INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. (2011)

Cross-layer error recovery in wireless access networks: The ARQ proxy approach

Dzmitry Kliazovich[1,*,†], Simone Redana[2] and Fabrizio Granelli[3]

[1] *University of Luxembourg, 6 rue Coudenhove Kalergi, L-1359 Luxembourg*
[2] *Nokia Siemens Networks, Munich, Germany*
[3] *Department of Information Engineering and Computer Science, University of Trento, via Sommarive 14, I-38050, Trento, Italy*

SUMMARY

Wireless networks are being increasingly employed to provide mobile access to network services. In most existing standards, reliable transmission on the wireless medium is achieved through the introduction of ARQ schemes at MAC layer, a strategy which is also employed by TCP for reliable end-to-end data delivery. The paper proposes an approach to overcome the performance degradation deriving from the duplicate ARQ strategies implemented at the transport and MAC layers by introducing a cross-layer solution to reduce un-necessary transmissions on the wireless medium. Furthermore, the paper describes how the proposed scheme, called ARQ Proxy, can be deployed in three different wireless technologies (3G Long-Term Evolution, Wi-Fi, and WiMAX) and provides extensive validation of the achievable improvement through simulations.

Received 13 August 2010; Revised 4 November 2010; Accepted 28 February 2011

KEY WORDS: performance enhancement; TCP; ARQ optimization; wireless networks; ARQ proxy; cross-layering; distributed protocol stacks

1. INTRODUCTION

Wireless communications clearly represent a fast-growing sector in the framework of data networks. Mainly, wireless technologies provide mobile access to networks and services—omitting the requirement for a cable (and fixed) infrastructure. This fact enables fast and cost-effective network organization, deployment and maintenance and allows users to easily join, leave or switch to other networks in a simple and possibly transparent way. However, wireless communications suffer from specific capacity limitations related to the very physical nature of wireless medium, such as limited shared bandwidth, time-varying behavior, interference, etc.

Nevertheless, wireless technologies are envisaged to be widely deployed in the last mile–connecting end-user to the core of the network, while leaving transport of data in the core to cable/optical architectures. Indeed, last mile is the most critical issue in today's network architectures, and DSL technology alone cannot satisfy such a rapidly growing market due to its limitations in terms of costs, capacity and distance.

As wireless networks created the field for new applications and services, at the same moment they increased the fruition of several existing services. Still, they suffer from several performance

*Correspondence to: Dzmitry Kliazovich, University of Luxembourg, 6 rue Coudenhove Kalergi, L-1359 Luxembourg.
†E-mail: kliazovich@ieee.org

limitations, in some case related to the excessive complexity deriving from the layering paradigm employed in protocol stack design and the related lack of effectiveness deriving for the standard TCP/IP protocol suite implementation (originally defined for cable networks).

This paper targets cross-layer design as a possible solution to the excessive overhead related to the layered TCP/IP protocol stack in order to improve the data transfer performance of wireless networks.

The organization of the paper is the following: Section 2 provides details about the motivation of the proposed scheme, Section 3 overviews related works in the topic, while Section 4 describes the proposed framework and its deployment in different technologies (3G Long-Term Evolution (LTE), Wi-Fi, and WiMAX). Section 5 provides validation of the proposed scheme by extensive simulation results, and Section 6 concludes the paper with final remarks.

2. MOTIVATION

Today, the dominant protocol stack is the TCP/IP reference model. TCP/IP was originally designed for wired links, characterized by high bandwidth, low delay, low probability of packet loss (high reliability), static routing, and no mobility. On the contrary, in the wireless domain, performance and resources are limited by the time-varying channel performance, the scarce availability of transmission spectrum, the employed modulation type and the limited transmission power. As a result, wireless packet delivery latency can be several orders of magnitude higher than the one achieved in wired networks.

Loss probability experienced by packet transmission on the wireless medium are in general higher than those on wired links: while Bit Error Rate (BER) varies from 10^{-6} to 10^{-8} for wired channels, it varies from 10^{-3} up to 10^{-1} for wireless channels.

Indeed, PER at the link layer derives from BER performance on the physical channel. For large packets (link layer segments) with a size of 1500 bytes, PER is significantly less than 1% in wired networks, while it can range from 1 to 12% in wireless. Such PERs are unacceptable for most of natively wired network protocols such as TCP, in which additive increase multiplicative decrease (AIMD) congestion control treats all losses as congestion losses and thus underestimates the actual available link capacity.

In order to counteract such range of variation in terms of BER / PER, channel coding schemes, such as turbo codes, can be employed at the link layer. This approach is known as Forward Error Correction (FEC). The FEC strength and added overhead cannot be dynamically adjusted and usually the selection of the proper FEC scheme is based on the worst case, thus introducing waste of resources (and corresponding throughput decrease) in case of good performance of the transmission channel.

For such reason, FEC is not the proper solution to provide reliable performance in wireless networks. The main drawback is due to the waste of transmission resources deriving from its employment in case of the absence of errors, therefore suggesting the usage of feedback information from the receiver in order to extrapolate information on the channel status. Thus, a traditional and widely implemented approach to increase the reliability of the wireless link is based on the usage of an Automatic Repeat reQuest (ARQ) protocol at the link layer.

ARQ provides a dynamic way to decrease the PER on the wireless links by increasing the delivery delay. The advantage derives from the fact that only corrupted packets are retransmitted. For that reason, the overhead deriving from ARQ protocols is automatically adapted to the conditions of the link.

Table I presents a brief overview of the main wireless technologies and their features to support the above assumption.

The most commonly used ARQ scheme in wireless is 'stop & wait': the sender is not allowed to send the next packet in the queue until the receiver positively acknowledges the successful delivery of the previous one. However, link layer is not the only layer which acknowledges packet delivery: the reliability of TCP communication is obtained through utilization of a positive

Table I. Characteristics of leading wireless technologies.

Technology	Nominal Range	Frequency Band	Channel Bandwidth	Physical Rate	TCP/IP Throughput	Mobility	ARQ
Wireless Wide Area Networks (WWAN)							
GSM (2G)	3–35 km	900 MHz, 1800 MHz (TDMA) 800 MHz, 1900 MHz (CDMA)	200 KHz (TDMA) 1.23 MHz (CDMA)	9.6 – 57.6 kbps	4 – 38 Kbps	Seamless global roaming	yes
(E)GPRS (2.5G)			200 KHz (TDMA)	56 - 115 kbps	32 – 84 Kbps		yes
EDGE				384 Kbps (48 – 60 Kbps per timeslot)	300 Kbps		yes
3G		1900 – 2025 MHz, 2110 – 2200 MHz	5MHz	large range 144 Kbps, medium range 384 Kbps, small range 2 Mbps	120 Kbps, 310 Kbps, 1.6 Mbps		yes
3G LTE		3G spectrum	1.25, 2.5, 5, 10, 20 MHz	DL: up to 100 Mbps, UL: up to 50 Mbps	Up to 80 Mbps		yes
Wireless Local Area Network (WLAN)							
IEEE 802.11	40 – 100 meters	2.4 GHz	22 MHz	1/2 Mbps	0.7/1.4 Mbps	Nomadic subnet roaming	yes
802.11b (Wi-Fi)				11 Mbps	5 Mbps		yes
802.11a		5 GHz		54 Mbps	25 Mbps		yes
802.11g		2.4 GHz		54 Mbps	25 Mbps		yes
802.11n		5 GHz	20/40 MHz	250+ Mbps	100+ Mbps		yes
Wireless Metropolitan Area Network (WMAN)							
IEEE 802.16 (WiMax)	Up to 50 km	11 – 66 GHz	20, 25, 28 MHz	32 – 134 Mbps		Fixed	no
IEEE 802.16a		2 – 11 GHz	1.75 – 20 MHz	4 - 75 Mbps	3.22 – 56 Mbps	Fixed	yes
IEEE 802.16e	1 – 4.5 km	2 – 6 GHz	5 MHz	15 Mbps		Pedestrian mobility – Regional roaming	yes

