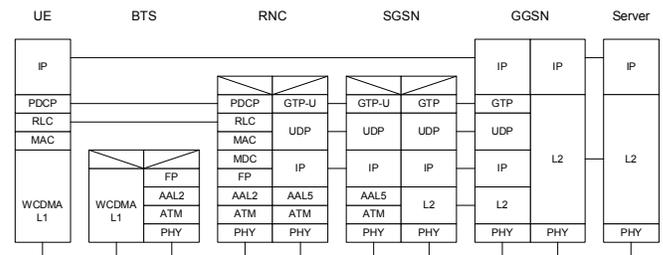


Fig. 1. WCDMA packet data user plane protocol stacks.

### A. PDCP and RLC Buffers

In the RNC, there are dedicated Packet Data Convergence Protocol (PDCP) [4] and Radio Link Control (RLC) [5] buffers for each RAB. The maximum RAB buffer length may depend on the traffic class. For example, 10 kB might be used for the conversational class and 30 kB for all other traffic classes. Naturally, there has to be a limit for the total amount of buffer memory. However, we do not consider that as a bottleneck.

Unacknowledged Mode (UM) and Acknowledged Mode (AM) RLC on Dedicated Traffic Channel (DTCH) and Dedicated Control Channel (DCCCH) have fixed-size buffers. The default size for both transmission and receiving buffer is the maximum AM window size times the maximum Protocol Data Unit (PDU) size. The same buffer size is used for UM RLC. For example, the following values might be used:

- DCCCH: maximum window size 256 PDUs, maximum PDU size 20 B
- DTCH: maximum window size 768 PDUs, maximum PDU size 42 B

It should be noted that the same data is stored in both PDCP and RLC buffers. This enables the use of various features in PDCP buffering, e.g., GPRS Tunnelling Protocol (GTP) packet reordering. Moreover, features that target in enhancing the performance of TCP can be implemented with the help of PDCP buffering.

### B. RLC/MAC Protocols

In Transparent Mode (TM) and UM RLC, higher layer packets are simply segmented and equipped with appropriate overhead before they are sent to the UE or the RNC. In AM RLC, however, RLC frames are not cleared from the retransmission buffer until they have been acknowledged. Moreover, the higher layer packet cannot be cleared from the PDCP buffer until its final RLC frame is cleared from the RLC buffer [5].

Acknowledgements can be polled in many different ways, e.g., by using counters and timers. When an RLC frame with the poll bit is received, the receiver answers either with a standalone acknowledgement or a piggybacked one. The standalone acknowledgements are put to the tail of the RLC buffer while the piggybacked acknowledgement information is added to the first non-pending RLC frame in the RLC buffer.

When an acknowledgement is received, the retransmission buffer is cleared from those frames whose identifier is found from the acknowledgement. All other pending RLC frames, whose identifier is lower than the highest identifier in the acknowledgement, are considered lost and they need to be retransmitted. Naturally, there is a limit for the number of retransmissions.

### C. HSDPA and its Flow Control

HSDPA [6] is a concept within WCDMA specifications whose main target is to increase user peak data rates and Quality of Service (QoS), and to generally improve spectral efficiency for downlink asymmetrical and bursty packet data services.

When implemented, the HSDPA concept can co-exist on the same carrier as the current Release'99 WCDMA services. Furthermore a user can download packet data over HSDPA, while at the same time having a speech call. HSDPA offers theoretical peak data rates on the order of 10 Mbps, and in practice more than 2 Mbps.

Compared to the Release'99 architecture, HSDPA introduces a short 2 ms transmission time interval (TTI), adaptive modulation and coding (AMC), multicode transmission, fast physical layer (L1) hybrid ARQ (H-ARQ), and it moves the packet scheduler from the RNC to the Node-B where it has easy access to air interface measurements. The latter facilitates advanced packet scheduling techniques, meaning that the user data rate can be adjusted to match the instantaneous radio channel conditions [7].

