Fernando Boavida
Edmundo Monteiro
Saverio Mascolo
Yevgeni Koucheryavy (Eds.)

LNCS 4517

Wired/Wireless Internet Communications

5th International Conference, WWIC 2007
Coimbra, Portugal, May 2007
Proceedings

Springer

Fernando Boavida Edmundo Monteiro
Saverio Mascolo Yevgeni Koucheryavy (Eds.)

Wired/Wireless Internet Communications

5th International Conference, WWIC 2007
Coimbra, Portugal, May 23-25, 2007
Proceedings

Springer

Volume Editors

Fernando Boavida
Edmundo Monteiro
University of Coimbra, Departamento de Engenharia Informatica
Polo II, Pinhal de Marrocos, 3030-290 Coimbra, Portugal
E-mail: {boavida, edmundo}@dei.uc.pt

Saverio Mascolo
Politecnico di Bari, Dipartimento di Elettrotecnica ed Elettronica
Via Orabona 4, 70125 Bari, Italy
E-mail: saverio.mascolo@gmail.com

Yevgeni Koucheryavy
Tampere University of Technology
P.O. Box 553, 33101 Tampere, Finland
E-mail: yk@cs.tut.fi

Library of Congress Control Number: 2007926907

CR Subject Classification (1998): C.2, D.4.4, D.2, H.3.5, H.4, K.6.4

LNCS Sublibrary: SL 5 – Computer Communication Networks and Telecommunications

ISSN 0302-9743
ISBN-10 3-540-72694-2 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-72694-4 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12068319 06/3180 5 4 3 2 1 0

Preface

WWIC 2007 was organized by the University of Coimbra, Portugal, and it was the fifth event of a series of International Conferences on Wired/Wireless Internet Communications, addressing research topics such as the design and evaluation of protocols, the dynamics of the integration, the performance trade-offs, the need for new performance metrics, and cross-layer interactions. Previous events were held in Berne (Switzerland) in 2006, Xanthi (Greece) in 2005, Frankfurt (Germany) in 2004, and Las Vegas (USA) in 2002.

As in 2005 and 2006, WWIC was selected as the official conference by COST Action 290 (Wi-QoST–Traffic and QoS Management in Wireless Multimedia Networks).

WWIC 2007 brought together active and proficient members of the networking community, from both academia and industry, thus contributing to scientific, strategic, and practical advances in the broad and fast-evolving field of wired/wireless Internet communications.

The WWIC 2007 call for papers attracted 257 submissions from 36 different countries in Asia, Australia, Europe, North America, and South America. These were subject to thorough review work by the Program Committee members and additional reviewers. The selection process was finalized in a Technical Program Committee meeting held in Malaga, Spain, on February 15, 2007.

A high-quality selection of 32 papers, organized into 8 single-track technical sessions made up the WWIC 2007 main technical program, which covered transport layer issues, handover and QoS, traffic engineering, audio/video over IP, IEEE 802.11 WLANs, sensor networks, protocols for ad-hoc and mesh networks, and OFDM systems. The technical program was complemented by two keynote speeches, by Henning Schulzrinne (Columbia University, New York, USA) and Nitin Vaidya (University of Illinois at Urbana-Champaign, USA), on New Internet and 4G Wireless Networks, and Multi-Channel Wireless Networks, respectively.

In addition to the main technical program, the two days preceding the conference were dedicated to two workshops: the 1st ERCIM workshop on eMobility (http://www.emobility.unibe.ch/workshop) and the 1st WEIRD workshop on WiMax, Wireless and Mobility (http://workshop.ist-weird.eu/).

We wish to record our appreciation of the efforts of many people in bringing about the WWIC 2007 conference: to all the authors that submitted their papers to the conference, we regret that it was not possible to accept more papers; to the Program Committee and to all associated reviewers for their careful reviews; to our sponsors and supporting institutions; to the University of Malaga, for hosting the TPC meeting. Finally, we would like to thank all the people that helped us at the University of Coimbra and all the volunteers from the Laboratory of Communications and Telematics.

