Detecting TCP Traffic Dynamical Changes in UMTS Networks

Ioannis Vasalos[1], Averkios Vasalos[2], and Heung-Gyoon Ryu[3]

[1] Newcastle University, UK
[2] University of Birmingham, UK
[3] Chungbuk National University, Korea
vasaloseng@googlemail.com, ecomm@cbu.ac.kr

Abstract. This paper presents a study of the methodology for the detection of congestion epochs and data transmission dynamical changes over mobile connections in the Universal Mobile Telecommunications System (UMTS) network. Dynamical changes in the data traffic occur in the Transmission Control Protocol (TCP), which is the protocol that regulates the data transmission inside the network. Using the concept of the recently introduced natural complexity measure of the Permutation Entropy (PE), the dynamical characteristics of the TCP inside the UMTS network are studied. It is shown that the PE can be effectively used to detect congestion epochs and the timely change in the dynamical pattern of the data transmission as regulated by the TCP. This is of crucial importance in order to prevent extended congestion epochs and the deterioration of the Quality of Service (QoS) in mobile networks.

Keywords: TCP, UMTS, Permutation Entropy, Data Traffic Dynamics, Network Congestion.

1 Introduction

Evolution in the field of communications has seen a phenomenal development in contemporary networks. Nowadays networks with different functional attributes and performance limits such as wired, wireless and mobile networks interoperate and form an extremely complex infrastructure. The emergence of this large scale network of networks has seen the formation of the Internet, which is one of the most complicated systems ever created. In particular UMTS networks, which are the latest generation of cellular mobile networks, amalgamate a wireless and wired infrastructure [1] in order to offer data services to the network's users. This heterogeneous network comprised by subnetworks with different fundamental architectures and different performance requirements is expected to function seamlessly allowing the smooth data traffic flow from the source to the destination node in order to offer users an increased QoS.

Teletraffic studies in wired communication networks [2] and mobile communication networks [3, 4] showed that during peak traffic time, chaos and

I. Tomkos et al. (Eds.): BROADNETS 2010, LNICST 66, pp. 242–255, 2012.

persistent oscillations appear through the network. The nature of these effects can be targeted on the Additive Increase Multiplicative Decrease (AIMD) dynamics of the Transmission Control Protocol (TCP) that networks use in order to offer a connection oriented reliable byte stream service. Crucially Veres and Boda [5] were the first to demonstrate how the TCP congestion control can show chaotic behaviour. Complementary research by Rao et al [6], analytically presented that TCP can generate bounded trajectories with a complicated attractor and its dynamics embed a tent like map for the update of the packet transmission window size, which generates chaotic dynamics. Relative research in mobile UMTS networks, as it is presented in [3, 4], showed that under increased traffic load in moderate radio conditions the TCP data traffic dynamics of Mobile Stations (MS) range from periodically stable to chaotic with a direct impact on the QoS of service of the network. Such impact is expressed by significantly decreased data throughputs, unfairness in resource allocation to the network users and increased delays in data application performance. Therefore timely detection of dynamical changes and of the appearance of chaotic dynamics in the UMTS network traffic profile is a crucial problem in order to maintain the network's integrity and guarantee the QoS as perceived from the users.

Research focused on the detection of anomalies of the network dynamics, among others, is presented in [7, 8] where space-time characteristics of congestion in large networks and the system behaviour relative to the levels of congestion is studied. A macroscopic level method is established to simulate a network of thousands of on-off traffic sources creating congestion in various locations of the network. The research is focused on how to locate network bottlenecks and how to detect possible Distributed Denial of Service (DDoS) attacks by studying the spatio-temporal traffic dynamics. Furthermore in [9] the phase transition between congested and non-congested phases in the Internet is studied to quantify the probabilities of network congestion according to the dynamic complexity of the Internet traffic. In [10] probability distributions of several network long-term traffic measures such as the source IP address, the destination IP address and data flow size are studied to create a unique fingerprint of the traffic profile of the network under consideration. In case this profile changes, as measured by the Information Entropy, anomalies in the network behaviour can be detected.