In all HSDPA flow control [8] implementations, it is the MAC-hs at Node-B that is controlling the flow over the Iub interface, i.e., MAC-hs allocates a certain number of credits per time unit for each HSDPA buffer in the RNC. Each credit allows HS-DSCH frame protocol in the RNC to send a single MAC-d

[9] PDU during the allocation interval. The implementation details of HSDPA flow control are left to the network equipment manufacturers.

HSDPA frame protocol is used between the RNC and the UE: 1 to 255 MAC-d frames are packed into a single FP frame every 10 ms (other TTIs can be used as well). The number of MAC-d frames in a single HSDPA FP frame naturally depends on flow control and on the number of buffered MAC-d frames [10].

## III. RELATED WORK

Here we describe different AQM mechanisms that could be applied in the RNC in order to reduce buffering delays.

### A. Random Early Detection

Probably the best-known method for TCP performance enhancement, RED (and Explicit Congestion Notification, ECN) [2, 3, 11, 12] aims at preventing TCP synchronization. However, RED does not usually work well if there is only a single flow or a couple of flows sharing a buffer: the buffer occupancy will vary a lot, and we are forced to use the instantaneous queue length instead of a slowly averaged one. This is exactly the case with RNC and its PDCP buffers. Thus, other possible methods should be tested, too.

### B. Adaptive RED Thresholds Based on RAB Rates, ARED

Static RED parameters can lead into decreased TCP goodput or too high IP packet delays. Variable *minTh* and *maxTh* would thus be preferred over static values. In the RNC we can simply utilize RAB rate (e.g., through the number of HSDPA credits received from the Node-B) and then compute minimum threshold by multiplying the RAB rate [B/s] by desired *delay* (e.g., 0.5 s) (1). Minimum *minTh* is limited to 7500 bytes. Maximum threshold is then given by a widely used rule of thumb (2).

$$\text{minTh} = \max(7500, \text{delay} * \text{rate}) \quad (1)$$

$$\text{maxTh} = 3 * \text{minTh} \quad (2)$$

### C. Packet Discard Prevention Counter

In [13] Sgfors, Ludwig, Meyer and Peisa have presented a technique called Packet Discard Prevention Counter (PDPC). They propose a deterministic packet dropping mechanism using a counter: after a packet drop we have to accept *N* (default: 20) packets before we can drop another packet – unless maximum threshold is exceeded. This technique assumes that we have adaptive *minTh* and *maxTh*. Details, however, are not provided in [13]. Thus, we shall simply utilize RAB rate here as presented in the previous section. Drop from tail shall be used instead of drop from front (suggested in [13]), since drop from front is not too easy in the RNC if the IP packet is already segmented into RLC blocks and given a sequence number – and not at all feasible if RLC blocks have already been sent.

## IV. TTLRED

In time-to-live based RED, i.e., TTLRED, there is no need to estimate the RAB rates. Instead, we provide the IP packets with

timestamps as they enter the PDCP buffer. This is a bit similar to assigning PDU lifetimes (the remaining time period that the PDU is considered as valid) in the 2G-SGSN [14] and the lifetime packet discard idea proposed by Gurtov and Ludwig [15]. Those schemes, however, have very little to do with AQM. Next, we present two alternative schemes that utilize the timestamp.

#### A. TTLRED for Incoming Packets (TTLRED1)

When we apply TTLRED for incoming packets, we simply replace the averaged queue size in the gentle RED algorithm [2] (see Fig. 2) with a packet lifetime. Whenever a packet arrives, we find the packet with the highest lifetime (current time less timestamp) from the PDCP buffer. Drop counter is also utilized; only every  $N^{\text{th}}$  packet can be dropped (or marked if ECN is used) – even if the high threshold is exceeded.

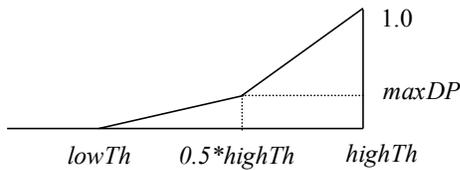


Fig. 2. TTLRED: dropping probability as a function of packet lifetime.