acknowledgement scheme which specifies TCP receiver to acknowledge successfully received data from sender. TCP header reserves special fields enabling it to carry acknowledgement information. As a result, the TCP receiver can produce a TCP acknowledgment (TCP-ACK) as a standalone packet or encapsulate it into outgoing TCP data packet in case bi-directional data exchange is taking place.

In a wireless scenario, whenever a TCP data packet is transmitted over wireless link, the sender first receives an acknowledgement at the link layer (LL-ACK). Then, receiver's TCP generates an acknowledgement at the transport layer. This acknowledgement represents ordinary payload for the link layer—which should be acknowledged by the link layer of sender node. A single TCP data packet delivery framework is presented in Figure 1.

In most wireless networks (Wi-Fi and WiMax), before any data frame transmission can take place over wireless link the sender node must: (i) first, contend for (or reserve) channel access; (ii) then, perform synchronization of its transmitter with the correspondent receiver. When the sender is able to deliver bits to the receiver, it transmits physical layer header (containing information specifying physical layer techniques, such as modulation type, data rate, etc., used for the rest of the packet transmission).

In summary, in most of the available wireless network architecture reviewed in the previous paragraphs, a single TCP data packet transmission is acknowledged three times: one at the transport level and two times at the link layer, and for each acknowledgement the corresponding physical and link layer overhead is added. This leads to a significant performance reduction, since most overhead is associated with the physical layer - which headers are often transmitted at the lowest rate supported by the cell (like in Wi-Fi), mainly for backward compatibility.

Moreover, between the TCP data packet transmission and TCP-ACK reception and the corresponding acknowledgment, a significant delay is introduced due to the wireless medium access or duplexity management.

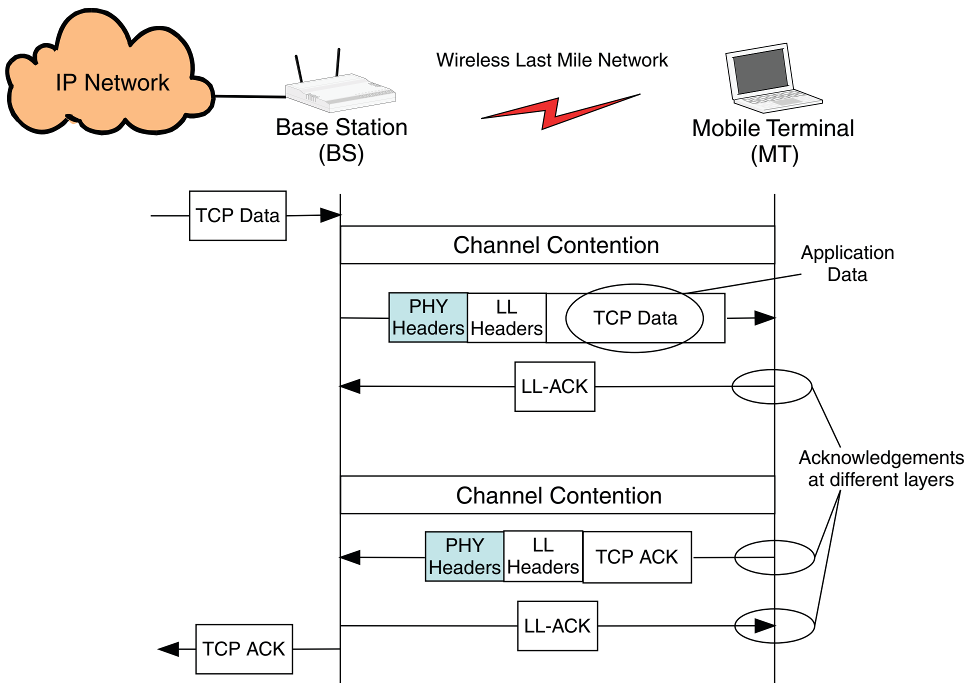

Figure 1. TCP data packet delivery over a wireless link.

Optimization of current acknowledgement scheme will obviously bring performance improvement through the reduction of medium-busy time with a subsequent reduction in overhead and delay. Moreover, it would require interaction between the transport and link layer (acknowledgements being generated at different layers of the protocol stack), thus requiring proper cross-layering schemes.

3. RELATED WORKS

Optimization of TCP/IP protocol suite in heterogeneous networks, and specifically on wireless links, has a long history. This section is not intended to provide a comprehensive overview of all existing solutions in the framework of TCP/IP optimization on wireless networks, but to underline those approaches which are most closely related to the proposed methodology.

One of the first surveys and comparison between existing solutions was presented by Balakrishnan *et al.* [1]. The authors categorize all solutions into three categories: end-to-end, link layer, and connection splitting, and propose the introduction of a software module in the protocol stack of the BS, called snoop agent. The snoop agent listens to ongoing TCP connections, keeps track of their state, and assists with local retransmissions (over the wireless section of the network). Starting from such proposal, cross-layer design and the usage of software agents represented the main direction for TCP/IP optimization in wireless environments.

As a result, many TCP/IP optimization solutions are available which require the introduction of a certain degree of awareness and cooperation among the protocol layers [2], making cross-layering a promising approach for the design of next generation protocol stacks. However, it must be underlined that, in order to be widely accepted, a significant effort is required from standardization bodies and industries for unification of the different cross-layer frameworks and signaling architectures.

An alternative and more generic TCP/IP adaptation approach is commonly referred to as ‘agent-based networking’. It aims at the introduction of active functionalities into the passive network core. Potentially, such agents can be implemented at any layer of the protocol stack. However,

DOI: 10.1002/dac

there is a clear trend for high-layer agents. For example, in the Explicit Congestion Notification (ECN) scheme [3], a specific agent running in an ECN-enabled router is able to inform TCP sender about growing network congestion by marking the ECN bit in the packet header. Based on the ECN feedback information, the TCP sender is able to properly adapt the outgoing data rate to the network congestion status.

The vast majority of network agents is implemented at the application layer and is commonly known as proxy agents. The functionalities implemented by the proxy agents depend on the goals for introducing the proxy. For example, the most widely used type of proxies is Web proxy, which serves as a gateway between a particular Intranet and Wide Area Network (WAN) and may include blocking or other content management functions.

