Packet concatenation at the IP level for performance enhancement in wireless local area networks

Dzmitry Kliazovich · Fabrizio Granelli

Abstract Wireless local area networks experience performance degradation in presence of small packets. The main reason for that is the large overhead added at the physical and link layers. This paper proposes a concatenation algorithm which groups IP layer packets prior to transmission, called PAC-IP. As a result, the overhead added at the physical and the link layers is shared among the grouped packets. Along with performance improvement, PAC-IP enables packet-based fairness in medium access as well as includes QoS support module handling delay-sensitive traffic demands. The performance of the proposed algorithm is evaluated through both simulations and an experimental WLAN testbed environment covering the single-hop and the widespread infrastructure network scenarios. Obtained results underline significant performance enhancement in different operating scenarios and channel conditions.

Keywords Packet concatenation · WLAN performance optimization · IEEE 802.11

1 Introduction

Wireless local area networks (WLANs) are becoming increasingly popular in the world of telecommunications, especially for the provisioning of mobile access to network services. The nature of wireless links involves aspects such

D. Kliazovich · F. Granelli (✉)
DIT - University of Trento, Via Sommarive 14,
I-38050 Trento, Italy
e-mail: granelli@dit.unitn.it
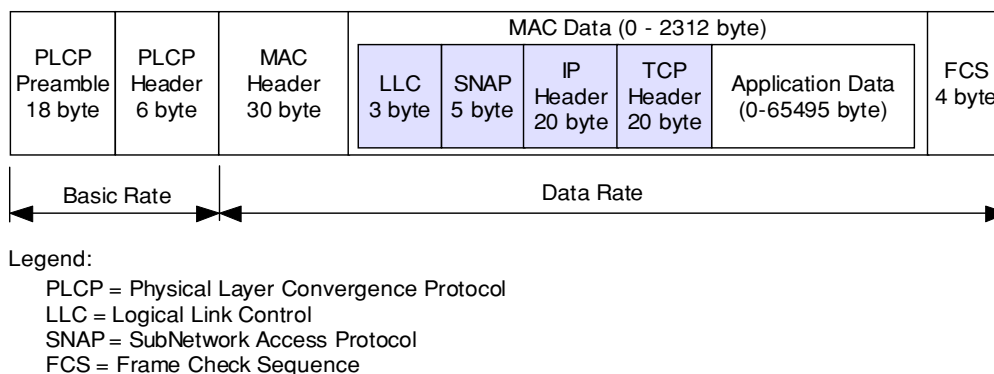
D. Kliazovich
e-mail: klezovic@dit.unitn.it

as limited bandwidth, increased latency, channel losses, mobility, etc., which influence the packet delivery process. In order to deal with such operating limitations, wireless link and physical layers introduce a number of functions such as receiver/transmitter synchronization, contention resolution, rate adaptation, and so on, performed for every packet transmission. As a result, WLAN performance is highly dependant on the packet size or, equivalently, on the relation between actual data payload and the overhead added at different layers of the protocol stack.

In addition, according to several Internet traffic studies, more than 50% of all the packets traversing paths is smaller than 100 bytes [1, 2]. The main sources producing small packets are acknowledgements generated by the dominant in Internet Transmission Control Protocol (TCP) [3], various multimedia and web (HTTP-based) applications.

The goal of this paper is to discuss performance problems of wireless local area networks (WLANs) which arise in the presence of small packets, defining major points for performance optimization (Section 2). Section 3 presents an approach for packet concatenation at the IP layer, pointing out its advantages and drawbacks. A Quality of Service (QoS) scheme—applicable for different packet concatenation methodologies—is introduced in Section 4. Performance evaluation (Section 5) describes the conducted simulation experiments and test-bed measurements for single-hop and infrastructure WLAN scenarios, which underline theoretical and practical benefits of the presented scheme. Finally, Section 6 concludes the paper drawing final remarks as well as directions for future work on the topic.

2 WLAN performance in presence of small packets

According to the statistics presented by researchers [2], the majority (over 85%) of Internet traffic is TCP-based. The

			MAC Data (0 - 2312 byte)					
PLCP Preamble 18 byte	PLCP Header 6 byte	MAC Header 30 byte	LLC 3 byte	SNAP 5 byte	IP Header 20 byte	TCP Header 20 byte	Application Data (0-65495 byte)	FCS 4 byte

Basic Rate | Data Rate

Legend:
PLCP = Physical Layer Convergence Protocol
LLC = Logical Link Control
SNAP = SubNetwork Access Protocol
FCS = Frame Check Sequence

Fig. 1 Packet encapsulation for IEEE 802.11 protocol stack

reliability of TCP is obtained through the utilization of positive acknowledgement schemes. Such acknowledgements are small-sized packets (40 bytes), meaning that TCP produces a bulk of small packets which traverse the communication networks between the sender and the receiver. Moreover, TCP is not the only protocol which produces small packets on the network. Multimedia applications, which commonly use UDP, also produce small packets, in order to decrease the packet propagation delay and reduce potential information loss associated to the loss of a single packet.

The transmission of application data over a wireless channel requires recursive encapsulation by lower (from transport to physical) layers of the protocol stack. Figure 1 provides an overview of packet encapsulation for an application which employs TCP as the transport layer protocol.

Link and physical layer headers specified by IEEE 802.11 standard [4] represent a relevant overhead if compared with wired network ones. Since this overhead does not depend on the size of the packet, for small packets it can be even several times greater than the actual application data. Thus, the authors of [5] showed that for an ordinary TCP ACK packet "the total overhead is four times the payload!"

IEEE 802.11 standard [4], currently the dominant and widespread solution for WLAN connectivity, together with its extensions (a, b, g) specifies different rates for data transmission ranging from 1 Mb/s to 54 Mb/s. However, the introduced rates are those which are achieved at the physical layer on the wireless channel and not the effective performance in terms of data delivery. The actual meaning of the term 'maximum rate' will change with the release of IEEE 802.11n standard scheduled at the end of 2007 [6], which aims to achieve bitrates exceeding 150 Mbps using MIMO (multiple input multiple output) architectures. In contrast to -a, -b, and -g physical layer extensions, IEEE 802.11n will provide the reported speed not at the physical layer but to the upper layers of the protocol stack.

