International Journal of Innovative Research in Information Security (IJIRIS)
Issue 02, Volume 3 (February 2016)
ISSN: 2349-7017(O)
ISSN: 2349-7009(P)
www.ijiris.com

Performing Network Simulators of TCP with E2E Network Model over UMTS Networks

M.Thejovathi *M.Tech (Software Engineering)*
MobileNo: +91 9908563865
Department of Computer science & Engineering

Abstract- Wireless links losses result in poor TCP throughput since losses are perceived as congestion by TCP with the evolution of 3G technologies like Universal Mobile Telecommunication System (UMTS), the usage of TCP has become more popular for a reliable end-to-end (e2e) data delivery. However, TCP was initially designed for wired networks and therefore it suffers performance degradation due to the radio signal getting affected by fading, shadowing and interference. There are many strategies proposed by the research community on how to improve the performance of TCP over wireless links such as introducing link-layer retransmission, explicitly notifying the sender of network conditions or using new variants of TCP. As UMTS network coverage and availability are currently experiencing rapid growth, optimization of various internal components of its wireless network is very important. One of the optimization is the introduction of High Speed Downlink Packet Access (HSDPA). This architecture not only allows higher data rates but also more reliable data transfer by the introduction of Hybrid ARQ (HARQ). With this enhancement to the UMTS network, it becomes vital to see the performance of TCP in such a network. Therefore in this thesis, we try to evaluate two aspects of UMTS networks: first, the impact of HSDPA parameters like scheduling algorithm and RLC/MAC-hs buffer size on overall performance of TCP and second, to study the behaviour of two categories of TCP rate and flow control: loss based and delay based. Our simulation shows that delay based TCP tends to perform better than loss based TCP in our selected scenarios. The simulations are performed using the network simulator NS-2 with an e2e network model for enhanced UMTS (EURANE).

Keywords: Network simulator (NS2), UMTS Model Network, TCP/IP, HSDPA level optimization

I. INTRODUCTION

Over the years, since first mobile cellular systems appeared, there has been much anticipation of the onslaught of data services, but the radio access platforms have been the inhibitor from making this a reality. Third generation (3G) is a term that has received and continues to receive much attention as the enabler for high speed data for the wireless mobility market. Since 1999, it has evolved as a successor to 2G and 2.5G mobile telecommunication system by providing far more data rates than its predecessors. The mobile network developers have shown strong interest in evaluating the performance of TCP in UMTS. The motivation behind it is to get information that can be used for optimizing UMTS radio access network design for better utilization of the scarce radio resources. It also broadens the understanding and optimization needed for TCP in UMTS, to support the high speed data rates offered by the wireless network. Optimizations in both TCP and UMTS internal parameters can offer efficient usage of network resource simultaneously providing better services to the customers at relatively lower maintenance. Most of the related work in the area of TCP performance over wireless network focuses Most of the related work in the area of TCP performance over wireless network focuses on Wireless LAN (WLAN). However, in this thesis we have chosen Universal Mobile Telecommunication System (UMTS) as our basis for wireless technology due to the fact that usage of Internet over mobile phones has emerged as one of the most popular trends of recent times. Also that the High Speed Downlink Packet Access (HSDPA) architecture introduce in UMTS is designed to provide much high data rates than any existing wireless technology [5]. We focus on evaluating the performance of two different categories of TCP: loss based and delay based, in UMTS network with HSDPA. The TCP variants included for loss based category were TCP Reno and TCP Newreno. And the TCP variants chosen for delay based category were TCP Vegas and TCP FAST. The motivation behind choosing two categories of TCP is to provide the comparison of how their basic congestion avoidance algorithm reacts to optimizations in UMTS network. We also focus on providing HSDPA level optimization like the impact of scheduling algorithm on the overall performance of UMTS network.

II. RELATED WORK

In the field of performance evaluation of TCP over UMTS, not many useful results from performance evaluations are available. It is also very hard to compare the results of different performance evaluations as they depend heavily on the used protocols and parameter settings. One of the examples of interesting TCP performance evaluation for UMTS is the network simulations carried out at the ETH. The thesis report focuses on performance evaluation of certain TCP protocols (all loss based) in optimized UMTS network. The report mainly evaluated the performance of TCP in respect to different modes of RLC layer with different number of local retransmission. They had published a paper about optimizing UMTS link layer parameters using TCP as the transport protocol *Ref.No* [3]. Their results also provide optimal values for the maximum number of allowed link-level retransmission and the RLC window size of the receivers.