Another group of proxies, for example caching proxies, aims at data transfer performance improvement. An interesting approach is presented in [4]. The authors discuss on the possibility of introducing customized proxies in the network, which can perform the functions of content filtering, compression, encryption, remote cashing, and other on a per-client basis.

Furthermore, both cross-layer design and agent-based networking are considered to constitute an essential basis for novel networking paradigms, such as active networking [5] and cognitive networking [6].

The ARQ proxy approach introduced in this paper combines all three design paradigms present in networking literature, i.e. layering, cross-layering, and agent-based networking. It allows abstraction of 'atomic' functions from a specific protocol layer and provides means for detaching the abstracted functional blocks from the protocol stack in order to relocate them within the network ('in-network').

In ARQ proxy, the functional element responsible for TCP ACK generation at the mobile node is abstracted and relocated at the BS. The operation of the relocated TCP ACK generation block is fully controlled using the link layer feedback channel maintained between the BS and the mobile node. Further details on the broader concept of distributed protocol stack can be found in [7].

4. ARQ PROXY APPROACH

4.1. *Proposed approach*

The basic idea behind the proposed ARQ proxy approach is to avoid the transmission of standalone TCP ACK packets over the radio link between the Base Station (BS) and Mobile Terminal (MT) together with associated overhead added at the 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 within the protocol stack: the ARQ Proxy and ARQ Client (see Figure 2).

ARQ Proxy is a software module located in the protocol stack of the wireless BS. Having access to TCP and IP headers of the in-transit traffic, ARQ Proxy generates a TCP ACK for every TCP data packet destined to MT (to confirm successful data reception up to the flow segment carried in this TCP data packet).

ARQ Proxy does not require any flow-related state information or TCP layer implementation in a conventional sense. Indeed, TCP ACK is generated using a simple memory copy operation applied to the fields (IP addresses, port numbers, and flow sequence numbers) of the received TCP data packet into a previously generated template of TCP ACK. As a result, there is no conventional TCP layer functionalities need to be supported at the BS, and no TCP state related information needs to be stored. This allows ARQ proxy approach scales well in large network systems supporting high load at the BSs.

The fact that no TCP flow state-related information is used in TCP ACK generation process implies the assumption that all the segments of a given TCP flow are successfully received at the destination node. Since this assumption is not always true, TCP ACKs generated by ARQ Proxy module are not released to the Fixed Host (FH) immediately, but stored in BS memory until requested by the ARQ Client.

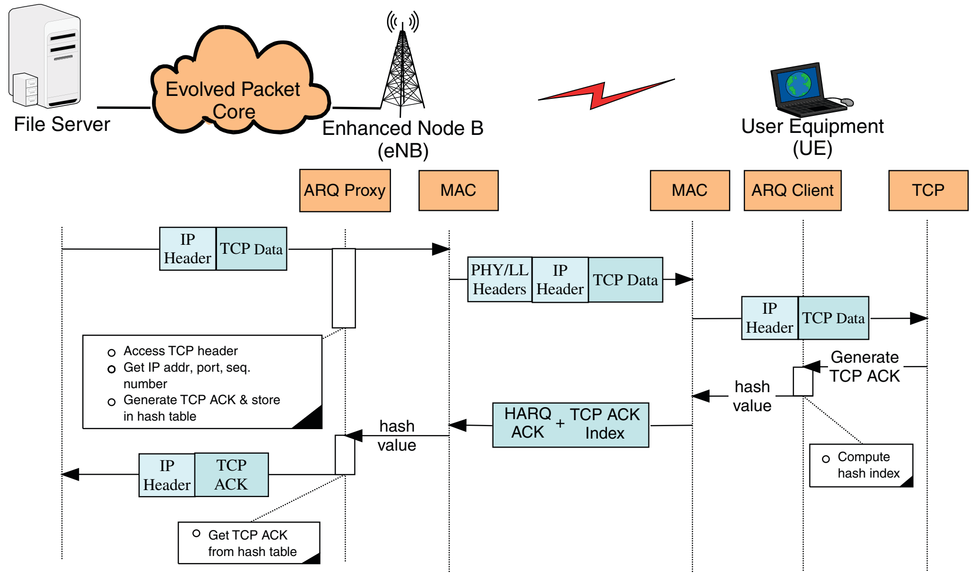

Figure 2. ARQ Proxy and ARQ Client functionality in 3G LTE.

ARQ Client is a software module whose logic position is between the link and transport layers of the MT protocol stack. The purpose of the module is to suppress all outgoing standalone TCP ACK packets and replace them with MAC layer requests for the release of appropriate TCP ACK segment by the ARQ Proxy.

In order to do so, whenever a standalone TCP ACK is produced at the MT transport layer, a TCP ACK suppression request is immediately scheduled for transmission at the link layer, while the original TCP ACK packet travels down the protocol stack (which involves corresponding processing at each layer, output queuing delay, shared medium access and other procedures).

Whichever comes first to the physical layer (the TCP ACK or the corresponding suppression request) will be transmitted, while the other one cancelled.

Figure 3 shows ARQ Client position and operation in MT's protocol stack. The design choice is made towards the use of ARQ/Hybrid ARQ (HARQ) signaling as bearer for TCP ACK suppression request. However, this does not represent a limitation of the proposed approach since other approaches could be implemented.

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

1. TCP ACKs which correspond to connection establishment or connection termination phases of TCP. Indicated by SYN and FIN header flags, such packets carry maximum window, initial flow sequence numbers and other crucial parameters for correct data flow set-up or termination.
2. TCP ACKs encapsulated into outgoing data segments. In case of bidirectional data transfer and delayed-ACK option enabled, the receiver is capable to encapsulate TCP acknowledgement, simply by setting TCP ACK bit and acknowledgement sequence number into outgoing data packet header. As a result, encapsulated TCP ACKs 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 BS are unidirectional where data packets are originated at content server.
3. Duplicate TCP ACKs. In case out-of-order TCP packet is received, MT should send duplicate TCP ACK for the last successfully received segment. However, BS does not maintain any flow related information and thus cannot 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 the related

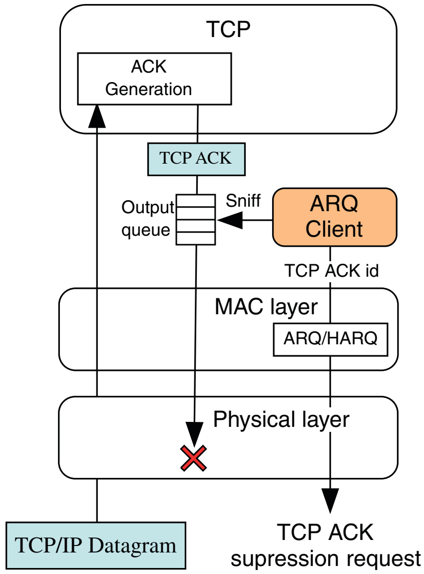

Figure 3. ARQ Client position within the MT protocol stack.

request to ARQ Proxy module. As a result, TCP ACK generated by ARQ Proxy and stored in BS's memory 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 BS is not running out of the receiver buffer space.

