Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers and Electrical Engineering 32 (2006) 241–265

Computers and Electrical Engineering

www.elsevier.com/locate/compeleceng

Study and performance analysis of transport layer mechanisms applied in military radio environment

Tomaso de Cola [a,*], Mario Marchese [b,1]

[a] CNIT – Italian National Consortium for Telecommunications, University of Genoa Research Unit, Genoa, Italy
[b] DIST – Department of Communications, Computer and System Science, University of Genoa, Genoa, Italy

Abstract

The need for reliable data communication performed in critical conditions and the respect of real-time constraints is an open issue in radio-military networks, where fixed and nomadic terminals communicate by means of tactical radio infrastructures. In this perspective the need for real-time constraints plays a fundamental role when the interested applications are remote control, radar traces acquisition and command/control message exchanging. In this paper, the difficulties of data transportation in wireless networks are considered by taking emphasis on the TCP-NewReno implementation that does not guarantee satisfying performance results in such environments. The whole investigation is addressed at designing and validating novel transmission mechanisms to be implemented at the transport layer, directly inherited from standard TCP transmission scheme. TCP-radio and TCP-radio-newRTO proposals are thus discussed and compared with TCP-Westwood+ and TCP-NewReno in order to highlight the necessity for an enhanced transport protocol for data communication performed in radio military environments, where the data transfer is heavily affected by hazardous conditions, such as scattering, shadowing and slow fading.

All the tests have been performed through the employment of a network emulator tuned to the radio characteristics in terms of end-to-end delay, available bandwidth and channel modelling.

Keywords: Military environment; Transport layer; TCP-IP; Tactical networks; Radio channel modelling

1. Introduction

Military communications are characterized by peculiar applications, as data retrieval from sensors, weapon control and sensor control, which require, on one hand, data transfer of high reliability but, on the other hand, have the need to use standard components (COTS – Components Off the Shelf), in particular concerning software interfaces.

* Corresponding author. Tel.: +39 010 3532806; fax: +39 010 3532154.
E-mail addresses: tomaso.decola@cnit.it (T. de Cola), mario.marchese@unige.it (M. Marchese).
[1] Tel.: +39 010 3532806; fax: +39 010 3532154.

0045-7906/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compeleceng.2006.01.018

From the network viewpoint, military infrastructures are often characterized by heterogeneity, because they include radio, satellite and cable portions, each of them deserving a special attention [1]. Tactical networks, which have a topical operative role, are mainly based on radio support. As a consequence the applications mentioned above need to be transported over tactical environment, possibly using standard software interfaces and getting a fully reliable service. A suitable choice is to use the TCP/IP suite and, in more detail, due reliability needs, the Transport Control Protocol (TCP) [2].

TCP, even if it still works, is not efficient over channels characterized either by large delay-bandwidth product or by losses due to channel errors. The large product makes the acknowledgement scheme on which TCP is based, very inefficient. That is the main characteristics of geostationary satellite links. On the other hand, TCP considers any loss as a congestion event and decreases the source bit rate entering the network. If a loss is due to a channel event (e.g. rain fading), reducing the rate does not bring any advantage. That is the typical case of radio networks.

Tactical networks are often affected both by large delay-bandwidth product and by channel errors. A reliable transport layer should be designed with great attention.

The basic idea of the paper is taking TCP-NewReno as reference, keeping standard interfaces towards application and IP layers, and proposing some simple modifications to make the transport protocol suited for tactical environment. Simplicity of implementation, together with efficiency [3], is part of the aim of the overall design. Actually, the schemes proposed are very simple to implement but guarantee a high degree of flexibility. Simplicity is allowed by one of the few advantages of tactical networks: they are normally managed by just one carrier and all the characteristics (e.g., available bandwidth and number of users connected) of the network are known so that the TCP (or, better, the modified TCP) may use all these information to set its parameters. This hypothesis is known in the literature with the name of "Complete Knowledge" approach, which has been introduced in [4] for satellite communication.

Many efforts [5] have been provided to mitigate the problems of TCP over wireless and the scientific community is still working about this topic [6].

In particular, [7] classifies the possible solutions to be applied in such environment in five main categories, namely: Pure Transport Layer, Hard State Transport Layer, Soft State Transport Layer Caching, Soft State Cross Layer Signalling and Pure Link Layer. The first category operates only on an end-to-end basis by modifying the transport layer implementations and without extra logic or change in intermediate nodes as based stations or radio access points. In this perspective, [8] allows a more effective recovery mechanism by exploiting the advantages provided by a selective repeat scheme; TCP Eifel [9] allows specifying a better timeout management that avoid spurious timeout and useless retransmissions. TCP-WestWood+ and TCP-Veno [10,11] employ a rate-based transmission algorithm inherited from the TCP-Vegas implementation and specify a more effective recovery mechanism able to estimate the available channel bandwidth without reducing the congestion window too drastically. TCP Migration and TCP-Freeze [12,13] allow, respectively, suspending and resuming the TCP session (Migration) and freezing the data communication by Zero Window Probe (Freeze). TCP Peach+ [14], even if designed for a specific satellite environment, implements a more effective recovery mechanism than the TCP-NewReno by exploiting the knowledge carried in "NIL Segments". It is able to assure satisfying results also when the channel state is very critical in terms of packet error rate. An alternative scheme tuned for wireless environments is A-TCP [15], which estimates the channel state by operating strong modifications on the TCP state machine.

Concerning the Hard-State Transport Layer, it includes all the schemes based on connection splitting as I-TCP and M-TCP [16,17], which allow a more effective management of the data communication over wireless links by opening a second connection at the transport layer in order to separate the wireless environment from the rest of the network. The main drawback of this approach is that the end-to-end semantic is often sacrificed and a heavy effort to maintaining a state of the whole communication is required on the intermediate nodes. In practice the idea is isolating the radio portion of the network by using Gateways that split the transport layer connection. PEPs (Performance Enhancing Proxies) are included in this class.

Regarding Soft-State Transport Layer Caching, a soft-state is maintained on the intermediate nodes or on the mobile hosts in order to perform local retransmissions [18] and to notify the loss of segments [19] to the terminal hosts. These schemes cannot be applied in a military framework, because they fail in presence of encryption mechanisms due to the necessary interpretation of the transport header.

Soft State Cross Layer Signalling consists of a coupled interaction between transport layer and underlying layers. TCP-F and TCP-BuS [20] exploit the knowledge of the channel state offered by the routing protocols and then by the network layer.

Finally Pure Link Layers includes all Forward Error Correction (FEC) and Automatic Retransmission reQuest (ARQ) schemes [21] and hybrids solutions. In particular, in radio-military environment, NATO STANAG 5066, 4538 and 4539 [22–24] are employed in over HF/VHF/UHF in order to decrease the frame loss error rate [25] by means of channel coding and local retransmissions to be performed at the datalink layer.

Alternative schemes belonging to the Pure Transport Layer approach are investigated in this paper. The schemes presented may be applied also within Hard State Transport Layers PEP structures relatively to the radio portion of the network. In particular, the modification of the recovery mechanism within TCP-NewReno has been considered together with the transmission window management in order to assure a reliable data communication and an effective resource use. The modifications proposed are very simple and operative. A FEC strategy implemented at the datalink layer is also adopted, in order to cope with the effective hazardous conditions typical of forest, rural and sea environments, where military applications are usually utilized. The proposed solutions have been evaluated with respect to the performance provided by TCP-NewReno [26] in different radio channel conditions and for various kinds of services directly related to military applications. In more detail, a file transfer has been considered for the tests. Its dimension ranges from a few kilobytes as in messaging applications, to 100 kbytes and up to 1 Mbytes, as in database access with web interface, radar images downloading, command and control operations. The performance analysis has been performed by using a network emulator [27], tuned to the radio environment characteristics and driving real traffic. The characteristics of actual military sites, got from available data sheets, have been used in the performance tests.

The paper is structured as follows. In Section 2, a short description about the network emulator employed during the tests is presented. The global scenario investigated in this paper together with the channel modelling employed in this work is presented in Section 3. Section 4 contains a short revision of TCP, while Section 5 is dedicated to the design and the analysis of the novel protocol solutions. Section 6 contains the performance analysis of the proposed transport protocol solutions by means of a comparison with TCP-NewReno and TCP-Westwood+. Conclusions are listed in section 7.

2. Network emulator

The reference architecture of the emulator is shown in Fig. 1, along with one possible system to be emulated enclosed in the cloud (a wireless system, where mobile users communicate through a base station, has been depicted in this case). Different units called Gateways (GTW) operate as interface among the emulator and the external PCs. Each GTW is composed of a PC with two network interfaces: one towards the external world (a 10/100 Mbps Ethernet card), and towards the emulator. An Elaboration Unit (EU), which has a powerful elaboration capacity, carries out most of the emulation, as the decisions about each PDU. The interface towards the external world concerns the GTWs; the loss, delay and any statistics of each PDU regards the EU; the real transport of the information PDU through the network concerns the input GTW and the output GTW. The various components are connected through a 100 Mbps network, completely isolated, by a full-duplex switch. The emulator has an available bandwidth much wider than the real system to be emulated, which should not overcome a maximum overall bandwidth of 10/20 Mbps (1 Mbps in this research). The architecture of the emulator is not exactly correspondent to the real system. Each terminal is divided, in the emulator, into two parts (GTW and EU). The network layer, the network interface towards the external world and the interface between network layer and radio modem are contained in the GTW. The other parts of the modem (i.e. the data link layer, protocol and encapsulation), the overall transmission characteristics (e.g. bit error ratio, channel fading, lost and delayed packets), as well as the queuing strategies are contained in the EU. In this way, the overall architecture is able to drive real-time traffic, with respect to the constraints imposed by the radio link, in terms of propagation delay, available bandwidth, and channel modelling.

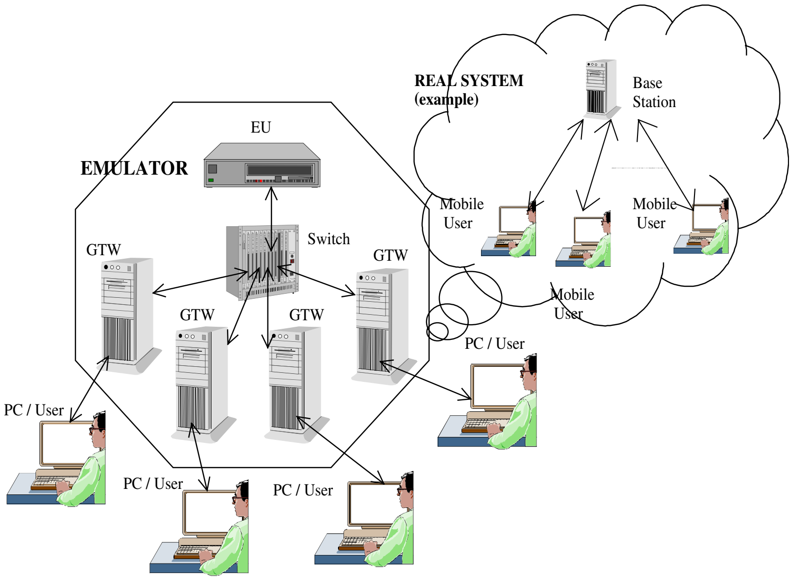

Fig. 1. Overall emulator architecture and real system.