In the current situation, most of the overhead is related to the PLCP (Physical Level Convergence Protocol) Preamble, which is used for synchronization of the wireless re-

Table 1 Achievable throughput of IEEE 802.11a, b

Physical layer	Link speed (Mb/s)	TCP throughput (Mb/s)	Efficiency (%)
802.11b	1	0.75	74.9
	2	1.41	70.7
	5.5	3.38	61.5
	11	5.32	48.4
802.11a	12	9.2	76.6
	24	16.2	67.5
	54	26.57	49.2

ceiver. This preamble as well as the PLCP header is always transmitted at the basic rate—regardless of the actual link speed. This requirement allows operation at different rates, since the information about the rate of the remaining portion of the PPDU (Physical Protocol Data Unit) is stored in the PLCP header. This implies that, for the transmission of any data frame over the wireless channel in IEEE 802.11b, PLCP preamble and header will take 192 μs (for a basic rate equal to 1 Mbps)—regardless of the achieved bit rate on the channel. This aspect has a relevant impact on the actual performance over the wireless link.

Table 1 summarizes the analysis of achievable (maximum) TCP throughput under the hypotheses of no collisions, no fragmentation, no RTS/CTS and no bursts. Experiments for IEEE 802.11b extension are presented in [7].

Results underline that a high percentage of the wireless link capacity is wasted for the transmission of supplementary information, and the bandwidth which is available for data transmission is far less than the capacity reported at the physical layer.

The maximum achievable TCP throughput is achieved for a TCP/IP datagram size which corresponds to the most common Maximum Transmission Unit (MTU) of 1500 bytes, used in Ethernet LANs. However, the size of packets sent on the networks is far from being fixed at MTU. More than a half of the packets in Internet are in fact smaller than 100 bytes [2], which means that the relationship between performance and packet size becomes a relevant issue. This is underlined
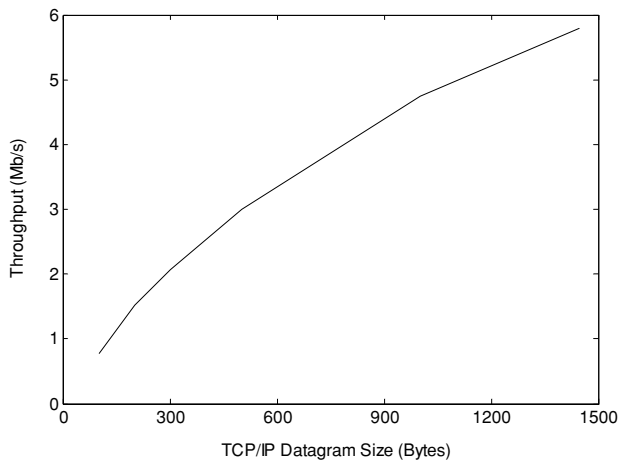
Fig. 2 IEEE 802.11b throughput versus packet size

in Fig. 2, where evaluation results of TCP throughput versus packet size are presented for IEEE 802.11b. IEEE 802.11b extension is chosen for the experiments as the most diffused implementation nowadays, supported by the majority of vendors. However, since they differ only for the physical interface, validity of the achieved results can be extended for conceptual similarity to other versions of the standard (802.11a and 802.11g). The details of the environment for the conducted experiments are described in Section 5.

Figure 2 underlines that for small packets (left part of the graph) the performance of IEEE 802.11b is dramatically decreased. Thus, for packet sizes less than 100 bytes, the throughput can be less than 10% of the available capacity.

As a consequence, the main idea for the optimization of the performance on the wireless channel is to increase the available channel capacity by enlarging the packet size via concatenation of small packets into a large *"group-packet"*. Several solution solutions have been proposed to perform data concatenation at either transport or at the link layer.

At the transport layer, one of the first solutions was introduced by Nagle in 1984—now known as Nagle algorithm [8, 17]. This algorithm aims at reducing the number of small packets which are generated by TCP-based applications (such as Telnet). The main idea of Nagle algorithm is to allow the TCP sender to collect more data coming from the application instead of immediate output of several small segments. The concatenation is limited by the maximum size of the packet that can be built, which corresponds to the maximum segment size of the TCP connection, as well as by the time required for the collection process. Nowadays, Nagle algorithm is a standard requirement for TCP implementations.

Nagle algorithm together with its modifications [9, 18], which do not change the core idea of its operation, forms the group of solutions which implement concatenation at the transport layer.

At the link layer, the concatenation is performed by exploiting the awareness of the physical channel characteristics.

Following this principle, Packet Frame Grouping (PFG) [5] groups small frames at the link level in order to share the header overhead within the whole group. Similar to the fragmentation technique specified in the IEEE 802.11 standard [4], PFG separates the sent data frames and their link level ACKs by Short InterFrame Space (SIFS).

An implementation of the approach requires only minor modifications to the link level protocol, such as a counter for the number of bytes sent in the current frame, for limiting the maximum size of the frame. In contrast to other schemes, PFG is not limited to packets destined to a particular host.

Another approach, called PAcket Concatenation (PAC) [10], concatenates MAC layer frames into a superframe. The selection of packets for such concatenation is based on the next hop address. Each concatenated module is a link layer frame, which includes MAC header and CRC field in order to provide error independence. PAC is able to concatenate up to 9 MAC data frames into a superframe, the delivery of which is acknowledged by the new type of ACK frame. An additional field of this new ACK supports selective acknowledgement of the subframes.

PAC provides more effective overhead reduction if compared with PFG approach, since a single physical preamble and header are shared by the entire superframe.

Summarizing, link layer solutions are designed for finer optimization, achieved by a concatenation scheme which is aware of the wireless medium characteristics. However, most of them imply modification of the standard link layer protocol which requires a big effort from the research community for standardization as well as from industry for the modification of the firmware of wireless devices.

