An Adaptive Cross-Layer Strategy for 4G Wireless Communications

Lin Cui
School of Info. Tech. Eng.
Tianjin Univ. of Tech. & Edu.
Tianjin, China
cuilin.academic@gmail.com

Xin Cui
School of Business
Shandong Univ. at Weihai
Weihai, China
chlgms@sdu.edu.cn

Woo Jin Lee
School of EECS
Kyungpook National Univ.
Daegu, Korea
woojin@knu.ac.kr

Qijia Zhang
School of Info. Tech. Eng.
Tianjin Univ. of Tech. & Edu.
Tianjin, China
jack_gxy@126.com

Abstract—**This paper introduces a cross-layer strategy on how to safely enable the corruption-aware transport protocols in the next generation wireless communications without disabling the link layer 32-bit CRC checksum mechanisms. Simulation results show that the proposed scheme can help the corruption-aware transport protocols significantly improve their performance compared to that of the existing schemes.**

Keywords-Transport Protocol; FCS; CRC; checksum

I. INTRODUCTION

With the advent of wireless networks, a high and variable packet corruption rate has been seen in wireless communications (e.g., 10-50% erasure rate [1]). Those corrupted packets have to be discarded due to failed checksum of frame check sequence (FCS) in current networks. This will significantly degrade the end-to-end throughput performance of the traditional transport protocols.

On the other hand, since it is difficult for human being's senses to find out some tiny differences resulted from random bit errors for both video and voice, a number of codecs (e.g., the AMR speech codec, the Internet Low Bit Rate Codec, and error resilient H.263+, H.264 and MPEG-4 video codecs) may be designed to cope better with errors in the payload than with loss of entire packets [5].

In order to achieve such a target, UDP-Lite [5] provides a checksum with an optional partial coverage in transport layer. When using this option, a packet is divided into a sensitive part (covered by the checksum) and an insensitive part (not covered by the checksum). Errors in the insensitive part will not cause the packet to be discarded by the transport protocol at the receiving end host. When the checksum covers the entire packet, which should be the default, UDP-Lite is semantically identical to UDP [3]. Compared to UDP [3], the partial checksum feature provides extra flexibility for applications that want to define the payload as partially insensitive to bit errors.

As UDP-Lite, DCCP [2] uses a header checksum to protect its header against corruption. DCCP applications can, however, request that the header checksum covers part of the application data, or perhaps no application data at all. For some noisy links, this can greatly improve delivery rates and perceived performance.

Both UDP-Lite and DCCP have been standardized as RFC documents by IETF, and neither of them introduces a feasible cross-layer strategy to ensure the corrupted packets reaching transport layer. Unfortunately, in most realistic wireless networks, such as IEEE 802.11 a/b/g links, it is mandatory to use frame check sequence (FCS) field for Cyclic Redundancy Check (CRC) checksum. Hence, the corruption-aware transport protocols, like UDP-Lite and DCCP, cannot work over such networks since all corrupted packets will be discarded in MAC layer forcedly due to failed CRC checksum before they are delivered to upper layers.

This paper introduces a cross-layer strategy about how to safely enable the corruption-aware transport protocols in the next generation wireless communications, without disabling the link layer 32-bit CRC checksum mechanisms. The rest of this paper is organized as follows. Section II introduces some related works. Section III describes the proposed scheme in detail. Section IV shows some simulation results and section V concludes this paper.

II. RELATED WORKS

Nowadays, the sole wireless standard that takes corruption into account in MAC layer is IEEE 802.16. In the specification of IEEE 802.16, the FCS field of a MAC protocol data unit (MPDU) can be used as an optional choice, as shown in Fig. 1. Whether or not the FCS field is optional can be indicated by CRC indication (CI), as shown in Fig. 2 [4].

Although the specification of 802.16 can enable the corruption-aware transport protocols to work over 802.16 links, it takes some risks to overload the network traffic and/or to deliver some garbage to application layer. This is because when FCS field is unused, the validity of a MAC header has to be done by checking the 8-bit header checksum (HCS) field as shown in Fig. 2, which is relatively weaker than 32-bit CRC checksum. Thus some packets with the corrupted MAC frame header may be delivered to the wrong destinations or be acknowledged to the wrong sources. More seriously, TCP may deliver some garbage to application layer since both IPv4 and TCP itself employ the 16-bit 1's complement checksum

CHINACOM 2010, August 25-27, Beijing, China
Copyright © 2011 ICST 973-963-9799-97-4
DOI 10.4108/chinacom.2010.46

scheme which is also weaker than 32-bit CRC checksum (IPv6 has removed the checksum field in order to reduce packet processing time in routers).