3. General framework

3.1. Global scenario

The reference scenario is depicted in Fig. 2. One single tactical radio network with nomadic users exchanging data each other is considered.

The attention is focused on the transmission schemes implemented at transport layer to be applied in a military wireless network deployed in different environments (forest, rural and sea). Nomadic users, whose position is variable in time, communicate through radio links in the HF/VHF/UHF bands as in most military applications. Each single mobile terminal implements the TCP/IP suite. Mobile base stations act as access points for the hosts in the network and may be mounted on tanks and helicopters for safety reasons. The protocol stack is aimed at guaranteeing efficient support to applications as data retrieval from sensor control, which require a high reliable data transfer service. Applying the TCP/IP stack, the transport layer is in charge

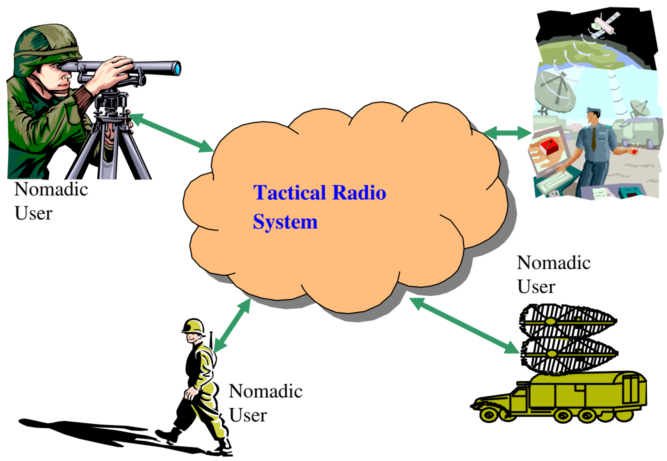

Fig. 2. Reference scenario.

for reliability. The base station, which may also implement a Performance Enhancing Proxy (PEP) so splitting the TCP connection, connects the Tactical Radio System with the rest of the military network, if the transmission is not within the tactical network. The work contained in this paper concerns only the tactical portion of the network. Should the data transfer have a destination outside the tactical portion, the transport layer connection is supposed to be limited to the PEP gateway (implemented in the base station).

The overall technical scenario is depicted in Fig. 3. The portion of the network under study is shown in light grey. If, on one hand, TCP is very inefficient over the tactical radio bearer, on the other hand, the limited dedicated scope allows assuring the full network knowledge and control. Consequently, the transport layer (the TCP or, better, the suitable modified version) may be really adapted to the network characteristics. Many studies in the literature [28] are dedicated to investigate mechanisms to find out if a loss is due either to congestion or to channel errors. Other researches improve the TCP performance having in mind QoS (Quality of Service) management over the Internet, as, for instance, Differentiated and Integrated Services [29–31]. The algorithms considered in this work do not restrict the scope of application (actually, they may applied together with a PEP architecture) and allows implementing simple strategies working at the transport layer.

It is straightforward that all the improvements operated on the TCP implementation are such to maintain the socket API semantic and are not visible to the application side.

Some more words are needed to take emphasis on the functionalities implemented in the underlying layers: the Internet Protocol (IP) operates at the network layer. The data link layer is supposed to perform Forward Error Control (FEC) or Automatic Retransmission reQuest (ARQ) operations. This requirement is justified by the fact that most military radio systems exhibit high bit error rates (up to 10^{-3} and possibly even higher) because of temporary outages due to terrain blocking of signals and interference of various nature. As a consequence the reliability of the communication cannot be guaranteed by the only employment of the TCP protocol but it is necessary to implement mechanisms at the lower layers, which assure a more robust data

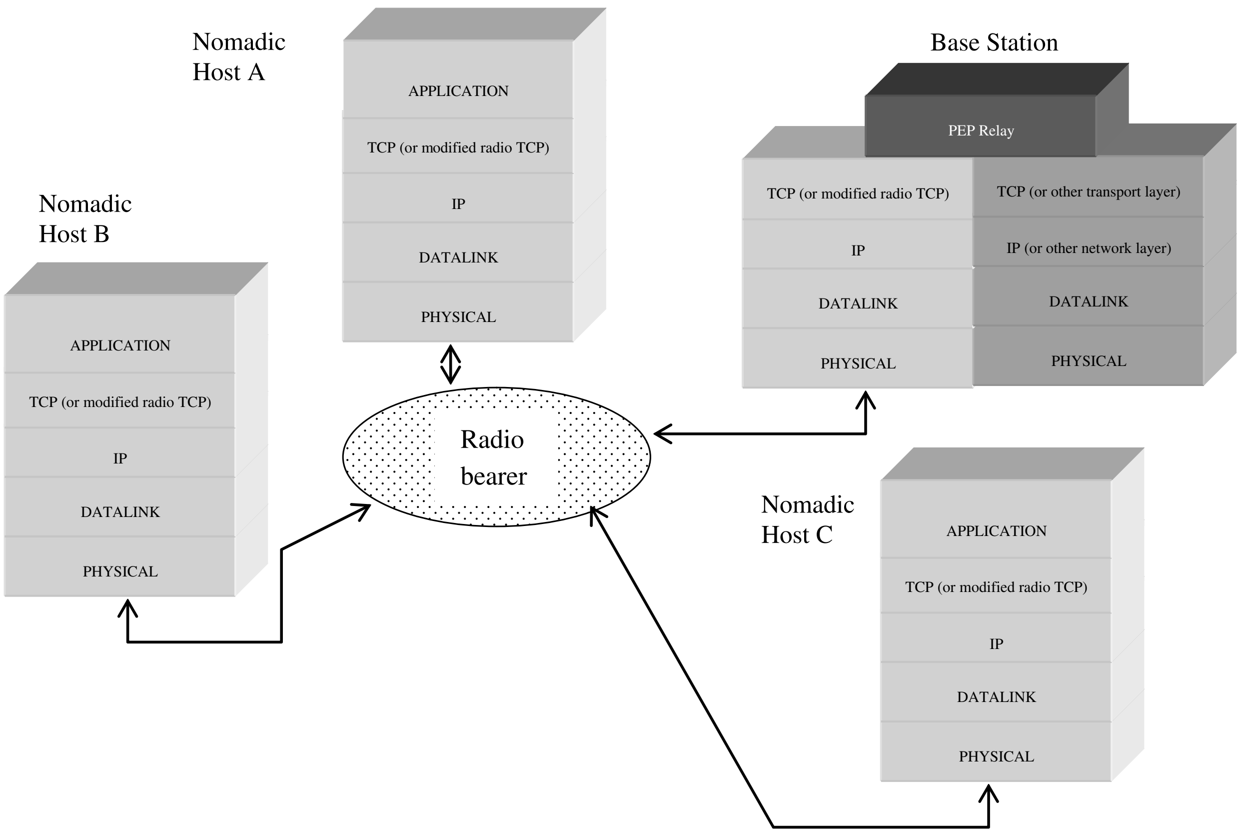

Fig. 3. Technical scenario.

communication. In particular, typical HF/VHF/UHF protocols like CLOVER-2000 [32], FED-STD-1052 [33] and NATO STANAG 5066 [22], which offer efficient performance when applied to a wireless environment are strongly recommended. Studies to investigate the effect of ARQ on the TCP performance are reported in the literature. For the sake of simplicity, but without loss of generality, a FEC mechanism operating with a block coding characterized by a couple (n, k) and code rate R_c is considered in this paper. The datalink layer entity is responsible of fragmenting the PDU coming from the upper layers into blocks of data, containing k symbols of information and $(n–k)$ of redundancy. In particular, in order to mitigate the effect of correlated bursty errors due to the communication over wireless channel, a RS (255, 223) code is employed together with an interleaving scheme. The aforementioned choice has been determined by its high corrective capacity and by its code rate, which determines an effective bandwidth value seen by the upper layer equal to

$$R_c \cdot B = 223/255 \cdot B \cong 0.81B \tag{1}$$

The physical radio bearer is modelled having in mind the HF/VHF/UHF environment. The model used to describe the physical and the data link layer is detailed below.

3.2. Wireless channel model

Data communication performed in VHF/UHF radio links is affected by diffraction and shadowing due, for example, to obstacles present in forests and rural environments, trees for example. On the other hand also the quality of data communication performed in HF band is strongly deteriorated by the multiple reflections of radio signals that may occur between the earth's surface and the ionosphere, giving rise to multihop propagation.

The employment of the i.i.d (where the errors are considered independent and identically distributed) model, which is accurate in the case of ideal interleaving assumption and fast fading has been widely considered in the literature [34] for wireless environments (mostly satellite). Unfortunately, the environment considered in this paper neither allows the implementation of ideal interleavers, which are very burdensome and imply extra latencies, nor can be limited by fast fading conditions because it is aimed at considering a general outdoor scenario where nomadic users communicate. In general, wireless channels are described by means of short term and large range fluctuations [35]. The former is due to the presence of scatterers close to the source; this situation is usually statistically represented by a Rayleigh distribution. The latter is described in the literature in terms of shadowing and characterized by an exponential autocorrelation function [36]. Models based on Finite State Markov Chains (FSMC) [37,38] have been considered to take into account the main peculiarities of HF subsystems.

Given these conditions, together with the application of the Reed-Solomon (255, 223) code, a good representation of the overall system is provided by a Two State Markov Chain (TSMC) [39]. The state 0, "good state", where the transmission is successful. The State 1 is called "bad state" in order to highlight the impossibility to have a successful transmission. In more detail, when a communication is performed during the "bad state", all the blocks transmitted will be corrupted by errors, without possibility of recovery by FEC capabilities. The behaviour of TSMC is thus described by the following transition probability matrix:

$$\Pi = \begin{bmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{bmatrix} \tag{2}$$

The TSMC steady state is described by the probabilities π_0 and π_1 [39]

$$\begin{aligned} \pi_0 &= \frac{\alpha}{\alpha+\beta}; \\ \pi_1 &= \frac{\beta}{\alpha+\beta} \end{aligned} \tag{3}$$

As a consequence, the probability to have a lost packet in the steady state is

$$P_{\text{pkt\_lost}} = P(\text{pkt\_lost}/\text{state}=0) \cdot P(\text{state}=0) + P(\text{pkt\_lost}/\text{state}=1) \cdot P(\text{state}=1) = \pi_1 \tag{4}$$

Different values of α and β have been considered during each test in order to analyse different channel conditions and their impact on the performance of data communication.

4. TCP

The Transmission Control Protocol (TCP) is a connection-oriented, end-to-end reliable transport protocol working between hosts in packet-switched networks, and between interconnected systems of such networks. The motivations, the philosophy and the functional specification of the protocol are contained in RFC 793 [2]. Some of the material contained in it is summarized in the following to identify the aim and the scope of the TCP. Most of this sub-section is taken from RFC 2582 [26], which describes the algorithms ruling the TCP behaviour in presence of congestion and highlights the main differences in congestion control between TCP-Reno and TCP-NewReno. It specifies four algorithms: slow start, congestion avoidance, fast retransmit and fast recovery. The definitions contained in Table 1 have been used throughout the subsection and may help understand the TCP transmission mechanism.