May 2007

Fernando Boavida
Edmundo Monteiro
Saverio Mascolo
Yevgeni Koucheryavy

Organization

Executive Committee

General Chairs Edmundo Monteiro, University of Coimbra, Portugal
Yevgeni Koucheryavy, Tampere University of Technology, Finland

Program Chairs Fernando Boavida, University of Coimbra, Portugal
Saverio Mascolo, Polytechnic of Bari, Italy

Local Organizing Committee

Jorge Sá Silva, University of Coimbra, Portugal
Paulo Simões, University of Coimbra, Portugal
Marilia Curado, University of Coimbra, Portugal
Fernando Velez, University of Beira Interior, Portugal

Steering Committee

Torsten Braun, University of Bern, Switzerland
Georg Carle, University of Tubingen, Germany
Giovanni Giambene, University of Siena, Italy
Yevgeni Koucheryavy, Tampere University of Technology, Finland
Peter Langendoerfer, IHP Microelectronics, Germany
Ibrahim Matta, Boston University, USA
Vassilis Tsaoussidis, Demokritos University, Greece
Nitin Vaidya, University of Illinois, USA

Supporting and Sponsoring Organizations

Project COST290
Cisco Europe
Alvarion
Advanced Resources
Critical Software
Fundação Calouste Gulbenkian
Câmara Municipal de Coimbra
University of Coimbra

Program Committee

Rui Aguiar, University of Aveiro, Portugal
Bengt Ahlgren, SICS, Sweden
Ozgur B. Akan, Middle East Technical University, Turkey
Khalid Al-Begain, University of Glamorgan, UK
Manuel Alvarez-Campana, UPMadrid, Spain
Chadi Barakat, INRIA, France
Carlos Bernardos, Universidad Carlos III de Madrid, Spain
Bharat Bhargava, Purdue University, USA
Fernando Boavida, University of Coimbra, Portugal
Torsten Braun, University of Bern, Switzerland
Wojciech Burakowski, Warsaw University of Techology, Poland
Maria Calderon, University Carlos III de Madrid, Spain
Georg Carle, University of Tübingen, Germany
Hermann de Meer, University of Passau, Germany
Ruy de Oliveira, CEFET-MT, Brazil
Michel Diaz, LAAS-CNRS, France
Sonia Fahmy, Purdue University, USA
Giovanni Giambene, University of Siena, Italy
Geert Heijenk, University of Twente, The Netherlands
Marc Heissenbüttel, Swisscom Mobile, Switzerland
Markus Hofmann, Columbia University, USA
Andreas Kassler, Karlstads University, Sweden
Ibrahim Khalil, RMIT University, Australia
Daniel Kofman, ENST, France
Yevgeni Koucheryavy, Tampere University of Technology, Finland
Rolf Kraemer, IHP Microelectronics, Germany
Peter Langendoerfer, IHP Microelectronics, Germany
Remco Litjens, TNO, The Netherlands
Pascal Lorenz, University of Haute Alsace, France
Christian Maihöfer, DaimlerChrysler, Germany
Saverio Mascolo, Politecnico di Bari, Italy
Paulo Mendes, DoCoMo Euro-Labs, Germany
Ingrid Moerman, Ghent University, Belgium
Dmitri Moltchanov, Tampere University of Technology, Finland
Edmundo Monteiro, University of Coimbra, Portugal
Liam Murphy, University College Dublin, Ireland
Marc Necker, University of Stuttgart, Germany
Guevara Noubir, Northeastern University, USA
Philippe Owezarski, LAAS-CNRS, France
George Pavlou, University of Surrey, UK
George Polyzos, AUEB, Greece
Utz Roedig, University of Lancaster, UK
Alexandre Santos, University of Minho, Portugal
Jochen Schiller, Free University Berlin, Germany
Andrew Scott, Lancaster University, UK

Patrick Sénac, ENSICA, France
Dimitrios Serpanos, University of Patras, Greece
Jorge Silva, University of Coimbra, Portugal
Vasilios Siris, University of Crete / ICS-FORTH, Greece
Dirk Staehle, University of Würzburg, Germany
Burkhard Stiller, University of Zurich / ETH Zurich, Switzerland
Ivan Stojmenovic, University of Ottawa, Canada
Phuoc Tran-Gia, University of Wuerzburg, Germany
Vassilis Tsaoussidis, Demokritos University, Greece
J.L. van den Berg, TNO / University of Twente, The Netherlands
Fernando Velez, University of Beira Interior, Portugal
Giorgio Ventre, University of Naples, Italy
Thiemo Voigt, SICS, Sweden
Miki Yamamoto, Kansai University, Japan
Chi Zhang, Florida International University, USA
Martina Zitterbart, University of Karlsruhe, Germany