Nevertheless, solutions especially designed for wireless links may mitigate the drawbacks of other upper layer approaches. For example, in most cases they do not introduce additional delay in a single packet delivery, since the concatenation process can be applied to packets already waiting for the medium to become idle in the transmission queue.

The choice of the protocol layer for packet concatenation defines the size of the overhead which may be shared by the entire superframe. From this point of view, the highest performance improvement is achieved by the transport layer concatenation which shares the headers added at all the layers of the protocol stack (from transport down to physical), while link layer concatenation can share physical layer headers only. However, transport layer concatenation is limited in data collection to a single transport layer connection (or socket in TCP/IP notation). On the other hand, the concatenation performed at the link layer considers the data generated by all running applications as well as control data produced internally at kernel level (like ICMP, ARP, etc.).

Following the considerations pointed above, we propose to concatenate IP packets at the network layer. Similar to link

Table 2 Overhead reduced by different concatenation schemes (for 802.11b)

Concatenation layer	Overhead suppressed		Size (Bytes)	Transmission delay (us)	Limitation
	Headers				
Link	PHY Header		24	192	Next-hop router or no limitations
Network	PHY Header · MAC Header		58	216.72	Next-hop router
Transport	PHY Header · MAC Header · IP Header · TCP Header		98	245.8	Particular application

layer concatenation, data collection is not limited to a single application, while overhead suppression is performed for the physical and link layer headers which can significantly waste resources in a WLAN environment.

Table 2 compares the overhead suppressed for every packet concatenated by schemes operating at different layers. Overhead size and transmission delay, calculated for IEEE 802.11b physical channel running at 11 Mb/s, show the benefits that can be achieved by concatenation at the network layer, which is able to achieve a reasonable tradeoff - providing higher overhead suppression than link layer schemes while overcoming the restriction to a particular application as a source of data of transport layer approaches.

3 IP packet concatenation (PAC-IP)

3.1 The concatenation technique

The main idea of PAC-IP is to concatenate network layer packets (IP header + IP payload) into a single *"group-packet"*, which will be considered as ordinary payload at the link layer. This concept is illustrated in Fig. 3.

A single *group-packet* contains only packets destined to the same (Level-2) host, which are chosen on the basis of the MAC-layer address. This means that not only packets with the same IP address can be packed into a *group-packet* but also IP packets which are routed to the same MAC-level device.

After the *group-packet* is built, it is forwarded to the link layer for transmission on the wireless channel. As a result, Link Layer (MAC) and Physical (PHY) layer headers are added only once for the entire collection. As a consequence,

the reduction of the medium busy time due to the elimination of MAC and PHY headers is given by:

$$\left(\frac{PHY\,header + PHY\,preamble}{Basic\,Rate} + \frac{MAC\,header + FCS}{Data\,Rate} \right) \cdot (n-1), \qquad (1)$$

where *n* is the number of IP packets in the *group-packet*.

According to (1), in the case of IEEE 802.11b running at 11Mbps, PAC-IP will save 216.72 μs for every concatenated packet. Further improvement (not covered by the formula) is in elimination of the exponential backoff algorithm and optional RTS/CTS exchange forming medium contention procedure included in the standard [4].

On the receiver side, a *group-packet* is separated into the original IP packets by using the *group-packet* size obtained from the MAC header as well as each IP packet size (a field in the standard IP header). Note that PAC-IP does not modify standardized headers (neither at the link nor at the IP level).

The functionality of PAC-IP has a conceptual similarity with Nagle algorithm [8]. The main difference between them is that Nagle algorithm operates at the byte stream level, while PAC-IP works with IP packets.

In a similar manner as link layer solutions, PAC-IP groups packets to be sent to the same destination. More in detail, it only considers the next hop of the network. There is a variety of scenarios where wireless networks are nowadays employed; among them, the most widely spread nowadays are: (1) wireless-cum-wired (where the wireless hop is the last hop of the network between the base station and wireless node), and (2) multi-hop networks (where the route of the packet goes through several wireless links). In both cases

Fig. 3 PAC-IP concatenation
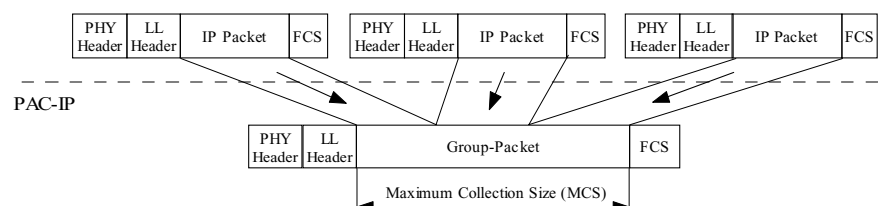
grouping packets by their next destination address (next hop) provides relevant advantages: it makes concatenation useful not only for the source node but also for nodes where traffic aggregation is performed, e.g. when the base station delivers packets from different sources of wired network to the same wireless node.

Furthermore, IEEE 802.11 MAC, through the specification of the medium contention algorithms, introduces a packet-based fairness among the nodes of the system: each node has roughly the same opportunity in medium access for a single packet transmission. However, there are no considerations related to the size of the packet. PAC-IP improves fairness of the system by providing equal opportunities for transmission of the same amount of data (in bytes).

3.2 PAC-IP implementation details

PAC-IP implementation requires the introduction of a software module inside the protocol stack. This module can be a part of the stack implemented in the operating system. Such implementation does not require any modification to wireless devices currently available on the market. On the other hand, PAC-IP module can be implemented inside a network interface driver or be a part of the network card's firmware, the latter enabling release of computational resources of the CPU of the node.

The main component of PAC-IP module is Packet Concatenator, which scans packets coming from the IP layer, identifying those traveling to the same destination or the same next hop router. This is accomplished by the analysis of the IP addresses and corresponding MAC addresses through the ARP look-up table.