A segment is considered lost either after the special timer (Retransmission Timeout – RTO) expires, or after three duplicated acknowledgements (four ACKs indicating the same sequence number) as explained in the following.

The slow start and congestion avoidance algorithms are used by a TCP sender to control the amount of outstanding data being injected into the network. The minimum between "cwnd" and "the minimum between the source buffer and rwnd" governs data transmission (the variable TW identifies the real transmission window). Another state variable, the slow start threshold (ssthresh), is used to determine whether the slow start or the congestion avoidance algorithm is used to control data transmission.

Slow start. The slow start algorithm aim is to probe the network to check the current available capacity so to limit the bit rate entering the network and to avoid congestion. IW, the initial value of cwnd, must be less than or equal to $2 \cdot \mathrm{SMSS}$ bytes. A non-standard, experimental TCP extension allows using a larger window, whose value is limited by Eq. (5)

$$\mathrm{IW} = \min(4 \cdot \mathrm{SMSS}, \max(2 \cdot \mathrm{SMSS}, 4380\ \mathrm{bytes})) \tag{5}$$

Table 1
Definition of TCP parameters

Parameter	Definition
SEGMENT	Any TCP/IP data or acknowledgment packet (or both)
SENDER MAXIMUM SEGMENT SIZE (SMSS)	Size of the largest segment that the sender can transmit. It depends of the type of network used and of other factors
RECEIVER MAXIMUM SEGMENT SIZE (RMSS)	Size of the largest segment the receiver can accept
FULL-SIZED SEGMENT	A segment that contains the maximum number of data bytes permitted (i.e. SMSS bytes of data)
RECEIVER WINDOW (rwnd)	The most recently advertised receiver window and a receiver-side limit on the amount of outstanding data. It corresponds, at least in the implementations checked, to half of the receiver buffer length, at the beginning of the transmission
CONGESTION WINDOW (cwnd)	A TCP state variable that limits the amount of data TCP can send. It is a limit on the amount of data the sender can transmit into the network before receiving an acknowledgment (ACK). Some implementations maintain cwnd in units of bytes, while others in units of full-sized segments
INITIAL WINDOW (IW)	Size of the sender's congestion window after the three-way handshake is completed
FLIGHT SIZE (FlightSize)	The amount of data that has been sent but not yet acknowledged. Actually it identifies the segments still "in flight", still inside the network
SYN_RETRIES	Maximum number of retransmissions for SYN segment
TCP_RETRIES2	Maximum number of retransmissions for data segment
MAX_RTO	Maximum value for the Retransmission TimeOut (RTO). It represents an upper bound to the timeout, which is updated by the backoff algorithm when TCP timer expires

The initial value of ssthresh may be arbitrarily high and it is reduced in response to congestion. The slow start algorithm is used when cwnd < ssthresh. During slow start, a TCP increases cwnd by at most SMSS bytes for each ACK received that acknowledges new data. The slow start phase ends when cwnd exceeds ssthresh or when congestion is observed.

Congestion avoidance. The congestion avoidance algorithm is used when cwnd > ssthresh. When cwnd and ssthresh are equal, the sender may use either slow start or congestion avoidance. Congestion avoidance continues until congestion is detected. Within the congestion avoidance phase cwnd is increased by 1 full-size segment after the arrival of a number of ACKs corresponding to the value of cwnd/SMSS is arrived (each (cwnd/SMSS) ACKs $\rightarrow$ cwnd = cwnd + 1 $\cdot$ SMSS). When a TCP sender detects segment loss by using the retransmission timer, the value of ssthresh must be set to no more than the value given in Eq. (6)

$$\text{ssthresh} = \max(\text{FligthSize}/2, 2 \cdot \text{SMSS}) \tag{6}$$

Furthermore, cwnd must be set to no more than the loss window, which equals 1 full-sized segment (regardless of the value of IW). Therefore, after re-transmitting the dropped segment the TCP sender uses the slow start algorithm to increase the window from 1 full-sized segment to the new value of ssthresh, where congestion avoidance is started again.

Fast Retransmit/Fast Recovery. A TCP receiver should send an immediate duplicate ACK when an out-of-order segment arrives. The purpose of this ACK is to inform the sender that a segment was received out-of-order and which sequence number is expected. From the sender's perspective, duplicate ACKs can be caused by a number of network problems (e.g. dropped segments, re-ordering of data, replication of data). Obviously, a TCP receiver will send an immediate ACK when the incoming segment fills in all or part of a gap in the sequence space. The TCP sender uses the "fast retransmit" algorithm to detect and recover loss, based on incoming duplicate ACKs. The algorithm uses the arrival of 3 duplicate ACKs (4 identical ACKs) as an indication that a segment has been lost. After receiving 3 duplicate ACKs, TCP performs a retransmission

Table 2
TCP-NewReno specification

Transmission algorithms/events	
"Slow Start"	*Setup:* IW = min(4 · SMSS, max(4380, 2 · SMSS)) CWND = IW ssthresh = Infinity MAX_RTO = 120 SYN_RETRIES = 10 TCP_RETRIES2 = 15 *Data communication:* Acknowledgment arrival: CWND = CWND + 1; TW = min(source buff, min(cwnd, rwnd))
"Congestion Avoidance"	cwnd = cwnd + SMSS · SMSS /cwnd
"Fast Retransmit/Fast Recovery"	sstresh = max(2, FlightSize/2) cwnd = sstresh + 3 · SMSS <retransmit data> <transmit new data if possible> cwnd = sstresh
3 Duplicated Acknowledgments	"Fast Retransmit" entered
"Fast Recovery" terminated	"Congestion Avoidance" entered
Timeout	cwnd = 1 · SMSS Backoff algorithm: RTO = min(2RTO, MAX_RTO) <retransmit data> sstresh = max(2, FlightSize/2) "Slow Start" entered

of what appears to be the missing segment, without waiting for the retransmission timer to expire. After the fast retransmit algorithm sends the missing segment, the "fast recovery" algorithm governs the transmission of new data until a non-duplicate ACK arrives. Finally the fast recovery phase is terminated when a segment acknowledging all the retransmitted and the new packets, is received. More details about the algorithm specification are shortly indicated in Table 2.

TCP researchers have suggested a number of loss recovery algorithms improving fast retransmit and recovery. Some of them are based on the TCP selective acknowledgment (SACK) option [8], which allows specifying exactly the sequence number of the missing segment, together with the FACK (Forward Acknowledgment) [40] algorithm currently implemented in the latest version of Linux Operating System.

More schematically, the procedures listed above may be summarized in Table 2; a C-like language is used for the description.

5. Analysis of protocol solutions

5.1. General framework

In this paper, the main attention is focused on the impact of bursty errors on upper layers and on the design of suited transmission mechanisms to be implemented within the transport layer. In this perspective, a wide part of the scientific community has studied how TCP behaves in presence of wireless links [5,6]. TCP misunderstands link errors, considers them as congestion [26] and does not offer satisfying performances. TCP action is not tuned for radio tactical infrastructures where congestion events are unusual and outage periods are frequent [41]. TCP reduces the congestion window and consequently the transmission rate, causing in this case the waste of the network available resources. Even in "good state" TCP does not assures the full exploitation of the bandwidth available because the transmission window is not sufficient to fill the "bandwidth pipe" of such channels, which may experience a large bandwidth delay product [42].

In this perspective the investigation of transport protocol solutions addresses two different key points, developed in the next subsections.

The first of them contains two different solutions (namely TCP-radio and TCP-radio-newRTO) based on TCP schemes, which differ from TCP-NewReno in the recovery mechanism and timeout management, have been proposed and discussed. The subsection after the next studies TCP-radio and TCP-radio-newRTO behaviours with reference to TCP-NewReno.

From the literature [5,6,18,19], it is well known that TCP is not efficient in wireless environments: not only because of the delay-bandwidth product TCP is able to deal with congestion events but is does not consider correctly packet losses caused by link errors, such as outage events and channel degradation. On the other hand the timeout management, which can be often invoked during long outage periods, as experienced in radio-military networks, causes a strong deterioration of the communication performance in terms of reduced throughput and high end-to-end service delay. The backoff algorithm does not help because it generates long and useless idle time between two successive retransmissions. Another important aspect of the timeout management within TCP implementation is the maximum number of retries both in the "three-way handshake" and in the data communication, which may cause a premature aborting of the connection. In order to cope with the aforementioned drawbacks, the two proposals indicated in the following as TCP-radio and TCP-radio-newRTO have been studied and implemented on the Linux operating system kernel 2.2.19.

The two solutions present have different transmission mechanisms with respect to TCP and offer an improved recovery mechanism able to deal more effectively with link errors rather than congestion events that are extremely rare in the considered environment. Moreover, TCP-radio-newRTO version presents a modified timeout management, able to deal with long outage periods.

5.2. Modified TCP versions

TCP-radio. In order to assure an effective exploitation of the network available resources in terms of bandwidth capabilities, the initial window is set to the bandwidth-delay product and the congestion window value is no longer increased when the acknowledgments arrive. It is straightforward that also the TCP

Table 3
TCP-radio specification

Transmission algorithms/events	
"Normal Transmission"	*Setup:* IW = <bandwidth-delay product> cwnd = IW ssthresh = Infinity MAX_RTO = 120 SYN_RETRIES = 10 TCP_RETRIES2 = 15 *Data communication:* Acknowledgment arrival: cwnd = cwnd + 1; TW = min(source buff, min(cwnd, rwnd))
"Fast Retransmit/Fast Recovery"	<retransmit data> <transmit new data if possible>
1 Duplicated Acknowledgment	"Fast Retransmit" entered
"Fast Recovery" terminated	"Normal Transmission" entered
Timeout	Backoff algorithm: RTO = min(2RTO, MAX_RTO) <retransmit data> "Normal Transmission" entered

buffers on the receiver and sender sides are properly tuned in order to make effective the modifications indicated above.

Also the TCP recovery mechanism has been modified: at the end of the recovery phase, the congestion window valued is not reduced and, as a consequence, the transmission rate is the maximum possible as set at the opening of the connection in order to "fill" the bandwidth pipe. Also the timeout management has been slightly modified. In particular, when a timer expires the congestion window is not reduced to one segment but, once terminated the retransmission phase, the data communication continues with the maximum rate possible as indicated in the previous case. A short summary about TCP-radio proposal is shown in Table 3.