Additional Reviewers

Imad Aad
Janet Adams
Mifdaoui Ahlem
Ivano Alocci
Paul Amer
Mina Amin
Houyou Amine
Berl Andreas
Omar Ashagi
Baris Atakan
Urtzi Ayesta
Alper Bereketli
Giuseppe Bianchi
Andreas Binzenhoefer
Erik-Oliver Blaß
Roland Bless
Thomas Bohnert
Nafeesa Bohra
Alister Burr
Orlando Cabral
Tiago Camilo
Maxweel Carmo
Brian Carrig
Ed Casey
Tim Casey
Marcel Castro
Francisco Cercas
Eduardo Cerqueira
Wei Chai
Roman Chertov
Luca Cicco
Paolo Costa
Marilia Curado
Saumitra Das
Daniel Dietterle
Stylianos Dimitriou
Roman Dunaytsev
Barbara Emmert
Ernesto Exposito
Sílvia Farraposo
Pedro Ferreira
Andreas Festag
John Fitzpatrick
Frances Frances
Bob Gaglianello
Thomas Gamer
Thierry Gayraud
Vasken Genc
Stelios Georgoulas
Wagenknecht Gerald
O. Gonzalez-Duque
Alfredo Grieco
Cataldo Guaragnella
Karim Guennoun
Antonis Hadjiantonis
Jarmo Harju
Robert Henjes
Kin-Hon Ho
Hans-Jochim Hof
Richard Holzer
Tobias Hossfeld
Michael Howarth
Michael Howarth
Christian Hübsch
Bernhard Hurler
Talha Isik
Dedinski Ivan
Aravind Iyer
Jakub Jakubiak
Oberender Jens
Ljupco Jorguseski
Guy Juanole
Eleni Kamateri
Alexander Klein
Gerald Koch
T. Kontonikolas
Jukka Koskinen
Andreas Kuntz
Jerome Lacan
Gabriela Leao
António Lebres
Emmanuel Lil
Luis Loyola
Andreas Maeder

Maurizio Maggiora
Damien Magoni
Apostolos Malatras
Lefteris Mamatas
Zoubir Mammeri
Dimitri Marandin
Brogle Marc
Portman Marius
Anwander Markus
Wälchli Markus
Wulff Markus
Antonio Marques
Rüdiger Martin
I. Martinez-Yelmo
Saverio Mascolo
Volkamer Melanie
Abdelhamid Mellouk
Michael Menth
Dmitri Moltchanov
Andreas Müller
Christian Müller
John Murphy
Sean Murphy
Marc Necker
Augusto Neto
Simon Oechsner
Olga Ormond
Evgeni Osipov
Vittorio Palmisano
Manuel Pedro
Tanguy Perennou
P. Papadimitriou
Giorgos Papastergiou
Wuechner Patrick
Steffen Peter
Krzysztof Piotrowski
Rastin Pries
Ioannis Psaras
Andre Rodrigues
Christian Rohner
Frank Roijers
Sabyasachi Roy
Christos Samaras
Venkatesh Saranga
Julian Satran
Michael Scharf
Isaac Seoane
Pablo Serrano
Bilhanan Silverajan
Paulo Simoes
Siva Sivavakeesar
Rute Sofia
Dirk Staehle
Mikael Sternad
Werner Teich
Staub Thomas
Christina Thorpe
Ageliki Tsioliaridou
Francesco Vacirca
Luc Vandendorpe
Luis Veloso
Lars Völker
Nicolas Wambeke
N. Wang
Ning Wang
Ning Wang
Wang Wenbo
Joerg Widmer
Yan Wu
Santiago Zapata
Xuan Zhong
Andre Zimmermann

Table of Contents

Transport Layer Issues

Handover and QoS

QoS and Traffic Engineering

Audio/Video over IP

IEEE 802.11 WLANs

Sensor Networks and Location-Aware Systems

Protocols for Ad Hoc and Mesh Networks

Efficient Spectrum Use

TCP Contention Control: A Cross Layer Approach to Improve TCP Performance in Multihop Ad Hoc Networks

Ehsan Hamadani and Veselin Rakocevic

Information Engineering Research Centre
School of Engineering and Mathematical Sciences
City University, London EC1V 0HB, UK
{E.hamadani, V.rakocevic}@city.ac.uk

Abstract. It is well known that one of the critical sources of TCP poor performance in multihop ad hoc networks lies in the TCP window mechanism that controls the amount of traffic sent into the network. In this paper, we propose a novel cross layer solution called "TCP Contention Control" that dynamically adjusts the amount of outstanding data in the network based on the level of contention experienced by packets as well as the throughput achieved by connections. Our simulation results show TCP Contention Control can drastically improve TCP performance over 802.11 multihop ad hoc networks.