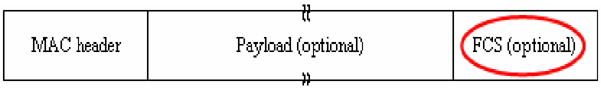

Figure 1. MAC PDU format of IEEE 802.16

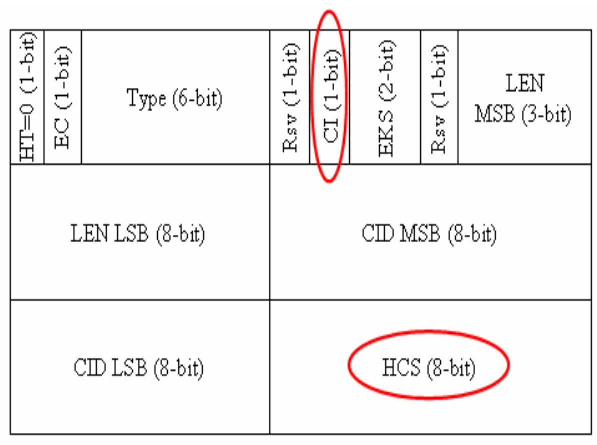

Figure 2. Format of IEEE 802.16 MPDU header

In addition, for the next generation wireless networks, the internetworking of heterogeneous networks (such as among wireless mesh network based on 802.15, wireless municipal area network based on IEEE 802.16, wireless local area network based on IEEE 802.11 a/b/g, etc.) is a big issue because they have different formats of MPDU and different mechanisms for MAC layer. In consequence, the throughput of a wireless traffic over heterogeneous networks (for example, heterogeneous network composed of 802.16 and 802.11 links) cannot be enhanced, by applying corruption-aware transport protocols only due to the restriction of non-802.16 MAC layer (e.g., 802.11 MAC layer).

In this regard, the link layers of the different wireless networks should have a common rule on how to use a strong integrity check (e.g., CRC-32) to protect either entire packet or header portion flexibly. If the underlying link supports this, the end-to-end corruption-aware transport protocols can benefit from permitting partially damaged IP packets to be forwarded, when requested.

TCP CAIAD [6] is one of such end-to-end corruption-aware transport protocols, which introduces a new error and congestion control scheme using corruption-aware adaptive increase and adaptive decrease algorithm. In [6], the corrupted segments will be further processed by the transport layer of the receiver, and a duplicate ACK is generated to explicitly indicate both a real-time corruption event and the congestion state of the link. Based on the feedback information, the sender estimates the current corruption strength and increases its *cwnd* by different increments instead of entering fast recovery phase as long as there is no concurrent loss event. Meanwhile, the slow start threshold will be estimated not only based on the received integral packets but also based on the received corrupted packets as per every round trip time and only updated when the lost but not the corrupted segment is detected. The authors argue that since the corrupted packets can still arrive at the receiver side, they should reflect some available bandwidth at a certain extent as well.

As mentioned before, nevertheless, there is no cross-layer scheme to enable the end-to-end corruption-aware transport protocols to effectively work over underlying CRC-32 checksum mechanisms with various FCS coverages up to now. This paper aims at this issue and proposed an adaptive cross-layer scheme for 4G wireless communications.

III. Proposed Scheme

Generally, while a packet travels an error-prone wireless channel, not only the part of user data but also the packet header suffers bit errors with various probabilities. In current networks, those corrupted MAC frames will be discarded directly by the intermediate routers because of failed FCS checksum. Thus normally the corrupted MAC frames cannot arrive at receiver side unless corruption occurs in the last hop of their traveling path.

In design of the next generation wireless communication system, one of important issues is to enable the corruption-aware protocols to effectively work over CRC checksum. Of the proposals, a cross-layer coordinator, through which multiple layers of either OSI model or TCP/IP protocol suite can exchange their relevant information anytime, is preferred. In this paper, we propose to use a "connection-based cross-layer access table" in memory to coordinate adaptive behaviors among various layers. Figure 3 shows an example format for the table, and we assume in this paper that the sender will automatically adopt such a table for an end-to-end connection when it is established over an error-prone wireless link.