TCP-radio-newRTO. In addition to the modifications suggested in the TCP-radio proposal, it is also recommendable to consider that long outage periods, caused by shadowing effects, determine TCP timer expiring and consequent long idle time between successive retransmissions because of the employment of backoff algorithm. In order to mitigate this aspect it is necessary to modify the overall management of TCP timeout [43]. For this purpose, backoff algorithm, which may cause long waiting periods before retransmitting a segment, is excluded and the upper bound for RTO (Retransmission TimeOut) value is reduced in order to avoid long idle periods, if the retransmission of data may still be possible [44]. Furthermore, the upper bounds of retransmission in the connection-opening phase and during data communication (namely SYN_RETRIES and TCP_RETRIES2) are raised up to assure the accomplishment of the communication. In this way, even if the transmission medium experiences long periods of unavailability, TCP-radio-newRTO is able to assure the establishment of the connection (thanks to the increased value of SYN_RETRIES) and the accomplishment of the data transaction (thanks to the modified recovery mechanism and, mainly, to the increased value of TCP_RETRIES2).

Some more details about the schemes implemented in TCP-radio-newRTO are indicated in Table 4.

A comparison among the two solutions presented above and TCP-NewReno, usually implemented in Linux kernels, is presented in the next subsection.

5.3. Evaluation of the modified TCP versions

In order to analyse the behaviour of the proposed solutions (i.e. TCP-radio and TCP-radio-newRTO) with respect to TCP-NewReno dynamics, several tests have been accomplished taking a reference scenario

Table 4
TCP-radio-newRTO specification

Transmission algorithms/events	
"Normal Transmission"	*Setup:* IW = <bandwidth-delay product> cwnd = IW ssthresh = Infinity MAX_RTO = 12 SYN_RETRIES = 500 TCP_RETRIES2 = 500 *Data communication:* Acknowledgment arrival: cwnd = cwnd + 1; TW = min(source buff, min(cwnd, rwnd))
"Fast Retransmit/Fast Recovery"	<retransmit data> <transmit new data if possible>
1 Duplicated Acknowledgment	"Fast Retransmit" entered
"Fast Recovery" terminated	"Normal Transmission" entered
Timeout	RTO_updating algorithm: RTO = min(RTO, MAX_RTO) <retransmit data> "Normal Transmission" entered

composed of just one wireless link whose bandwidth, for the sake of completeness, has been assumed in each test equal to 16, 64, 256 kbps and 1 Mbps. Concerning the latency imposed in each emulation, a delay of 1 s has been considered, to take into account not only the propagation delay of the channel but also the extra latencies introduced by lower layer processing (mainly due to interleaving operations). At a first instance, a probability of packet loss equal to 0.10 (i.e. $\pi_1 = 0.10$), which is quite close to the values reported in real radio systems, has been assumed.

In order to have a qualitative description of the behaviours experienced by the solutions considered, the time-sequence diagram for a long data communication (transfer of 1 Mbytes) has been studied for each bandwidth case (i.e. 16, 64, 256 kbps, 1 Mbps). For each analysed case one single realisation has been considered.

Concerning the tests performed with bandwidth values equal to 1 Mbps and 256 kbps, the graphs shown in Fig. 4 and in Fig. 5 (showing the number of transmitted bytes versus time for TCP-NewReno, TCP-radio and TCP-radio-newRTO, having bandwidth of 1 Mbps and 256 kbps, respectively) highlight that the recovery mechanisms adopted in the modified TCP versions allow a more effective retransmission of lost data segments. The transfer size 1 Mbytes in both cases. Moreover, a more aggressive transmission scheme based on the knowledge of bandwidth-delay product allows a better utilization of the network available resource. In these cases there is also a difference between the behaviours experienced by TCP-radio and TCP-radio-newRTO. This is due to the modified timeout management implemented in TCP-radio-newRTO that avoids long and useless idle periods between successive retransmissions. TCP-radio even if it adopts an effective recovery mechanism, is not able to cope with long outage periods that cause frequent timer expirations. It is important to highlight that the performance difference between TCP-NewReno and modified versions of TCP is due also to the large "bandwidth pipe" that TCP-NewReno is not able to exploit completely because of the maximum congestion window value ruled by the buffer size on sender and receiver sides.

Figs. 6 and 7 show the same quantities of Figs. 4 and 5, but having a bandwidth availability of 64 kbps and 16 kbps, respectively. The improved performance of the modified TCP versions is still evident because of the changed recovery mechanism and timeout management. In the case with bandwidth of 16 kbps the enhancement registered with respect to TCP-NewReno behaviour is less clear than in the previous tests because of the limited bandwidth-delay product. It is equal to few segments in this case and also TCP-NewReno is able to offer satisfying performance if timeouts are not so frequent, as in the example reported in Fig. 7. Differently, in Fig. 6, the high number of timeouts causes a severe degradation of TCP performance.

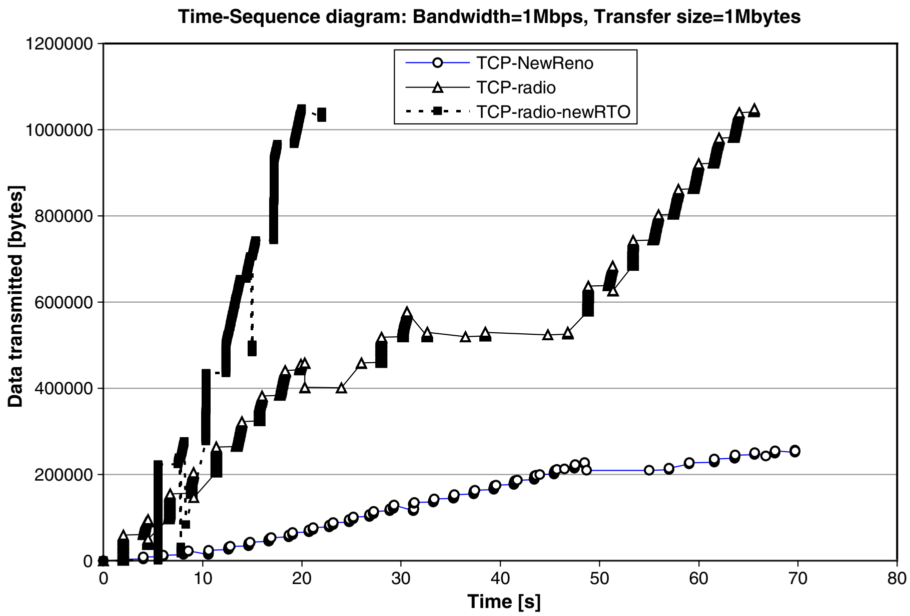

Fig. 4. Comparison of sequence diagrams for the proposed transport solutions (bandwidth of 1 Mbps).

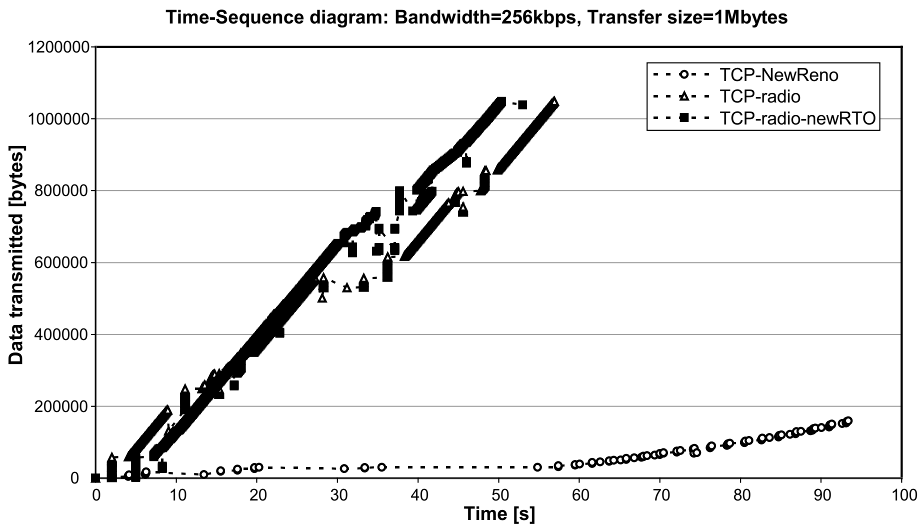

Fig. 5. Comparison of sequence diagrams for the proposed transport solutions (bandwidth of 256 kbps).

In order to analyse the role played by the timeout management in data communications performed in wireless environment more deeply, the case where the channel state is very critical is considered, experiencing a probability of packet lost equal to 0.50 ($\pi_1 = 0.50$, in practice a useless channel). It is straightforward that it concerns a particularly extreme case because the probability of receiving both a correct and corrupted packet is the same. This investigation is however important if we consider that in radio-military networks the partial unavailability of the transmission link and, then, the high number of data packet lost is not a rare event, because of the hazardous conditions in which the transaction has to be accomplished. As shown in the time-sequence diagram depicted in Fig. 8 (containing the transmitted number of bytes versus time for the three

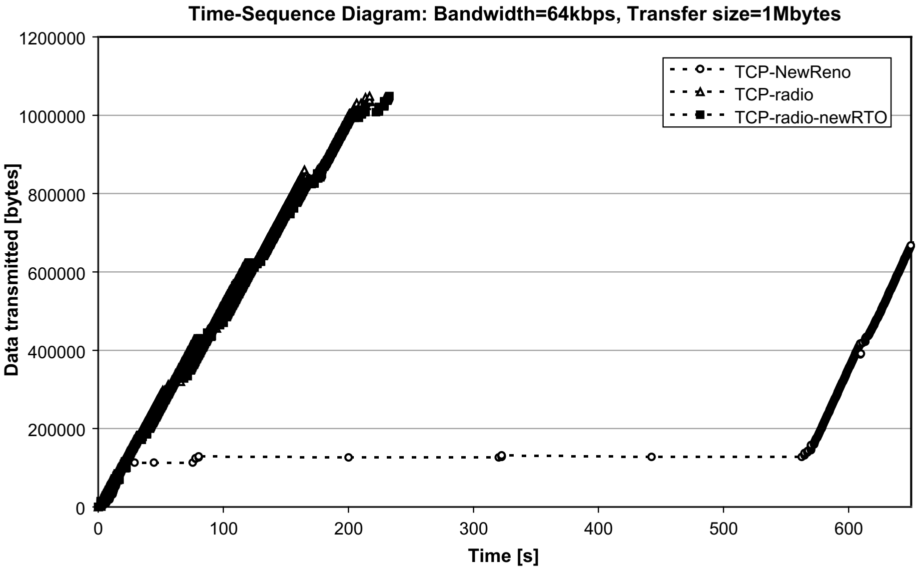

Fig. 6. Comparison of sequence diagrams for the proposed transport solutions (bandwidth of 64 kbps).