Methods presented above for the detection of dynamical changes in the network traffic regard macroscopic network models and require long-term observation of network dynamics. Therefore according to our literature review there is an open question for algorithms that can detect fast and efficiently real-time dynamical changes and anomalies in the traffic profile of communication networks. To answer this question in this paper the methodology of the Permutation Entropy (PE) is considered. In the seminal paper of Band and Pompe [11], the concept of the PE is introduced as a complexity measure for time series analysis. The methodology is verified in [12] to be a conceptually simple and computationally very fast algorithm for the quantification of the complexity of a time series produced by a complex system and the detection of dynamical changes in the time series of the system.

Following the aforementioned research [3, 4] on the chaotic dynamics developed in the traffic profile of the TCP traffic flow inside the UMTS network, research in this paper is focused on the quantification of the complexity and the detection of dynamical changes of the TCP traffic profile time series in the UMTS network. We propose the utilisation of the PE to detect dynamical changes and network anomalies in the TCP traffic flow. The framework of the methodology is based on the hypothesis that when normal traffic with fairness and QoS exist inside the network, the traffic profile should have a certain measure of complexity as determined by the PE. Consequently, possible variations of the PE could detect dynamical changes and anomalies inside the network. Such information along with the impact of this behaviour on the QoS of the UMTS network can be used as a real-time, computationally fast algorithm to detect anomalies in the network and provision the network in order to guarantee the QoS and ensure the network's functionality.

The remainder of this paper is organized as follows. Section 2 presents the TCP algorithm and its dynamical behaviour analysis. Additionally the motivation of the research on the UMTS network as the network currently used globally for mobile communications is explained. Section 3 introduces the concept of PE as a measure of time series complexity and Section 4 presents the network simulation model. Section 5 presents the results of the PE for characterisation of the complexity and the detection of dynamical changes in the TCP traffic flow time series along with the impact of dynamic complexity on the QoS of the network. Finally Section 6 concludes the paper.

2 TCP Dynamical Behaviour

TCP is the network protocol that provides a connection-oriented, reliable byte stream service over the Internet. Reliability in the TCP connection between the sender and the receiver in the network is guaranteed through a set of rules that for each packet sent by the sender the receiver acknowledges each packet sent. When packets are lost in the network the receiver using timer and packet sequence number information notifies the sender that packets that have not arrived on time are lost in the network. Fundamentally the TCP relies on two mechanisms to regulate the flow of data: flow control and congestion control. The flow control ensures that the sender does not overflow the receiver buffer and the congestion control ascertains that the sender does not congest the network by exceeding the connection bandwidth and the buffer space at the network's queues. Assuming large receiver buffers, the dynamics of TCP traffic are generated by the congestion control mechanism. Fundamentally the congestion control mechanism uses strict deterministic rules in order to regulate the transmission rate of the sender by controlling the congestion window $w(t)$. This is the maximum number of unacknowledged packets the sender can send to the destination before it stops and waits for an acknowledgment. From the congestion window, the throughput s of each TCP is specified by the number of packets over the Round Trip Time (RTT).

$$s(t) = \frac{w(t)}{RTT} \quad . \tag{1}$$

RTT is the time from when a packet was sent from the sender until its acknowledgment was received. If a network is assumed with a total of N nodes and $n \in N$ number of queues with queue size q and $n \in N$ number of relative links with bandwidth B and propagation delay D, then the RTT is defined as the sum of propagation delays D and queuing delays $\frac{q^n}{B^n}$.

$$RTT = \sum_{n \in N} D^n + \frac{q^n}{B^n} \ . \tag{2}$$

The strict deterministic rules that define the data flow and the TCP dynamical behavior are categorized into four main dynamical modes. These are *slow-start*, *congestion-avoidance*, *Fast Retransmit/ Fast Recovery (FR/FR)* and *timeout* modes and form the Additive Increase Multiplicative Decrease (AIMD) congestion control mechanism. The detailed rules for each dynamical mode vary along with the version of the TCP used. In this paper the dynamical behavior of the TCP SACK protocol is studied. TCP SACK has being tested over various wireless networks [13, 14] to achieve the best throughput over wireless links. So it is the preferred version of the protocol to be used in the UMTS network modeled in this paper. A brief presentation of the four dynamical models is presented.