Keywords: Contention, Multiple ad hoc Networks, TCP Congestion Control.

1 Introduction

Multihop ad hoc networks are autonomous systems of mobile devices connected by wireless links without the use of any pre-existing network infrastructure or centralized administration. During recent years ad-hoc networks have attracted considerable research interest thanks to their easy deployment, maintenance and application variety. To enable seamless integration of ad hoc networks with the Internet (for instance in ubiquitous computing applications), TCP seems to be the natural choice for users of ad hoc networks that want to communicate reliably with each other and with the Internet. However, as shown in many papers (e.g. [1,2]), TCP exhibits serious performance issues such as low and unstable throughput, high end-to-end delay and high jitter. This is because most TCP parameters have been carefully optimized based on assumptions that are specific to wired networks. For instance, since bit error rates are very low in wired networks, nearly all TCP versions assume that packet losses are due to congestion and therefore invoke their congestion control mechanism in response to such losses. On the other hand, because of wireless medium characteristic and multihop nature of ad hoc networks, such networks exhibit a richer set of packet losses, including medium access contention drops, random channel errors and route failure where in practice each are required to be addressed differently. In particular, as we have shown in [3], when TCP runs over 802.11 MAC in multihop ad hoc networks, frequent channel contention losses at the MAC layer are wrongly

F. Boavida et al. (Eds.): WWIC 2007, LNCS 4517, pp. 1–16, 2007.

perceived as congestion and are recovered through TCP congestion control algorithm. This phenomenon severely degrades the performance of TCP as it leads to unnecessary TCP retransmission, unstable and low throughput, unfairness, high end-to-end delay, and high jitter. As we concluded there, a high percentage of MAC layer contention drops can be eliminated by decreasing the amount of traffic load in the network. This observation in addition to the results derived in [2,4], motivated us to propose a novel cross layer solution called "*TCP Contention Control*" that will be used in conjunction with TCP Congestion Control algorithm. In simple words, when TCP Contention and TCP Congestion Control are used together, the amount of outstanding data in the network is tuned based on the level of contention and channel utilization as well as level of congestion in the network. More precisely, while TCP Congestion Control adjusts the TCP transmission rate to avoid creating congestion in the intermediate network buffers, TCP Contention Control adjusts the transmission rate to minimize the level of unnecessary contention in the intermediate nodes. Therefore, when two algorithms are used jointly in the network, the TCP sender sets its transmission rate not merely based on the amount of congestion in the network and available buffer size at the receiver but also by the level of medium contention in intermediate nodes along the data connection. Our simulation results over a variety of scenarios confirm that the proposed scheme can dramatically improve the TCP performance in multihop networks in addition to substantial decrement in number of packet retransmission in the 802.11 link layer.

The rest of the paper is organized as follows. In section 2, we will give a brief overview of TCP congestion control algorithm. In section 3, the main problem of TCP congestion control in ad hoc networks are discussed in fine details. Then based on the drawn facts, we propose the new cross layer solution in section 4, which aims to improve TCP performance in multihop ad hoc networks. This is followed by the simulation model and the key results obtained by simulating the proposed model against the default TCP protocol in section 5. Finally, in section 6, we conclude the paper with some outlines towards future work.

2 TCP Congestion Control

TCP Congestion Control was added to TCP in 1987 and was standardized in RFC2001 [5] and then updated in RFC2581 [6]. In a broad sense, the goal of the congestion control mechanism is to prevent congestion in intermediate router's buffer by dynamically limiting the amount of data sent into the network by each connection. To estimate the number of packets that can be in transit without causing congestion, TCP maintains a congestion window (*cwnd*) that is calculated by the sender as follows: when a connection starts or a timeout occurs, *slow start* is performed where at the start of this phase, the *cwnd* is set to one MSS (Maximum Segment Size). Then the *cwnd* is increased by one MSS for each acknowledgment for the new data that is received. This results in doubling the window size after each window worth of data is acknowledged. Once *cwnd* reaches a certain threshold (called the slow start threshold, *ssthresh*), the connection moves into the *congestion avoidance* phase. Ideally, a TCP connection operating in this phase puts a new packet in the network only after an old one leaves. The TCP in congestion avoidance also probes the network for resources that might have become available by continuously increasing the window, albeit at a lower rate than in slow start. In the start of this phase, TCP gently probes the available bandwidth by increasing the *cwnd* by one packet in every round trip time (Additive Increase). During this time if the TCP detects packet loss through duplicate acknowledgments, it retransmit the packet (fast retransmit) and decreases the *cwnd* by a factor of two (Multiplicative Decrease) or it goes to slow start according to the TCP version used. Alternatively, if the sender does not receive the acknowledgment within retransmission time out (RTO), it goes to slow start and drops its window to one MSS. In both occasions, the *ssthresh* is set to half the value of *cwnd* at the time of loss.