For concatenation, upon IP packet arrival the Concatenator initializes a timer with the Maximum Concatenation Time (MCT) value while putting this packet into the buffer. If timeout occurs, the concatenation process is immediately stopped, and the collected *group-packet* is forwarded to the link layer for transmission. The limitation of the concatenation time is required to provide an upper bound to the delay which can be introduced by the concatenation algorithm.

PAC-IP concatenation is limited not only in time by MCT, but also in size by the Maximum Collection Size (MCS) parameter. MCS is bounded to the Maximum Transmission Unit (MTU), which is equal to 1500 bytes for the most common network interface (Ethernet). Such limitation avoids performance drawbacks deriving from fragmentation at the link level.

Another important point of PAC-IP implementation is that a concatenation process requires data copy. The first incoming IP packet becomes a *group-packet* and its data buffer is extended to MCS. Then, next incoming IP packet is concatenated to the *group-packet* using simple data copy process, avoiding the requirement for an additional memory realloca-
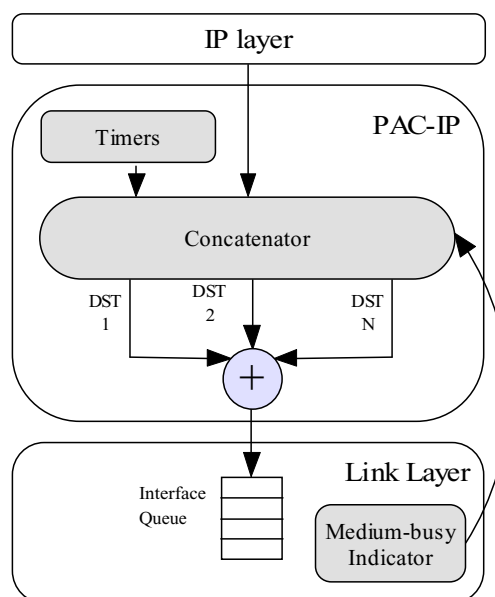

Fig. 4 PAC-IP software module and its position in the protocol stack

tion. The described technique is well suited with sk_buff representation of packets within the open protocol stack model available in Linux operating system.

In Section 2 it was outlined that transmission of bigger packets leads to better performance from the point of view of overhead reduction, which is the main aim of PAC-IP. However, wireless channels suffer from problems such as collisions, hidden nodes, signal interference and so on. When the channel error rate is high, it is reasonable to reduce the size of the packets in order to achieve lower data loss probability—especially when RTS/CTS exchange is not used before data transmission. Experimental studies show that a trade-off can be achieved between physical overhead reduction deriving from enlarging the packet and frame error rate [11]. As a result, MCS can be dynamically adjusted to the optimal value by the algorithm proposed in [11], thus avoiding performance sensitivity to varying channel error rates.

PAC-IP is designed for operation on relatively slow channels (like IEEE 802.11), where the wireless medium is contended by many stations and where single packet transmission takes a relevant time (at least 200 μs for IEEE 802.11b from the considerations described in Section 2). This enables further optimization through the reduction of unnecessary delay, i.e. to produce concatenation when the medium is busy and no data can be transmitted anyway.

For this purpose, PAC-IP module receives a state of the wireless medium from the Medium-busy Indicator (MI) operating at the link layer. MI is a passive element which provides a binary (medium busy/idle) feedback. The medium is considered to be busy in any of the three following cases:

- *Pending transmission.* This includes medium contention, RTS/CTS exchange, packet transmission and link layer ACK reception phases.
- *Another node transmission.* According to IEEE 802.11 standard a node can determine if the medium is occupied by another node physically by carrier sense (CSMA) or virtually through the Network Allocation Vector (NAV) carried by every frame in the MAC header.
- *Interface queue is not empty.* The availability of packets waiting for the transmission means the *group-packet* which is currently being concatenated in the Concatenator can be scheduled for transmission only when interface queue will become empty—which is equivalent to the medium-busy case.

Relying on the medium-busy feedback presented above, PAC-IP does not output a *group-packet* upon MCT timeout occurrence until medium becomes free. Nevertheless, it does output the *group-packet* in case MCS is reached. Consequently, MI feedback allows to exploit the medium-busy time for concatenation.

4 Quality of service (QoS) extension

The core idea of all methods for data concatenation is the same—to concatenate the data (byte-stream or packets) into units of bigger size. This is exactly the opposite with respect to the fragmentation idea implemented in the majority of network interfaces—to divide (fragment) the whole data unit into several packets of fixed size.

Concatenation and fragmentation concepts are both designed to adjust the outgoing traffic to network data units of a given size. Such adjustment represents a relevant issue, which helps to control performance in wireless networks.

Most concatenation techniques introduce additional delay in packet delivery—the time required for the packet concatenation procedure. This means that, in case of Maximum Collection Time (MCT) timeout expiration, the first packet of the collection will have the delivery delay increased by the MCT value, while other packets of the collection will have an additional delay between 0 (if a packet was added exactly before the timeout occurrence) and the MCT value. Such situation is not suitable for a variety of applications which require guarantees on the delivery delay. In order to satisfy delay requirements, we think that Quality of Service (QoS) could be directly enforced by concatenation algorithms.

Figure 5 presents a QoS concatenation module which uses two classes for the incoming packets differentiating them according to the delay requirements. The packets generated by non-delay sensitive applications follow "Concatenate" class which produces concatenation for up to MCT expiration (including details outlined in Section 3) and then un-
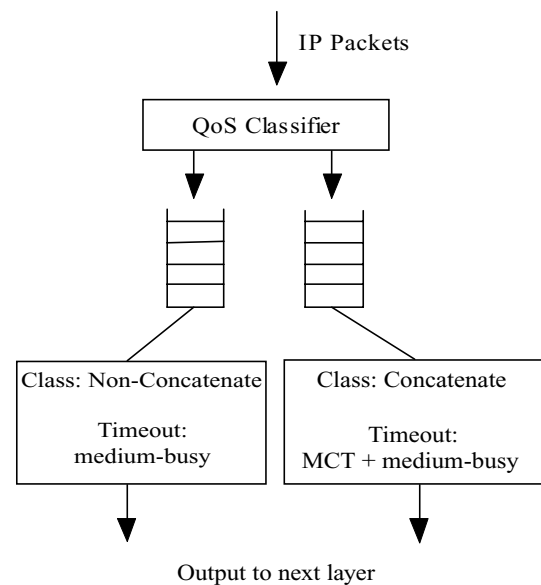

Fig. 5 QoS-enabling module for packet concatenation schemes

til the medium becomes available for the transmission. On the other hand, delay-sensitive traffic is classified as "Non-Concatenate" class. The packets belonging to this class are not concatenated and immediately ready for transmission. However, in case medium-busy indication is received, PAC-IP tries to concatenate packets waiting in the queue until transmission becomes possible, in order to provide performance enhancement by exploiting available (and known) waiting time for packets in the transmission queue.