1) *Slow-start* mode: When a connection is initiated the TCP enters *slow-start* mode and the congestion window is increased very rapidly with an exponential rate being multiplied by 2 every RTT. This is modelled by

$$\frac{d}{dt} w(t) = \frac{\log 2}{RTT} w(t) \ . \tag{3}$$

2) *Congestion-avoidance* mode: When the congestion window reaches the advertised window of the receiver the system switches to congestion-avoidance mode. During congestion control the congestion window size increases linearly with an increase equal to the packet-size *L* every RTT

$$\frac{d}{dt} w(t) = \frac{L}{RTT} \ . \tag{4}$$

Slow start and congestion avoidance modes last until a drop or a timeout are detected. Detection of a drop leads the system to *FR/FR* mode, whereas the detection of a timeout leads the system to timeout mode.

3) *Fast Retransmit/Fast Recovery (FR/FR)* mode: When congestion occurs and packets are dropped, TCP SACK *FR/FR* mode immediately decreases the congestion window by half and immediately retransmits the missing segments. Once acknowledgements for the retransmitted packets arrive the sender leaves fast recovery mode and switches again to congestion avoidance mode. It has to be noted, that for our system, since selective acknowledgement is used in the protocol, for every

window of packets dropped TCP enters only one time fast recovery, since it knows exactly which packets are dropped. Considering w^{CA} as the last value of the congestion window in the congestion avoidance mode, the evolution of the congestion window is expressed as

$$w(t) = \frac{w^{CA}}{2} \; . \tag{5}$$

4) *Timeout mode*: Finally a timeout occurs when the retransmission timer of a packet sent expires. The congestion window is reset to 1, and *slow-start* as described before repeats.

In general apart from the initiation of the connection, TCP mainly operates between *congestion-avoidance* and *FR/FR* modes. *Timeout* and *slow-start* are used when severe packet losses occur in a network. The dynamics of the TCP algorithm can be described by two main dynamical modes. Mode 1 *congestion-avoidance* and mode 2 *FR/FR*. In mode 1 TCP starts sending packets with a rate proportional to the reception of acknowledgments from the receiver. Mode 2 is entered when packet loss occurs and the congestion window is halved. The TCP dynamics are due to the "gluing together" of mode 1 and mode 2 [6]. These are directly related to the level of traffic congestion inside the network, expressed by the number of TCP traffic flows sending traffic through the network and the packets buffered at the queues inside the network.

From [15] it is known that chaos is the aperiodic, long-term behaviour of a bounded system that exhibits sensitive dependence to initial conditions. As it is analytically and experimentally studied for a wired network in [6] and for a UMTS network in [3, 4], in case of stable RTT delays, dynamics alternate between congestion avoidance and *fast retransmit/fast recovery* mode in a stable cycle and hence create periodic trajectories. However when the traffic load increases, queues get congested much faster and multiple losses occur that makes the RTT exhibit large variations. When the RTT loses stability due to high levels of lost packets in the network the trajectories in one mode will enter the other generating bounded trajectories, which create aperiodic routes, which under circumstances can be chaotic.

2.1 TCP over UMTS

In mobile networks the problem of the complexity of the network dynamics is aggravated by the wireless nature of the system. As analysed in [3, 4] instability in the network is further invoked by the variable restricted wireless bandwidth, the user mobility and the interference of the physical environment. These inherent characteristics of the mobile network cause increased round trip time and increased packet loss probability, which is well-known to severely affect the throughput performance of the TCP. In UMTS the Radio Link Control (RLC) protocol is used to recover lost packets from the radio access network by using an Automatic Repeat Request (ARQ) algorithm. Although this minimises packet loss probability, it increases RTT because of the link layer retransmission of lost frames. Since the TCP protocol remains unchanged for wired and wireless applications, these fundamental parameters are not implemented in the protocol. Under such interference it is possible that TCP measures and adapts to congestion erroneously, inserting more instability to the network dynamics due to nonlinear chaotic behaviour and unpredictability due to stochasticity from packet losses in the whole system.

For the reasons discussed above it is very crucial to study the TCP behaviour inside the wireless system in order to quantify the complexity of the dynamics developed in the TCP traffic profile. Such knowledge would allow the creation of congestion control mechanisms able to detect changes in the network dynamics and act accordingly to control and suppress such anomalies inside the UMTS. In this paper we focus on the implementation of the PE for the quantification and detection of dynamical changes and postpone the discussion of mechanisms for the control of such phenomena for a future study.

