Computing TCP Throughput in a UMTS Network

Kunde Anand and Vinod Sharma
Department of Electrical Communication Engineering
Indian Institute of Science, Bangalore, India.
Email: {anandk, vinod}@ece.iisc.ernet.in

***Abstract*— We compute the throughput obtained by a TCP connection in a UMTS environment. For downloading data at a mobile terminal, the packets of each TCP connection are stored in separate queues at the base station (node B). Also due to fragmentation of the TCP packets into Protocol Data Units (PDU) and link layer retransmissions of PDUs there can be significant delays at the queue of the node B. In such a scenario the existing models of TCP may not be sufficient. Thus, we provide a new approximate TCP model and also obtain new closed-form expressions of mean window size. Using these we obtain the throughput of a TCP connection which matches with simulations quite well.**
***Keywords*: UMTS, TCP, throughput computation, mean window size.**

I. Introduction

We consider the problem of computing throughput of a TCP (Transmission Control Protocol) connection through a UMTS (Universal Mobile Telecommunication System) network. UMTS is the most widely used 3G wireless cellular network. If one sends data from a host (server) in the wired backbone through a UMTS network then he will most likely be using TCP for transmission. For providing minimum throughput requirement guarantees to a user (the usual quality of service requirement for a data application), one needs to compute the throughput of the TCP connection through the UMTS network (possibly also through the wired part of the Internet).

Over the past few years, extensive research has been done on modelling the TCP. However TCP is a very complicated protocol and there are still important gaps in understanding its behaviour. Thus in this paper we develop an approximate model for TCP which can be particularly useful in studying the performance of 3G cellular systems such as UMTS.

The main performance index for TCP connections is their mean throughput. It has been obtained in [2], [6], [11], [13]-[16], [21] and [24] under various assumptions. For example, [1], [11], [13] and [15] assume iid packet loss. [13] and [14] assume a constant RTT (Round Trip Time). Hence the random queueing delays are ignored. Also in [2], [7], [16] and [21] knowledge of average RTT is required. In [1] RTT is not assumed constant and a detailed model of TCP and channel are analyzed and it presents a TCP throughput expression. However, a closed form expression is not available and analysis is quite involved. In [6], [19] it is assumed that the queue (router) does not become empty. This may be a realistic assumption if the bandwidth-delay product is small and /or a large number of TCP connections are sharing the queue. In [13] and [15] it is assumed that there is no upper bound on the window size. If the receiver limited window size is not very large then this assumption can really affect the throughput approximations.

In a UMTS network a TCP packet is fragmented into several PDUs (Protocol Data Unit). Each PDU is transmitted on the wireless link. The wireless link is error prone. If the received PDU is in error then it is retransmitted on the wireless link. The maximum number of transmission attempts allowed for a PDU is N_1 which can be configured. If the PDU is not received without error in N_1 transmissions then it is dropped. For a TCP packet to be received successfully, all of its PDUs should be received without error. Thus fragmentation and retransmission of PDUs can result in a significant overall transmission time of a TCP packet on the wireless link. Therefore, the resulting queueing delays of a TCP packet in the wireless queue can be significant as compared to the propagation delay. We consider such a scenario and also when the queue may sometimes becomes empty. This scenario seems to have been studied only in [1] where a closed-form expression for throughput is not available. Other studies mentioned above do not provide satisfactory approximation in this scenario. We obtain a closed-form formula for the TCP throughput for this case and our model is very different and much simpler than that in [1].

In our computation of TCP throughput we will use time stationary mean window size. It is obtained in [13]-[15], [17], [20]. But [13] and [15] assume that there is no limit on the window size. Receiver advertised window size limitation can have significant impact on the mean window size. In [14] the authors consider bounded window size and also the losses due to time-outs and triple duplicate acknowledgements. But they do not provide a closed form formula. [17] and [20] gave approximate expressions for TCP time stationary mean window size as a function of the packet loss probability by modifying the formula in [16] which is for the window evolution process considered just before the loss epochs.