© 2014-16, IJIRIS- All Rights Reserved

Also conducted evaluation studies of TCP over UMTS by considering a simple model of the RLC protocol. The packet loss was generated by randomly marking the packets as erroneous which resulted in a requirement for retransmission at the RLC layer. They also tried to study the impact of RLC layer retransmissions ont he overall performance of the UMTS network. Several TCP variants were evaluated using the network simulator NS-2, simulating different traffic scenarios with FTP and Telnet *Ref.No* [17]. It highlighted that type of traffic also had a significant impact on overall performance of TCP in UMTS network.

A research group at the INT France Telecom had investigated throughput in UMTS using the UMTS module of the OPNET network simulator *Ref.No* [13]. They propose a resource scheduling scheme to improve throughput and fairness for non-real-time packet data traffic on the downlink shared channel (DSCH) in UMTS. Another research group at the University of Texas evaluated TCP performance over UMTS networks by carrying out simulations with different retransmission settings at the Medium Access Control (MAC) layer *Ref.No*[12]. This approach is targeted for very delay sensitive applications like real-time applications. To avoid the delay associated with retransmissions at the RLC, a lower layer fast MAC retransmission is introduced in the paper. Thus, retransmission is done on at least two layers (MAC, RLC and even TCP if used as the transport layer protocol).

The main difference between our work and related work is that we not only focus on optimization of UMTS wireless components but also try to provide an analytical view on the impact of these optimizations on two categories of TCP (loss based and delay based). We find the idea of studying the TCP in terms of categories very interesting as the internal mechanism of our selected TCP variants makes them behave differently from each other in same wireless conditions. This provides an opportunity to not only optimize the wireless parameters of UMTS network but also to optimize TCP parameters. One of most interesting parts of this thesis is the analysis of TCP FAST *Ref.No*[16] in a wireless network which, to the best of our knowledge, is one of the first attempts. Our analysis of TCP FAST shows that its performance is very much dependent on the configuration of its fast convergence parameter (α). We also show in general that delay based TCP tends to better than loss based TCP in reacting to wireless network conditions.

In the UMTS part we show that schedulers at the Node B play a very important role in distributing the radio resources of the network. We provide a comparison between two of the most common scheduling strategies i.e. Round Robin and Max C/I. We conclude stating the requirement for Fair Fast Channel Dependent Scheduling as both the scheduling strategies do not justify fairness and fast channel dependent scheduling at the same time. We also try to analyze the impact of RLC/MAC-hs buffer size and conclude that high buffer capacity need not improve the performance of TCP in UMTS network; instead we suggest having optimum range of buffer size. We acknowledge the fact that exhaustive study of RLC and MAC-hs layer retransmission policies and varying Block Error Rate (BLER) network conditions are currently under observation and are out of the scope of this report.

TCP IN WIRELESS NETWORKS:

Transmission Control Protocol (TCP) is one of the most popular and robust data transfer protocols. During the early days of evolution of the Internet, e2e reliability was a major concern for the web based application. The issue of e2e reliability was addressed in TCP's design with the support for data rates from a few Kbit/s up to hundreds of M bit/s. TCP took packet losses as the indication of congestion and reacted to it by performing retransmissions and decreasing sending window to one MSS (Maximum Segment Size). With the advent of wireless technologies, as the usage of TCP for end to end data transfer became more and more common, the performance issues also became more critical. TCP, in wireless networks, is impacted by different environmental properties like radio signal related with fading, shadowing, interference, mobility, handovers aspects, etc. which changes the network conditions dramatically *Ref.No*[15]. Packet losses and delays caused in such various environmental conditions are seen by TCP as congestion. As a result, it reacts wrongly to the situation, eventually degrading the overall performance. Many researchers have tried to propose different strategies to optimize TCP performance in wireless networks *Ref.No*[15], [16], [17]. Some of the different types of TCPs and different strategies incorporated with TCP are briefly discussed as follows. We classify them into four main categories:

LINK LAYER SOLUTIONS:

