Cross-Layer Error Control Optimization in 3G LTE

Dzmitry Kliazovich[1] and Fabrizio Granelli
DIT - University of Trento
Via Sommarive 14, I-38050 Trento, Italy
E-mail: [klezovic,granelli]@dit.unitn.it

Simone Redana and Nicola Riato
Nokia Siemens Networks S.p.A COO RTP PT RST
Via Monfalcone, 1, I-20092 Milan, Italy
[Simone.Redana,Nicola.Riato]@nsn.com

***Abstract*—3G Long-Term Evolution (LTE) is a recent effort taken by cellular industries to step into wireless broadband market. The key enhancements target an introduction of new all-IP architecture, enhanced link layer and radio access with OFDM modulation and multiple antenna techniques.**

In this study, we focus on the overhead deriving from the multilayer ARQ employed at the link and transport layers. To the aim of reducing unnecessary burden on the wireless link, we propose a cross-layer ARQ approach, called ARQ Proxy, which substitutes the transmission of TCP ACK packet with a short MAC layer request on the radio link. Packet identification is achieved through association of a hash function to the raw packet data.

Performance of the ARQ Proxy is evaluated using EURAE extensions for ns2 simulator. Results demonstrate significant improvements in terms of system capacity, TCP throughput performance, and higher tolerance to transmission errors.[2]

Keywords-3G LTE, cross-layer, ARQ proxy.

I. Introduction

The 3G Long-Term Evolution (3G LTE) is an attempt to step into wireless broadband taken by cellular providers and equipment vendors [1]. 3G LTE introduces evolved radio interface with major enhancement coming from the use of Orthogonal Frequency Division Multiplexing (OFDM) and multiple antenna techniques. These technologies are already available on the market and employed in WiMAX as specified in IEEE 802.16 standard [2].

Along with the evolved radio interface, 3G LTE specifies the evolution of network architecture. It is designed to be packet-based and contain less network nodes, thus reducing protocol processing overhead and leading to reduced latency and network deployment costs. Moreover, the evolved architecture is focused on packet data communications using TCP/IP protocol suite.

However, TCP/IP (designed in early days of ARPANET specifically for wired networks) shows poor performance over wireless channels, mainly due to high error rates [3]. In order to compensate wireless errors, 3G LTE employs error recovery techniques at the link layer, which partially overlap with error recovery performed at the transport layer of TCP/IP.

In this paper, we present a novel cross-layer ARQ optimization technique, called *ARQ proxy*, which jointly optimizes error recovery mechanisms operating at different layers. ARQ proxy reduces the demand for network resources consumed by the employed error recovery techniques leading to the increase of network capacity as well as throughput performance.

II. 3G LTE Overview

In the effort to challenge advanced communication technologies in the long run, 3rd Generation Partnership Group (3GPP) initiated the study item *Evolved UTRA and UTRAN* [3], aiming to complete the set of specifications for the evolved radio access in 2007 with initial product deployment in 2009.

The requirements for the long-term evolution (LTE) can be summarized as follows:

- Peak data rates of 100 Mb/s in the downlink and 50 Mb/s in the uplink leading to spectrum efficiency of up to 5 bit/s/Hz.
- Reduced user- and control-plane latency to less than 10 ms and less than 100 ms, respectively.
- Mobility is supported for up to 350 km/h.
- Spectrum flexibility and reduced network costs.

To achieve the above mentioned targets, new updates to the radio interface as well as modifications to the network architecture are considered.[3]

A. Evolved Architecture

Reduced latency and costs are two main driving requirements for architecture evolution which are accomplished with reduced number of network nodes. Fewer nodes reduce protocol-related processing along the data path as well as call setup time and control-plane related processing.

Fig. 1 illustrates high level architecture for the evolved system specified in [5]. In the Evolved UTRAN, the functions of Radio Network Controller (RNC) are distributed between evolved Node B (eNB) and network core. The evolved core provides local mobility management, internetworking with previously released 3GPP systems as well as non-3GPP wireless technologies. 3G LTE relies on all-IP communications as opposed to the widely used circuit-switched approach.