We will theoretically obtain time stationary closed form formulae for mean window size which are more accurate than those in [17] and [20]. Our formulae can be used to provide QoS to different applications in a network. In particular we will do this for the UMTS system in this paper.

Due to error prone nature of a wireless link, TCP can perform quite poorly on it. Thus there have been several studies on improving the performance of TCP over wireless networks through various methods. Authors in [8] study the performance of TCP with a link layer automatic repeat request (ARQ) protocol. [7], [8] and [23] study the throughput performance of TCP in UMTS. [23] assumes selective repeat ARQ as the link layer protocol and study its effect on TCP performance in UMTS. In [7] and [8] authors assume a constant RTT in wired portion of the network and obtain delay in the wireless portion using a model for the RLC (Radio Link Control) layer. However, they neglect queueing delays in the end-to-end network. This may not be realistic in UMTS like networks

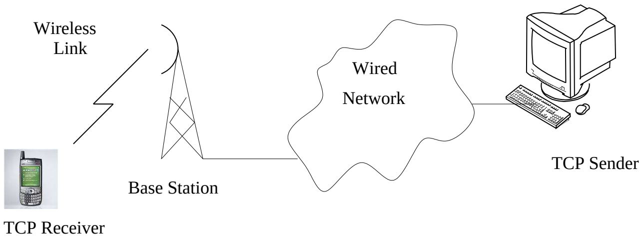

Fig. 1. Typical UMTS reference network scenario

where the queueing delays may be non-negligible compared to the overall RTT. [7] and [8] assume Go-back-N ARQ as the RLC protocol. Authors in [24] obtain bounds on the TCP throughput assuming a selective repeat ARQ protocol. [23] studies the throughput performance without neglecting the queueing delays. But they do not provide closed-form expressions for throughput. In this paper, we study the system with selective repeat ARQ as the link layer protocol which is used in the actual system. We also consider queueing delays at the wireless link. Also, unlike in [23] we obtain closed-form simple expressions for the TCP throughput.

The paper is organized as follows. Section II describes the UMTS and provides preliminary computations. Section III studies the TCP model and discusses the TCP throughput available under different bandwidth-delay product regimes. Section IV obtains new formulae for mean TCP window size. Section V provides a new approximate model of TCP and develops throughput formulae when the queue length is not negligible but the link also may not be utilized fully. Section VI utilizes these formulae to obtain throughput for a TCP connection in a UMTS network. Comparison with simulation results is also provided. Section VII concludes the paper.

II. Model and Reference Network Scenario

Figure 1 depicts a typical UMTS network environment. Different mobile nodes establish TCP connections with their hosts on the wired network. We assume that the MSs (Mobile Stations) are downloading files. We will assume these are persistent connections. However, our formula can be used for ON-OFF TCP connections also, as in [10] and [17]. The TCP connections pass through separate queues at the base station. We assume that the base station allocates sufficient buffer for each queue such that there is no overflow. The base station segments each TCP packet into a fixed number of PDUs and transmits them.

We use the following notation for our analysis. Each TCP packet is segmented into N PDUs of L bits each. The PDUs are transmitted successively one after the other. If a PDU is not successfully received in its first attempt, the sending RLC entity retransmits the PDU until it succeeds. However, there is a limit N_1 on the number of transmission attempts per PDU. If a PDU is not successful in transmission even after N_1 attempts, the PDU and the corresponding TCP packet are discarded from the RLC buffer. The TCP sender has to retransmit such discarded packets.

In the UMTS TDD (Time Division Duplex) radio interface, transmission of PDUs takes place in frames of duration 10 msec. Each frame is further subdivided into 15 time slots. Uplink and downlink transmissions are separated in time within a frame. We allocate the first 8 slots for the uplink to transmit the acks, capacity request messages and other control information and the next 7 slots for the downlink direction to transmit the PDUs and capacity allocation messages. In a slot 4 PDUs can be transmitted. Hence the effective link speed C in the downlink direction is 28 PDUs/frame$=$ 896 kbps. A set of variable spreading factor orthogonal codes are used to spread QPSK modulated data symbols in every slot. We assume these parameters are such that all the PDUs of a TCP packet can be transmitted in the allocated slots for the downlink in a frame. Node B receives acks at the begining of the next frame for the successfully received PDUs and it retransmits the erroneous PDUs along with the PDUs of the next TCP packet. Thus if a TCP packet is discarded, at least some of its PDUs would have been transmitted in N_1 consecutive frames. We assume that each ack/nack is delivered without error.