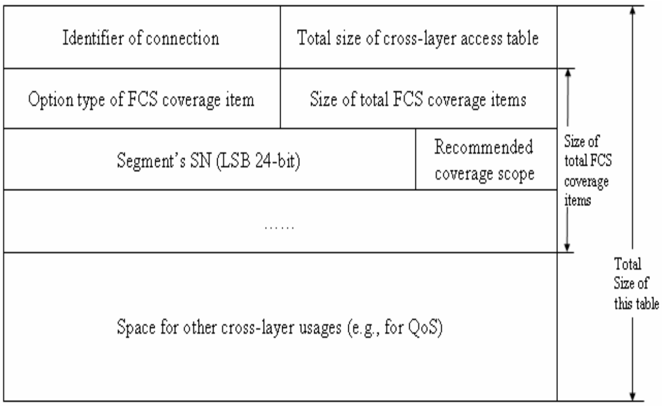
Identifier of connection
Total size of cross-layer access table
Option type of FCS coverage item
Size of total FCS coverage items
Segment's SN (LSB 24-bit)
Recommended coverage scope
……
Space for other cross-layer usages (e.g., for QoS)
Size of total FCS coverage items
Total Size of this table

Figure 3. Format of "cross-layer access table"

Moreover, when the connection is established, the sender and the receiver should decide whether or not to use corruption-aware protocol at transport layer by exchanging the relevant information based on application's type of service. Once a corruption-aware protocol is chosen, after encapsulating a layer 4th protocol data unit (PDU), the sender should access the "cross-layer access table" and append an FCS coverage item for this segment. At the same time, in order to indicate the lower layers, a currently unused flag (e.g., one of the six reserved bits in TCP base header) should be set in every segment's header at transport layer.

From Fig. 3 we can see that each item consists of two fields; they are the segment's sequence number field and the recommended coverage scope field. The former will be filled with the least significant 24-bit of the segment's sequence number and we believe it is enough for the lower layers to identify every packet/frame from each other since sending buffer is not so large. The later will be filled with the recommended coverage scope, except the link frame's header. In detail, this field is available for one of the following four statues: "00", "01", "10" and "11". Wherein "00" recommends the link frame's FCS to check entire frame as normal (hereinafter, refers to normal FCS); "01" recommends that the FCS partially checks transport protocol PDU's base header and IPv4's base header in addition to its frame header (variable length for different network standards); In the same way, besides the frame header, "10" recommends the link frame's FCS partially checks transport protocol PDU's base header, IPv4's base header and IPv4's option field; "11" recommends the link frame's FCS partially checks transport protocol PDU's base header and IPv6's base header (hereinafter, all of "01", "10" and "11" refer to partial FCS).

It is noted that for the different transport protocols, the sizes of their base headers are different. In particular, TCP base header is 20-byte in length, SCTP common header is 12-byte, DCCP generic header is 16-byte and UDP-Lite header is only 8-byte. For simplicity of the intermediate nodes' implementation, the size of transport protocol PDU's base header is fixed to the maximum 20-byte in this paper. Also, if only any IPv4 option is used, the size of the used IPv4's option space is assumed as 40-byte as well.

The other reason that we propose to calculate the partial FCS based on above scope is that generally the intermediate nodes have no the detailed knowledge of every passing MAC frame and some IP headers may contain option fields which extend their header scopes. Thus, in order to ensure the integrity of transport protocol PDU's base header, the partial FCS has to be calculated based on the possibly maximum header scope. However, TCP option field can be omitted since the both useful sequence number and checksum fields lie on TCP base header.

When the transport layer PDU is forwarded to IP layer, the IP protocol realizes that upper protocol is "corruption-aware" from the special flag labeled in its base header. Thus it accesses the "cross-layer access table" and updates the corresponding "recommended coverage scope" depending on IP version as well as the usage of IPv4 option field. After that, a MAC frame will be constructed at the source. The FCS of this frame will be calculated based on the different checksum scopes, which is indicated by the "recommended coverage scope" of the "cross-layer access table".