Moreover, traffic differentiation performed by the QoS Classifier is based not only on the application-specific delay requirements, but also on protocol performance issues. As an example, QoS-enabled PAC-IP will specify a "Non-Concatenate" class for such TCP packets which could be considered as control packets. TCP protocol, in fact, starts data exchange with connection establishment, where no data are usually transmitted but connection-related information between end-to-end communicating stations. In this phase, TCP produces several standalone packets of small size which should be delivered as fast as possible in order to let the data flow begin, and therefore they should be classified in the "Non-Concatenate" class. In order to support such functionality, however, PAC-IP should have an access to TCP header information (especially flags SYN, FIN, ACK, etc.).

5 Performance evaluation

The performance of the proposed solution is analyzed by simulations with the ns-2 (version 2.29) network simulator [12] as well as by measurements on a real IEEE 802.11 WLAN test-bed. The results are presented for wireless single-hop and infrastructure networks scenarios.

5.1 Single-hop scenario

(a) *Simulation results*. The simulation scenario contains two static nodes located within the transmission range of each other. A TCP source is attached to one of them, while TCP sink is running on the other one. TCP source continuously sends data limited by Maximum Segment Size (MSS) variable within the experiment.

In order to have simulation results comparable with those achieved on the test-bed, IEEE 802.11b was chosen as the physical layer standard. The simulation parameters are summarized in Table 3.

The main reason for choosing TCP for performance evaluation is due to the availability of flow control and congestion avoidance mechanisms, which adapt the throughput level to the available capacity. Consequently, in order to avoid performance drawbacks derived from the multiplicative decrease phase when TCP outgoing rate is reduced by its half, the outgoing buffer is set greater than bandwidth delay product of the communication path. As a result, the buffer of such size ensures full capacity utilization even in cases when TCP outgoing rate is less than the physical path capacity.

Drawing on the considerations presented in the introductory section of the paper, we decided to compare the proposed PAC-IP with the standard TCP/IP on IEEE 802.11b [3].

Figure 6 presents the throughput comparison of PAC-IP against IEEE 802.11 standard. The maximum collection size is set to 1500 bytes (most common MTU in wired LANs) leading to the concatenation of packets smaller than 750 bytes. In simulations, TCP source is in saturation state (i.e. it has always a packet to send).

Table 3 Simulation parameters

Parameter name	Value
Slot	20 μs
SIFS	10 μs
DIFS	50 μs
PLCP preamble + header	192 μs
Data rate	11 Mb/s
Basic data rate	1 Mb/s
Propagation model	two-ray ground

In the concatenation area (left part of the graph), the resulting throughput level is close to the highest throughput achieved by standard IEEE 802.11 (i.e. when 1500-bytes datagrams are transmitted).

(b) *Test-bed experiments*. For the purpose of further investigation of PAC-IP behavior in real scenarios, we designed a simple IEEE 802.11b test-bed which consists of two laptop computers located within transmission range. Both computers, equipped with wireless IEEE 802.11b Orinoco Silver cards, are running Redhat Linux 9.0 (kernel version 2.4.20-8). In order to support PAC-IP functionality, Orinoco_cs wireless driver version 0.13 d was modified. TCP throughput measurements are obtained using Iperf (version 1.7.0) [13] performance measurement tool in all the experiments.

Both laptops stationary located within wireless cards transmission range are connected through the "ad-hoc" connection mode of IEEE 802.11. Therefore, before TCP connection can be established, a route discovery phase should be passed—which includes address resolution in order to find link layer addresses which correspond to IP addresses on the nodes.

Fig. 6 PAC-IP throughput (simulations and test-bed) vs IEEE 802.11
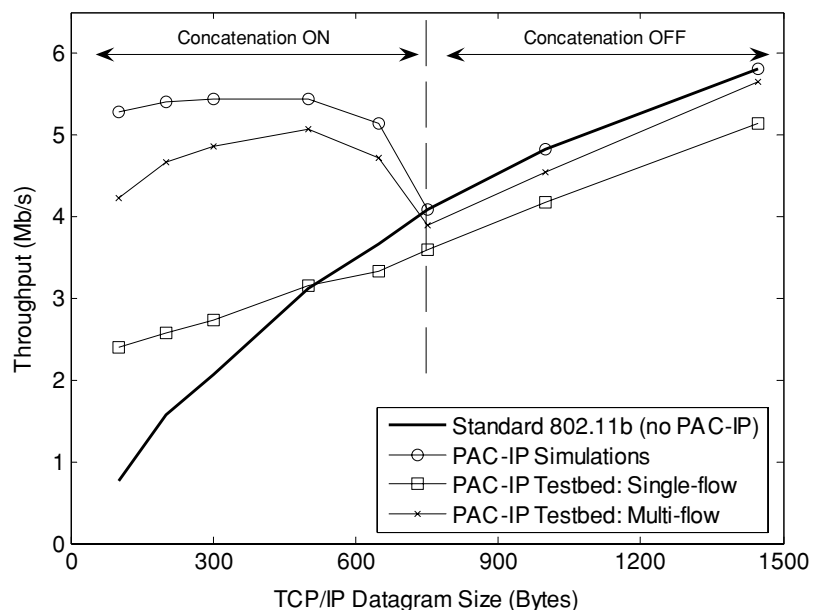
Moreover, according to TCP specification [3], before any data flow can take place, a TCP connection should be established and then the connection should be terminated when data delivery is concluded. Connection establishment and termination phases contain an exchange of control packets which do not transport any data from the application. Moreover, the slow start phase begins data delivery during which the TCP window is increasing.