The probability of successful reception of a PDU depends upon the transmit power, receiver noise, the channel gain (assumed constant here although it is time varying in a wireless channel), the spreading factor and the interference from the other users. These are the important system parameters which should be tuned to obtain proper performance. In the following we obtain the system performance assuming these have been fixed. Our study can be used to tune these parameters. Let p_l denote the PDU error probability in a single transmission attempt. Then $p_u = p_l^{N_1}$ is the probability that a PDU is lost after making N_1 transmission attempts.

We are interested in computing the throughput of a TCP connection in this system. In this section we will compute the first and second moments of the transmission time a TCP packet takes in the wireless link. These will be used later on in computing the throughput.

Let s_k be the number of transmission attempts on the k^{th} PDU of a TCP packet. A TCP packet is *successful* in transmission if each of its PDUs is successfully transmitted within N_1 transmission attempts. Let p be the probability that a TCP packet is lost. Let S_s be the total time taken in transmission of a TCP packet if it is successfully transmitted and let S_l be the time spent if it is not successfully transmitted. Let S be the time taken in transmission by a TCP packet (successful or lost). Then we have

$$p = 1 - (1-p_u)^N, \; S = \sum_{k=1}^{N} s_k T. \tag{1}$$

Where $T = L/C$. Let A be the event that a PDU is successful in transmission within N_1 transmission attempts and let B be the event that a TCP packet is successfully transmitted. Then

$$P(s_k = n|A) = \frac{p_l^{n-1}(1-p_l)}{1-p_l^{N_1}}, \; n = 1, \ldots, N_1 \tag{2}$$

and

$$E[s_k|A] = \sum_{n=1}^{N_1} n \frac{p_l^{n-1}(1-p_l)}{1-p_l^{N_1}}. \tag{3}$$

Hence $E[S_s] = NE[s_k|A]T$. Let S_d be the number of transmission attempts by a dropped TCP packet. Then we have

(because of the specific selective ARQ mentioned above)

$$E[S_d] = \sum_{m=1}^{N} P(m \text{ PDUs are lost}|B^c) (N_1.m + (N-m)E[s_k|A]) \quad (4)$$

where

$$P(m \text{ PDUs are lost}|B^c) = \frac{P(m \text{ PDUs are lost})}{P(B^c)} \quad (5)$$

where $P(m \text{ PDUs are lost}) = \binom{N}{m} p_u^m (1-p_u)^{N-m}$ and $P(B^c) = 1-(1-p_u)^N$.

Also, $E[S_l] = E[S_d]T$. Thus we have

$$E[S] = pE[S_l] + (1-p)E[S_s]. \quad (6)$$

We will also need the second moments,

$$E[S^2] = pE[S_l^2] + (1-p)E[S_s^2] \quad (7)$$

where

$$E[S_s^2] = T^2 E[(\sum_{i=1}^{N} s_i)^2|B] \quad (8)$$

$$= T^2[\sum_{i=1}^{N} E[s_i^2|A] + N(N-1)(E[s_1|A])^2]$$

$$E[S_l^2] = E[S_d^2]T^2 \quad (9)$$

$$E[S_d^2] = \sum_{m=1}^{N} P(m \text{ PDUs are lost}|B^c). \quad (10)$$

$$(N_1.m + (N-m)E[s_k|A])^2 \quad (11)$$

III. TCP Model and Throughput Computation

In this and following sections we provide a TCP model and its mean throughput computation which can be useful not only in the UMTS context but also else where. Thus our model may be more general than needed in the present context.

We consider a system with several TCP connections passing through one or more routers / queues (Fig.2). The TCP connections are long-lived and they always have packets to transmit. Our results can be generalized to ON-OFF TCP connections [10].