The basic idea of link layer solutions is to trigger packet loss recovery mechanism by providing faster local retransmission without informing the TC Player *Ref.No*[18]. That is, TCP is not involved in handling wireless losses and the errors over wireless links, which are instead recovered by the link layer mechanisms. In the majority of the current wireless system, the use of ARQ protocols *Ref.No*[6] allows recovery from wireless link errors and provides relatively reliable transfer of packets to the upper layers. However, the interaction between ARQ and TCP may result in poor performance due to spurious retransmissions [22] caused by an incompatible setting of timers at the two layers. To address these issues, there are several enhancements proposed for the link layer solutions: Snooping protocol, Delayed Duplicate Acknowledgement, TULIP

SPLIT TCP SOLUTIONS:

The main idea behind this solution is to split one TCP connection into two connections at the edge of wired and wireless network. The argument supporting this mechanism is that the wired and wireless networks have different characteristics and as well as different transmission rates.

© 2014-16, IJIRIS- All Rights Reserved

Normally the splitting is performed at the wireless gateway or base station but the cellular networks are different. In cellular networks as the base station or Node B is not IP-capable the splitting has to been designed at Radio Network Controller (RNC) node. Due to the splitting the TCP sender on the wired side is only concerned about the congestion and wireless losses. They impose a great amount of complexity and dependency on the node in which the split is implemented.

EXPLICIT NOTIFICATION:

The explicit notification proposals contain a different philosophy compared to most of aforementioned schemes. The sender identifies congestion through data loss over the wireless network, since it explicitly receives information about the transmission status from intermediate routers, which helps the sender to make decisions on the congestion.

END TO END SOLUTIONS:

The end-to-end solutions generally make some changes to the original TCP behaviour on the sender side and demands more cooperation between the sender and receiver to distinguish wireless losses from the network congestion. Optimizations may also be placed entirely in the end-hosts to avoid adding complexity to the network. The intermediate node need not be TCP-aware and pass through the same intermediary node, similar to many other proposals described above. Available e2e solutions are shown as follows: Eifel algorithm, TCP Real Freeze TCP, TCP Westwood

UMTS Network Architecture

The Universal Mobile Telecommunication System (UMTS) is one of the new breed of standards for 3G (third generation) mobile communications. Figure2.1 depicts the architectural components of a UMTS network *Ref.No[*1]. The network architecture of UMTS is formed of three logical entities: the Core network, the UTRAN and the UE.

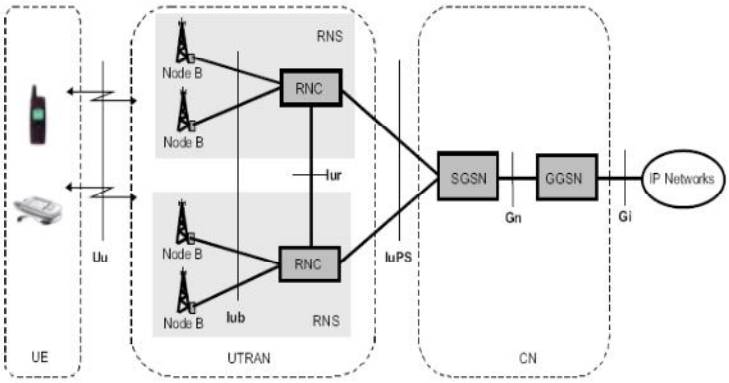


Figure 2.1 UMTS Network Architecture

These logical entities have several functional nodes within them to serve the UMTS network. The Core network is composed of GGSN (Gateway GPRS Support Node) and SGSN (Serving GPRS Support Node), which are responsible for switching/routing calls and data connections to the external networks. The UTRAN part can contain one or more Radio Network Subsystem (RNS). An RNS consists of one Radio Network Controller (RNC) and one or more Node Bs (also known as base station). The Node Bs is responsible for performing air interface layer1 processing (channel coding and interleaving, rate adaptation, spreading etc.) based on the Wideband Code Division Multiple Access (WCDMA) technology [1]. Between RNC and MSC/VLR, used for circuit switched traffic and IuPS between RNC and SGSN, used for packet switch traffic.

UMTS PROTOCOL ARCHITECTURE:

So far, we introduce the basic UMTS network architecture. In this section, we describe shortly the protocol architecture. There are three layers involved in the architecture: network layer, MAC layer and data link layer. From figure 2.2 it can be seen that the channel in UMTS is organized in layers: Logical channels, Transport channels, and Physical channels. Physical channels provide the medium for transmission by a specific carrier frequency, scrambling code, and channelization code and time duration. The physical layer offers services to the MAC layer via transport channels that are characterized by how and with what characteristics data is transferred. The MAC layer, in turn, offers services to the RLC layer by the means of logical channels. The logical channels are characterized by the type of data which is transmitted.