In order to trigger TCP ACK transmission, the ARQ Client module implemented at MT should clearly specify which of the TCP ACKs generated by the ARQ Proxy should be transmitted to the fixed host. For that reason, TCP ACK suppression request contains a TCP ACK identification (afterwards referred to as TCP ACK ID).

TCP ACK ID should be easily computable at both BS and MT without the need for direct communication. There are three alternative approaches that satisfy this property which correspond to different ARQ Proxy implementation in different wireless network environments: hash values are used in 3G LTE, link layer frame sequence number are adapted for Wi-Fi, and additional HARQ codewords are considered for WiMAX networks.[‡]

4.2. ARQ proxy in 3G LTE networks

3G LTE is an attempt to step into wireless broadband taken by cellular providers and equipment vendors [8] offering peak data rates of up to 100 Mb/s and reduced connection setup latency while offering high mobility at low deployment and operation costs [9].

The first design choice introduced at the evolved link layer favoring ARQ Proxy implementation is related to variable-size RLC protocol data units in 3G LTE while the second comes from the introduction of HARQ at the evolved link layer designed to compensate high error rates of the radio channel, which cause dramatic performance reduction of TCP protocol.

In HARQ, each erroneously transmitted packet is not dropped at the receiver and negative feedback is sent to the sender. Each retransmission contains additional redundancy information (incremental redundancy) which in case of a subsequent error is considered jointly with the previously transmitted erroneous packet in an attempt for combined recovery. As a result, HARQ

[‡]In the following three sections we use the following terms indicating Mobile Terminal (MT) in different network technologies interchangeably: User Equipment (UE) for 3G LTE, Mobile Node (MN) for WiFi, and Mobile Station (MS) for WiMAX; as well as the following terms for indicating Base Station (BS): enhanced NodeB (eNB) for 3G LTE and Access Point (AP) in WiFi.

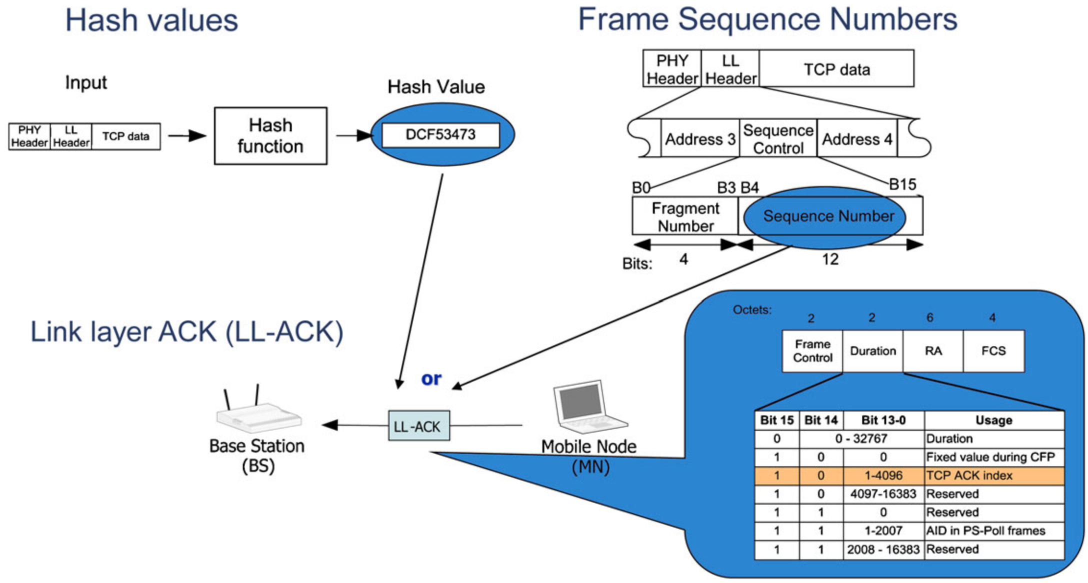

Figure 4. Use of hash functions and frame sequence numbers for packet identification.

implements a stop-and-wait retransmission scheme, producing positive or negative feedback for every frame transmitted at the link layer.

In this framework, ARQ Proxy implementation in 3G LTE networks integrates the feedback channel for sending TCP ACK IDs with HARQ feedback, with the main purpose of saving system resources and reducing communication overhead.

Figure 2 illustrates a single TCP data packet delivery in a 3G LTE network using the ARQ Proxy approach. Whenever a TCP data packet is detected, the ARQ Proxy implemented at the Enhanced Node B (eNB) generates a TCP ACK confirming the reception of TCP data packet stored in a local hash table.

The TCP ACK index in this hash table (which corresponds to TCP ACK ID) is computed by applying a selected hash function to TCP data packet headers based on which the TCP ACK was generated (see Figure 4).

The use of hash tables and hash functions for packet identification allows independent TCP ACK ID calculation at eNB and UE, in order to enable the UE to obtain it for every generated TCP ACK packet by simply applying the same hash function to the fields of the received TCP data.

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 lifetime is exceeded, the packet is silently dropped from the hash table (without taking any other action). By this mechanism, eNB ensures that the memory resources are freed for TCP ACKs not requested by UE. This happens in case the TCP data segment 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 [10].

Traditionally, hash functions are used in cryptography, data storage and search applications. In networking, the use of hash functions is mostly limited to integrity check, error detection and error correction techniques—commonly performed using Cyclic Redundancy Check (CRC) or MD5 algorithms—and peer-to-peer lookup services.

At the UE side (see Figure 3), whenever the substitution request is required to be transmitted, the ARQ Client passes the computed hash value to the link layer. The link layer uses HARQ ACK message to acknowledge successful delivery of each PDU for fixed bandwidth reservation requests extending it with a one-bit value. If set, it requires eNB to extend the resource grant in order to allow the UE to transmit a fixed size hash value during the next transmission opportunity.

Although HARQ is requesting bandwidth resources for hash value transmission, the originally generated TCP ACK packet travels down the protocol stack: whichever comes first to the physical layer for 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 ARQ Proxy disabled).

4.3. ARQ proxy in Wi-Fi networks

ARQ Proxy implementation in Wi-Fi network differs from its implementation in other networks by the packet identification technique used for TCP ACK identification in suppression request as well as the way the suppression request is transmitted.

IEEE 802.11 standard specifies that every sender needs to mark outgoing frames with continuously incremented, 12-bit long sequence numbers at the link layer (see Figure 4). In case of 4TCP/IP datagram fragmentation at the link layer, frame sequence number remains the same for all the fragments. As a result, ARQ Client located at the MN can indirectly identify TCP ACK generated by ARQ Proxy, by referring to the frame sequence number added by the BS at the link layer to the TCP data packet used in TCP ACK generation.

The transmission of TCP ACK suppression requests is performed at the feedback channel organized between MT and BS at the link layer. Specifically, 12 of 14 bits reserved in the 'duration' field of IEEE 802.11 link layer acknowledgement (LL-ACK) are required as shown in Figure 4. Such technique does not require modification of the frame structure specified by Wi-Fi standard, thus favoring interoperability and incremental deployment of the scheme.

The use of the reserved portion of LL-ACK frame favors incremental deployment of the proposed technique enabling operation in a mixed network environment where nodes implementing ARQ Proxy co-exist with those not implementing the proposed approach.