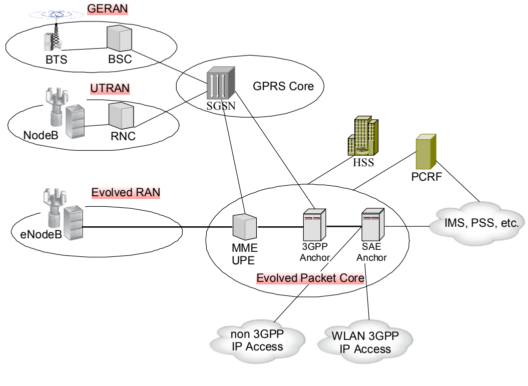

Figure 1. 3G LTE System Architecture.

[1] This work was done while Dzmitry Kliazovich was pursuing an internship focused on 3G LTE research at Siemens (now Nokia Siemens Networks), Italy.

[2] The ARQ Proxy approach is currently patent-pending under EP 07425087.9 "Cross-Layer Error Recovery Optimization for 3G LTE Systems."

[3] At the moment of writing this paper 3GPP group has just initiated 3G LTE working phase. For that reason, the final details of the evolved system may differ from those presented in current paper. However, it is indented to provide the reader with the key concepts which are already agreed by the standardization committee and will most likely become a part of the standard.

1930-529X/07/$25.00 © 2007 IEEE

B. *Evolved Radio Access*

OFDM is an attractive technology to reach high data rate requirements. It provides high performance in frequency-selective channels, enables frequency domain adaptation, and is well suited for MIMO techniques. Additionally, OFDM will avoid royalties which are currently a considerable fraction of overall costs in CDMA-based systems.

In the uplink, Single-Carrier OFDM (SC-OFDM), also referred as DTF-spread OFDM, is chosen to enable frequency-domain equalization at the receiver side. SC-OFDM can be implemented in "localized", i.e. sub-carriers are mapped on the consecutive portion of the spectrum, or in "distributed" mode. Indeed, both schemes provide low peak-to-average power ratio (PARP), ensuring power-efficient communication for mobile terminals.

C. *Evolved Link Layer*

The 3G LTE link layer is split into three sublayers: Medium Access Control (MAC), Radio Link Control (RLC), and Packet Data Convergence Protocol (PDCP).

One of the main innovative aspects is related to the introduction of variable size RLC protocol data units (PDUs) which helps to avoid padding overhead.

Additionally, 3G LTE similar to other types of wireless networks suffers from high error rates at radio channel (physical) level, which can generate significant performance degradation of TCP protocol.

In order to counteract such variation of BER, Forward Error Correction (FEC) or Automatic Repeat reQuest (ARQ) techniques can be applied. 3G LTE considers both of them introducing Hybrid ARQ (HARQ) at the MAC layer as well as ARQ at the RLC layer.

In case of ARQ, the receiving system checks Cyclic Redundancy Check (CRC). In case CRC is correct, an ACK is sent back to sender and the received packet is forwarded to upper layers of the protocol stack for processing. In case CRC check fails, the packet is dropped silently or with negative ACK (NACK) notification sent to the sender. The ARQ considered for implementation in 3G LTE is of Go-Back-N type, which positively acknowledges every N successfully received packets with a status report.

In HARQ, in case CRC is not correct, the packet is not dropped at the receiver side. It is stored for the case when retransmitted packet will be still erroneous and combined recovery can be attempted. This technique is known as *chase combining*. Eventually the error is resolved or the maximum number of retransmissions is reached. Each following retransmission contains additional redundancy information (*incremental redundancy*). A stop-and-wait HARQ is considered in MAC implementation, which provides the sender with a binary ACK/NACK feedback for every transmitted packet.