We will consider a single router. Our results have been generalized to multiple routers in [4]. We obtain the TCP throughput under the following different scenarios:

1) The router queue never becomes empty.
2) Queueing delay is negligible compared to overall RTT.
3) Queueing delay may not be negligible compared to RTT and also the queue may sometimes become empty.

Together, these three cases cover all possible scenarios.

We use the following notation in our analysis. There are N long-lived (persistent) TCP connections. For the i^{th} TCP connection, $W_{\max}(i)$ is the maximum window size, Δ_i is the total propagation delay (outside the queue) and s_i is the packet length in bits (assumed $i.i.d$ for each connection, independently of packet lengths of other connections). Packets of i^{th} TCP connection are dropped independently with probability p_i. The link speed is C bps. Then from [19] we have a

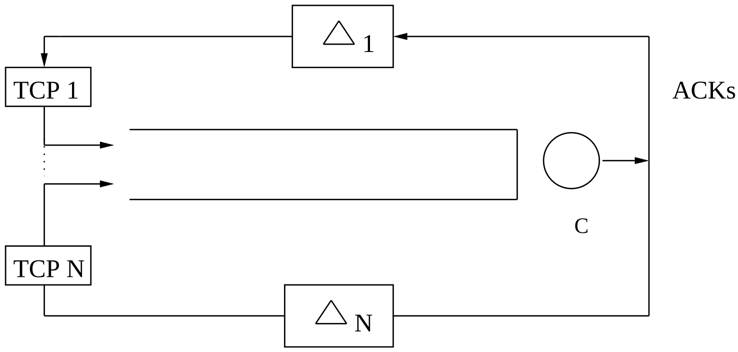

Fig. 2. Multiple TCP flows through a single router with fixed rate C

stationary distribution for the system. Let $E[W_i(p_i)]$ be the time stationary mean window size and $E[q_i]$ be the mean number of packets at the router queue for the i^{th} connection.

Let λ_i be the throughput obtained (in packets/sec) for the i^{th} TCP connection. From Little's law, $\lambda_i \Delta_i$ is the mean number of TCP i packets propagating outside the queue. Thus,

$$E[q_i] = E[W(p_i)] - \lambda_i \Delta_i. \quad (12)$$

We approximate the mean sojourn time $E[S]$ at the router queue by $\sum_{j=1}^{N} E[q_j]E[s_j]/C$. This is approximately same for all packets. Thus the throughput λ_i obtained by the i^{th} TCP connection can be approximated by

$$\lambda_i = \frac{E[W_i(p_i)]C}{\sum_{j=1}^{N} E[q_j]E[s_j] + \Delta_i C} \text{ packets/sec}, \ i = 1, \ldots, N. \quad (13)$$

This formula was shown to be reasonably accurate in [19].

In (13) we have N equations with N unknowns $\lambda_1, \ldots, \lambda_N$. However these are not linearly independent and hence require more constraints to provide a unique solution. These constraints depend upon the particular scenario one is involved in.

In scenario 1 when the queue never becomes empty, the total throughput from the queue is C bps. Thus,

$$\sum_{i=1}^{N} \lambda_i E[s_i] = C. \quad (14)$$

Along with the N equations (13), (14) provides a unique solution λ_i , $i = 1, \ldots, N$ provided the $E[W_i(p_i)]$, $i = 1, \ldots, N$ are known.

In the setup of the second scenario, the throughput of connection i can be approximated by $E[W_i(p_i)]/(\Delta_i + \frac{E[s_i]}{C})$ if $E[W_i(p_i)]$ is available.