After calculating the current value of *cwnd*, the effective limit on outstanding data (i.e. flight size), known as '*send window*' *(swnd)*, is set as the minimum of the *cwnd* and available receiver window *(rwnd)*. The *rwnd* is the amount of available buffer size in the receiver side and is taken into account in order to avoid buffer overflow at the receiver by a fast sender (flow control). Therefore:

$$swnd = \min\{rwnd, cwnd\} \tag{1}$$

3 Problem Description

As we mentioned in section 2, the performance of TCP directly depends on the *swnd*. It is well known that the optimal value for swnd should be proportional to the bandwidth-delay product of the entire path of the data flow [4]. It is important to note that the excess of this threshold does not bring any additional performance enhancement, but only leads to increased buffer size in intermediate nodes along the connection. As shown in [1,7,8], the bandwidth-delay product of a TCP connection over multihop 802.11 networks tends to be very small. This is mainly because in 802.11, the number of packets in flight is limited by the per-hop acknowledgements at the MAC layer. Such property is clearly quite different from wireline networks, where multiple packets can be pushed into a pipe back-to-back without waiting for the first packet to reach the other end of the link. Therefore, as compared with that of wired networks, ad hoc networks running on top of 802.11 MAC, have much smaller bandwidth-delay product. However, as shown in [2], TCP grows its congestion window far beyond its optimal value and overestimates the available bandwidth-delay product. To get a better understanding of TCP overestimation of available bandwidth-delay product in ad hoc networks, consider a simple scenario in fig.1 where all nodes can only access their direct neighbors. Here a TCP connection is running from node A to E and all nodes have at least one packet to send in the forward direction.

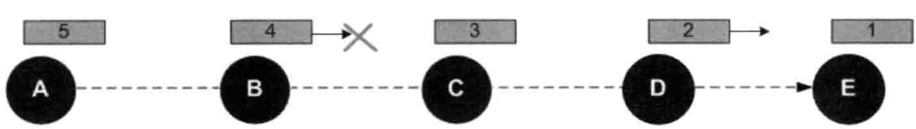

Fig. 1. 4 hop chain topology

Let us assume nodes B and D initially win the channel access and start to transmit their data into the network at the same time. Soon after both stations start transmitting their data, the packet from B to C is collided with the interference caused by D→E transmission. Following this case, node A is very likely to win the access to the channel and starts transmitting several consecutive packets towards B before releasing the channel [9]. Meanwhile, since B is unable to access the channel it buffers the new packets in addition to packet(s) already in its buffer and starts building up its queue (figure 2).

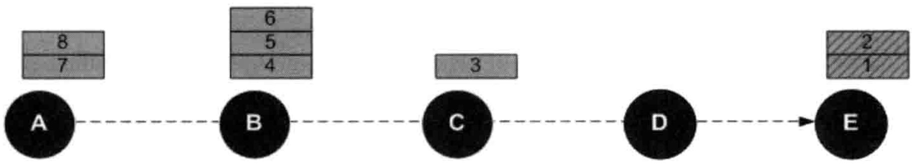

Fig. 2. Queue build up in network

This results in an artificial increase of the RTT delay measured by the sender as node B now becomes the bottleneck of the path. Such situation leads to an overestimate of the length of available data pipe and therefore an increase of the TCP congestion window and hence network overload in the next RTT. To have a better understanding of the effect of network overload on the TCP performance, fig.3 summarizes the chain of actions that occur following a network overload. In particular, increasing the network overload causes higher amount of contention among nodes as all of them try to access the channel (stage 2). On the other hand, when the level of contention goes up, more packets need to be retransmitted as the probability of collision increases with the increasing level of contention (stage 3). This in turn introduces extra network overload and therefore closing the inner part of the cycle (stage 1→stage2→sage3→stage1).