3 Permutation Entropy Definition

Permutation entropy is introduced in the paper of Band and Pompe [11] as the methodology of transforming a time series of a dynamical system into a symbolic sequence. The method relies on the embedding theorem [15], which states that when a dynamical system has an attractor with box-counting dimension D_f the complete dynamical and topological properties of the original system can be studied from a single scalar time series using an embedding dimension $m > 2D_f$. In this case the TCP protocol is considered as the dynamical system, whose dynamical properties through the PE measurements will be studied from the single scalar time series of the congestion window. The methodology as presented in [11, 12] includes the embedding of a scalar time series creating vectors of the form $X_i = (x_i, x_{i+L}, \dots x_{i+(m-1)L})$ where m is the embedding dimension, L is the delay time and $(m-1)L$ is the embedding window. In the scalar time series we consider all the $m!$ permutations. In order to create a symbolic sequence for any vector each real value of the X_i vector is rearranged in an increasing order and it is assigned to a symbolical corresponding j value with increasing order where $j = j_0, j_1, j_2, \dots, j_{m-1}$. Finally a new vector X_j is created where each symbolical value of j is rearranged in the position of the original X_i vector. Hence as it is explained in [12] any vector X_i is uniquely mapped onto $j_0, j_1, j_2 \cdots j_{m-1}$, which is one of the $m!$ permutations of m distinct symbolical values $(0,1,2,\dots,m-1)$. When each such permutation is considered as a symbol, the reconstructed trajectory in the m-dimensional space is represented by a symbol sequence. The Permutation Entropy (PE) is defined as a measure of the probabilities of the appearance of the permutations for each of the different $m!$ permutations. Considering the probability distribution for the appearance of distinct symbols as $P_1, P_2, \dots P_l$ where $l \le m$. The Permutation Entropy is defined as [11, 12]

$$H_m = -\sum_{j=1}^{l} P_j \ln P_j \ . \tag{6}$$

For example let us consider the embedded time series x= 10, 5, 7, 12, 4, 8, creating vectors X_i with three values. The vectors with three consecutive values that are created are: (10, 5, 7), (5, 7, 12), (7, 12, 4), (12, 4, 8). Following the methodology as described above, these correspond to the symbolic X_j vectors: (2, 0, 1), (0, 1, 2), (1, 2, 0), (2, 0, 1). Hence from eq. 6 the PE of order $m = 3$ is

$$H_3 = -\frac{2}{4}\ln\left(\frac{2}{4}\right) - 2\cdot\frac{1}{4}\ln\left(\frac{1}{4}\right) \approx 1.0397 \ .$$

It is clear that for a completely random system where the time series of the system will be an Independent and Identically Distributed (i.i.d) sequence the probability distribution for any symbol would be $P_j = 1/m!$. Then for a sequence of values the PE would be $H_m = \ln(m!)$ which is the maximum value H_m can acquire. Accordingly when a time sequence has only a certain value sequence or a constantly decreasing or increasing time series the PE will be $H_m = 0$. In order to define a specific upper bound, H_m is normalized by $\ln(m!)$. Hence the range of values of the normalized Permutation Entropy H_m is $0 \leq H_m \leq 1$. This is verified in [11, 12] where application of the PE to an i.i.d sequence measured $H_m = 1$. Accordingly application of the PE to the chaotic time series data of the transient logistic chaotic map and the transient Lorenz chaotic attractor measured $H_m < 1$. In general the PE is a measure of the dynamics of the time series, and the more irregular the time series is, the larger value the PE acquires.

4 Methodology

In order to evaluate the dynamics of the TCP protocol inside the UMTS network, we select to measure each TCP's congestion window $w(t)$ as a representative of the protocol behaviour. The congestion window is directly related to the nonlinear equations that govern the TCP data rate and congestion avoidance. The network model is built using the OPNET modeller and consists of a typical UMTS network architecture [1]. The simulation parameters settings for the network model scenario are detailed in Table.1. The most important parameters for the research objectives include settings for the TCP algorithm, the radio-air interface, the QoS configuration and the RLC channel.

We focus our study on the scenario where *2* Mobile Stations (MSs) are considered to send information on the uplink using the TCP SACK, sending data files with the File Transfer Protocol (FTP) to an FTP server in the wired part of the network. In order to simulate a network with increased traffic load a drop-tail router inside the network is considered as a bottleneck in the network, served by a 200 kbps throughput link with a bottleneck queue of 40 packets. The simulations are run for 800 seconds.