#### B. TTLRED for Buffered Packets (TTLRED2)

Another alternative is to utilize timers. In this approach, the packets are given a random dropping (or marking) time when they enter the PDCP buffer. Packet marking algorithm is illustrated in Fig. 3 and it is somewhat analogous to gentle RED [2]. Moreover, a higher lifetime is always assigned when the buffer is empty or whenever a new TCP flow is detected, as it could be a SYN packet.

```

if (U(0, 1) < maxDP) {
    lifetime := U(lowTh, 0.5*highTh)
} else {
    lifetime := U(0.5*highTh, highTh)
}

```

Fig. 3. TTLRED packet marking algorithm.

The packets are checked every  $T$  seconds (e.g., 10 ms). If current time exceeds the packet timestamp, the packet is dropped (or marked if ECN is used and the flow supports it). Since some packets are stored both in the RLC buffer (size usually around 30 kB) and in the PDCP buffer simultaneously, TTLRED will remove only such PDCP packets that do not have any corresponding RLC blocks in flight yet. Moreover, we stamp the PDCP sequence number only when the PDCP data is copied to the RLC buffer, and these sequence numbers are reassigned in the event of packet drop as otherwise we would have problems with in-sequence delivery.

## V. PERFORMANCE EVALUATION

We use a modified version of the ns-2 simulator [16]. Six simulations are run in each test case in order to get 95% confidence intervals. Simulation time is 1200 seconds. The different AQM mechanisms are tested under numerous different

conditions. We vary the downlink bearer rate (from 64 kbps to 3.6 Mbps, DCH or HSDPA; uplink is always a 64 kbps DCH RAB), number of parallel TCP connections (one or three), TCP’s advertised window (30 or 60 packets) and PDCP buffer size (30 kB or 100 kB). Moreover, different file sizes are tested (250 kB vs. 2.5 MB) in the case of single TCP connection.

#### A. Simulation Parameters

One-way core network delay between server and RNC is set to 70 ms. The only bottleneck (excluding TCP performance) in our system is the RNC. There are no packet drops or variable delays elsewhere in the system. Acknowledged Mode (AM) RLC with polling- and timer-based retransmissions as well as duplicate detection and in-sequence delivery are assumed. HSDPA flow control is simple: the radio capacity is simply divided equally among all currently active flows and multiplied by 0.9. For DCH air interface packet loss, a simple Gilbert model is used. Average packet loss rate is 1.5%<sup>1</sup>. This model is not applied with HSDPA but local retransmissions are modeled as additional delay: with 10% probability a packet has to wait until the next scheduling period, i.e., 0 – 10 ms.

For static RED (as well as for ARED and PDPC, where applicable), we apply the following parameter values:

- $minTh = 10$  kB (33.3 kB with 100 kB PDCP buffer)
- $maxTh = 30$  kB (100 kB with 100 kB PDCP buffer)
- $maxDP = 0.1$
- $w_{AQS} = 1.0$
- Drop from tail

For TTLRED, we apply the following parameter values:

- $lowTh = 0.25$  s (TTLRED1) or 0.5 s (TTLRED2)
- $highTh = 1.5$  s (TTLRED1) or 3.0 s (TTLRED2)
- $maxDP = 0.1$

We simulate two kinds of TCP traffic: web browsing and “peer-to-peer” file downloading. For both types, we utilize the following hierarchical HTTP traffic model:

- HTTP/1.0 utilizing four parallel TCP connections with web browsing or HTTP/1.1 [17] utilizing single TCP connection and pipelining with file downloading.
- Session length equals simulation time.
- Page reading time is six seconds.
- Inline objects per page: three in web browsing and zero in file downloading (single file, i.e., main page only).
- Request size [bytes]: lognormal distribution with a mean of 5.93 and a standard deviation of 0.32.
- Object size for the main page and the inlines: 65 kB in web browsing and 250 kB / 2.5 MB in file downloading.

<sup>1</sup> Since AM RLC (with retransmissions) is used, packet loss on the air interface is seen as additional delay.

- TCP NewReno [18] with the ns-2 default values. The performance of TCP Reno [19] is not studied here, because most current real-life TCP implementations are already closer to NewReno.
- MTU of 1500 bytes, overhead of 60 bytes (TCP+IPv6).