In order to avoid the influence of the mentioned phases on the performance bounds of PAC-IP, the measurement phase was chosen to be within the data delivery time interval when TCP flow is in steady state. TCP data packet size was fixed within a single flow, while NO_DELAY option was turned on in order to avoid data concatenation by Nagle algorithm [8]—a transport layer solution described in Section 3 of this paper.

Figure 6 presents measurement results obtained with a single TCP flow connection. It shows a relevant difference with the results obtained from simulations: the TCP flow with small data packet sizes (left part of the curve) experiences lower throughput than expected. The reason is that, in case of simulation, it was assumed that TCP could fulfill the entire collection during concatenation, while practical implementations of TCP can output only several packets (depending on the congestion window size) and after that they wait for an acknowledgement from the receiver. However, even the reduced size of the collected packet brings a significant throughput improvement comparing with legacy TCP/IP over IEEE 802.11.

Furthermore, the implementation of PAC-IP requires data copy for concatenation of the incoming packet. In order to evaluate the influence of data copy process to the delivery time, in our implementation all the packets (even those which size is more than a half of MCS) are copied before releasing them to the link layer. As a result, the time required for the data copy process reduces the throughput in case of large packets (right part of the graph) of a single TCP data flow. In order to reduce the delay introduced by data copy (releasing the resources of the host), PAC-IP should be implemented inside the firmware of the wireless device, exploiting efficient copy facilities such as DMA (Direct Memory Access).

In multi-flow scenario, the client node opens multiple TCP connections with the server node. The total bandwidth is then divided among TCP flows. For that reason, the resulting throughput curve in Fig. 6 represents cumulative throughput achieved by all the flows. If compared with single-flow scenario, the throughput achieved by multi-flow connection closely approximates simulation results, thus confirming the relevant level of improvement enabled by PAC-IP even in a real WLAN scenario.
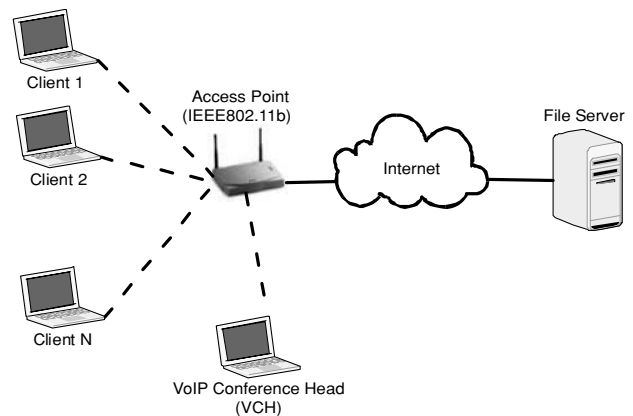

Fig. 7 Infrastructure WLAN scenario

5.2 Infrastructure WLAN scenario

The results obtained in the single-hop scenario presented in previous section show good performance improvement enabled by PAC-IP concatenation. However, most of wireless networks deployed nowadays are more complex than a single-hop connection. For this reason, PAC-IP performance is evaluated also in the most widely used WLAN scenario, which consists of an IEEE 802.11b cell composed of multiple client nodes and a single access point having a wired interface to Internet connection.

In Fig. 7 one VoIP Conference Head (VCH) and N client nodes are connected to the Access Point (AP) running IEEE 802.11b protocol. There are N VoIP bidirectional calls carried between client nodes and VCH. Each client node carries an uplink VoIP flow to VCH, and VCH carries N downlink flows for N stations. This comes from the fact that VCH provides VoIP conference multiplexing function. Both uplink and downlink flows are performed through the AP.

For each call, we use the ITU G711 64 Kb/s codec [14]. The 40 bytes RTP/UDP/IP Layer header is then added to each VoIP packet. At the MAC layer, IEEE 802.11 DCF basic access mode without RTS/CTS is used, while physical layer parameters are fully consistent with ones presented in Table 3.

In this scenario, there are $N + 1$ bidirectional VoIP flows constantly present in the system which consume a constant portion of system's capacity. For that reason, we evaluate the rest of the capacity left unoccupied. In order to do so, along with taking a part in VoIP conference, each client node downloads a file from the File Server located in the Internet. As a result, there are N TCP connections flowing from the File Server to the client nodes. In order to emulate Internet connection the File Server is connected with AP using 20 Mb/s, 40 ms delay link running Ethernet protocol at the link layer.
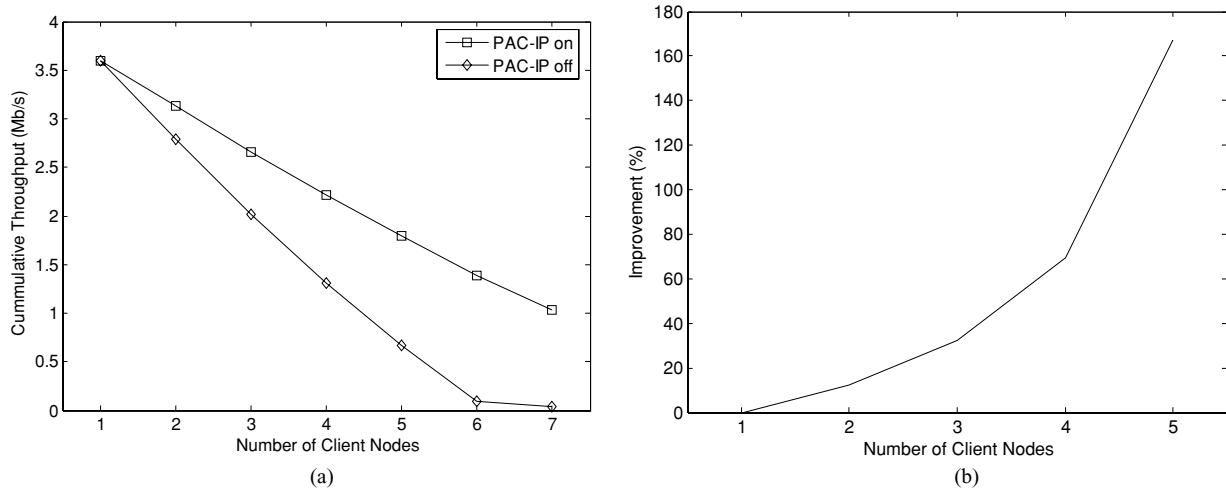
Fig. 8 Infrastructure WLAN performance vs. number of client nodes (with no link errors): (a) throughput of data (TCP) connections and (b) the corresponding improvement level