In the UMTS setup of Section II, the probability of packet loss p_i will be known (or controllable) based on other system parameters (as explained in the last section). Often the probability of loss p for a TCP connection depends upon the queue length. For example if RED (Random Early Detection) [9] is used in the queue then the probability p depends upon the RED parameters (for a TCP connection) and the average queue length. But the average queue length also depends upon the other connections (see [19] for analysis of such a system). In more usual scenario, when there is a finite buffer and tail-drop, the probability with which the packets of a TCP connection are

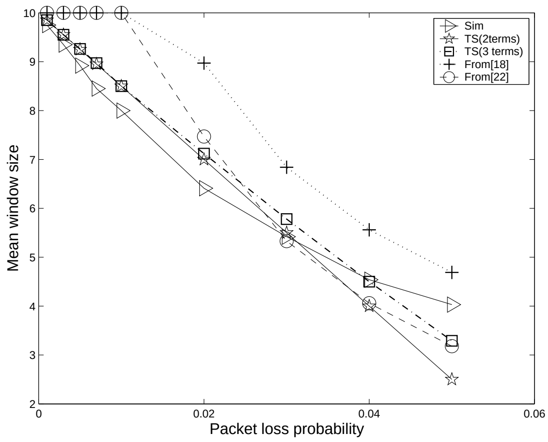

Fig. 3. Comparison of $E[W(p)]$: Taylor series

dropped depends upon the other system parameters. In such cases the throughputs obtained by different TCP connections are more complicated to compute (see, e.g., [1]).

We consider the scenario 3 in Section V, where the queueing delay at the router may not be negligible compared to the overall RTT and also the queue may become empty sometimes (and hence (14) may not provide a good approximation). This is a likely scenario in UMTS (especially because at the base station each connection passes through a separate queue).

As remarked earlier even for scenarios 1 and 2 our approximation (13) for throughput requires computing time stationary $E[W_i(p_i)]$. The $E[W_i(p_i)]$ can be computed approximately from [17], [20]. We do not know any other closed form formulae in the literature. For our purposes, it will be useful to have closed form expressions. The formulae for $E[W(p)]$ in [17], [20] are adhoc and provide reasonable approximation only for low values of p. Thus in the next section we will theoretically obtain better formulae for $E[W(p)]$.

IV. Computing $E[W(p)]$

In this section we obtain the time-stationary distribution π_p of the TCP congestion window size and then the TCP steady state mean window size $E[W(p)]$ as a function of the packet loss probability p.

We will assume that the packets are dropped independently of each other. This assumption is not satisfied in a Tail-drop queue but is satisfied if RED is deployed at the router or in a UMTS system considered in Section II when sufficient interleaving is present in the physical layer. We consider only one TCP connection in this section. It is persistent and uses Reno. The key assumptions in this section are:

1) There is a receiver buffer limitation on the TCP sender's congestion window with the maximum window size $W_{\max}$.
2) TCP ack packets are always successfully delivered to the sender.
3) TCP packet losses are independent events with equal probability p.
4) The effect of all packet losses in a window is seen at the end of transmission of a window of packets (for simplicity one could think of this as a scenario when the RTT is constant and much larger than $W_{\max}s$ i.e., bandwidth-delay product is large).

Most of the above assumptions are commonly made in literature. TCP-Reno version is the most used version in practice. The slow start phase lasts for a negligible time. Also, assuming TCP packet losses to be $i.i.d$ is of immediate relevance to us.

Under the above assumptions the TCP window evolution process seen after each (constant) RTT is a discrete time Markov chain (DTMC) with the transition probabilities

$$\begin{aligned} W_{k+1} &= \min(W_k+1, W_{\max}) \text{ with probability}(1-p)^{W_k} \\ &= \max(1, \lfloor W_k/2 \rfloor), \text{ otherwise.} \end{aligned} \tag{15}$$

We denote its transition matrix by P_p. The first equality in (15) represents the probability of no packet loss in a window when the window size is W_k. In this case the TCP increments the congestion window by one packet. The second equality represents the case where atleast one packet loss occurs. Hence TCP halves the congestion window. Here we ignore the probability that more than one packet is lost in a window. Thus time-outs are ignored. This will be a good approximation for a small p. We will consider refinements later on.