Table 1. Simulation parameters settings values

Network Parameter	Setting
TCP	
TCP Flavour	SACK
Maximum TCP/IP Packet Size	1500 bytes
Maximum Received Window	65535 bytes
Initial Window	1
Radio-Air Interface	
Cell Pathloss Model	Outdoor to Indoor Pedestrian Environment
Shadow Fading Standard Deviation	10 dB
QoS Configuration	
QoS Traffic Class	Background class
Maximum Bit Rate (Uplink/Downlink)	128 kbps
RLC Channel Configuration	
RLC Mode (Uplink/Downlink)	RLC Acknowledged Mode
Packet Unit (PU) Segmentation	Yes
In-Sequence Delivery of PU	No
Transmitting/Receiving Window	32 PU

The objective of the simulation is to test the network under different levels of congestion inside the network, which have proportional levels of packet drops. This is because we want to study how the PE measures the complexity of the traffic profile time series relative to the occurring congestion; and if it can detect dynamical changes in the traffic profile due to the different levels of the occurring congestion. Therefore the router is considered to serve on-off background traffic from a source sending traffic through the bottleneck buffer using the UDP protocol. As shown in [16], use of 10% UDP traffic is adequate in order to ensure random packet loss in the router queues. The experimental procedure consists of increasing the level of background traffic from 40 to 80 kbps served by the router and examining the dynamic behaviour of the congestion window of each mobile station's TCP as well as QoS measurements of MS throughput inside the UMTS network.

In order to acquire the PE measurement from the time series of the TCP congestion window, initially the 800 seconds time series is partitioned into small vectors of data. The data blocks are embedded using the optimal embedding dimension m and delay time L, which are calculated using the methodology as presented in [17]. For the TCP time series presented in this paper the optimal embedding parameters are

$m = 6$ and $L = 10$. From the embedded vectors, the PE H_6 is calculated as a function of time according to the blocks of data considered. The variations of the PE are expected to detect changes and anomalies in the TCP congestion window time series.

5 Results

In Fig. 1 and Fig. 2 plots of the temporal evolution of the TCP congestion windows along with the complexity measure of the PE of MS1 and MS2 are respectively presented. Plots (a) and (b) in each figure correspond to different levels of congestion for on-off UDP background traffic of 40 and 80 kbps respectively. In the plots windows as indicated by the dashed vertical lines are drawn to indicate a change of the dynamics in the evolution of the congestion window. The windows are drawn to indicate when apparent dynamical changes occur inside the network.

As it is presented in Fig. 1.a, from the congestion window temporal evolution graphs for 40 kbps background traffic, apart from the initiation of the connection where *slow-start* is performed, the congestion windows of MS1 and MS2 evolve in a stable pattern. This is portrayed in the temporal evolution of the PE measurement, which for the complete time duration of the simulation acquires values around 0.2. Additionally one small increase in the congestion window for MS1 as shown in Fig. 1.a with a relative decrease in the PE around the window of 400-450 seconds is observed. A similar phenomenon occurs for MS2 in Fig. 2.a in the window of 600-640 seconds.

As congestion increases for 80 kbps background traffic, it is observed from Figs. 1.b of MS1 and Figs 2.b of MS2 that the congestion window temporal evolution becomes much more irregular. As congestion increases and more packet drops occur inside the network, the size of the congestion windows gets significantly decreased. From the windows drawn by the dashed vertical lines it is observed that when congestion increases and the dynamics of the congestion window become irregular a sharp increase in the PE is observed. As an example we refer to the window from 230-400 seconds as observed for MS1 in Fig 1.b where a sharp increase in the PE is measured. Respectively MS2 in the same time window gets a decrease in the PE and although a timeout occurs around 310 seconds much larger values of congestion windows are observed. This indicates that the MS2 will achieve a better throughput with an increased QoS in this time window.