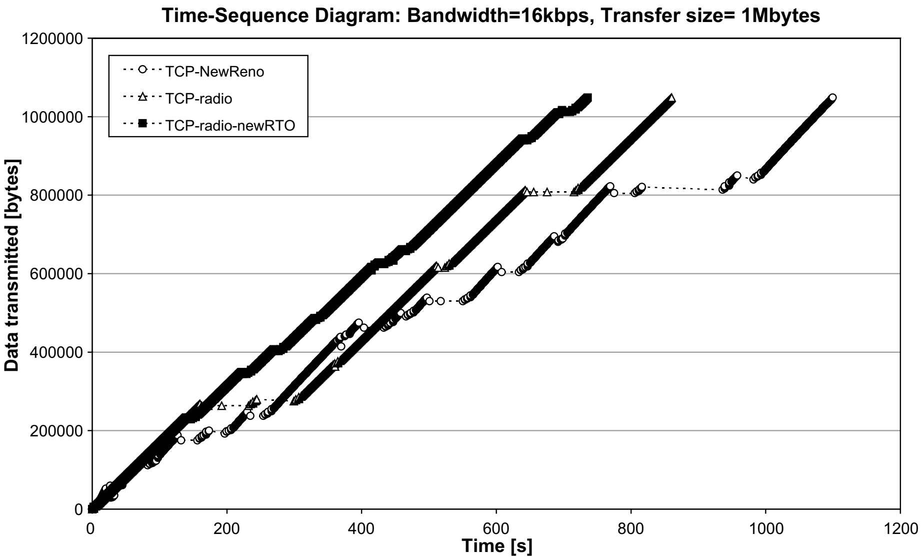

Fig. 7. Comparison of sequence diagrams for the proposed transport solutions (bandwidth of 16 kbps).

transport layers, a transfer size of 100 kbytes and a bandwidth of 64 kbps), the behaviour experienced by the TCP-radio-newRTO is more performant than the other solutions, thanks to the modified timeout management that assures acceptable performance results even if the channel state is very critical. It is also noticeable that at the beginning of the transaction the modified TCP versions (i.e. TCP-radio and TCP-radio-newRTO) present similar behaviours because no timeouts have occurred yet. On the contrary, in the following of the communication the difference is more evident because timers expire several times and only TCP-radio-new-RTO is able to manage them efficiently.

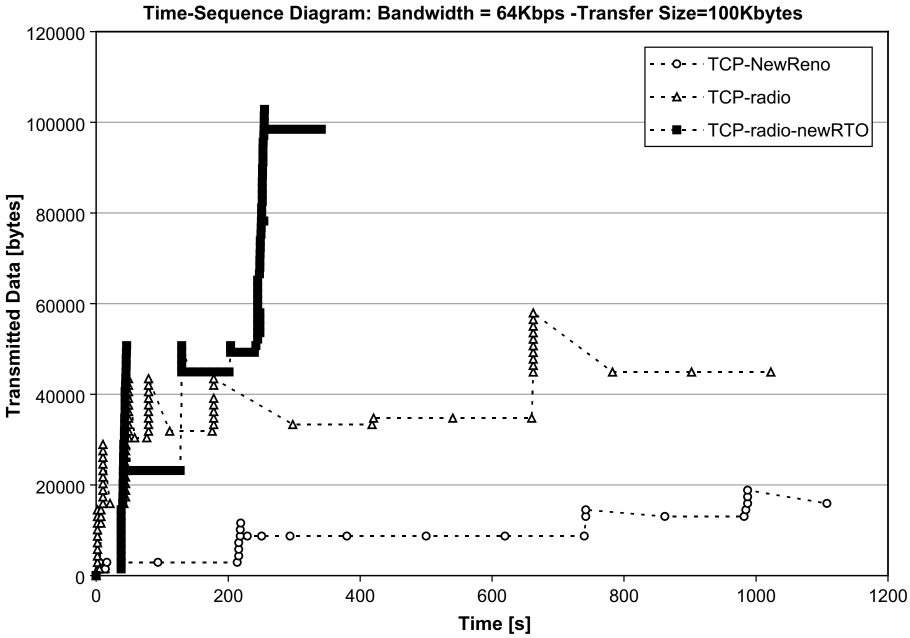

Fig. 8. Comparison of sequence diagrams for proposed transport solutions.

6. Performance evaluation

As introduced in the previous sections, the reference scenario is composed of one wireless link whose bandwidth has been assumed, as in the previous section, equal to 16, 64, 256 kbps and 1 Mbps. Concerning the latency imposed in each emulation, a delay of 1 s has been considered. In order to evaluate the overall behaviour of the system, different traffic loads have been considered. For each test, file transfers of 3 k, 10 k, 100 k, 500 k, 1 Mbytes have been performed. The overall analysis has been achieved considering three different values of packet loss probability, namely $\pi_1 = 0.10$ (indicated as *test 1*), 0.30 (indicated as *test 2*) and 0.50 (indicated as *test 3*), in order to show the behaviour of the introduced protocol solutions when the wireless environment is under very hazardous conditions. In particular the first tests implements a value of 0.10 that is quite common in typical radio military systems; in next tests this value raises up to 0.30 and 0.50. For the sake of the completeness and to increase the statistical validation, each test has been performed by considering three different couples of values of the transition probabilities (α, β), which rule the average duration of outage periods, as mentioned in the test description below.

In order to take more detailed and precise indications about the performance, two different metric parameters have been employed

- The normalised throughput (7), computed as the ratio between the registered average throughput and the network available bandwidth,

$$\text{Normalized\_throughput} = \frac{\text{Throughput}}{\text{Bandwidth}} \tag{7}$$

- The gain (8) in terms of end-to-end delay service reported for each analysed protocol solution with respect to TCP-NewReno implementation. The gain (%) has been calculated as

$$\text{Gain}(\%) = \frac{T_{\text{TCP-NewReno}} - T}{T_{\text{TCP-NewReno}}} \cdot 100 \tag{8}$$

where T stands for the end-to-end service delay reported by the analysed solutions and $T_{\text{TCP-NewReno}}$ for the same quantity got by using the TCP-NewReno solution, taken as reference.

The solutions introduced in the paper have been compared with other schemes proposed by the scientific community. In this perspective TCP-Westwood+ has been considered, in order to exploit its capabilities of bandwidth estimation that may be very profitable in a wireless environment. To establish a maximum performance bound, an "ideal" transmission protocol, has been considered. It knows the channel state a priori and, as a consequence, is able to transmit data segments when the channel really allows it. It is straightforward that it represents a non-realistic case but it provides an upper bound of the performance results that may be achieved by real transport solutions when the communication is performed in the same channel conditions. And the packets' overhead is the same.

Concerning the test performed with TCP-Westwood+ (implemented without SACK), the Ns-2 [45] simulator has been employed; the "ideal" algorithm has been tested through a simple simulator written in C language.

6.1. Test 1 ($\pi_1 = 0.10$)

Different emulations has been performed by setting the following transition probability (α, β): (0.001, 0.01), (0.01, 0.1) and (0.1, 1). The registered results have been averaged. The normalised throughput and the gain are the performance metrics.

The cases where a bandwidth of 1 Mbps and 256 kbps is available are considered in Figs. 9 and 10, respectively. The normalised throughput versus the transfer size is shown for the different TCP alternatives and for the ideal case. It is important to notice that a high bandwidth-delay product is present and, as a consequence, a poor result is expected from TCP-NewReno, which is not able to assure high performance in environments where high delay and bandwidth capacity are experienced. The more effective behaviour of TCP-radio and TCP-radio-newRTO with respect to TCP-NewReno and TCP-Westwood+ is clear when big transfers are considered. Actually the two novel protocols are able to fill the bandwidth pipe and even in presence of bursty losses they can effectively recover without heavy deterioration of the overall performance. In more detail, TCP-radio and TCP-radio-newRTO offer normalised throughput values (last column of Figs. 9 and 10) just

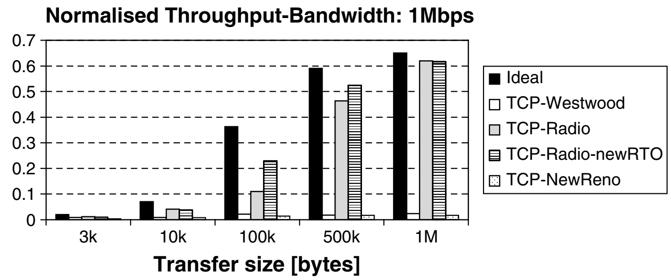

Fig. 9. Normalised throughput with a bandwidth of 1 Mbps.

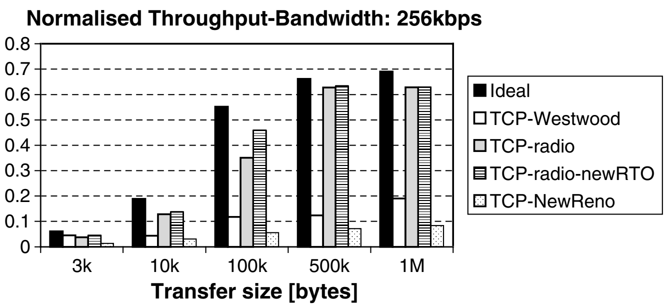

Fig. 10. Normalised throughput with a bandwidth of 256 kbps.

over 0.6 when a transfer of 1 Mbytes is analysed. For lower transfer sizes, the advantage is progressively less evident. In particular, for size of 3 kbytes and 10 kbytes the difference among the different solutions is very limited, because the reduced number of segments entering the network. In short, the performance of TCP-radio-newRTO for big sizes of transfer is quite close the ideal one; in particular, for 500 kbytes and 1 Mbytes sizes, TCP-radio-newRTO guarantees throughput values of 0.52 and 0.61 while the ideal solution ranges from 0.59 to 0.65.

Another important aspect is that TCP-Westwood+ always behaves better than TCP-NewReno: this effect is mainly due to the employment of a bandwidth estimation mechanism, which allows proper adjustments of cwnd values, rather than drastically halving the packets in flight, as TCP-NewReno does. On the other hand, the performance experienced by TCP-Westwood+ is lower than TCP-radio and TCP-radio-newRTO, because it is not able to fill the bandwidth pipe completely.

The cases of bandwidths 64 kbps and 16 kbps, are shown in Figs. 11 and 12, respectively. The differences between the solutions are small. The bandwidth-delay product is limited and also TCP-NewReno and TCP-Westwood+ are able to offer quite satisfying results. This aspect is still more evident if the behaviour experienced with a bandwidth of 16 kbps is analysed more deeply. In this case all the solutions present normalised throughput values close to the Ideal case, especially when a transfer size of 3 kbytes and 10 kbytes is considered. From Fig. 12, it is clear that also TCP-NewReno offers effective performance results because the bandwidth pipe in this case corresponds to the transmission of just three segments and, as a consequence, it can be quickly filled by the standard TCP transmission scheme. The normalised throughput registered by the TCP-NewReno in Fig. 12 (last column, namely 0.52) is however lower than the other solutions because the recovery mechanism has the effect of approximately halving the congestion window (cwnd), setting it to the half the packets in flight, and thus, of deteriorating the performance. TCP-Westwood+ still outperforms TCP-NewReno because of its enhanced capabilities in the recovery phase.

The end-to-end delay service gain (%) with respect to TCP-NewReno one is the metric now. Differently from the analysis reported above (i.e. normalised throughput), the performance results for each transfer size is analysed by modifying the channel bandwidth value.

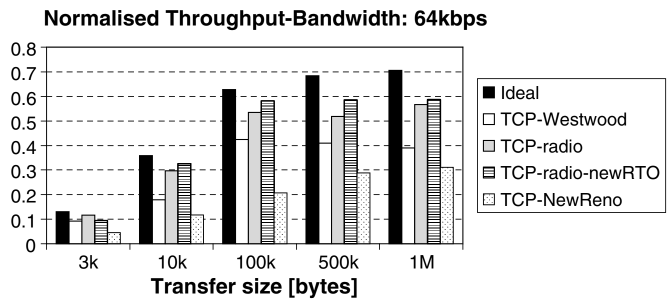

Fig. 11. Normalised throughput with a bandwidth of 64 kbps.