© 2014-16, IJIRIS- All Rights Reserved

ISSN: 2349-7017(O)
ISSN: 2349-7009(P)
www.ijiris.com

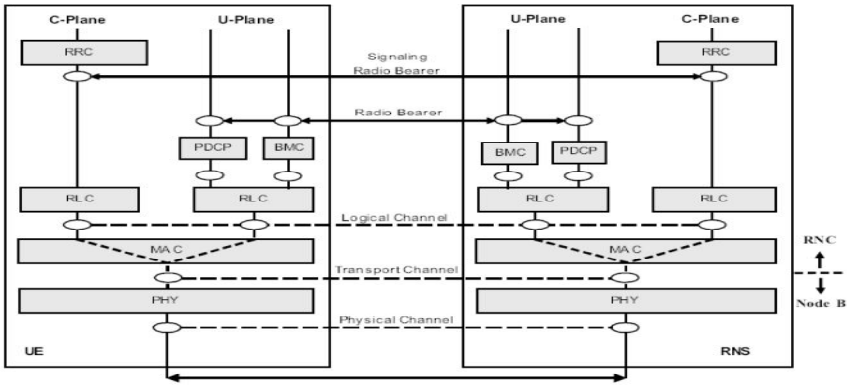


Figure 2.2 UMTS Protocol Stack

SIMULATION PARADIGM:

NS2

The performance evaluation of TCP over UMTS network can be done either by conducting field tests or by performing simulation tests. The field testing which sounds more authentic over simulation testing is very expensive and almost non-iterative exercise due to infeasibility of performing exhaustive testing with different parametric settings. For this reason we choose to perform simulation based performance evaluation using NS-2 *Ref.No*[7] as our basis. The rationale behind selection of NS-2 over other network simulators like OMNET++ or OPNET is the flexibility of NS-2 and the research support that is available for testing different TCP variants (especially TCP FAST).

EURANE Model

EURANE (Enhanced UMTS Radio Access Network Extensions for NS-2) is one of the main outcome of SEACORN project (Simulation of Enhanced UMTS Access and Core Networks), which investigates enhancements to UMTS for UTRAN and Core Network through simulation. EURANE is an end-to-end extension which adds three extra radio link nodes to NS-2, the Radio Network Controller (RNC), Base station (BS) and the User Equipment (UE).

Propagation Model

The propagation model for the physical layer of wireless channel of UMTS network is modelled considering following characteristics: *Interference,* in WCDMA is a sum of intra-cell and inter-cell interference, having noise-like character. This is mainly due to large number of sources adding to the signal, which are similar in signal strength. The intra-cell interference is added at the input of the channel model and the inter-cell interference is added at the input of the receiver. *Channel Quality Indicator (CQI),* is a 5-bit feedback from the receiving UE to the transmitting Node B. Each CQI value represents a specific combination of the Number of Codes, Modulation and Code Rate, resulting in a specific Transport Block Size (TBS). The relation between CQI and SNR for a BLER of 10% is approximated through a linear function.

$$CQI = \begin{cases} 0 & SNR <= -16 \\ SNR + 16.62 & -16 < SNR < 14 \\ 1.02 & 14 <= SNR \\ 30 & \end{cases}$$

The probability a block is received correctly depends on the SNR, the CQI of the block and the implementation of the receiver. *Receiver,* models the soft combining part of the HARQ process. The model applies chase combining (as described in the previous section) for retransmission. Chase combining applies soft combining of at the most three receptions. The resulting SNR is the square root of the sum of the squares of the SNRs of the individual receptions.

$$SNR = 10 * \log \left(\sum_{n \in N} 10^{SNRn/10} \right)$$

After one or two transmissions, if the signal is received correctly than the receiver sends an ACK and thereby initiates the transmission of a new block. If after three transmissions the signal is not received correctly the block is delivered in error to the RLC.