Similar observations are made for all the plots in Fig. 1 and Fig. 2. In general a sharp increase in the PE indicates the onset of a congestion epoch inside the network with more irregularity in the time series of the congestion window and more packet drops inside the network. That is why in the case of 80 kbps background traffic the measurements of PE temporal evolution mainly vary from 0.3 to 0.4. Additionally the decrease in the PE captures the transition from an increased congestion epoch to a more stable congestion window evolution where larger congestion windows are observed. Hence it is concluded that the PE can capture anomalies and changes in the dynamics of the TCP traffic flow and can be used as a measure of complexity for the quantification of the dynamics of the TCP traffic flow. Additionally it is highlighted that measurements of the PE for the traffic flows is very fast. For each simulated 800 second TCP time series, the processing of the PE lasted for less than one minute on a 2.26 GHz laptop pc. This is a seminal advantage of the PE as it can be used for online traffic monitoring applications for real-time detection of congestion epochs inside the network.

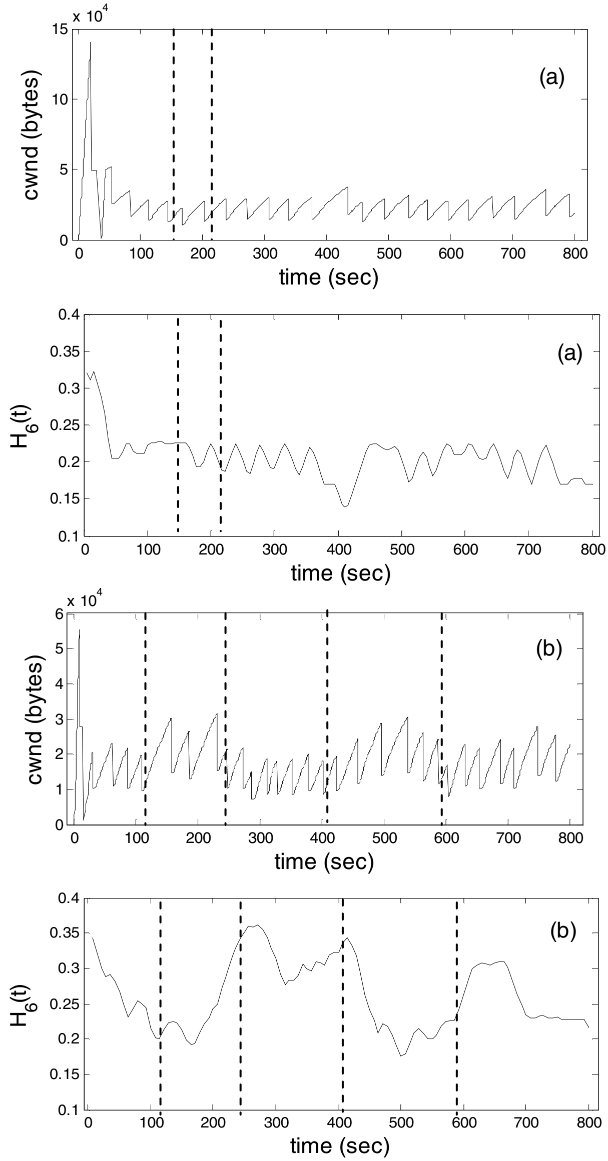

Fig. 1. MS1 temporal evolution of the TCP congestion window (cwnd) $w(t)$ and normalized Permutation Entropy H_p for background traffic of: (a) 40 kbps, (b) 80 kbps