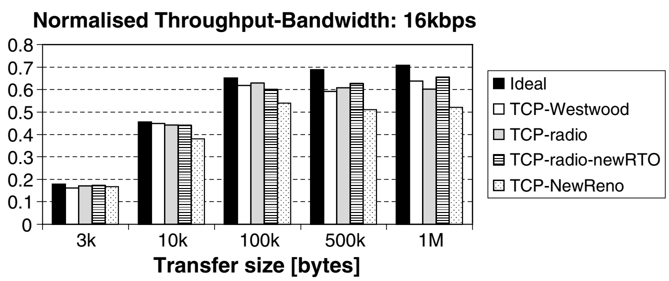

Fig. 12. Normalised throughput with a bandwidth of 16 kbps.

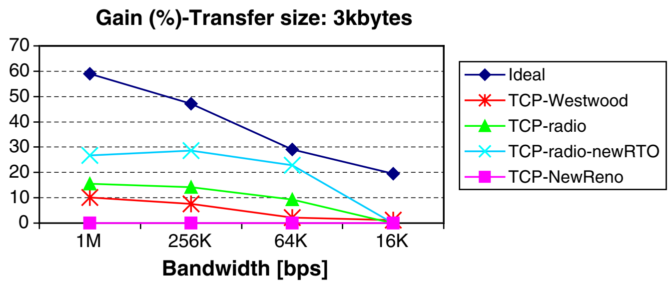

Fig. 13. Gain (%) with a transfer size of 3 kbytes.

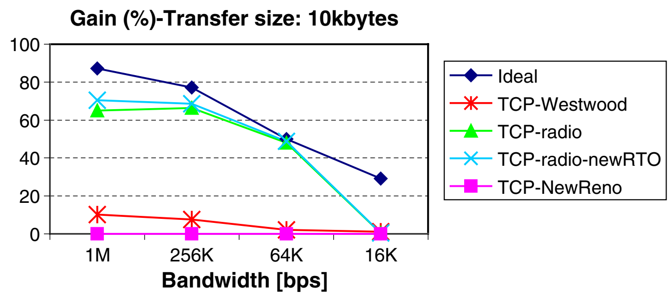

Fig. 14. Gain (%) with a transfer size of 10 kbytes.

Figs. 13 and 14 show the Gain in percentage versus the bandwidth available for a transfer size of 3 kbytes and 10 kbytes, respectively. The number of segments transmitted is small and the gain expected with respect to TCP-NewReno cannot be too high. From Fig. 13: TCP-radio-newRTO and TCP-radio, which provide the best results, have a gain ranging from 26% to 0.5% and from 15% to 0.3%, respectively, as the available bandwidth value decreases. The reason of it is twofolds: on one hand the limited number of segments transmitted throughout the network does not allow modified TCP versions to take a significant gain over TCP-NewReno; on the other hand, by reducing the available bandwidth, the results collected for each protocol solution are very close the reference because of limited bandwidth-pipe. Furthermore, from Fig. 14: the behaviour is similar. The gain ranges from 70% to 0.8%, for TCP-radio-newRTO, and from 65% to 0.5%, for TCP-radio. TCP-Westwood+ provides a maximum gain of about 10% when a bandwidth of 1 Mbps is available. This slight improvement with respect to TCP-NewReno is due to the fact that the employed implementation does not support SACK option and it cannot guarantee results as satisfying as the other modified TCP versions.

By further increasing the transfer size (100 kbytes, 500 kbytes and 1 Mbytes, as shown in Figs. 15–17, respectively), it is important to note that TCP-radio-newRTO and TCP-radio provide higher gain results;

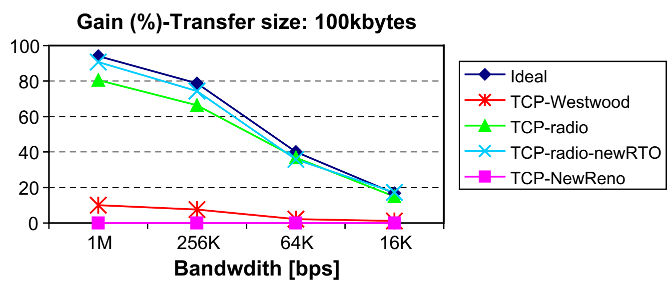

Fig. 15. Gain (%) with a transfer size of 100 kbytes.

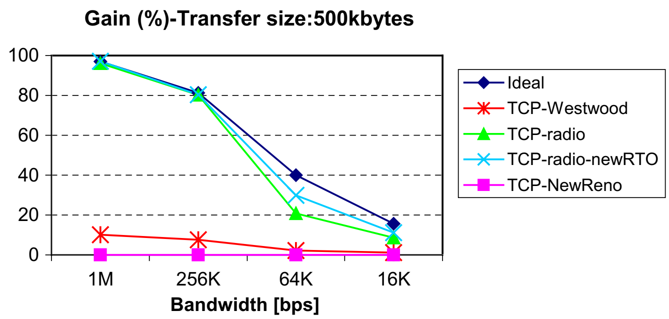

Fig. 16. Gain (%) with a transfer size of 500 kbytes.

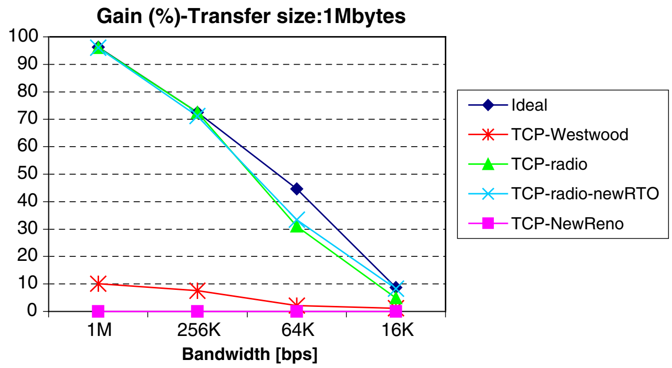

Fig. 17. Gain (%) with a transfer size of 1 Mbytes.

in particular a maximum gain well over 90% for TCP-radio-newRTO in all the shown cases of 1 Mbps of bandwidth is available. Concerning TCP-radio, the maximum gain ranges from 80% (transfer size of 100 k) to 96% (transfer size of 1 Mbytes). The main element of this investigation is that both solutions provide performance results very close to the Ideal ones.

6.2. Test 2 ($\pi_1 = 0.30$)

In this case, in order to increase the statistical validation of the performed emulation, three different couples of values for the transition probabilities have been considered: (0.428, 1), (0.0428, 0.1) and (0.00428, 0.01). It is straightforward that the performance results are heavily affected by the packet error rate equal to 0.3. For the sake of simplicity but without loss of generality, in the following, only the case of 1 Mbps is considered, because it provides the most interesting results in terms of normalised throughput and gain. Fig. 18 shows the normalised throughput in logarithmic scale versus the transfer sizes for the considered transport algorithm. Fig. 19 shows the gain in the same conditions. Very low throughput values are provided for all the solutions. In general TCP-radio and TCP-radio-newRTO provide the best performance values, as in the previous cases; providing numerical results ranging from 0.002 to 0.006, for TCP-radio, and from 0.002 to 0.06 for TCP-radio-newRTO, by increasing the transfer size. Concerning TCP-Westwood+ and TCP-NewReno, the values registered are very low and are the consequence of the implemented recovery mechanism, which does not perform well in presence of long bursts of errors. In particular, from Fig. 18 the values range from 0.0006 to 0.0004 for TCP-NewReno and from 0.002 to 0.004 for TCP-Westwood+. Ass outlined in the previous analysis, the limited performance of TCP-Westwood+ is also determined by the unemployment of SACK option, which is recommended in highly noisy environments. For transfer size of 1 Mbytes, the TCP-NewReno value is not indicated, because the registered value is too low due to the high number of retransmissions and it is not

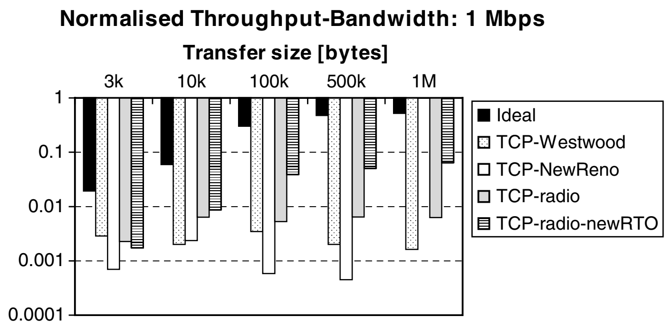

Fig. 18. Normalised throughput with a bandwidth of 1 Mbps.

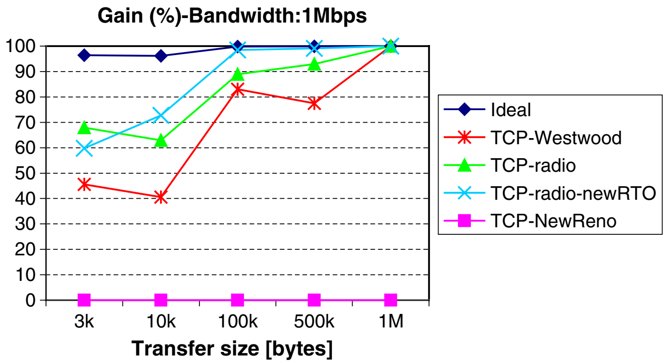

Fig. 19. Gain (%) with a bandwidth of 1 Mbps.

significant. It is obvious from the above observations and from the diagram shown in Fig. 18, that the Ideal case outperforms all the prospected solutions.

From the point of view of the gain in terms of end-to-end delay service with respect to TCP-NewReno, some important indications arise from Fig. 19. In particular, the advantages provided by the different solutions with respect to the reference are meaningful. All the solutions offer gain over 40%, growing as the transfer size increases. This aspect is mainly due to the fact that, effectively, TCP-radio and TCP-radio-newRTO implement an enhanced recovery mechanism; on the other hand, TCP-NewReno performance is very unsatisfying. TCP-Westwood+ provides very satisfying results with respect to TCP-NewReno. Finally, as outlined before, the numerical results corresponding to the case of 1 Mbytes are not really significant, because TCP-NewReno experiences a very high end-to-end service delay and for all the protocol solutions similar gain results, close to 100%, are observed.

6.3. Test 3 ($\pi_1 = 0.50$)

Finally the case of useless channel (a packet error rate equal to 0.50) is considered. This case deserves a particular attention, because it describes the case where radio-military communications are performed in extremely hazardous conditions.

All the emulations have been achieved by considering as transition probability values the couple (0.1, 0.1). The statistical validation has been completed by performing repeated tests with the same transition probabilities.