The combined usage of ARQ and HARQ at different layers improves radio channel error rate by several orders: BER is typically equal to 10^{-2} for no retransmissions, 10^{-3} after HARQ at the MAC layer, and 10^{-6} after ARQ at the RLC layer [9].

While 3G LTE link layer implements multilayer ARQ, it is not the only layer employing error recovery: TCP reliability is obtained through the utilization of a positive acknowledgement scheme which specifies TCP receiver to acknowledge data successfully received from the sender. TCP header reserves special fields for enabling it to carry acknowledgement information. As a result, the TCP receiver can produce a TCP acknowledgment (TCP-ACK) as a standalone packet or, in case of bi-directional data exchange, encapsulate it into outgoing TCP segments.

Fig. 2 illustrates a single TCP data packet delivery in 3G LTE network. The eNB after the reception of a TCP data packet from the packet core forwards it over the radio link to the appropriate UE. This downlink transmission includes also an overhead associated with PHY, MAC, and RLC headers. In case of successful reception, UE generates HARQ ACK response to the eNB forwarding the received TCP data packet up to the TCP layer of the protocol stack. Consequently, TCP layer generates TCP ACK. This TCP ACK represents ordinary payload for the UE link layer: before it can be transmitted on the wireless link, uplink resources should be requested and corresponding assignment grant should be received at the MAC layer. Moreover, TCP ACK transmission requires an acknowledgement of HARQ entity which resides at the eNB.

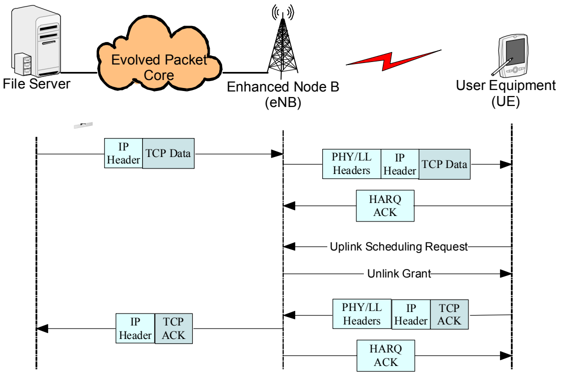

Figure 2. TCP packet delivery in 3G LTE network.

Summarizing, a single TCP data packet transmission is acknowledged three times: once at the transport level and two times at the link layer. Moreover, while HARQ ACK is a single bit feedback, the TCP ACK together with protocol overhead added at the link and physical layers consumes a relatively large amount of uplink bandwidth resources – reducing overall system capacity.

Next section proposes a solution which produces joint optimization of ARQ/HARQ entities located at different layers of the protocol stack with the objective to reduce the demand for bandwidth resources in uplink by enabling signaling and cooperation between entities at different levels of the protocol stack.

III. ARQ PROXY

The basic idea of the proposed approach is to avoid the transmission of standalone TCP ACK packets over the radio link between eNB and UE together with associated overhead added at physical and link layers, including scheduling request/response exchange at the MAC layer. In order to support this functionality, no changes are needed to the TCP protocol, but new software entities need to be introduced: the ARQ proxy and ARQ client.

1930-529X/07/$25.00 © 2007 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2007 proceedings.

A. *ARQ Proxy*

ARQ proxy is a software module located in the eNB protocol stack. It sniffs the packets destined to various UEs, assuming to have an access to the network and transport layer headers. In contrast to previous cellular systems, in 3G LTE network layer packets become explicitly visible at eNB due to the employed mapping when one network layer packet is encapsulated into one RLC PDU.

Whenever a TCP data packet is detected, the ARQ proxy generates a TCP ACK confirming the reception of TCP data packet (assuming the case it is received in sequence). eNB does not require an implementation of TCP layer in a conventional sense. For TCP ACK generation, ARQ proxy simply copies the required fields (IP addresses, port numbers, and flow sequence numbers) into the corresponding fields of a TCP ACK packet template previously allocated in the memory. The generated TCP ACK is not immediately released into the evolved network core, but stored locally in a hash table at the eNB. As soon as TCP ACK is generated and stored, the original TCP data packet continues its path towards the UE.