The big problem is that in the course of traveling the heterogeneous wireless networks, not only source and destination but also every intermediate node has to correctly differentiate the partial FCSs from the normal one. Consequently, how to let the intermediate nodes know the exact FCS checksum scope is crucial. Our solution is quite simple, that is, using two currently unused bits or values in the frame header indicates the FCS coverage scope that is recommended by "cross-layer access table". Since different wireless standards support different frame format, the indication may be signed in different fields and the exact length of FCS checksum scope may be different as well in various wireless standards.

As an example, in the wireless networks that are connected over the 802.11 wireless links, the Duration/ID field of MAC frame is not fully used, plenty of values are reserved (e.g., when MSB 2-bit is equal to "10", plenty of LSB 14-bit's values, which are from 1 to 16383, are reserved). We can use two of these reserved values to indicate the different FCS checksum scope. Further, the partial FCS of a 802.11 MAC frame can be calculated based on its first 70 bytes (the sum of 30 bytes' 802.11 MAC frame header, 20 bytes' IPv4 base header and 20 bytes' maximum transport segment's base header), 110 bytes (70 bytes plus 40 bytes' possibly maximum IPv4 option field) or 90 bytes (use 40 bytes' IPv6 base header instead of IPv4's base header in the first case), respectively.

As another example, in the 802.16 wireless networks, the header checksum (HCS, 8-bit) field of the 802.16 MAC frame is useless in our scheme since the more powerful CRC-32 checksum has covered frame header. We can indicate the different FCS checksum scope in this field.

Similarly, the partial FCS of a 802.16 MAC frame can be calculated based on its first 46 bytes (the sum of 6 bytes' 802.16 MAC frame header, 20 bytes' IPv4 base header and 20 bytes' maximum transport segment's base header), 86 bytes (46 bytes plus 40 bytes' possibly maximum IPv4 option field) or 66 bytes (use 40 bytes' IPv6 base header instead of IPv4's base header in the first case), respectively.

In such a way, only those MAC frames with the corrupted header will be dropped, and the frames with the valid header will successfully arrive at the receiver side without traffic overload as well as wrong delivery. Therefore, the corruption-aware transport receiver can process these corrupted segments and feed the detailed corruption information back to the sender so as to avoid deflation of *cwnd*.

The integrity of the rest part of the frames can be taken charged by the transport layer checksum. Notice that Internet checksum is relatively weaker than CRC mechanism. In order to prevent some garbage to be delivered to application layer, TCP may need an additional CRC checksum in its option field for checking the entire segment. In contrary to TCP, SCTP natively uses CRC scheme and DCCP also already has a CRC checksum option too.

The main drawback introduced by the partial FCS scheme is the little overhead. As described before, TCP employs Internet checksum in transport layer, which is relatively weaker than CRC mechanism. In order to prevent some garbage to be delivered to application layer, TCP may need an additional CRC checksum in its option field. This will result in some extra overhead. Nevertheless, the overhead could be minor, compared to the improved throughput.

IV. SIMULATION RESULTS

We use TCP CAIAD [6] as the corruption-aware transport protocol in our experiments, and select TCP Westwood+ [7] as the reference protocol from which TCP CAIAD is evolved.

We argue in this paper that when partial FCS is applied, the packet dropped rate incurred by corruption should be set in simulations by the proportion between the possibly maximum header scope and the whole size of the MAC frame over an error-prone wireless link. This is because MAC frame is the service data unit (SDU) in wireless channel. For example, if each IP packet has a fixed size of 1040-byte and packet corruption rate is ß, the packet drop rate caused by corruption could be considered as 110ß/1074 (that is, 1074=30+1040+4) approximately in 802.11 WLAN.

In particular, each IP packet uses a fixed size of 1040-byte, and the packet drop rate incurred by corruption is set to 110/1074 of packet corruption rate for the proposed scheme. On the other hand, TCP Westwood+ [7] regards packet corruption as packet loss. Thus its packet drop rate is equal to the packet corruption rate in simulations.

Moreover, in simulations we only devote our mind to the impacts of header corruption and neglect the checksum procedure of partial FCS. To minimize other impacts on performance comparison, e.g., network layer's congestion, we use a simple simulation topology as shown in Fig. 4. In the figure, the wired link has the link bandwidth of 10 Mbps and the transmission delay of 35ms, whereas the wireless link has the bandwidth of 2 Mbps and the transmission delay of 1ms.