### B. Simulation Results

Fig. 3–12 and Table 1 illustrate the main results (95<sup>th</sup> percentile delay and TCP goodput) of our simulations. Based on our simulations, we have made the following observations:

- The main benefit of AQM is decreased delay with lower RAB rates. This can be important if the same RAB carries different flows, some of which are delay-sensitive. Big PDCP buffers, narrow bandwidth, big advertised windows and tail dropping can lead to high delays.
- PDCP buffers should be large enough for maximized TCP goodput. Bandwidth-delay product formulas (see, e.g., [20]), however, should be used with caution, as PDCP buffers are not normal router buffers (AM RLC, HSDPA flow control). 100 kB seems to be enough even for HSDPA RABs (assumed RTT of 80 ms and theoretical

maximum bandwidth of 10 Mbps). If the PDCP buffers are large enough, AQM does not increase TCP goodput – things would most probably be differently with very small PDCP buffers (see, e.g., [13]). AQM may actually decrease TCP goodput, but this depends on the selected parameters. We chose to limit the delay and pay the price of slightly decreased goodput.

- Static RED may unnecessarily lower TCP goodput with high RAB rates, especially with big files (see Fig. 12).
- ARED, PDPC and TTLRED give the best results.
- TTLRED for incoming packets can result in higher delays than the other AQM schemes, since there is no protection against bigger bursts (see Fig. 5).
- We also noticed that if the parameters are non-optimal, TTLRED for buffered packets could result in lower TCP goodput especially with the following combination: high bandwidth, multiple TCP flows per RAB, big buffers and big advertised windows. With multiple flows, we could think of downgrading the advertised windows a bit – but this kind of action would break the end-to-end principle.

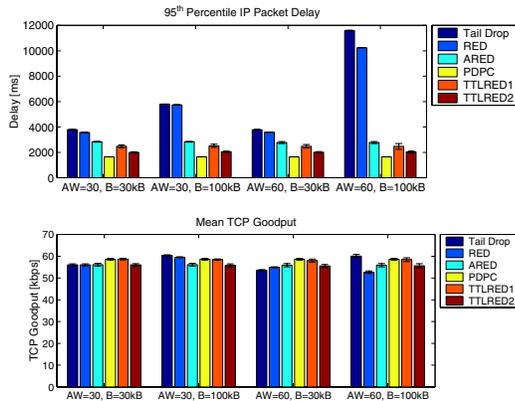


Fig. 4. R99 64 kbps, single flow: delay and goodput.

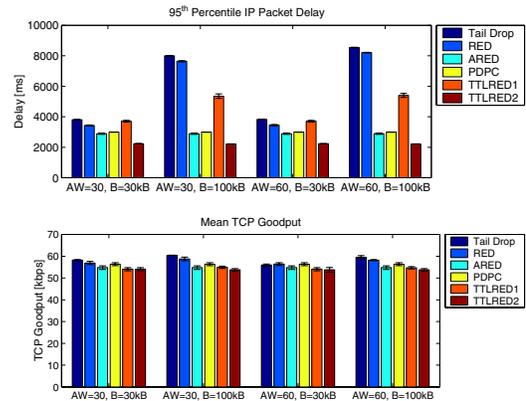


Fig. 5. R99 64 kbps, four flows: delay and goodput.

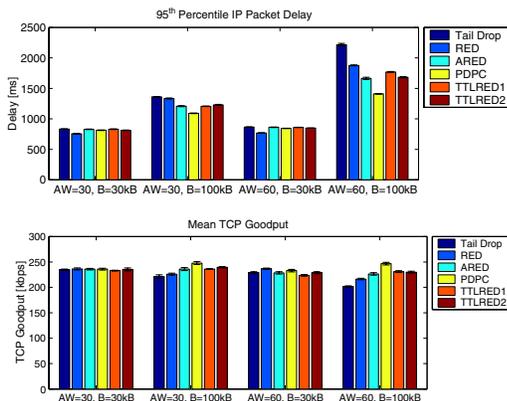


Fig. 6. R99 384 kbps, single flow: delay and goodput.