4.4. ARQ proxy in WiMAX networks

Similar to other wireless systems, in particular to 3G LTE, WiMAX implements several error recovery technologies at different layers of the protocol stack operating over the wireless link, i.e. cumulative or selective ARQ at the link layer and multichannel stop-and-wait HARQ at PHY-MAC layers. In the latter case, the sender should wait for receiver feedback for every packet sent on any particular HARQ channel.

In WiMAX, a single TCP DATA packet delivered from the WAN requires MT to send an HARQ ACK response for every successfully received TCP data packet forwarding it to the upper layers of the protocol stack for processing. Consequently, the transmission of TCP ACK generated at MT transport requires bandwidth reservation, transmission of full TCP/IP datagram over the wireless link, and HARQ ACK from the BS.

In terms of the overhead produced by acknowledgements generated at different layers:

- A standalone TCP ACK requires at least 46 bytes (20 for TCP header, 20 for IP header, and 6 for MAC header). If ARQ is enabled, there are another 4 bytes for the ARQ CRC.
- An Uplink (UL) HARQ acknowledgement occupies considerable portion of the uplink slot, whereas a Downlink (DL) HARQ ACK is only one bit placed in the downlink HARQ ACK IE contained in the DL-MAP message.
- The size of an ARQ acknowledgement depends on the employed acknowledgement scheme and is quite variable. However, its minimum value is 4 bytes per data block.

The overhead analysis presented above shows theoretical benefits of the proposed ARQ Proxy approach in WiMAX networks.

Similar to 3G LTE, WiMAX networks use hash values obtained by application of a predefined hash function on the TCP data packet headers, which allows independent hash value computation at both BS and MT.

The main difference comes from the way TCP ACK suppression requests are passed to the BS. Following the analysis of the error recovery schemes implemented by the IEEE 802.16 standard, the HARQ positioned at the physical/link layers appears to be suitable for this task, as it follows a multichannel stop-and-wait approach.

HARQ information is transmitted by the MS in the appropriate HARQ ACK region where only the uplink PUSC permutation scheme is allowed. According to IEEE 802.16 standard, with this permutation scheme an OFDMA slot is made up of six tiles. The even tiles (Tile0, Tile2, and Tile4) are used for one HARQ ACK and the odd tiles (Tile1, Tile3, and Tile5) are used for another HARQ ACK that can belong to a different MS.

In WiMAX, the importance of HARQ information is high since its incorrect reception could cause the loss and further retransmission of a large amount of data. For this reason, HARQ ACK region it protected with the most robust modulation scheme (QPSK). Moreover, instead of Forward Error Correction (FEC), proprietary error protection scheme is designed for HARQ ACK channel with the purpose to provide high redundancy. Specifically, the HARQ ACK and NACK are encoded with a 3-symbol codeword where each symbol ranges between '0' and '7': 0.0.0 for ACK and 4.7.2 for NACK.

For each of the eight symbols, the IEEE 802.16 standard defines the exact modulation pattern to use in the tile where it is carried, i.e. which QPSK symbol has to be transmitted on each subcarrier of the tile. Therefore, error protection of the HARQ acknowledgements is given by two components:

1. Redundancy at the level of modulation patterns: only eight modulation patterns are used although 65536 (48) of them are available within a tile.
2. The redundancy at level of code-words: there are 512 (83) codeword available, but from this set only two codewords (0.0.0 for ACK and 4.7.2 for NACK) are used.

In order to enable HARQ acknowledgments to carry additional information required by ARQ Proxy approach, additional combinations of codewords may be used at the expense of reduced redundancy in HARQ error protection.

Since each tile contains one symbol among eight, it can carry three bits only and since a HARQ ACK region consists of three tiles, this last one can contain only nine bits.

Therefore, a HARQ acknowledgement cannot directly carry a TCP ACK ID employed by ARQ Proxy scheme, which is usually larger than nine bits in size. However, additional codewords can be utilized to carry uplink bandwidth reservation requests. In fact, assuming fixed size for the TCP ACK ID, the MS can simply specify the number of TCP ACK IDs it wishes to transmit including it into the HARQ acknowledgement. This can be accomplished using a subset of the 512 codewords and by associating to each codeword of the subset a specific number of TCP ACK IDs that the MS has to transmit. However, by increasing the number of used codewords, the HARQ ACK error rate, or more generally the codeword error rate (CWER), can increase and therefore the set of used codeword should remain relatively small.

The number of new codewords to introduce depends on the number of TCP data packets transmitted in the downlink HARQ packet. Since TCP packets usually have a big size and this commonly corresponds to Ethernet MTU of 1500 bytes, it is unlikely to have more than eight TCP packets in a single downlink frame. Therefore, it seems reasonable to extend the number of used codewords in order to permit to a MS to require the bandwidth resources needed for the transmission of eight TCP ACK ID values. This choice does not sensibly affect the HARQ ACK error rate if the new codewords are chosen so that to maximize the Euclidean distance between the used codewords.

4.5. *Benefits and limitations*

End-to-End (E2E) semantics: 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

BS are triggered by the MT. Moreover, MT requests the transmission of those ACKs which are consistent with the employed acknowledgement strategy (delayed or selective acknowledgement [11]), while TCP ACKs which are not requested are silently dropped at the eNB when their lifetime is exceeded.

Performance and system capacity: Reduction of overhead deriving from fine-tuning multiple ARQ schemes implemented at different layers of the protocol stack releases uplink network resources, which can be further reused by the same MT or other nodes operating in the same cell.

Channel error rate: ARQ Proxy approach substitutes the transmission of a relatively large TCP ACK packet with a small (several bytes long) substitution request over the radio channel. As a result, the effect of channel error propagation is reduced for such packets, meaning that with the same error rate the corruption is more likely to happen with big rather than with a small packet.

Round trip time (RTT): TCP throughput and error recovery performance (and in general its reactiveness) depends on the round trip time (RTT) between the sender and the receiver [12]. In 3G LTE, RTT is composed by 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 state (active or idle), and employed framing. This reduction is typically in the order of tens of milliseconds [13].

Mobility: The support of full inter- and intra-network mobility is one of the most crucial requirements in modern wireless networks. Moreover, the handover of MTs should be performed with minimum delay to ensure uninterrupted service execution at guaranteed quality. For that reason, approaches maintaining MT state or profile at the BS are not desirable due to the delay overhead associated with the state transfer during handover.

In our scheme, in case MT changes its location and registers with another BS, the hash table stored by the ARQ Proxy at old BS is not transferred to the new BS and can be deleted. In this way, after location update, the MT will send hash values for only those TCP ACKs which correspond to packets received from the new BS.

Incremental deployment: ARQ Proxy can be incrementally deployed in already operational networks, where MTs and BSs implementing the proposed approach co-exist with those that do not.

For example, in case MT does not include ARQ Client module, none of the TCP ACKs generated at the BS will be requested using their hash values and will be simply destroyed after their lifetime expiration. On the other hand, in case BS does not implement ARQ Proxy approach, all bandwidth allocation requests sent by MTs 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.