Simulation Parameters Settings and Performance Metrics:

We have used two different set of parameters in this report. The first set is consisting of fixed parameters which are same for all simulation scenario considered in this report. For example, we have chosen the RLC layer operating mode as Acknowledge Mode (AM) as is standard operating mode and performs better than Transparent Mode or Un-acknowledge Mode (UM) the other values chosen for the simulation are according to the 3Gspecifications

© 2014-16, IJIRIS- All Rights Reserved

Fixed parameters:

RLC Mode	AM
RLC PDU size	40 bytes
RLC Poll Timeout	170 ms
HS-DSCH Max. Data Rate	1.2 mbps
HS-DSCH TTI	2 ms
MAC-hs No. of HARQ transmissions	3
Inter-cell Interference	30 dB
Intra-cell Interference	-70 dB
Transmission Power of BS	50 dBm
Transmission Power of UE	50 dBm
Block Error Rate (BLER)	0.1
Simulation duration	200 s

Considered TCP Variants for the simulation:
The following TCP variants are taken into consideration based on their wide popularity, stable behaviour and also as fall into two main categories: Loss based and Delay based.

Packet Loss based congestion control
The TCP variants like Reno and Newreno uses packet loss as the indication for congestion.

Reno: The main features of TCP Reno are slow start, congestion avoidance, fast retransmit and fast recovery. It retained the enhancements incorporated into Tahoe, but introduced fast recovery in conjunction with fast retransmit. Using fast recovery significantly improves performance compared to TCP Tahoe when a single packet is lost from a window, but can negatively impact performance for multiple packets drop from a single window of data, which is often true in case of a wireless link.

Newreno: TCP Newreno has the same behaviour like TCP Reno, but includes a small change to the TCP Reno algorithm at the sender side. The change concerns the sender's behaviour during fast recovery when a partial acknowledgement is received that acknowledges some, but not all of the packets that were outstanding at the start of that fast recovery procedure. In TCP Reno, partial acknowledgements take TCP out of fast recovery by decreasing the size of the usable window back to the size of the congestion window. In TCP Newreno, partial acknowledgements do not take TCP out of fast recovery, in fact it waits till all the packets within the current window size are ACKed and then it goes back to the congestion window. TCP Newreno can wait for a timeout in case of multiple losses per window or it can recover without a retransmission timeout by retransmitting one lost packet per round trip time (RTT) until all of the lost packets from that window have been retransmitted.

TCP Vegas: The basic idea of TCP Vegas is to estimate the level of congestion before it happens, and consequently avoid the sending of unnecessary packet that might get dropped. Based on observed RTT, and the size of the sending window, the sender calculates the throughput rate every RTT. This rate is compared to an expected rate, which is calculated based on minimum observed RTT for the connection. If RTT becomes larger, the source will shrink its congestion window, thus reducing its transmission rate. The opposite effect takes place if RTT becomes shorter.
TCP FAST: TCP FAST tries to maintain a constant number of packets in the queues throughout the network. The number of packets in queues is estimated by measuring the difference between the observed RTT and the base RTT, defined as the round trip time when there is no queuing In this respect, it is a direct descendant of TCP Vegas.

III. PERFORMANCE OF DIFFERENT TCP VARIANTS IN UMTS NETWORKS

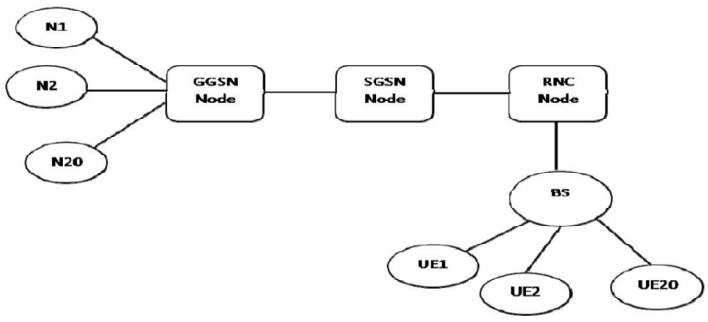


Figure 4.1: The basic simulation topology.

© 2014-16, IJIRIS- All Rights Reserved

Impact of MAC-hs layer Schedulers on TCP

The main component of HSDPA architecture, MAC-hs, is located into Node B. As HS-DSCH is a shared channel, a scheduling algorithm plays a critical role in the allocating the transport channel to different UEs associated with a Node B. At every TTI the MAC-hs checks the status of its flow/priority buffer and Channel Quality (CQI) value of the respective UE, and depending on the scheduling algorithm, determines which packets will be used to construct a MAC-hs PDU for transmission.