The TCP ACK is associated with an index in the hash table computed by passing IP and TCP header fields as an input for the hash function.

The use of hash tables and hash functions for indexing allows independent index calculation at eNB and UE, such that UE can obtain an index of TCP ACK generated at eNB for every received TCP data packet by simply applying the same hash function to the fields of the received TCP data packet. Using this index, the ARQ client can request eNB to release a given TCP ACK into the network core.

Each TCP ACK located in the hash table at the eNB has a lifetime which is assigned at the moment of TCP ACK generation. In case the lifetime is exceeded, the packet is silently dropped from the hash table. By this mechanism, eNB ensures the memory resources are freed for TCP ACKs not requested by UE. This happens in case TCP data packet arrives out-of-order or delayed-ACK scheme is implemented by UE receiver. The lifetime value can be set to be equal to TCP timeout.

The choice of appropriate hash function is left out of the scope of the paper. However, taking into account the extensiveness of the information available on this topic the interested reader is directed to [8]. Furthermore, we leave the design of a proper hash function which will include specifics of input data (IP and TCP data headers) for future work. In fact, there is already a hash function available in TCP/IP which corresponds to cyclic redundancy check.

B. *ARQ Client*

The ARQ Client is a software module which resides in the protocol stack of the UE. It suppresses all outgoing standalone TCP ACK packets and replaces them with MAC layer requests for the appropriate TCP ACK transmission initiated from eNB. In order to do so, ARQ Client computes an index of the desired TCP ACK in the hash table of the eNB by applying the hash function to the received TCP data packet. The computed hash value is passed to the eNB using an interface between ARQ Proxy and ARQ client defined at the MAC layer.

In our scheme, the HARQ ACK message acknowledging successful delivery of each PDU is extended with a hash value resource reservation bit. If set, it requires eNB extending resource grant in order to allow UE to transmit a fixed size hash value during the next transmission opportunity.

We identify two possible positions for ARQ client in UE's protocol stack: enhanced transport layer (Fig. 3a) and standalone (Fig. 3b).

In case ARQ client is implemented inside the transport layer, whenever a TCP ACK is produced as a standalone packet it generates the corresponding hash value passing it to the HARQ entity at the MAC layer. While HARQ is requesting bandwidth resources for hash value transmission, the originally generated TCP ACK packet travels the stack down which involves corresponding processing at every layer, output queuing delay, and other procedures commonly performed by the protocol stack.

Whichever comes first to the physical layer for the transmission (the TCP ACK or the corresponding hash value) will be transmitted, while the other one cancelled. This technique enables the design of hash value-based ACK generation not synchronized with a particular HARQ ACK message. For example, in case a hash value transmission was not requested with HARQ ACK generated for TCP packet recently received, it can be requested with the next outgoing HARQ ACK. However, in case MAC layer did not succeed to pass the hash value and TCP ACK arrives at the physical layer, it is transmitted like in the legacy implementation (i.e. with no ARQ proxy enabled).

Transport layer enhancement can be costly or sometimes not possible due to the large variety of TCP/IP enabled devices already deployed on the market. In this case, ARQ Client can be implemented as a standalone module on top of the MAC layer. Such implementation does not require the modification of the protocol stack core (TCP and IP layers) and can be introduced at driver level or inside network interface firmware.

In standalone implementation (Fig. 3b), ARQ Client sniffs all packets produced at the transport layer. In case a standalone TCP ACK is detected, it simply computes the corresponding hash value for such packet passing it to the HARQ module, while the original packet continues its outgoing path down the protocol stack. The rest of the ARQ Client functionality is similar to the enhanced transport layer case illustrated above.