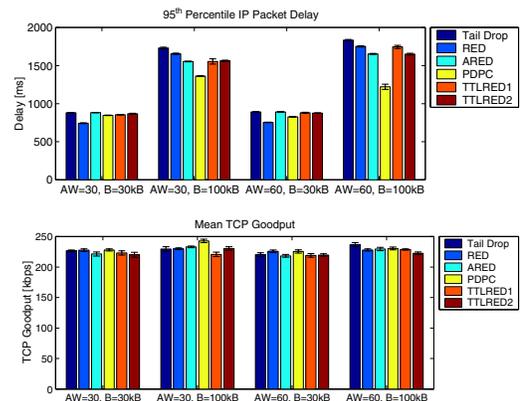


Fig. 7. R99 384 kbps, four flows: delay and goodput.

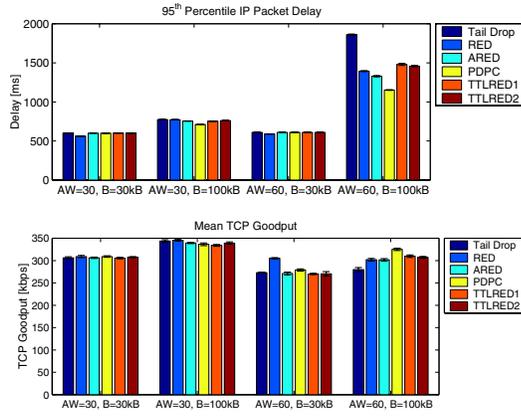


Fig. 8. HSDPA 500 kbps, single flow: delay and goodput.

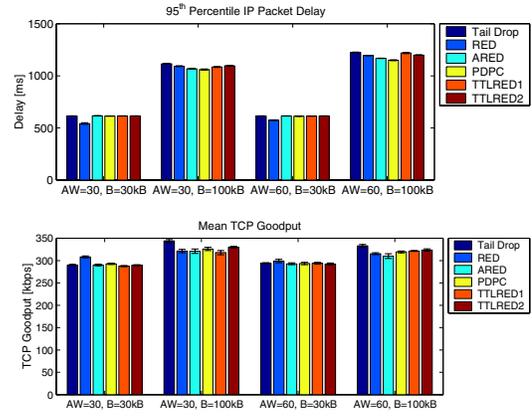


Fig. 9. HSDPA 500 kbps, four flows: delay and goodput.

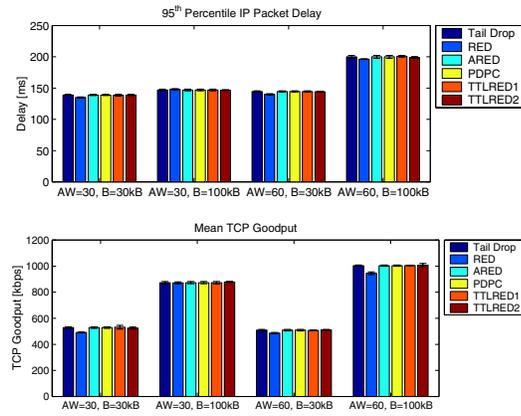


Fig. 10. HSDPA 3.6 Mbps, single flow: delay and goodput.

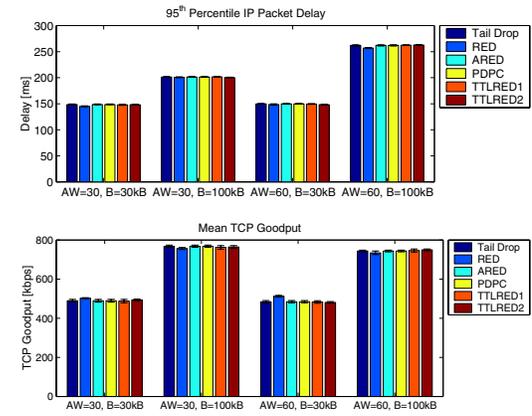


Fig. 11. HSDPA 3.6 Mbps, four flows: delay and goodput.