In the scenario presented in Fig. 7, PAC-IP functionality is enabled in all wireless devices (client nodes, AP and VCH). However, the configuration of $MCT = 20$ ms and $MCS = 1500$ bytes leads to concatenation of VoIP packets traversing the link between AP and VCH, i.e. the concatenation affects only VoIP packets received by AP from client nodes and VoIP packets generated by VCH for client nodes.

In addition, in order to minimize the delay in voice packet delivery VoIP packets are classified into "Non-concatenate" class by the QoS module. As a result, the concatenation of the voice packets is performed in case the outgoing queue is not empty (as described in Section 4). However, in the presented scenario most of the time interface queues contain TCP packets flowing over the network.

Figure 8(a) shows the cumulative throughput obtained by TCP connections, which corresponds to the network capacity left unoccupied by VoIP flows. The measure is equal for the case with a single client node in the system since no VoIP packet can be concatenated within 20 ms interval. The difference grows with the increase of the number of client nodes—where PAC-IP concatenation reduces VoIP capacity demands. In this experiment, the curve of the achieved improvement level (see Fig. 8(b)) grows rapidly hitting 160% for the number of client nodes equal to 5.

Another important point is the performance in error-prone environment. PAC-IP concatenates small packets building a large *group-packet* which becomes more sensitive to link errors due to its increased size. Indeed, the probability of losing a large packet is much greater than in case of small packets. This fact is discussed in details in [15] and [16] for wireless and optical communications, respectively. More in details, the authors of [15] designed a tight combination of MAC-layer FEC with retransmission technique, while the burst assembly proposal described in [16] has conceptual similarity with the proposed approach. However, the difference with the proposed PAC-IP scheme lies in the extremely large size of the assembled burst with respect to the IP packet size as well as in the support of differential services in optical networks. As a result, in [16] packet loss triggers multiple retransmissions at the link layer, thus consuming the available bandwidth.

In order to evaluate PAC-IP performance in an error prone environment, experiments are run by varying Packet Error Rate (PER) (see Fig. 9(a)) and Bit Error Rate (BER) (see Fig. 9(b)) on the wireless link. Bit errors are chosen to be uniformly distributed (which corresponds to the worst case—causing higher PER/BER ratio). The results (presented for four client nodes) show the performance level of PAC-IP is unaffected when error rate is low (PER $= 10^{-9} \div 10^{-4}$ or BER $= 10^{-9} \div 10^{-6}$). In case of higher error rates (PER $= 10^{-3} \div 10^{-1}$ or BER $= 10^{-5} \div 10^{-2}$), the performance of PAC-IP is slightly decreased as expected. However, the performance (throughput) in presence of PAC-IP is never lower than the case corresponding to no concatenation.

Furthermore, it is important to observe that the different performance of PAC-IP in the two figures (9(a) and (b)) derives from the fact that the same BER causes higher PER in the "PAC-IP ON" scenario due to the higher size of the *group-packets*.

Finally, an additional point to be addressed is PAC-IP implementation complexity. In infrastructure WLAN scenario there is a low-cost requirement which makes an AP a simple device implementing only the physical and link layers of the protocol stack. However, PAC-IP implementation does not require full IP support at the AP. For packet concatenation, it requires access to a single network header field ("Total length") which can be simply read with an offset from the beginning of the IP headers.
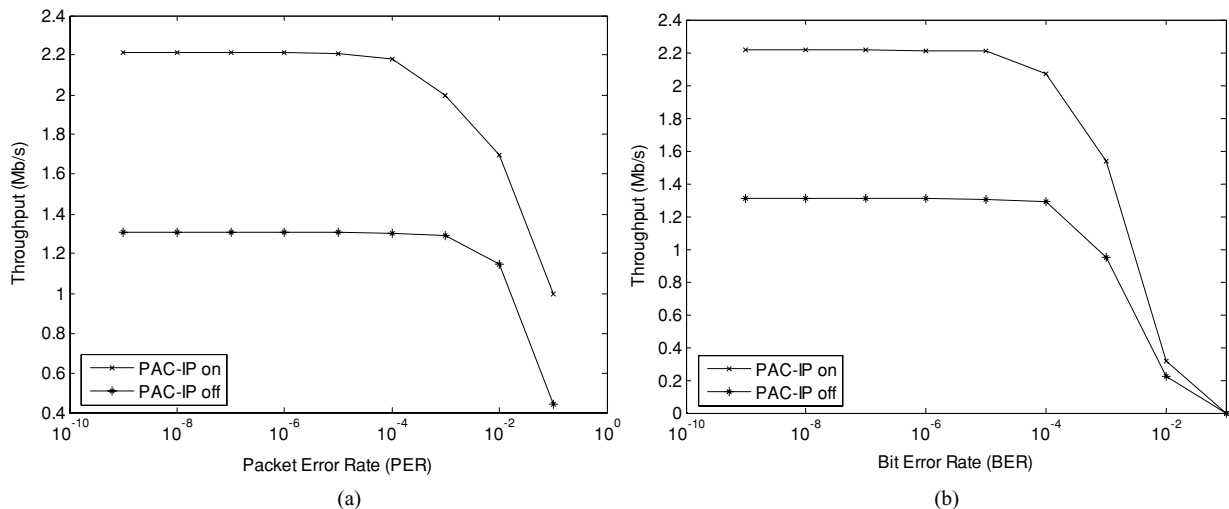
Fig. 9 PAC-IP performance in error-prone environment: (a) against Packet Error Rate (PER) and (b) against Bit Error Rate (BER)

In case of prioritized treatment of TCP connection open/close packets the QoS-enabled PAC-IP additionally requires an access to some transport layer flags (SYN, FIN, ACK). However, similar to other approaches which require an access to TCP header, this technique will not work with end-to-end encrypted traffic such as IPSec [19].