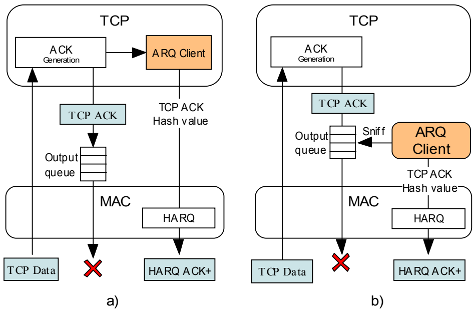

Figure 3. ARQ Client position in UE's protocol stack.

Generally, ARQ Client requests TCP ACK generation at eNB for all TCP ACKs outgoing from the UE as standalone packets and only for data transfer phase of the TCP connection. TCP ACKs which are not subject to replacement with HARQ requests are the following:

1930-529X/07/$25.00 © 2007 IEEE

1. TCP ACK which corresponds to connection establishment or connection termination phases of TCP. Indicated by SYN and FIN header flags, these packets carry maximum window, initial flow sequence numbers and other parameters crucial for the data flow setup or termination.
2. TCP ACK which is sent along with outgoing data packet. In case of bidirectional data transfer and delayed-ACK option enabled, the receiver is capable to encapsulate TCP acknowledgement which is simply TCP ACK bit and acknowledgement sequence number into outgoing data packet. As a result, TCP ACKs which are encapsulated in such a way do not create any additional overhead and thus may be left unaffected by ARQ proxy. However, it should be noted that most of TCP flows initiated by UE are unidirectional where data packets are originated at content server.
3. Duplicate TCP ACKs. In case out of order TCP packet is received, UE should send duplicate TCP ACK for the last successfully received segment. However, eNB does not maintain any flow related information and thus can not distinguish an out-of-order TCP packet. In order to overcome this limitation, in case of out-of-order delivery, TCP receiver will send duplicate TCP ACK generated by its protocol stack over the radio channel and will not send hash value request to ARQ proxy module. As a result, TCP ACK generated by ARQ proxy and stored in eNB's hash table will expire after lifetime is exceeded.
4. TCP ACK where the receiver advertises for a window (*rwnd* field in TCP header) different from the last advertised one. This will ensure UE is not running out of the receiver buffer space.

C. *Benefits and Limitations*

Fig. 4 summarizes the functionality of the proposed ARQ proxy and ARQ client modifications.

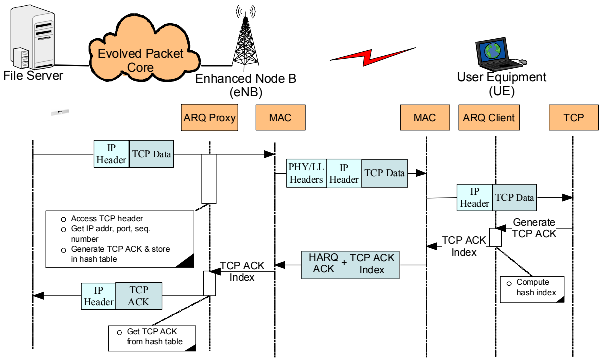

Figure 4. ARQ Proxy and ARQ Client functionality.

Along with network capacity increase, we identify the following benefits of the proposal:

End-to-End (E2E): ARQ proxy substitutes TCP ACKs on the radio link. However, this does not violate E2E semantics of the TCP protocol, since all TCP ACK transmissions originated at the eNB are triggered by the UE. Moreover, UE requests the transmission of those ACKs which are consistent with the employed acknowledgement strategy (delayed or selective acknowledgement [10]), while not requested TCP ACKs are silently dropped at the eNB when their lifetime is exceeded.

Performance and System Capacity: Joint reduction of ARQ overhead releases network resources in the uplink, which can be further reused by the same UE or other nodes operating in the same cell.

Channel error rate: ARQ proxy approach substitutes a relatively large TCP ACK packet with a several bytes long hash value transmitted over the radio channel. As a result, the effect of channel error propagation is reduced for such packets.