Performance comparison results

Through our simulation we prove that there is an optimum range of buffer size and MSS that can elevate performance of given TCP variant significantly in the wireless network. The RTT for the end to end TCP connection can be divided into RTTTCP = RTT wired + RTTUTRAN, where RTT wired is the RTT of the connection in the wired part and RTTUTRAN is the RTT experienced by the connection in UTRAN part. The RLC/MAC-hs buffer size has a significant impact on the RTTUTRAN. As RTTUTRAN increases, the overall RTTTCP also increases. When a TCP segment is received at the RLC layer, it is fragment into *n* RLC PDUs. The buffer at RLC and MAC-hs layers works in conjunction with bandwidth supported by the transport channels at the base station (Node B). As the bandwidth of transport channels at the base station is limited to 1.2Mbits, high RLC/MAC-hs buffer sizes results in a mismatch which creates a bottleneck at the base station. Due to this mismatch high delays are introduce in the UTRAN part which results in high RTTUTRAN. This delay adds to the queuing delay resulting in even longer RTTUTRAN (see figure 4.3). We observe from the figure 4.3 that the average end to end delay of TCP variants increase rapidly after the size of 250PDUs (except for TCP Vegas).

TCP Reno and Newreno are loss based congestion avoidance algorithms that follows Additive Increase Multiplicative Decrease (AIMD) paradigm. The pseudo code for window adjustment is

$$ACK: W \leftarrow W + 1/W$$
$$Loss: W \leftarrow W - \tfrac{1}{2} W$$

.

TCP Reno and Newreno manage their sending window based on packet loss, which is indicated by duplicate ACKs (3) or RTO timeout. For these protocols there is a linear increase in end to end delay when the RLC layer buffer increases (see the figure 4.3). With the gradual increase in observed RTT their RTO steadily grows larger. The main reason is the way RTO timeout is calculated:

$$SRTT = \alpha * SRTT + (1-\alpha)* RTT$$
$$RTTVar = \beta*|SRTT-RTT| + (1-\beta)*RTTVar$$
$$RTO= SRTT + 4*RTTVar$$

Here SRTT is smoothed RTT and RTT is the observed RTT. RTTVar is the variation in RTT. While RTO steadily increases to higher values, the protocol is forced to stays in congestion avoidance for longer duration which results in the slow and steady increment of sending window in a linear fashion. This long waiting results in effective underutilization of network bandwidth ultimately giving fewer throughputs. When the buffer size is too small, the excess of packet are dropped without informing the sender. For loss based protocol this means that they need to wait till RTO timeout or 3 duplicate ACKs from the receiver to indicate them the occurrence of such packet loss. Vegas calculate its sending window based on the observed RTT. The pseudo code for window adjustment is as follows

$$Expected = W/Base\ RTT,$$
$$Actual = W/Observed\ RTT$$
$$Diff = (Expected - Actual)* Base\ RTT$$

If difference Diff < α, than W ← W + 1 or Diff > β, than W ← W - 1 and Diff in the range of (α,β) the congestion is said to be equilibrium as remain unchanged. The equilibrium is the state for the protocol to reach the best data sending rate with minimum possible delay.TCP FAST like TCP Vegas, uses observed RTT for its window calculation.

As the observed RTT increases, TCP FAST assumes it due to be network congestion and decreases the sending rate. TCP Fast is originally designed for high bandwidth delay network (long fat network) [16]. The pseudo code for window adjustment is as follows $W \leftarrow (W^* \text{ Base RTT})/\text{Observed RTT} + \alpha$

It has an inbuilt mechanism for fast converge (through α) helps in reaching the maximum bandwidth utilization stage as soon as possible. Additionally TCPFAST implements an estimation component which computes two pieces of feedback for each data packet sent: 1) a multi-bit queuing delay and 2) one bit loss indication.