As a result, the most expensive process involved in data concatenation from the complexity point of view is data copy. However, this process can be speeded up with the use of appropriately designed dynamic arrays or linked lists data structures which copy only pointer to packet data but not the data itself.

Summarizing, PAC-IP concatenation implemented even in a single WLAN cell increases network capacity for the benefit of all the users, with the only drawbacks of requiring some additional computational power on the wireless devices (mainly for implementing data copy) and access to some network/transport layer header information (IP Total Length field, TCP flags).

6 Conclusions

The paper highlights the problems related to performance of wireless LANs, due to presence of small packets as well as to the overhead introduced at the link and physical layers of IEEE 802.11. After a classification of existing approaches, packet concatenation at the IP level (PAC-IP) is proposed as a possible solution designed to improve throughput performance, especially in case of transmission of small packets over the wireless link. The improvement of the packet-based fairness between nodes is also discussed. Evaluation of PAC-IP is performed through both simulations as well as on an experimental IEEE 802.11b-enabled test-bed in single-hop and infrastructure network scenarios. The results show good agreement with PAC-IP design principles.

Ongoing activities deal with performance evaluation of the implementation of the QoS-enabling module within PAC-IP, through simulation and experiments using the test-bed presented in the paper.

In addition, even if PAC-IP is presented as a solution for WLANs, it is worthwhile to underline that the basic paradigm can be extended and employed on any network where relevant overhead is introduced at the link and/or physical layers.

References

1. C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely and S.C. Diot, Packet-level traffic measurements from the sprint IP backbone, IEEE Network 17(6) (2003) 6–16.
2. K. Thompson, G.J. Miller and R. Wilder, Wide-area internet traffic patterns and characteristics (Extended Version), IEEE Network 11(6) (1997) 10–23.
3. J. Postel, Transmission control protocol, RFC 793 (September 1981).
4. Wireless LAN medium access control (MAC) and physical layer (PHY) Specifications, IEEE 802.11 standard (1997).
5. J. Tourrilhes, Packet frame grouping: Improving IP multimedia performance over CSMA/CA, *IEEE International Conference on Universal Personal Communications* Florence, Italy (1998).
6. Official IEEE 802.11 Working Group Project Timelines, http://grouper.ieee.org/groups/802/11/ 802.11_Timelines.htm
7. The Norwegian academic and research data network, http://www.uninett.no/wlan/throughput.html
8. J. Nagle, Congestion control in IP/TCP internetworks, RFC 896 (1984).
9. J. Mogul and G. Minshall, Rethinking the TCP Nagle algorithm, Computer Communication Review 31 (2001) 6–20.
10. K. Yeung, 802.11a modeling and MAC enhancements for high speed rate adaptive networks, Technical Report UCLA (2002).
11. P. Lettieri and M. Srivastava, Adaptive frame length control for improving wireless link throughput, range, and energy efficiency, *Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies*, Vol. 2, (March/April 1998) pp. 564–571.

12. NS-2 simulator tool home page. http://www.isi.edu/nsnam/ns/ (2000).
13. Iperf performance measurement tool, http://dast.nlanr.net/Projects/Iperf
14. F. Ohrtman, Voice over 802.11, Artech Hose (2004).
15. S. Choi, Y. Choi and I. Lee, IEEE 802.11 MAC-Level FEC with re-transmission combining, IEEE Trans. on Wireless Communication 5(1) (2006) 203–211.
16. V.M. Vokkarane, Q. Zhang, J.P. Jue and B. Chen, Generalized burst assembly and scheduling techniques for QoS support in optical burst-switched networks, in: *IEEE Global Telecommunications Conference* (November 2002).
17. J. Nagle, *Personal Communication* (February 1999).
18. G. Minshall, Y. Saito, J. Mogul and B. Verghese, Application performance pitfalls and TCP's Nagle algorithm, in: *Second Workshop on Internet Server Performance* Atlanta, GA (1999).
19. S. Kent and R. Atkinson, Security acrchitecture for the Internet protocol, IETF, RFC 2401 (1998).

Dzmitry Kliazovich received his Masters degree in Telecommunication science from Belarusian State University of Informatics and Radioelectronics in 2002. He is currently working towards the Ph.D. degree in University of Trento, Italy. From September 2005 to February 2006 he was a visiting researcher at the Computer Science Department of the University of California at Los Angeles. He is an author of more than 20 research papers published in international books, journals and conference proceedings. His main research interest lies in field of wireless networking with a focus on performance optimization and cross-layer design.

Fabrizio Granelli was born in Genoa in 1972. He received the "Laurea" (M.Sc.) 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 Ph.D. in Telecommunications from the same university, in 2001. Since 2000 he is carrying on his teaching activity as Assistant Professor in Telecommunications at the Dept. of Information and Communication Technology—University of Trento (Italy). In August 2004, he was visiting professor at the State University of Campinas (Brasil). He is author or co-author of more than 60 papers published in international journals, books and conferences, and he is member of the Technical Committee of the International Conference on Communications (from 2003 to 2007) and Global Telecommunications Conference (GLOBECOM2003 and GLOBECOM2004). Dr. Granelli is 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 IEEE Workshop on Computer-Aided Modeling, Analysis, and Design of Communication Links and Networks (CAMAD'06). His main research activities are in the field of networking and signal processing, with particular reference to network performance modeling, medium access control, wireless networks, cognitive radio systems, and video transmission over packet networks. He is Senior Member of IEEE and Associate Editor of IEEE Communications Letters.