Round Trip Time (RTT): TCP throughput and error recovery performance depends on the Round Trip Time (RTT) between the sender and the receiver. In 3G LTE, RTT is composed of packet transmission delay in the network core as well as over the radio link. The proposed ARQ optimization technique reduces RTT for the time associated with TCP ACK transmission over the radio link, including uplink bandwidth reservation delay, which depends on the UE's state (active or idle), and employed framing. This reduction is typically in the order of tens of milliseconds [11].

Mobility: The support of full inter- and intra-network mobility is one of the most crucial requirements in cellular networks. Moreover, the handover of UEs should be performed with minimum delay to ensure uninterrupted service execution at guaranteed quality. For that reason, approaches which maintain UE state or profile at eNB are not desirable due to the delay overhead associated with the state transfer during handover. In our scheme, in case UE changes its location and registers itself with another eNB, the hash table stored by the ARQ proxy at old eNB is not transferred to the new eNB and can be deleted. In this way, after location update, the UE will send hash values for only those TCP ACKs which correspond to packets received from the new eNB.

Incremental deployment: ARQ proxy can be incrementally deployed in already operational network where UEs and eNBs implementing ARQ proxy approach co-exist with those which are not. For example, in case UE does not include ARQ client module, none of the TCP ACKs generated at eNB will be requested using their hash values and will be simply destroyed after their lifetime expiration. On the other hand, in case eNB does not implement ARQ proxy approach, all bandwidth allocation requests sent by UEs will be rejected and original TCP ACK packets will transit over the radio channel. This is an outcome of the principle described above: whichever comes first at the physical layer (TCP ACK or its hash value) will be transmitted over the radio channel.

IV. PERFORMANCE EVALUATION

In order to perform the evaluation of the proposed solution, Enhanced UMTS Radio Access Extensions (EURAE) [6] for ns2 network simulator [7] are used.

In order to make EURAE closely approximate 3G LTE behavior, we combined Node B and RNC into a single node as well as we modified the link layer accordingly to avoid fragmentation and support one-to-one mapping of IP packets into RLC PDUs. The physical medium is configured to use HSDPA extension with default parameters, and Rayleigh fading for the propagation model with a trace data generated for UE located within 300 meters from the Node B. The TCP

1930-529X/07/$25.00 © 2007 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2007 proceedings.

connection is initiated between the file server located in the network core and the receiver sink attached to the UE.

In this scenario, ARQ proxy requires almost 21 times less channel resources than the original TCP scheme. This value is basically an outcome of the size difference between TCP ACK and its hash value: while TCP ACK is comprised of TCP (20 bytes), IP (20 bytes), PDCP (1 byte), RLC (2 bytes) headers, and PHY CRC (2 bytes) bringing it to 45 bytes in total, the hash value used in simulations is only 16 bits long plus one additional bit required for predefined reservation sent along HARQ ACK.

Along with the uplink capacity improvement, the proposed ARQ proxy scheme aims at RTT reduction due to faster TCP ACK feedback. Fig. 5 presents the RTT measurements experienced by the TCP flow. The ARQ proxy-enabled case shows the RTT close to the delay experienced in the network core, which one-way propagation delay is set to 30 ms. The original TCP scheme requires more than additional 20 ms, which are consumed for the uplink resource reservation and TCP ACK transmission. According to [4], TCP throughput performance is reversely proportional to the RTT of the connection. This corresponds to an added value for ARQ proxy implementation for TCP flow performance improvement.

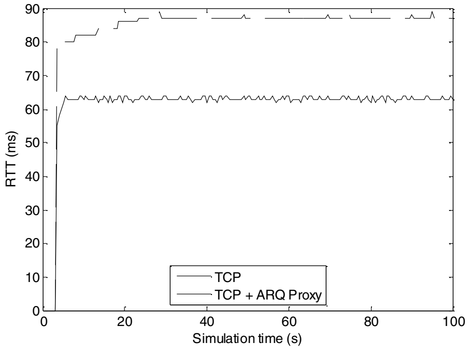

Figure 5. Round Trip Time (RTT) measurements.