TABLE I  
SIMULATION RESULTS: 65 OR 250 kB FILES, ADVERTISED WINDOW = 30 PKTS AND PDCP BUFFER SIZE = 100 kB

Raw numbers	End-to-End 95 <sup>th</sup> Percentile IP Packet Delay [ms]						Mean TCP Goodput [kbps]					
	TD	RED	ARED	PDPC	TRED1	TRED2	TD	RED	ARED	PDPC	TRED1	TRED2
R99 64 kbps, 1 flow	5793	5737	2829	1646	2513	2049	60	60	56	59	58	56
R99 64 kbps, 4 flows	8000	7646	2892	2996	5353	2211	60	59	55	56	55	54
R99 384 kbps, 1 flow	1363	1335	1207	1092	1206	1230	221	226	236	248	236	239
R99 384 kbps, 4 flows	1730	1655	1553	1362	1554	1562	230	230	233	243	221	230
HSDPA 500 kbps, 1 flow	774	773	756	711	752	762	344	345	339	336	334	339
HSDPA 500 kbps, 4 flows	1116	1092	1069	1060	1085	1096	344	321	321	326	318	330
HSDPA 3.6 Mbps, 1 flow	147	148	147	147	147	147	873	872	873	873	873	879
HSDPA 3.6 Mbps, 4 flows	201	201	201	201	202	200	768	757	768	768	763	765
Improvement over TD [%]	TD	RED	ARED	PDPC	TRED1	TRED2	TD	RED	ARED	PDPC	TRED1	TRED2
R99 64 kbps, 1 flow	N/A	-1	-51	-72	-57	-65	N/A	-1	-7	-3	-3	-7
R99 64 kbps, 4 flows	N/A	-4	-64	-63	-33	-72	N/A	-3	-9	-7	-9	-11
R99 384 kbps, 1 flow	N/A	-2	-11	-20	-11	-10	N/A	2	7	12	7	8
R99 384 kbps, 4 flows	N/A	-4	-10	-21	-10	-10	N/A	0	2	6	-4	0
HSDPA 500 kbps, 1 flow	N/A	0	-2	-8	-3	-2	N/A	1	-1	-2	-3	-1
HSDPA 500 kbps, 4 flows	N/A	-2	-4	-5	-3	-2	N/A	-7	-7	-5	-8	-4
HSDPA 3.6 Mbps, 1 flow	N/A	1	0	0	0	0	N/A	0	0	0	0	1
HSDPA 3.6 Mbps, 4 flows	N/A	0	0	0	0	0	N/A	-1	0	0	-1	0

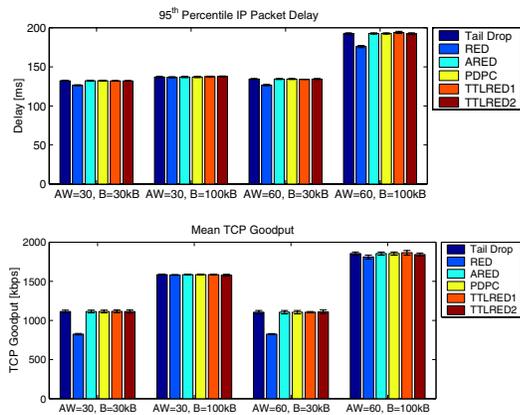


Fig. 12. HSDPA 3.6 Mbps, single flow, 2.5 MB files: delay and goodput.

## VI. CONCLUSIONS

We have studied the end-to-end performance of a single RAB (with single or multiple TCP flows) in a WCDMA network using four different AQM schemes. We have shown that AQM is quite useful to reduce the queuing delays caused by the PDCP buffering. Providing enough buffer space, on the other hand, will maximize TCP goodput.

Static RED (or most probably any AQM scheme with static parameters), is not enough but we need to adjust the parameters according to RAB rate. ARED and PDPC both utilize RAB rate in their minimum/maximum threshold setting – and they clearly outperform tail drop. Nevertheless, it may not be that straightforward to obtain the rate information for PDPC and ARED. Moreover, determining the parameters based on the rate information is not too easy either.