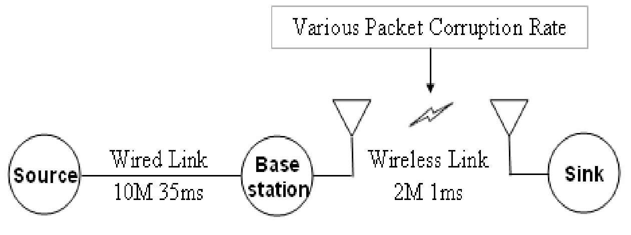

Figure 4. Simulation topology

We compare the average end-to-end throughputs during 100 seconds by performing the file transfer application in ns2 simulator. In the comparison, each data segment carries 1000 bytes' user data by TCP Westwood+, whereas TCP CAIAD only sends 990 bytes' user data by every data segment (we assume that each data segment contains two additional checksums in option field. One is CRC checksum option for entire segment and the other is internet checksum option for header portion. Both options need 10 bytes in all). Simulation results are shown in Fig. 5.

A note on notation: in the Figure the "proposed scheme" refers to TCP CAIAD with partial FCS scheme, "TCP CAIAD" refers to TCP CAIAD with disabling FCS checksum scheme and "TCP Westwood+" refers to TCP Westwood+ with the normal FCS scheme.

From Fig. 5, we can see that both the proposed partial FCS scheme and TCP CAIAD with disabling FCS scheme can provide the better performances while the corruption rate is higher than 0.1%. Especially, when the wireless link experiences the packet corruption rates ranged from 0.1% to 1%, it seems that both corruption-aware schemes can almost fully utilize the link bandwidth, while TCP Westwood+ with the normal FCS scheme already gets the drastic performance degradation then.

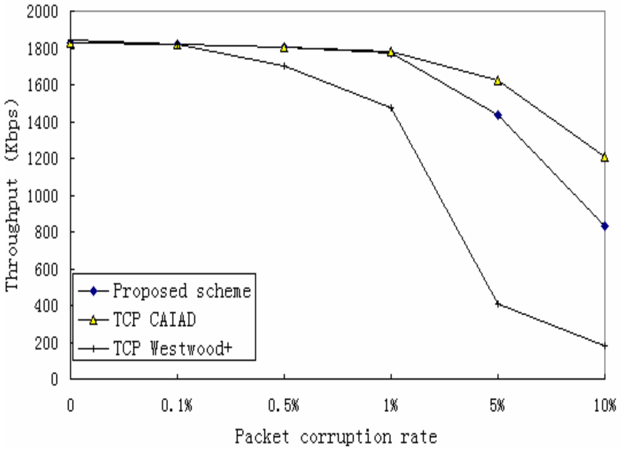

Figure 5. Throughput comparison

From Fig. 5, we can see that both the proposed partial FCS scheme and TCP CAIAD with disabling FCS scheme can provide the better performances while the corruption rate is higher than 0.1%. Especially, when the wireless link experiences the packet corruption rates ranged from 0.1% to 1%, it seems that both corruption-aware schemes can almost fully utilize the link bandwidth, while TCP Westwood+ with the normal FCS scheme already gets the drastic performance degradation then.

The figures from Fig. 6 to Fig. 8 show the comparisons of bandwidth utilization, evolutions of congestion window and slow start threshold with 10% packet corruption rate, respectively. From the figures, we can see that although all TCP variants get severe performance degradation due to the heavy packet corruption rate (e.g., 10%), the performance gains of the corruption-aware schemes (including both the partial FCS scheme and the skip flag scheme) are still prominent, compared to that of TCP Westwood+.