In original scheme, UE uses an acknowledged mode which involves HARQ at the MAC and ARQ at the RLC layers for TCP ACK transmission on the uplink. On the contrary, ARQ proxy does not involve any techniques to ensure reliable hash value delivery. This motivated us to evaluate the influence of errors on the hash value transmission (see Fig. 6). In case the eNB receives a corrupted hash value, no TCP ACK is generated from the hash table toward TCP sender.

The results show the TCP flow is able to maintain a nominal throughput level for up to 40% of erroneous TCP ACKs. Such stability is motivated by the cumulative nature of TCP acknowledgements, i.e. every successfully received TCP ACK acknowledges all the data received by the receiver up to a moment of this TCP ACK generation. Taking into account TCP high robustness to ACK errors, we consider no need to introduce any additional reliability into hash value transmission.

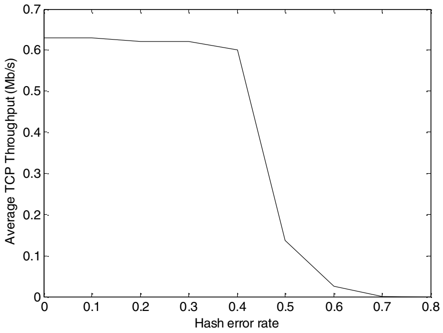

Figure 6. TCP throughput in the presence of hash value errors.

V. Conclusions and Future Work

3G Long-Term Evolution (LTE) is a recent effort taken by cellular industries to step into wireless broadband market. In this framework, the paper proposes a cross-layer optimization of error recovery strategies for 3G LTE networks. The proposal substitutes the transmission of TCP ACK packets with a short MAC layer request sent over the radio link. Specifically, TCP ACKs are generated by ARQ proxy module located at eNB based on transit traffic analysis and stored in the buffer until requested by ARQ client module which resides at UE's protocol stack. TCP ACK identification which is included into ARQ client request is obtained by applying a hash function onto raw packet headers' data.

Performance evaluation results demonstrate good agreement with the design objectives, i.e. ARQ proxy approach is able to provide uplink capacity increase, sustain high error rate, and bring TCP performance increase due to reduced RTT.

Future work on ARQ proxy includes application of the presented technique in WiFi (IEEE 802.11) and WiMAX (IEEE 802.16) environments.

References

[1] "Clash of the Titans – WiMAX and 4G: The Battle for Convergence is joined," Market report, Maravedis Research and Analysis, July 2006.

[2] IEEE Std. 802.16-2001, IEEE Standard for Local and Metropolitan Area Networks, part 16, "Air Interface for Fixed Broadband Wireless Access Systems," IEEE Press, 2001.

[3] 3GPP, RP-040461, "Proposed Study Item on Evolved UTRA and UTRAN," www.3gpp.org.

[4] Teunis J. Ott, J.H.B. Kemperman, Matt Mathis, "The Stationary Behavior of Ideal TCP Congestion Avoidance, " Internetworking: Research and Experience, v. 11, p.115-156, 1992.

[5] 3GPP, TR 23.882, "3GPP System Architecture Evolution: Report on Technical Options and Conclusions," www.3gpp.org.

[6] Enhanced UMTS Radio Access Extensions (EURAE) home page, http://www.ti-wmc.nl/eurane/

[7] The network simulator NS2, http://www.isi.edu/nsnam/ns/

[8] Partow, A. "General Purpose Hash Function Algorithms," http://www.partow.net/programming/hashfunctions/

[9] 3GPP, TS 36.300, "E-UTRA and E-UNTRAN: Overall description," www.3gpp.org.

[10] D. Clark, "Window and acknowledgement strategy in TCP," RFC 813, IETF, July 1982.

[11] 3GPP, R2-061866, "Non-Synchronized Random Access in E-UTRAN," Ericsson, www.3gpp.org.