Due to the very high packet error rate experienced in the transmission links, the expiration of TCP timers is a very frequent event. So it is expected that TCP-based transmission mechanisms behave very poorly. Moreover, because of the high number of retransmissions, in these conditions it is also possible to assist to connection aborts. Given the high impact on transport protocol design that such an environment may have, all the bandwidth values tested will be punctually focused.

At the beginning, the investigation is addressed to the study of the normalised throughput. Bandwidth values equal to 1 Mbps and 256 kbps are considered in Fig. 20 and in Fig. 21 that shows the normalised throughput versus the transfer size for all the schemes. All the solutions offer unsatisfying results, because of the heavy impact of the error rate on data communication. In particular TCP-NewReno and TCP-Westwood+ for the cases of transfer sizes ranging from 3 kbytes to 100 kbytes provide normalised throughput results ranging from 0.0001 to 0.0008; when greater transfers of data are explored, the related connections are aborted or terminated after a very long time; the values are not reported because not significant. Concerning TCP-radio, improved results with respect to the previously discussed ones are experienced; however, globally the performance result is not satisfying. This behaviour, apparently in contradiction with the tests performed at packet error rates of 0.1 and of 0.3, is due to the timeout management that is based on the backoff algorithm and, causes long idle periods between successive retransmissions. For what concerns TCP-radio-newRTO, the results show that its behaviour is far from Ideal, but more performant that the other solutions considered. This aspect is mainly due to the enhanced timeout management, which is no longer based on the backoff mechanism and allows a better utilization of the available bandwidth, without wasting too long time between successive retransmissions.

Also for other bandwidth cases (i.e. 64 kbps and 16 kbps, reported in Fig. 22 and in Fig. 23) similar conclusions hold. In more detail, from Figs. 22 and 23: for transfers of small size (namely 3 kbytes and 10 kbytes) the results reported for each protocol solution are quite close each other. When the dimension of the transfer increases and consequently also the number of the retransmissions, the efficacy of TCP-radio-newRTO and TCP-radio is straightforward. Taking the case with a bandwidth of 16 kbps, as an example, TCP-Westwood+ and TCP-NewReno for transfer of 100 kbytes have a normalised throughput of about 0.01. On the other hand, with TCP-radio the throughput is 0.065 and, with TCP-radio-newRTO, is 0.165. Moreover, when

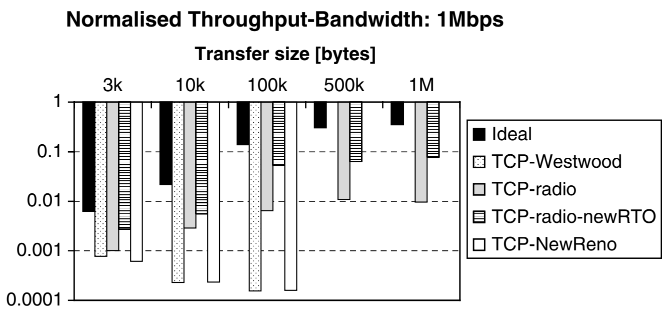

Fig. 20. Normalised throughout with a bandwidth of 1 Mbps.

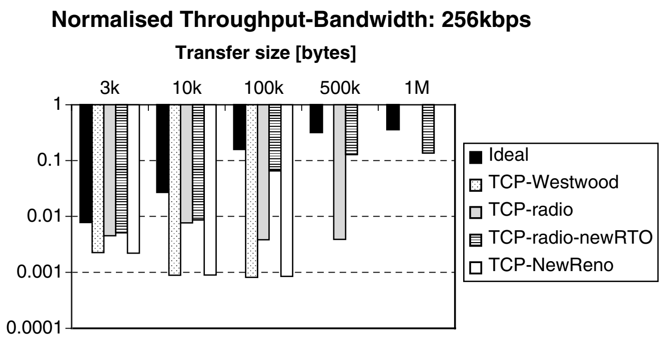

Fig. 21. Normalised throughput with a bandwidth of 256 kbps.

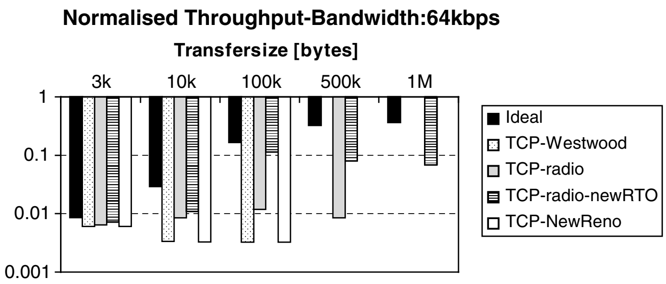

Fig. 22. Normalised throughput with a bandwidth of 64 kbps.

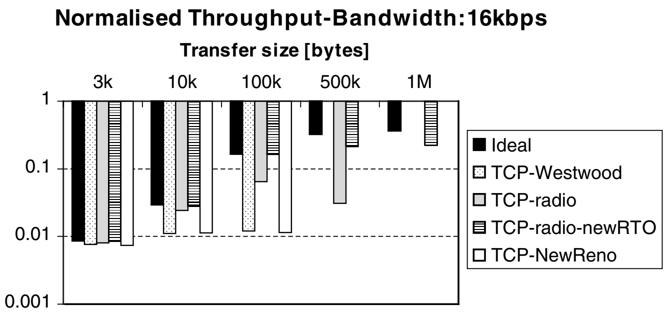

Fig. 23. Normalised throughput with a bandwidth of 16 kbps.

the transfer size increases (i.e. cases of 500 kbytes and 1 Mbytes) only TCP-radio-newRTO is able to complete the data communication, while in the other cases the connections are aborted or successfully terminated after a very long time.

Concerning the gain in term of end-to-end service delay, with respect to TCP-NewReno, transfer sizes of reduced dimension such as 3 kbytes and 10 kbytes are shown in Fig. 24 and in Fig. 25. The gain obtained with TCP-radio and TCP-radio-newRTO is significant when bandwidth values of 1 Mbps and 256 kbps are considered; in particular, taking the maximum gain, numerical values range from 77% to 95%. It is straightforward that, for transfer of small dimensions as the one considered here, if the channel bandwidth decreases also the gain is lower because the performance are very close to TCP-NewReno.

Concerning the case of transfer size equal to 100 kbytes depicted in Fig. 26, the impact of a packet error rate of 0.50 is more evident. In this configuration, TCP-NewReno and TCP-Westwood+ are not efficient at all. TCP-radio and TCP-radio-newRTO offer gain values ranging from 72% to 97.5%. TCP-Westwood+ does not offer any gain. This is due to the missing employment of the SACK option in the considered Westwood+ implementation, which does not assure an effective retransmission procedure.

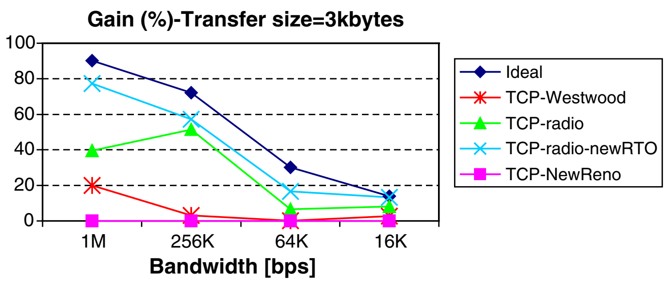

Fig. 24. Gain (%) with a transfer size of 3 kbytes.

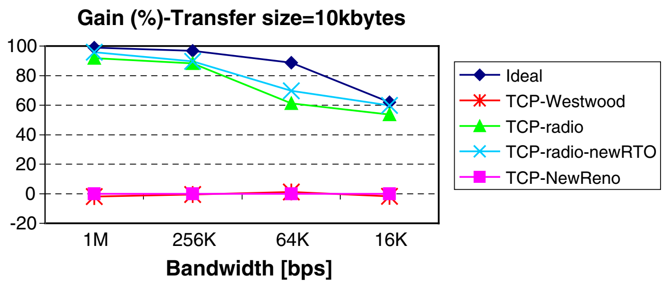

Fig. 25. Gain (%) with a transfer size of 10 kbytes.

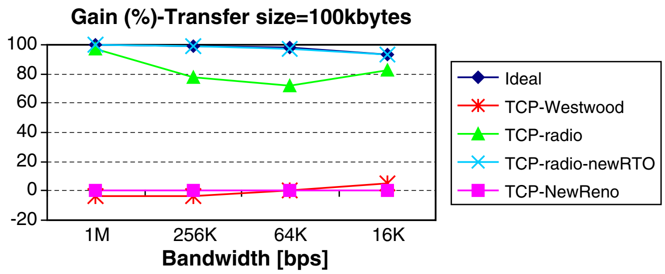

Fig. 26. Gain (%) with a transfer size of 100 kbytes.

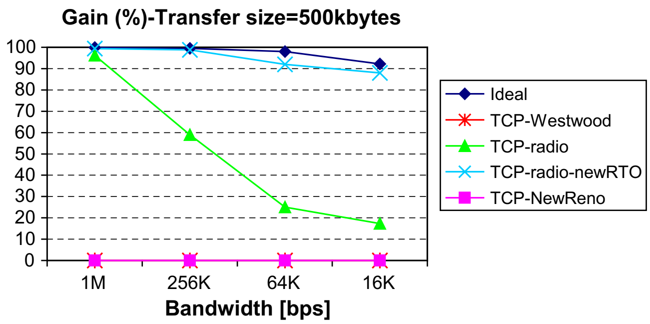

Fig. 27. Gain (%) with a transfer size of 500 kbytes.

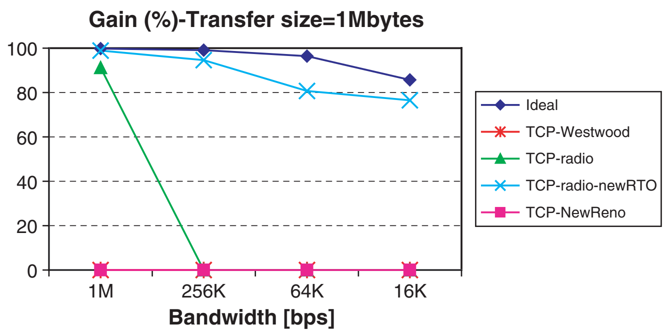

Fig. 28. Gain (%) with a transfer size of 1 Mbytes.

Finally, regarding configurations with transfer sizes of 500 kbytes and 1 Mbytes, whose results are contained in Fig. 27 and in Fig. 28, respectively, the situation is more critical: as previously pointed out, except for TCP-radio-newRTO, transfers require long time and, in most cases, the connections are aborted because the number of retransmissions exceeds the maximum allowed by TCP. In this case, TCP-radio-newRTO is the more satisfying solution, because offers results very close to the Ideal solution and makes the accomplishment of the communication possible, while the other solutions cannot always match it.