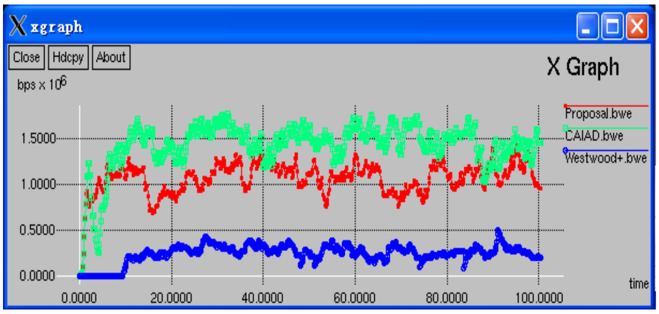

Figure 6. Bandwidth utilization comparison for 10% packet corruption rate

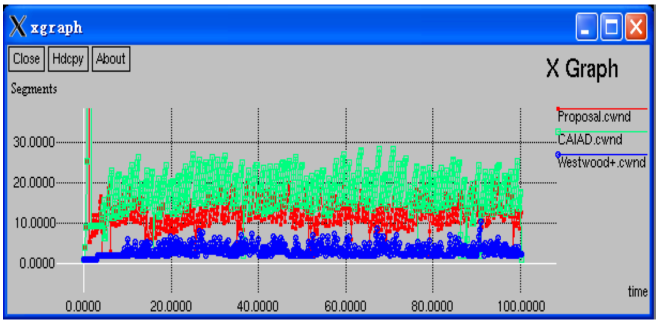

Figure 7. Congestion window comparison for 10% packet corruption rate

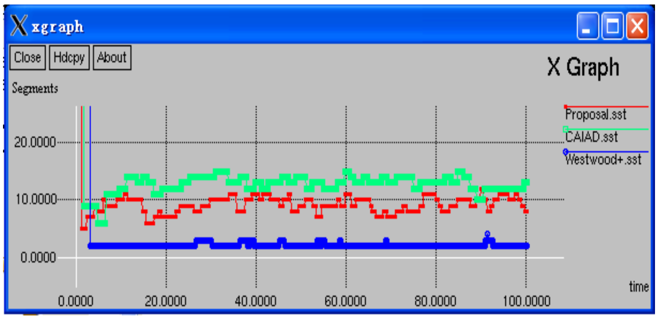

Figure 8. Slow start threshold comparison for 10% packet corruption rate

More seriously, once the initial control segments are corrupted during connection establishment phase, the traditional TCP will back-off the initial time exponentially (see Fig. 6). This will further degrade TCP performance. On the contrary, both corruption-aware schemes can retransmit the initial control segments immediately so as to ensure the TCP connection to be established in time (see Fig. 6).

Based on the comparisons above, we can draw a conclusion as follows. The adaptive partial FCS scheme can help corruption-aware protocols improve their end-to-end throughput obviously, compared to the traditional transport protocols. This is because the end-to-end corruption-aware transport protocols, which run over the proposed adaptive partial FCS scheme, can keep their native characteristic of differentiating packet corruption from packet loss. On the other hand, although the proposed adaptive partial FCS scheme cannot outperform TCP CAIAD with disabling FCS, the reasons stated in section II make the proposed partial FCS scheme more feasible for implementation in the next generation wireless communication systems.

V. CONCLUSIONS

In this paper, we present an adaptive cross-layer strategy to enable the corruption-aware transport protocols to work in the next generation wireless communication system without the necessity to disable the link layer CRC checksum mechanisms. From simulation results, we can see that the proposed partial scheme can still perform far better, compared to the traditional transport protocols with the normal FCS scheme in wireless environment with high BER.

On the other hand, even if disabling the link layer CRC checksum can lead to the higher throughput in simulations, the reasons mentioned before make the proposed scheme more feasible to implement in realistic networks in the next generation wireless communication system.

ACKNOWLEDGMENT

This research was supported by Startup Fund under the supervision of Tianjin University of Technology and Education (KYQD09001).

REFERENCES

[1] D. Aguayo, J. Bicket, S. Biswas, G. Judd and R. Morris, “Link-level Measurements from an 802.11b Mesh Network,” in Proceedings of the SIGCOMM 2004, Aug. 2004.

[2] E. Kohler, M. Handley and S. Floyd, “Datagram Congestion Control Protocol,” IETF, RFC 4340, Mar. 2006.

[3] Postel, J., "User Datagram Protocol", IETF, RFC 768, August 1980.

[4] IEEE 802.16 Working Group, “Draft IEEE Standard for Local and metropolitan area networks--Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems,” IEEE P802.16e/D11, September 2005.

[5] L-A. Larzon, M. Degermark, S. Pink, et al., "The Lightweight User Datagram Protocol (UDP-Lite)," RFC3828, July 2004.

[6] Lin Cui and Seok J. Koh, “Corruption-aware Adaptive Increase and Adaptive Decrease Algorithm for TCP Error and Congestion Controls in Wireless Networks,” International Journal of Communication Systems (IJCS), Vol. 22, pp. 543 - 564, May 2009.

[7] Westwood+ TCP – Modules for ns2 from
http://193.204.59.68/mascolo/tcp%20westwood/modules.htm.