© 2014-16, IJIRIS- All Rights Reserved

ISSN: 2349-7017(O)
ISSN: 2349-7009(P)
www.ijiris.com

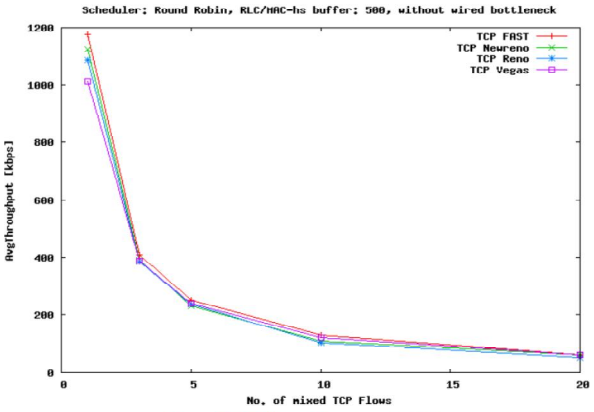


Figure 4.5: Without bottleneck link in wired part

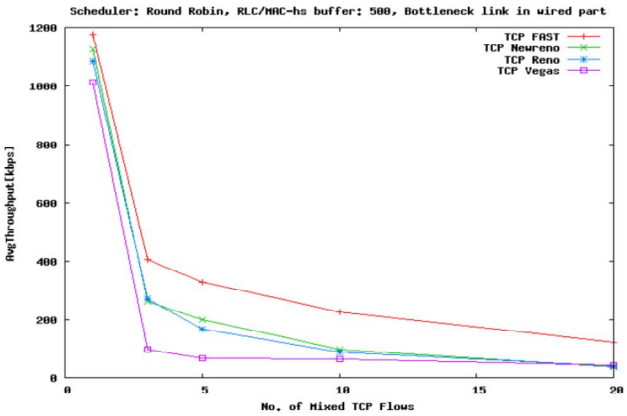
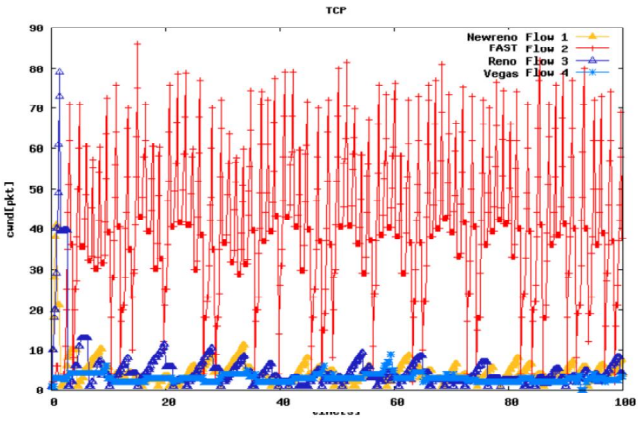
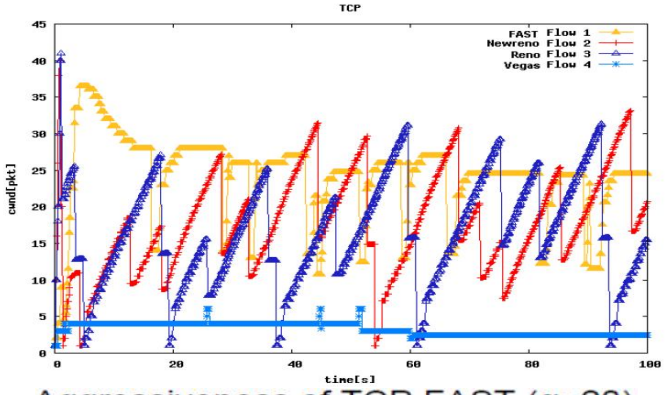


Figure 4.6: With bottleneck link in wired part



Aggressiveness of TCP FAST (α=100)



Aggressiveness of TCP FAST (α=20)

© 2014-16, IJIRIS- All Rights Reserved

CONCLUSION:

We observed that none of the two considered schedulingjustifies both fairness and channel quality utilization at the same time. Also theconcept flow-id based priority is not justified especially in case Max C/Ischeduling. Therefore a new scheduling algorithm is need which allocates radioresources depending upon the variation in the radio channel conditions whilealso implementing some degree of fairness to avoid flow starvation. We also feelthat it is more appropriate to introduce a special Quality of Service (QoS)parameter for high priority flows rather than flow-id based prioritization.we observed that in high buffer scenario delay based TCP tends to perform better than loss based TCP due to their ability to maintain the equilibrium state. In low buffer scenario only TCP FAST performs relatively poor than all the other variants. We observe that in absence of congestion all the TCP variants share the wireless channel bandwidth equally. We also observed that TCP FAST is highly aggressive protocol which needs to be tuned with its fast convergence parameter (α) if it is to be used in wireless long thin networks (LTN). As the future for UMTS unfolds there are lots of opportunities to investigate and optimize the impact of seamless mobility and handovers. The queuing aspect of the MAC-hs will need a revision when handover are considered