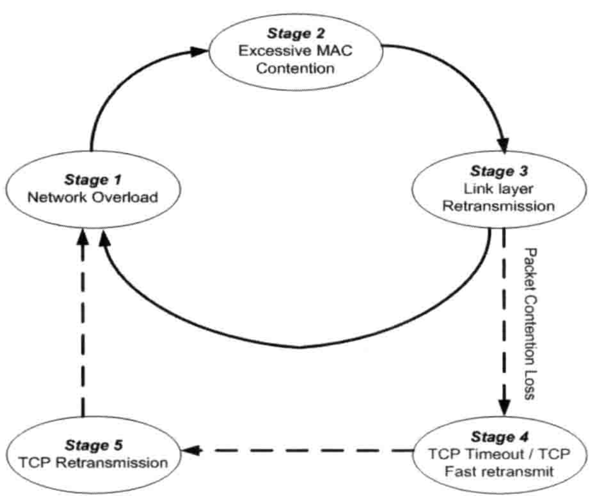

Fig. 3. TCP Instability cycle

This cycle is continued until one or more nodes cannot reach its adjacent node within a limited tries (specified by MAC_Retry_Limit in 802.11 MAC standard [10]) and drop the packet (packet contention loss). This packet loss is then recovered by the TCP sender either through TCP fast retransmit or through TCP timeout (stage 4). In both cases, TCP drops its congestion window resulting in a sharp drop in number of newly injected packets to the network (stage 5) and therefore giving the network the opportunity to recover. However, soon after TCP restarts, it creates network overload again by overestimating the available bandwidth-delay product of the path, and the cycle repeats.

Fig.4 shows the change of *cwnd* and the instances of TCP retransmission in a 4 hop chain topology as shown in figure 1 using 802.11 MAC. Here, the only cause of packet drop in the network has been set to contention losses to verify the problem of TCP and link layer interaction in ad hoc networks. The results fully support the above argument and confirm that TCP behavior towards overloading the network causes extensive packet contention drops in the link layer. These packet drops are wrongly perceived as congestion by the TCP and result into false trigger of TCP congestion control algorithm and frequent TCP packet retransmissions.

This observation is also confirmed in many studies such as [1,2,11] by showing that TCP with a small congestion window (e.g., 1 or 2) tends to outperform TCP with a large congestion window in 802.11 multihop networks. To enforce the congestion window to a small value, the authors in [4] showed that the bandwidth-delay product of ad hoc networks is limited to round trip per hop count (RTHC). They then refine this upper bound based on the 802.11 MAC layer protocol, and show that in a chain topology, a tighter upper bound of approximately 1/5 of the round trip hop count of the path outperforms in comparison to default TCP. The authors in [2] impose a hard limit of 1/4 of chain length based on transmission interference in 802.11.

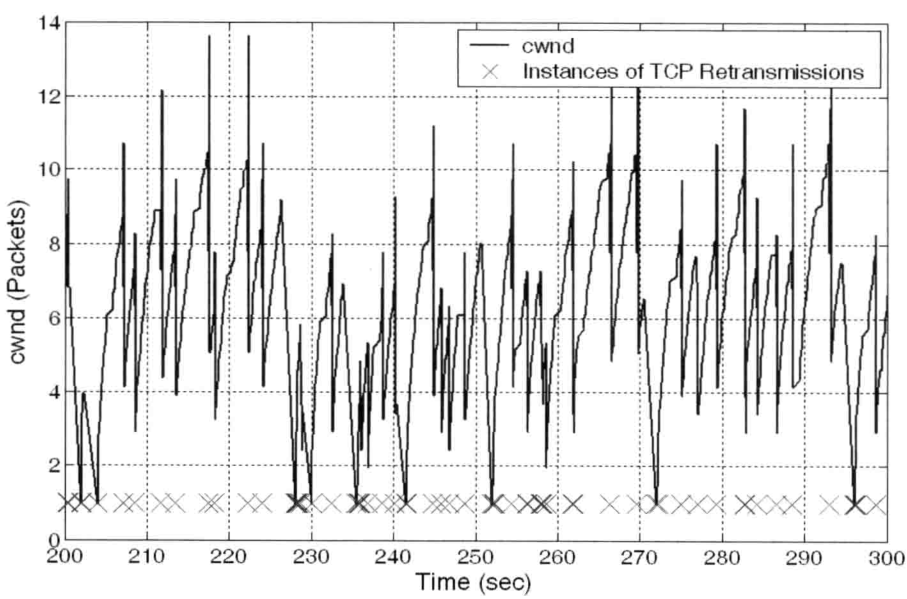

Fig. 4. Change of *cwnd* and the instances of TCP retransmission in a 4 hop chain topology