5. PERFORMANCE EVALUATION

5.1. Wi-Fi scenario

In order to analyze the performance of ARQ Proxy in Wi-Fi network, additional modules of the NS-2 network simulator (version 2.31) [14] are added supporting ARQ Proxy and ARQ Client functionality. ARQ Proxy module is attached to the BS, while ARQ Client is located in MN protocol stack. The configuration of the wireless link between BS and MN follows IEEE 802.11b specification parameters with 11 Mb/s physical data rate. Parameters of the wired link (100 Mb/s, 15 ms) model the situation when a mobile user is connecting to an Internet server physically located within the same metropolitan area. The BS ingress buffer is limited to 700 packets, and RTC/CTS exchange is turned off at the MAC layer as the most appropriate configuration widely used in infrastructure network scenario. Initial results of ARQ proxy implementation in WiFi scenario are provided in [15].

TCP NewReno is chosen for performance evaluation as the most widespread TCP version in Internet. However, it is important to underline that ARQ Proxy approach is not constrained to any specific TCP implementation.

Connection throughput and RTT are chosen as main performance metrics of TCP flow evaluated against variable TCP/IP datagram size as well as PER on the wireless link. The obtained 95%

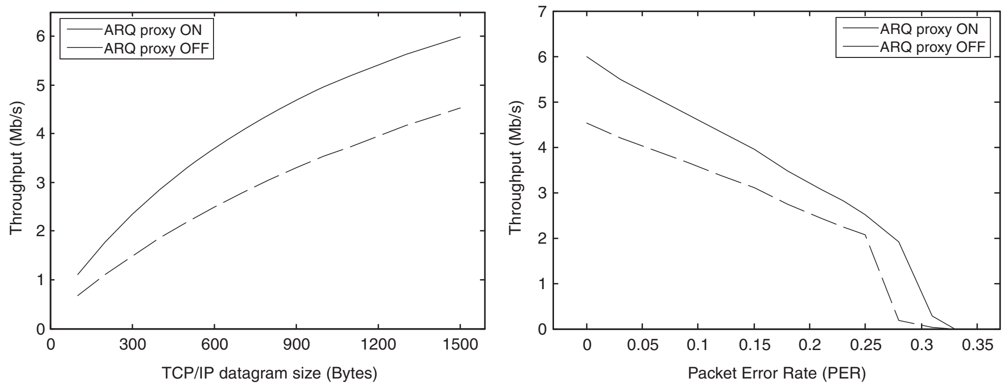

Figure 5. Throughput vs datagram size (left) and throughput vs error rate (right) performance of ARQ Proxy in WiFi networks.

confidence intervals appear to be less than 1/100th of the reported values, due to the throughput averaging over simulation time, and thus have not been included in the graphs.

Throughput: Figure 5 shows the throughput level achieved by TCP NewReno for different TCP/IP datagram sizes. The level of ARQ Proxy improvement is reversely proportional to the size of TCP data packet. For large TCP/IP datagram size of 1500 bytes, the achieved throughput improvement is around 25–30%. However, while for small packets (40–200 bytes) it is in the range of 60–70%, the general rule is that for TCP data packets which tend to be similar in size to TCP ACK packets the throughput improvement can reach 100%.

RTT: Along with throughput performance improvement, ARQ Proxy reduces RTT of TCP connection. TCP ACKs generated at the BS by ARQ Proxy agent avoid transmission, propagation, and queuing delays experienced at the wireless link. This delay is typically in the order of several milliseconds for IEEE 802.11b.

RTT reduction leads to TCP flow performance increase due to faster window evolution and faster reaction to packet drops performed by the AIMD flow control mechanism [16].

Error tolerance: Figure 5 illustrates TCP throughput with variable link error rate and TCP/IP datagram size equal to 1500 bytes. Although the throughput level is linearly decreasing, ARQ Proxy performance improvement remains constant and corresponds to around 30% for PERs of up to 0.25. Additionally, by enabling ARQ Proxy, TCP NewReno is able to sustain higher PERs (see Figure 4 for PER $>$ 0.25). This is motivated by the fact that no wireless link errors propagate into TCP ACKs generated at the BS in such scenario.

5.2. 3G LTE scenario

For ARQ proxy performance evaluation in 3G LTE networks Enhanced UMTS Radio Access Extensions (EURAE) [17] for NS2 network simulator (version 2.31) [14] 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 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 m from the Node B. The TCP connection is initiated between the file server located in the network core and the receiver sink attached to the UE. Initial results of ARQ proxy implementation in 3G LTE systems are provided in [18].

Throughput: 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 composed 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 the HARQ ACK.

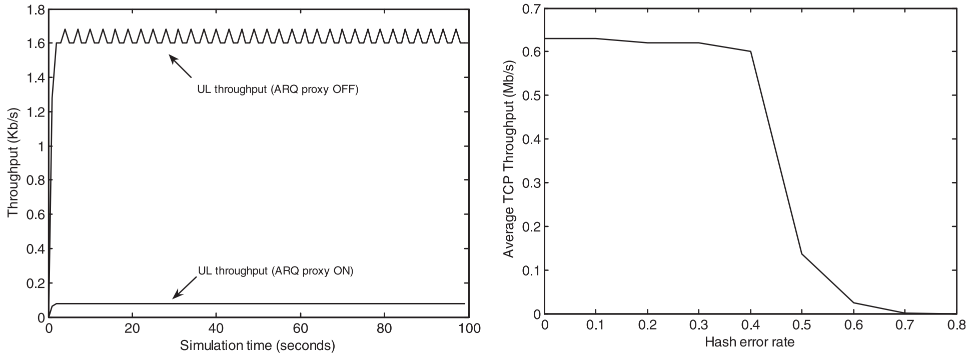

Figure 6. Throughput performance of ARQ Proxy in 3G LTE networks (left) as well as its tolerance of errors of the hash values (right).

RTT: Along with the uplink capacity improvement, the proposed ARQ proxy scheme aims at RTT reduction due to faster TCP ACK feedback. The ARQ Proxy-enabled case shows the RTT close to the delay experienced in the network core, in 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.

Error tolerance: 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 motivates the evaluation of the influence of errors on the hash value transmission (see Figure 6). In case the eNB receives a corrupted hash value, no TCP ACK is generated from the hash table toward the TCP sender.

The results show that 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 protection into hash value transmission.

5.3. WiMAX scenario

In order to evaluate ARQ proxy performance in WiMAX networks NS-based [14] system-level simulator compliant to IEEE 802.16e standard was developed by Nokia Siemens Networks. This simulator implements the OFDMA PHY mode with TDD duplexing by using the frame parameters specified in the MTG profile [19]. At the MAC layer, ARQ and HARQ are extended with the functionalities required by ARQ Proxy support. Initial results of ARQ proxy implementation in WiMAX systems are provided in [20].

The simulated network is a regular hexagonal cellular topology which includes a central tri-sectorial target site for statistics measurement and two additional interference rings, i.e. 19 independent sites in total. An intra-site frequency reuse-3 is used thus making available in each sector of a site 5 MHz of bandwidth around the central frequency of 2.5 GHz. In this scenario, a downlink TCP connection is established between a file server located in the fixed network and each of the 15 MSs placed in the traced cell of the simulated network.