REFERENCES:

[1]. C. Zhang and V. Tsaoussidis. TCP Real: Improving real-time capabilities of TCP over heterogeneous networks. Proceedings of the 11th IEEE/ACMNOSSDAV, 2001.
[2]. K. Tan, J. Song, Q. Zhang and M. Sridharan. A Compound TCP Approach for High-Speed and Long Distance Networks. In Proceedings of IEEE INFOCOM, April 2006.
[3]. F. Lefevre and G. Vivier. Optimizing UMTS Link Layer Parameters for a TCP Connection, VTC 2001.
[4]. I.C.C de Bruin, G. Heijenik, M. El Zarki and J. Lei Zan.Fair Channel Dependent Scheduling in CDMA Systems. 12th IST Summit on Mobile and Wireless Communications, Averio, Portugal, June 15-18 2003, pp. 737-741.
[5]. R. Yavatkar and N. Bhagwat. Improving end-to-end performance of TCP over mobile internetworks. Workshop on Mobile Computing Systems and Applications (Mobile 94), December 1994.
[6]. W. Chen, Y. Hsiao, J. Hou, Y. Ge, and M. Fitz. Syndrome: a light-weight approach to improving TCP performance in mobile wireless networks. Wireless Communications and Mobile Computing, (2):37.57, 2002.
[7]. A. Chockalingam, M. Zorzi, and V. Tralli. Wireless TCP performance with link layer FEC/ARQ. In Proceedings of IEEE International Conference on Communications, Vol. 2, 1999.
[8]. R. Yavatkar and N. Bhagwat. Improving end-to-end performance of TCP over mobile internet works. Workshop on Mobile Computing Systems and Applications (Mobile 94), December 1994.
[9]. H. Balakrishnan and R. Katz. Explicit loss notification and wireless web performance. In Proceedings Globe com Internet Mini-Conference, Sydney, Australia, November 1998.
[10]. 3GPP TS 25.322 V5.4.0 Technical Specification Group Radio Access Network; High Speed Downlink Packet Access; Overall Description, Release 5,June 2003
[11]. N. M. Chaskar, T. V. Lakshman, and U. Madhow. TCP over wireless with link level error control: analysis and design methodology. In IEEE Transactions on Networking, vol. 7, no. 5, pp. 605–615, Oct. 1999.
[12]. J. Mo, R. La, V. Anantharam and J. Warland. Analysis and Comparison of TCP Reno and Vegas. In Proceedings of IEEE INFOCOM, March 1999.
[13]. H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz. A comparison of mechanisms for improving tcp performance over wireless link. InProceedings of the AMC Annual Conference of the Special Interest Group on Data Communication (SIGCOMM)96, Stanford, CA, 1996.
[14]. B. Bakshi, P. Krishna, N. H. Vaidya, D. K. Pradhan. Improving Performance over Wireless Networks. 17th Int. Conf. Distributed Computing Systems, Baltimore, and May 1997.
[15]. B. Badrinath and A. Bakre. I-TCP: Indirect TCP for mobile hosts. 15th International Conference on Distributed Computing Systems, Vancouver Canada, May 1995.
[16]. R. Ludwig and R. H. Katz. The Eifel algorithm: Making TCP robust against spurious re-transmission. ACM Computer Communication Review 30, no. 1, pp. 30–36, January 2000.
[17]. R. Ludwig and M. Meyer. The Eifel detection algorithm for TCP. Internet Draft, IETF, February 2002.
[18]. N. Vaidya, M. Mehta, C. Perkins, and G. Montenegro. Delayed duplicate acknowledgements: A TCP-unaware approach to improve performance of TCP over wireless. Technical Report, Computer Science Dept., Texas A&M University, February 1999

AUTHORS PROFILE:

She recieved **B.Tech** and **M.Tech** Degrees in **Computer Science &Information Technology** from Jawaharlal Nehru Technological University,Hyderabad ,AP, INDIA. 2004 & 2007. Currently working as an Assistant Professor of computer science in KING KHALID University K.S.A. Previously worked as Senior Lecturer in JNTU Affliated Engineering Colleges, INDIA

.

© 2014-16, IJIRIS- All Rights Reserved