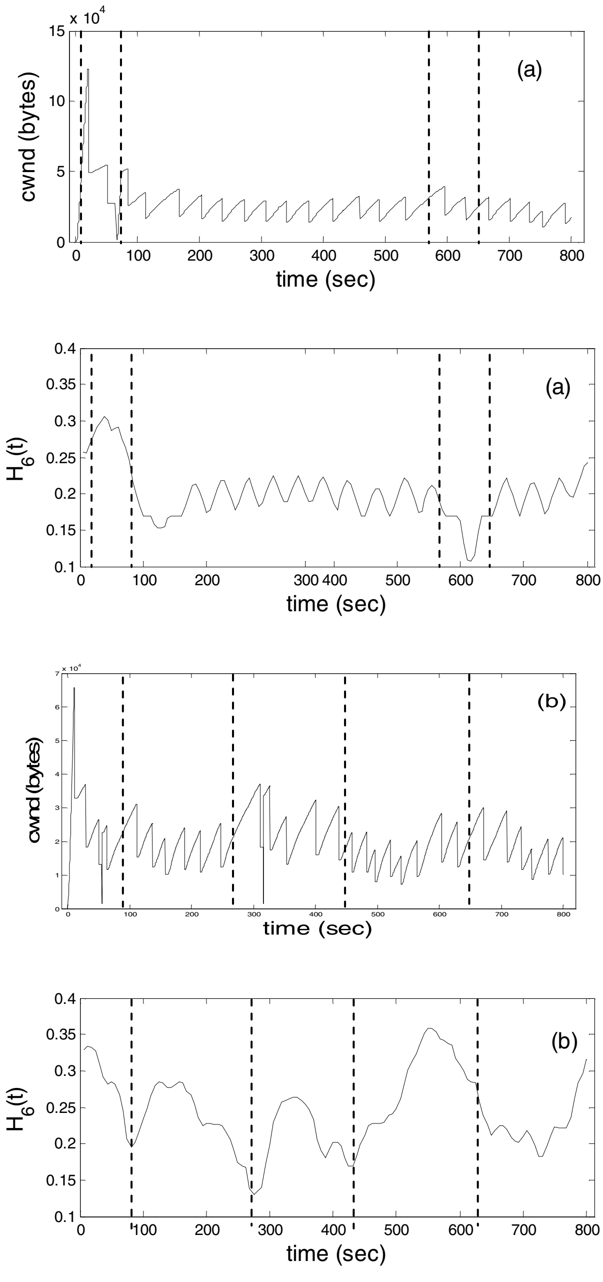

Fig. 2. MS2 temporal evolution of the TCP congestion window (cwnd) $w(t)$ and normalized Permutation Entropy H_p for background traffic of: (a) 40 kbps, (b) 80 kbps

The stable behaviour of the congestion windows for 40 kbps background traffic has a direct relation to the QoS as it is perceived from the average throughput for MS1 and MS2. As presented in Fig. 3.a both MSs average throughputs have close values along the temporal evolution of the simulation. Such a result indicates that both MSs use the network resources fairly and efficiently. On the contrary, as shown above from the PE measurements, the complexity of the dynamics of the congestion window increases as the traffic congestion increases at the simulations with background traffic of 80 kbps. As a consequence in Fig. 3.b MS1 has an average throughput reduced to around 20 kbps in comparison to MS2. Relatively in Fig. 3.c MS2 has an averaged throughput reduced to around 10 kbps in comparison to MS1. Therefore it is understood that the complex dynamics of the traffic profile have a direct impact on the QoS as perceived by the users in the network. Additionally loss of valuable radio resources of the UMTS network is observed as in both simulations MS1 is not transmitting packets up to its highest fair potential. This outcome is directly related with the appearance of chaotic dynamics that develop in the TCP traffic profile in congested networks as studied in [3, 4]. The chaotic dynamics are verified by the increased dynamic complexity as proved by the PE measurements.

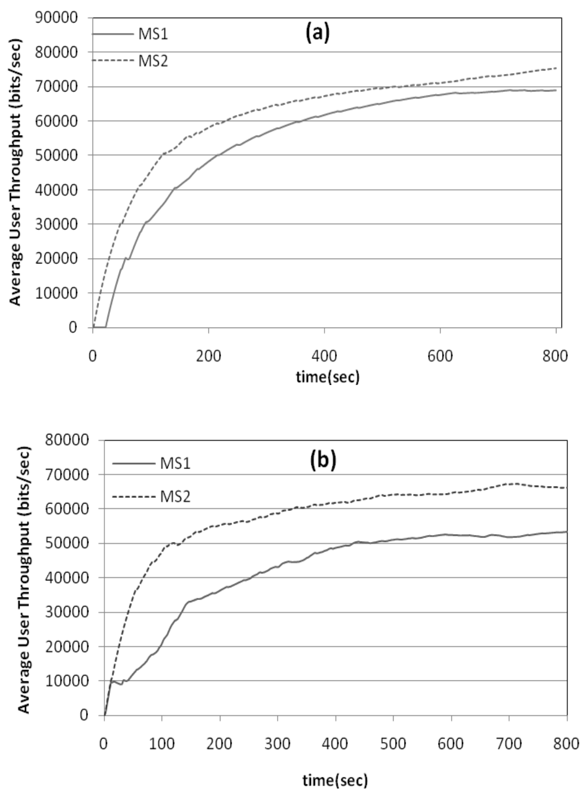

Fig. 3. Average MS1 and MS2 user throughput for background traffic of: (a) 40 kbps, (b) 80 kbps