The wireless channel is modeled by considering the SIEMENS Ray-Tracing@ 2.5 GHz model for the path loss, a log-normal shadowing with 8 dB of standard deviation and fast fading. The latter is simulated run-time by a trace previously produced through link level simulation of the SCME channel model developed in the context of the WINNER project. The BS antenna (one for each sector) is modeled assuming a 65° (−3 dB) with 50 dB of front-to-back ratio pattern and it is characterized by a gain of 17.5 dBi and a transmission power of 40 dBm. MSs instead are supposed to be equipped with omnidirectional antennas. All system parameters are summarized in Table II.

Figure 7 underlines ARQ proxy improvements in WiMax network in terms of system capacity increase and higher error rate tolerance.

Table II. System parameters.

Parameter	Value
Number of sites	19 (two interference rings)
Number of sectors per site	3
BS-BS distance	0.9 km
Center frequency	2.5 GHz
Reuse scheme	Reuse-3
Total channel bandwidth	15 MHz
Channel bandwidth per sector	5 MHz
NFFT	512
Frame duration	5 ms
BS PHY model	
BS Tx power/sector	40 dBm
BS antenna height	30 m
BS antenna pattern	65° (−3 dB), 50 dB front-to-back ratio
BS antenna gain	17.5 dBi
TS PHY model	
TS antenna height	1.5 m
TS antenna pattern	Omnidirectional
TS noise figure	7 dB
TS mobility	fixed

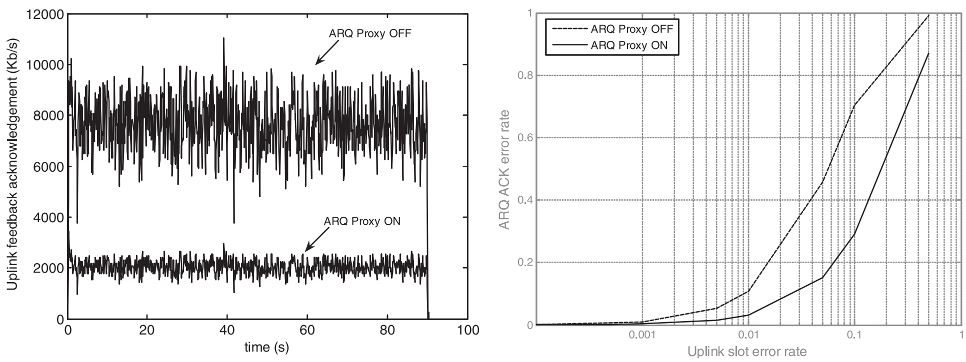

Figure 7. System capacity (left) and error tolerance (right) performance of ARQ Proxy in WiMax networks.

Throughput: In this scenario, total uplink capacity is mostly dedicated for TCP ACK feedback making it a good indicator for evaluation of ARQ Proxy performance in terms of system capacity increase. Numerically, the results derive from the difference in size between a standalone TCP ACK and its corresponding hash value, and equals approximately to 5.6 Kb/s.

Error tolerance: In WiMAX, hash values are embedded in MAC PDUs, so that they may be protected by error recovery procedures. Moreover, small in size hash values reduces exposure to the link errors if compared with large TCP ACK frames. As a result, fewer ACKs are getting lost which corresponds to higher uplink capacity as well as an increased TCP flow performance. This fact is confirmed by simulation results presented in Figure 7 which shows PER of the ARQ packets sent by a MS, obtained by varying the uplink SLER (slot error rate).

RTT: Smooth RTT reported for the downlink TCP connection is composed of backbone network delay (between the fixed server and BS) and wireless (between BS and MS) components. The ARQ Proxy approach reduces the time required for TCP ACK transmission over the wireless links from MSs by reducing its following components: (i) uplink transmission errors and hence the retransmission of acknowledgements in uplink, and (ii) time required for performing uplink bandwidth requests. The latter point is justified by the fact that bandwidth requests made using ARQ or HARQ acknowledgements are faster than those made using conventional reservation.

DOI: 10.1002/dac

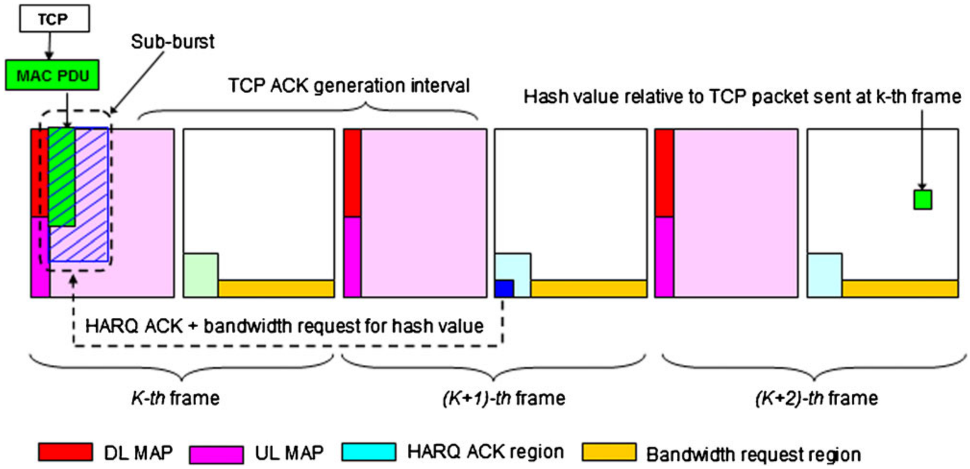

Figure 8. HARQ-based bandwidth request using only one HARQ channel.

In fact, in case of an HARQ-based request, the transmission request for a hash value (corresponding to a TCP data packet) can be made by using the HARQ ACK related to the HARQ packet which carries the TCP data packet itself in downlink, as shown in Figure 8.

A further reduction in the bandwidth request delay occurs when using multi-channel HARQ. It allows sending request for the hash value relative to the TCP packet received by the MS at frame *k*th is sent just in that frame by employing the HARQ acknowledgement of the HARQ packet received at (*k*-1)th frame.

6. DISCUSSION AND CONCLUSIONS

Most of the wireless technologies aimed at the provisioning of last-mile data access implement stop-and-wait ARQ protocols in order to counteract high error rates of the wireless channels. Being commonly introduced in all wireless standards, link-layer ARQ overlaps in functionality with other ARQ protocols operating at higher layers (like TCP). Specifically, a single TCP data packet transmission can be acknowledged three times: one at the transport level and two times at the link layer, and for each acknowledgement the corresponding physical and link layer overhead is added.

In this paper we consider such overlap in ARQ functionalities as a potential point for optimization and propose the ARQ Proxy approach. ARQ proxy is a cross-layer solution which substitutes the transmission of standalone TCP ACK packets with a short request encapsulated into link layer exchange over the wireless link. This request includes packet identification information which is obtained by application of a predefined hash function onto the raw packet data.

Furthermore, the paper describes how the proposed scheme, called ARQ Proxy, can be deployed in three different wireless technologies (3G LTE, Wi-Fi, and WiMAX) and provides extensive validation of the achievable improvement through simulations.