The above TCP congestion window evolution process $\{W_k\}$ is an irreducible, finite state, aperiodic Markov chain. Hence this DTMC has a unique steady state probability vector π_p (all vectors will be taken as column vectors). It is easy to compute $E[W(p)]$ from π_p for any p. However, we are interested in developing closed form expressions for $E[W(p)]$. This can be useful not only in the present context but also, for example, in [10], [17] and [19]. For this we expand π_p as a Taylor series around the point $p=0$. This is done because p is usually small in practice (at least should be) and π_o is known: $\pi_o(i)=0$ except when $i=W_{\max}$. We will show below that π_p has derivatives of all orders w.r.t. p. The Taylor series expansion for π_p is

$$\pi_p = \pi_0 + p\frac{\partial \pi_p}{\partial p}|_{p=0} + \frac{p^2}{2!}\frac{\partial^2 \pi_p}{\partial p^2}|_{p=0} + \ldots + \frac{p^n}{n!}\frac{\partial^n \pi_p}{\partial p^n}|_{p=0} + \ldots \tag{16}$$

From (15) we see that P_p is differentiable in p. Thus, the derivatives of π_p can be obtained as

$$\frac{\partial \pi_p}{\partial p} = \frac{\partial \pi_p}{\partial Vec(P_p)}\frac{\partial}{\partial p}Vec(P_p) \tag{17}$$

where the Vec operator ([12]) is defined as follows. If

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \tag{18}$$

then $Vec(A)$ denotes $(a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}, a_{31}, a_{32}, a_{33})^{\mathrm{T}}$. The derivative of a vector $X=(x_1, x_2, \ldots, x_n)^{\mathrm{T}}$ with respect to a vector $Y=(y_1, y_2, \ldots, y_m)^{\mathrm{T}}$ is the Jacobian ([12]).

However π_p is obtained from $\pi_p^{\mathrm{T}} P_p = \pi_p^{\mathrm{T}}$ along with the constraint $\sum_i \pi_p(i) = 1$. Thus we can not directly obtain $\frac{\partial \pi_p}{\partial Vec(P_p)}$. We use the Implicit Function Theorem [18] on the equation

$$f(P_p, \pi_p) = (I-P_p)^{\mathrm{T}}\pi_p + (\pi_p{}^{\mathrm{T}} e - 1)e \ . \tag{19}$$

where e is a column vector with all 1's and I is the identity matrix. Then π_p is a stationary distribution of P_p only if (P_p, π_p) is a zero of the function $f(P_p, \pi_p)$. The term $(\pi_p^{\mathrm{T}} e - 1)e$ on the right hand side corresponds to the constraint $\sum_i \pi_p(i) = 1$.

We can show that the conditions for the implicit function theorem are satisfied by $f(P_p, \pi_p)$ and

$$\begin{aligned} \frac{\partial \pi_p}{\partial p} &= -\nabla_{\pi_p} f(P_p, \pi_p)^{-1} \nabla_{Vec(P_p)} f(P_p, \pi_p). \quad (20) \\ & \nabla_p Vec(P_p) \\ &= ((I - P_p + A)^{-1})^T_{n \times n} (\pi_p^T \otimes I)_{n \times n^2}. \\ & (\nabla_p P_p)_{n^2 \times 1} \end{aligned}$$

where A is an $n \times n$ matrix of all 1's and $A \otimes B$ is the Kronecker product of matrices A and B ([12]).

Using above results, the Taylor series expansion for π_p with two terms and $W_{\max} = 10$, is

$$\pi_p = (0, 0, 0, 0, 10p, 10p, 10p, 10p, 10p, 1 - 50p)^T \quad (21)$$

and $E[W(p)] = 10(1 - 15p)$. For $W_{max} = M$,

$$\begin{aligned} E[W(p)] &= \frac{p}{2}(-0.5 + \frac{3M}{4})M^2 + M(1 - 0.5pM^2) \\ & \quad \text{if } M \text{ is even,} \\ &= 1.5\lfloor M/2 \rfloor (1 + \lfloor M/2 \rfloor) pM + \\ & \quad M(1 - \lceil M/2 \rceil Mp) \\ & \quad \text{otherwise.} \quad (22) \end{aligned}$$

This formula gives reasonable results for small p and/or small M (≤ 20). We can obtain better accuracy for π_p if we use the third term also in (16). Then for $W_{max} = 10$, $E[W(p)] = 5(2 - 30p + 63p^2)$.