7. Conclusions

The paper has been addressed to the study of transmission mechanisms to be implemented at the transport layer and to be applied in a wireless environment, where constraints of specific radio-military applications have to be respected. In this perspective, by taking the TCP protocol implementation as a reference, two novel solutions (TCP-radio and TCP-radio-newRTO) have been designed and compared with TCP-NewReno and TCP-Westwood+ in order to check their effectiveness. For this purpose, the investigated protocol solutions have been deeply analysed by considering transfers of different sizes performed in critical channel conditions. The study has focused on the normalised throughout and on the gain in terms of end-to-end delay with respect to TCP-NewReno. The performed emulations pointed out that in presence of packet error rate equal to 0.1 and 0.3 and with availability of high bandwidth values, such as 1 Mbps and 256 kbps, the employment of the novel solutions allows registering high performance results. On the other hand, if there is a lower availability of network resources (i.e. 64 kbps and 16 kbps), the utilization of TCP-Westwood+ or TCP-NewReno provides still effective results, because of the limited bandwidth pipe to fill. When the channel status is extreme and a packet error rate equal to 0.5 (useless channel) is assumed, TCP-Westwood+, TCP-NewReno and TCP-radio performances are heavily deteriorated, because of high number of retransmissions. In this configuration, even if the channel state cannot be more critical and does not allow high performance in terms of normalised throughput, TCP-radio-newRTO guarantees better results than other solutions and can manage the accomplishment of the data communication when transfers of big size are performed (namely 1 Mbytes).

It is important to say that the two modified versions of TCP are able to assure high performance results maintaining the socket API semantic and applying only minor modifications on the transmission mechanisms implemented in TCP-NewReno for what concerns congestion window ruling, recovery mechanism (both TCP-radio and TCP-radio-newRTO) and TCP timeout management (only TCP-radio-newRTO).

Acknowledgements

The work is part of a scientific project between Selex Communications, Italy; the Department of Communication, Computer and System Science of the University of Genoa; and the Italian National Consortium for Telecommunications (CNIT). The material of the project that concerns TCP-based services in radio military networks (the object of this paper) is currently tested over real tactical environments in the field and partially used within standardization groups in military environment.

References

[1] Kim Y, Jeong BJ, Chung J, Hwang C-S, Ryu JS, Kim K-H, et al. Beyond 3G: vision, requirements, and enabling technologies. IEEE Commun Mag 2003;41(3):120–4.
[2] Information Sciences Institute, University of Southern California, Transmission control protocol – Darpa Internet program – protocol specification, IETF RFC 793, September 1981.
[3] Kenneth Miller C. Data distribution over IP in high error rate military radio environments. In: MILCOM 1998 – IEEE military communications conference, Boston (MA), October 1998, p. 1067–71.
[4] Marchese M. TCP/IP-based protocols over satellite systems: a telecommunication issue. In: Stavroulakis P, editor. Chapter of the book 'Reliability, Survivability and Quality of Large Scale Telecommunication Systems'. Chichester, England: Wiley & Sons, Ltd.; 2003. p. 167–98.
[5] Dawkins S, Montenegro G, Kojo M, Magret V, Vaidya N. End-to-end performance implications of links with errors. IETF RFC 3150, August 2001.
[6] Inamura H, Montenegro G, Ludwig R, Gurtov A, Khafizov F. TCP over second (2.5G) and third (3G) generation wireless networks. IETF RFC 3481, February 2003.

[7] Ludwig R. A case for flow-adaptive wireless links. Technical report UCB//CSD-99-1053, University of California at Berkeley, May 1999.
[8] Floyd S, Mahdavi J, Mathis M, Podolsky M. An extension to the selective acknowledgment (SACK) option for TCP. IETF, RFC 2883, July 2000.
[9] Ludwig R, Meyer M. The Eifel detection algorithm for TCP. IETF RFC3522, April 2003.
[10] Ferorelli R, Grieco LA, Mascolo S, Piscitelli G, Camarda P. Live internet measurements using westwood+ TCP congestion control. In: GLOBECOM 2002 – IEEE global telecommunications conference, Taipei, Taiwan, November 2002, p. 2590–4.
[11] Fu CP, Liew SC. TCP Veno: TCP enhancement for transmission over wireless access networks. IEEE J Select Areas Commun 2003;21(2):216–28.
[12] Snoeren A, Balakrishnan H. TCP connection migration. Internet Draft, IETF, November 2000.
[13] Goff T, Moronski J, Phatak DS, Gupta V. Freeze-TCP: a true end-to-end TCP enhancement mechanism for mobile environments. In: IEEE INFOCOM 2000 – the conference on computer communications, Tel Aviv, Israel, March 2000. p. 1537–45.
[14] Akyildiz IF, Zhang X, Fang J. TCP-peach+: enhancement of TCP-peach for satellite IP networks. IEEE Commun Lett 2002;6(7):303–5.
[15] Fu Z, Greenstein B, Meng X, Lu S. Design and implementation of a TCP-friendly transport protocol for ad hoc wireless networks. In: ICNP 2002 – IEEE international conference on network protocols, Paris, France, November 2002. p. 216–25.
[16] Balakrishnan H, Padmanabhan VN, Seshan S, Katz RH. A comparison of mechanisms for improving TCP performance over wireless links. IEEE/ACM Trans Network 1997;5(6):756–69.
[17] Brown K, Singh S. M-TCP: TCP for mobile cellular networks. ACM Comput Commun Rev 1997;27(5):19–43.
[18] Balakrishnan H, Seshan S, Katz EAR. Improving TCP/IP performance over wireless networks. In: Proceedings of the 1st ACM conference on mobile communications and networking (Mobicom), Berkeley (CA), November 1995. p. 2–11.
[19] Balakrishnan H, Katz R. Explicit loss notification and wireless web performance. In: Proceedings of the IEEE Globecom 1998, Internet mini-conference, Sydney, Australia, November 1998.
[20] Kim D, Toh C-K, Choi Y. TCP-BuS: improving TCP performance in wireless ad-hoc networks. In: ICC 2000 – IEEE international conference on communications, New Orleans, LA, June 2000, p. 1707–13.
[21] Fairhurst G, Wood L. Advice to link designers on link Automatic Repeat reQuest (ARQ). IETF RFC3366, August 2002.
[22] STANAG 5066. Profile for maritime high frequency (HF) radio data communications. NATO, 2000.
[23] STANAG 4538. Technical standards for an automatic radio control system for HF communications links. NATO, 2000.
[24] STANAG 4539. Technical standard for non-hopping HF communications waveforms. NATO, 2000.
[25] Rostami M, Angeja J, Tavares J, Navarro A. HF channel modeling for real time packet transmission. SPIE – Digital wireless communications, Orlando (FL), USA, July 2003. p. 181–91.
[26] Floyd S, Henderson T. The NewReno modification to TCP's fast recovery algorithm. IETF RFC 2582, April 1999.
[27] Marchese M, Perrando M. A packet-switching satellite emulator: a proposal about architecture and implementation. In: ICC 2002 – IEEE international conference on communications, New York City, NY, April 2002, p. 3033–37.
[28] Krishnan R, Allman M, Partridge C, Sterbenz JPG. Explicit transport error notification (ETEN) for error-prone wireless and satellite networks. BBN Technologies, Technical Report No. 8333, March 2002.
[29] Wroclawski J. Specification of the controlled load network element service. IETF RFC2211, September 1997.
[30] Shenker S, Partridge C, Guerin R. Specification of guaranteed quality of service. IETF RFC2212, September 1997.
[31] Nichols K, Blake S, Baker F, Black D. Definition of the differentiated services field (DS Field) in the IPv4 and IPv6 headers. IETF RFC2474, December 1998.
[32] HAL communications: CLOVER-2000 high performance protocol. Available from: http://www.halcomm.com/clvr2k.
[33] Telecommunications: HF Modems, FED-STD-1052, August 1996.
[34] Zorzi M, Rao RR, Milstein LB. Error statistics in data transmission over fading channels. IEEE Trans Commun 1998;46(11):1468–77.
[35] Chiani M, Milani E, Verdone R. A semi-analytical approach for performance evaluation of TCP-IP based mobile radio links. In: GLOBECOM 2000 – IEEE global telecommunications conference, Taipei, Taiwan, November 2000. p. 937–42.
[36] Gudmundson M. Correlation model for shadow fading in mobile radio systems. IEE Electron Lett 1991;27(23).
[37] Tsai S. Markov characterization of the HF channel. IEEE Trans Commun Technol 1969;17(1):24–32.
[38] Fei H, Neeraj K Sharma. The quantitative analysis of TCP congestion control algorithm in third-generation cellular networks based on FSMC loss model and its performance enhancement. In: IEEE INFOCOM 2002 – the conference on computer communications, New York City, NY, June 2002. p. 407–16.
[39] Wang HS, Chang PC. On verifying the first-order Markovian assumption for a Rayleigh fading channel model. IEEE Trans Veh Technol 1996;45(May):353–7.
[40] Mathis M, Mahdavi J. Forward acknowledgment: refining TCP congestion control. ACM SIGCOMM Comput Commun Rev 1996;26(October):281–91.
[41] Nieto JW. An investigation of throughput performance for STANAG 5066 when utilizing STANAG 4539 high data rate HF waveforms. In: MILCOM 2002 – IEEE military communications conference, vol. 21, no. 1, October 2002. p. 120–4.
[42] Jacobson V, Braden R, Borman D. TCP extensions for high performance. IETF, RFC 1323, May 1992.
[43] Braden R. Requirements for Internet hosts – communication layers. IETF RFC 1122, October 1989.
[44] Paxson V, Allman M. Computing TCP's retransmission timer. IETF RFC 2988, November 2000.
[45] Ns-2 network simulator (ver 2). LBL, URL: http://wwwmash.cs.berkeley.edu/ns.

Tomaso de Cola was born in Manosque, France, on April 28, 1977. He received the "Laurea" degree (summa cum laude) in telecommunication engineering from the University of Genoa, Genoa, Italy, in 2001 and the Qualification degree as Professional Engineer in 2002. Since 2002, he has been with the Italian Consortium of Telecommunications (CNIT), University of Genoa Research Unit, where he is a scientist/researcher. His main research activity concerns: TCP/IP protocols, satellite networks, transport protocols for wireless links, interplanetary networks and delay tolerant networks.

Mario Marchese (S′94–M′97–SM′04) was born in Genoa, Italy in 1967. He received the Laurea degree (cum laude) from the University of Genoa, Genoa, Italy, in 1992, his qualification as a Professional Engineer in 1992, and a Ph.D. degree in telecommunications from the University of Genoa in 1996. From 1999 to 2004, he worked with the University of Genoa Research Unit, Italian National Consortium of Telecommunications (CNIT), where he was the head of research. Since 2005, he has been an Associate Professor in the Department of Communication, Computer and Systems Science (DIST), University of Genoa. He is the founder and still technically responsible for the CNIT/DIST Satellite Communication and Networking Laboratory (SCNL), University of Genoa, which contains high-value devices and tools, and implies the management of different units of specialized scientific and technical personnel. He is the Vice-Chair of the IEEE Satellite and Space Communications Technical Committee. He is the author and coauthor of more than 80 scientific papers in international magazines, international conferences and book chapters. His main research interests include satellite networks, transport layer over satellite and wireless networks, quality of service over ATM, IP, and MPLS, and data transport over heterogeneous networks.