6 Conclusions

This paper presented the feasibility of using the natural complexity measure of the PE in order to detect anomalies and dynamical changes in the temporal evolution of data transmission in UMTS networks. By analysis of the dynamics of the TCP traffic profile it is shown that the PE can accurately detect such dynamical changes using a computationally very quick algorithm. The PE is able to measure the complexity of the TCP time series as simulation results showed the increasing PE measurements relative to the increasing congestion levels in the network. Consequently the PE concept can be used for real-time monitoring of data traffic flows inside the UMTS, in order to prevent the prolonging of congestion epochs, or detect data traffic anomalies.

References

1. Walke, B.H.: Mobile Radio Networks, 2nd edn. Wiley, England (2002)
2. Pointon, C.T., Carrasco, R.A., Gell, M.A.: Complex Behaviour in Nonlinear Systems. In: Modelling Future Telecommunications Systems. BT Telecommunications Series, pp. 311–344. Chapman & Hall, England (1996)
3. Vasalos, I., Carrasco, R.A., Woo, W.L., Soto, I.: Nonlinear Complex Behaviour of TCP in UMTS Networks and Performance Analysis. IET Circuits Devices Syst. 2(1), 69–79 (2008)
4. Vasalos, I., Carrasco, R.A.: Dynamic Complexity of TCP in UMTS Networks and Performance Evaluation. In: IEEE ICWMC 2008, The Fourth International Conference on Wireless and Mobile Communications, pp. 253–259 (August 2008)
5. Veres, A., Boda, M.: The Chaotic Nature of TCP Congestion Control. In: IEEE INFOCOM, pp. 1715–1723 (March 2000)
6. Rao, N.S.V., Gao, J., Chua, L.O.: On Dynamics of Transport Protocols over Wide-Area Internet Connections. In: Kocarev, L., Vattay, G. (eds.) Complex Dynamics in Communication Networks, Springer Complexity 2005. Springer, Heidelberg (2005)
7. Yuan, J., Mills, K.: Exploring Collective Dynamics in Communication Networks. Journal of Research of the National Institute of Standards and Technology 107(2), 179–191 (2002)
8. Yuan, J., Mills, K.: Monitoring the Macroscopic Effect of DDoS Flooding Attacks. IEEE Trans. on Dependable and Secure Computing 2(4), 324–335 (2005)
9. Takayasu, M., Tretyakov, A., Fukuda, K., Takayasu, H.: Phase Transition and 1/f Noise in the Internet Packet Transport. In: Schreckenberg, M., Wolf, D.E. (eds.) Traffic and Granular Flow, pp. 57–67. Springer, Singapore (1998)
10. Nucci, A., Bannerman, S.: Controlled Chaos. IEEE Spectrum, 37–42 (December 2007)
11. Band, C., Pompe, B.: Permutation Entropy – A Natural Complexity Measure for Time Series. Phys. Rev. Lett. 88, 174102 (2002)
12. Cao, Y., Tung, W.W., Gao, J.B., Protopopescu, V.A., Hively, L.M.: Detecting Dynamical Changes in Time Series. Phys. Rev. E 70, 046217 (2004)
13. Vangala, S., Labrador, M.A.: The TCP SACK aware Snoop Protocol for TCP over Wireless Networks. In: IEEE 58th Vehicular Technology Conf., vol. 4(6-9), pp. 2624–2628 (October 2003)

14. Huang, J.J., Chang, J.F.: A New Method to Improve the Performance of TCP SACK over Wireless Links. In: IEEE 57th Vehicular Technology Conference, vol. 3, pp. 1730–1734 (April 2003)
15. Alli, K.T., Sauer, T.D.: Chaos an Introduction to Dynamical Systems. Springer, New York (1996)
16. Bohacek, S., Hespanha, J.P., Lee, J., Obraczka, K.: A Hybrid Systems Modeling Framework for Fast and Accurate Simulation of Data Communication Networks. In: ACM SIGMETRICS (June 2003)
17. Gao, J.B., Zheng, Z.M.: Direct dynamical test for deterministic chaos and optimal embedding of a chaotic time series. Phys. Rev. E 49, 3807–3814 (1994)