REFERENCES

1. Balakrishnan H, Padmanabhan VN, Seshan S, Katz RH. A comparison of mechanisms for improving TCP performance over wireless links. *IEEE/ACM Transactions on Networking* (*TON*) 1997; **5**(6):756–769.
2. Srivastava V, Motani M. Cross-layer design: a survey and the road ahead. *IEEE Communications Magazine* 2005; **43**(12):112–119.
3. Ramakrishnan K, Floyd S, Black D. The addition of explicit congestion notification (ECN) to IP. *RFC 3168*, September 2001.
4. Steinberg J, Pasquale J. Improving user relocatability, practicality, and deployment in the web stream customizer System. *Proceedings of the Tyrrhenian International Workshop on Digital Communications* (*TIWDC*), Ischia, Italy, September 2007.

5. Tennenhouse DL, Smith JM, Sincoskie WD, Wetherall DJ, Minden GJ. A survey of active network research. *IEEE Communications Magazine* 1997; **35**(1):80–86.
6. Mahmoud Q. *Cognitive Networks*: *Towards Self-Aware Networks*. Wiley: New York, 2007.
7. Kliazovich D, Granelli F. Distributed protocol stacks: a framework for balancing interoperability and Optimization. *International Workshop on Computer Aided Modeling*, *Analysis and Design of Communication Links and Networks*, Beijing, China, September 2008.
8. Clash of the Titans—WiMAX and 4G: the Battle for Convergence is joined. *Market Report*. Maravedis Research and Analysis, July 2006.
9. 3GPP RP-040461. Proposed Study Item on Evolved UTRA and UTRAN.
10. Partow A. General purpose hash function algorithms. Available from: http://www.partow.net/programming/hashfunctions/.
11. Clark D. Window and acknowledgement strategy in TCP. *RFC 813*, IETF, July 1982.
12. Ott TJ, Kemperman JHB, Mathis M. The stationary behavior of ideal TCP congestion avoidance. *Internetworking*: *Research and Experience* 1992; **11**:115–156.
13. 3GPP R2-061866. Non-synchronized random access in E-UTRAN. Ericsson.
14. NS-2 simulator tool home page, http://www.isi.edu/nsnam/ns/, 2000.
15. Kliazovich D, Ben Halima N, Granelli F. Cross-layer error recovery optimization in WiFi networks. *Tyrrhenian International Workshop on Digital Communication* (*TIWDC*), Ischia island, Naples, Italy, September 2007.
16. Kelly F. Mathematical modelling of the internet. In *Mathematics Unlimited—2001 and Beyond*, Engquist B, Schmid W (ed.). Springer: Berlin, 2001; 685–702.
17. Enhanced UMTS Radio Access Extensions (EURAE) home page. Available from: http://www.ti-wmc.nl/eurane/.
18. Kliazovich D, Granelli F, Redana S, Riato N. Cross-layer error control optimization in 3G LTE. *IEEE Global Communications Conference* (*GLOBECOM*), Washington, DC, U.S.A., December 2007.
19. WiMAX Forum™ Mobile System Profile, Release 1.0 Approved Specification, November 2006.
20. Kliazovich D, Beniero T, Dalsass S, Serrelli F, Redana S, Granelli F. Cross-layer error control optimization in WiMAX. *IEEE Global Communications Conference* (*GLOBECOM*), New Orleans, LA, U.S.A., December 2008.

AUTHORS' BIOGRAPHIES

Dzmitry Kliazovich is an ERCIM Fellow at the University of Luxembourg and VTT Technical Research Centre of Finland. He holds an award-winning PhD in Information and Telecommunication Technologies from the University of Trento (Italy).

In 2005/2006 he was a visiting researcher at the Computer Science Department of the University of California at Los Angeles (UCLA). In 2006 he was a Research Fellow at Nokia Siemens Networks setting up the research activities in the field of 3G Long-Term Evolution (LTE). He is a holder of several scientific awards, including Best Paper Award received at the IEEE/ACM International Conference on Green Computing and Communications (GreenCom) for his work in energy-efficient network-aware scheduling in data centers, as well as grants provided by the European Research Consortium for Informatics and Mathematics (ERCIM), IEEE Communication Society, Italian Ministry of Education, and the University of Trento.

He is an author of more than 50 research papers, active member of IEEE ComSoc Technical Committee on Communication Systems Integration and Modeling (CSIM), and an organizer of the CSIM Flagship Workshop CAMAD in 2006, 2010, and 2011. His main research activities are in the field of networking with the focus on cognitive protocols, energy efficiency and next-generation networking.

Dr Simone Redana is a Senior Research Engineer in the Radio Research unit of Nokia Siemens Networks, leading research and standardization projects on relaying beyond LTE Rel.10. He received his PhD degree in Telecommunication Engineering from Politecnico di Milano, Italy in 2005. In 2005 and 2006 he was with Azcom Technology as a consultant for Siemens Communication working on WIMAX demonstrator. In 2006, he joined Siemens Communication in Milan, Italy, which merged in 2007 with Nokia Networks to become Nokia Siemens Networks. Since 2008, he is with Nokia Siemens Networks in Munich, Germany. He contributed to the EU WINNER II project (relaying concept), to the Eureka Celtic project WINNER+ (system concept design) and to EU ARTIST4G project leading the work package on advanced relay concept design. He is currently contributing to the 3GPP RAN standardization of Relaying in Rel.10 and beyond as well as involved in the business case analysis for relays.

Fabrizio Granelli received the Laurea (MSc) degree in Electronic Engineering from the University of Genoa, Italy, in 1997, with a thesis on video coding, awarded with the TELECOM Italy prize, and the PhD in Telecommunications from the same university, in 2001. Since 2000 he is carrying on his research and didactical activities at the Department of Information Engineering and Computer Science—University of Trento (Italy), where he is currently Associate Professor in Telecommunications. In August 2004 and August 2010, he was a visiting professor at the State University of Campinas (Brasil). He is author or co-author of more than 100 papers published in international journals, books and conferences. Moreover, Dr Granelli is a guest-editor of ACM Journal on Mobile Networks and Applications, special issues on 'WLAN Optimization at the MAC and Network Levels' and 'Ultra-Wide Band for Sensor Networks' and Co-Chair of 10th IEEE Workshop on Computer-aided Modeling, Analysis, and Design of Communication Links and Networks (CAMAD'04). Dr Granelli is General Vice-Chair of the First International Conference on Wireless Internet (WICON'05) and General Chair of the 11th and 15th IEEE Workshop on Computer-aided Modeling, Analysis, and Design of Communication Links and Networks (CAMAD'06 and CAMAD'10). He is Co-chair of GLOBECOM 2007–2009 Symposia on 'Communications QoS, Reliability and Performance Modeling.'

His main research activities are in the field of networking and signal processing, with particular reference to network performance modeling, cross-layering, medium access control, wireless networks, cognitive radio systems, and video transmission over packet networks.

He is a Senior Member of IEEE, former Chair of IEEE ComSoc Technical Committee on Communication Systems Integration and Modeling (CSIM) and an Associate Editor of IEEE Communications Letters, IEEE Communications Surveys and Tutorials, International Journal on Communication Systems, Journal of Wireless Communications and Networking.