TTLRED does not need any rate information, which is a nice property. Moreover, the delay/goodput performance of TTLRED is more or less the same as the performance of ARED or PDPC. It is also worth noting that TTLRED can be used with adaptive buffer size, i.e., when the buffer starts to fill up, we can allocate more buffer space for the RAB in question. Nevertheless, further research (on TTLRED for incoming packets) may still be needed, e.g., in order to deal with sudden packet bursts in a more sophisticated way than tail drop.

## REFERENCES

[1] 3GPP, “3GPP TS 23.107, V5.9.0; Technical Specification Group Services and System Aspects; Quality of Service (QoS) concept and architecture (Release 5),” Jun. 2003.

[2] Sally Floyd’s RED page. <http://www.icir.org/floyd/red.html>

[3] S. Floyd and V. Jacobson, “Random Early Detection Gateways for Congestion Avoidance,” *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397–413, 1993.

[4] 3GPP, “3GPP TS 25.323, V6.0.0; Technical Specification Group Radio Access Network; Packet Data Convergence Protocol (PDCP) specification (Release 6),” Dec. 2003.

[5] 3GPP, “3GPP TS 25.322, V5.5.0; Technical Specification Group Radio Access Network; Radio Link Control (RLC) protocol specification (Release 5),” Jun. 2003.

[6] 3GPP, “3GPP TS 25.308, V5.4.0; Technical Specification Group Radio Access Network; High Speed Packet Access (HSDPA); Overall description; Stage 2 (Release 5),” Mar. 2003.

[7] J. Wigard, T. Kolding, K. Pedersen, H. Holma, and P. Mogensen, “High Speed Downlink Packet Access (HSDPA) for WCDMA,” Nov. 2003. <http://www.nokia.com/research>

[8] 3GPP, “3GPP TS 25.435, V5.5.0; Technical Specification Group Radio Access Network; UTRAN Iub Interface User Plane Protocols for Common Transport Channel data streams (Release 5),” Jun. 2003.

[9] 3GPP, “3GPP TS 25.321, V5.5.0; Technical Specification Group Radio Access Network; Multi Access Control (MAC) protocol specification (Release 5),” Jun. 2003.

[10] 3GPP, “3GPP TS 25.877, V5.1.0; Technical Specification Group Radio Access Network; High Speed Downlink Packet Access: Iub/Iur protocol aspects (Release 5),” Jun. 2002.

[11] Sally Floyd’s ECN page. <http://www.icir.org/floyd/ecn.html>

[12] K. Ramakrishnan, S. Floyd and D. Black, “The Addition of Explicit Congestion Notification (ECN) to IP,” RFC 3168, Sept. 2001.

[13] M. Sâgfors, R. Ludwig, M. Meyer and J. Peisa, “Queue Management for TCP Traffic over 3G Links,” *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC’03)*, Mar. 2003.

[14] 3GPP TS 48.018. 3<sup>rd</sup> Generation Partnership Project; Technical Specification Group GSM, EDGE Radio Access Network; General Packet Radio Service (GPRS); Base Station System (BSS) – Serving GPRS Support Node (SGSN); BSS GPRS Protocol (BSSGP) (rel’6).

[15] A. Gurtov and R. Ludwig, “Lifetime Packet Discard for Efficient Real-Time Transport over Cellular Links,” *ACM Mobile Computing and Communications Review*, vol. 7, no. 4, pp. 32–45, October 2003.

[16] UCB/LBNL/VINT, Network Simulator – ns (version 2), Jan. 2005. <http://www.isi.edu/nsnam/ns/index.html>

[17] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach and T. Berners-Lee, “Hypertext Transfer Protocol – HTTP/1.1,” RFC 2616, Jun. 1999.

[18] S. Floyd and T. Henderson, “The NewReno Modification to TCP’s Fast Recovery Algorithm,” RFC 2582, Apr. 1999.

[19] M. Allman, V. Paxson and W. Stevens, “TCP Congestion Control,” RFC 2581, Apr. 1999.

[20] G. Appenzeller, I. Keslassy and N. McKeown, “Sizing Router Buffers,” *Proceedings of SIGCOMM’04*, Aug. 30–Sept. 3, 2004, Portland, Oregon, USA.