Similarly one can include higher order terms of the Taylor series to get better accuracy for higher packet loss probabilities p.

The accuracy of the model can be further improved for high p by including the effect of time-outs and multiple packet losses within a window of transmission in our Markov chain model. With the new P_p, one can again use the above methods to efficiently obtain π_p and $E[W(p)]$.

A. Simulations

In this section we verify the accuracy of $E[W(p)]$ obtained in the previous section via the different methods. We use NS2 simulations. We consider a single TCP (Reno) connection between two nodes with a packet size 1000 bytes, link speed 1 Mbps, total propagation delay (excluding queueing delay) 150 msec and receiver buffer limitation $W_{\max} = 10$. We consider packet loss probabilities p from 0.001 to 0.05 and compare the simulated $E[W(p)]$ with the $E[W(p)]$ obtained via the Taylor series around p=0. We also consider the other formulae used in [17] and [20]. These results are provided in Fig 3 which shows that the Taylor series provides much more accurate formulae than those in [17] and [20].

V. TCP Throughput for Non negligible Queueing Delay and Idle time

In this section we consider the scenario 3 of Section III:

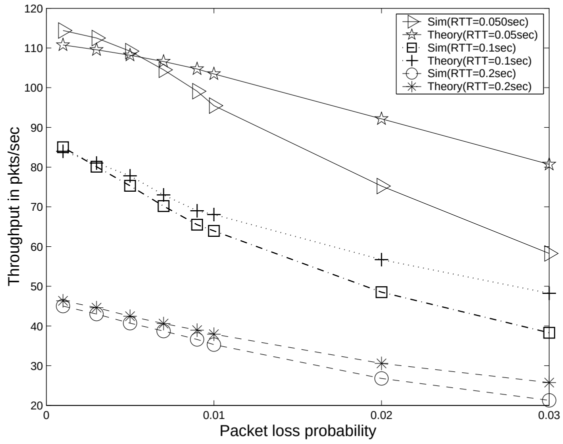

Fig. 4. Comparison of throughput in pkts/sec from $M/GI/1$ approximation

1) Queueing delay and propagation delay both are non-negligible.
2) Queue can become empty for non-negligible time.

We limit ourselves to a single queue. Generalization to multiple routers will be considered elsewhere. We will use the following notation. The TCP packet lengths will be $i.i.d$ with a generic length s bits, the propagation delay is Δ sec, and the link speed is C bps. The stationary mean sojourn time in the queue is $E[S]$ sec and the throughput obtained is λ packets/sec.

We approximate the router queue by an $M/GI/1$ queue. However its arrival rate will be obtained by taking into account the TCP dynamics. Then ([5])

$$E[S] = \frac{\lambda E[(s/C)^2]}{2(1-\rho)} + \frac{E[s]}{C} \quad (23)$$

is the mean sojourn time of the packets in an $M/GI/1$ queue with the given arrival rate and the packet size s where $\rho = \frac{\lambda E[s]}{C}$. Thus, when the window size of the TCP connection is fixed at $W_{\max}$ we can approximate the throughput by

$$\lambda = \frac{W_{\max}}{\frac{\lambda E[(s/C)^2]}{2(1-\rho)} + E[\frac{s}{C}] + \Delta} \quad \text{packets/sec.} \quad (24)$$

This equation has a unique positive solution which provides us the throughput λ attained by the TCP connection.

If the packets are getting dropped with probability p, then we use the approximation with $E[W(p)]$ replacing $W_{\max}$ in (24) where $E[W(p)]$ is obtained from Section IV.

In [4] we have extended this approximation to the system where multiple TCP connections pass through multiple links/routers.

A. Simulations

We compare the throughput obtained via the $M/GI/1$ approximate model of TCP studied in this section with NS simulations. We consider a single TCP connection with 1 Mbps link speed, $W_{max} = 10$ and 1000 byte packet size. We show the results for propagation delays (excluding the queueing delays) 50 msec, 100 msec and 200 msec and p from 0.001 to 0.03. This way we will cover all the three scenarios considered above. The results are provided in Fig 4. One sees a good match with simulations till $p = 0.01$.

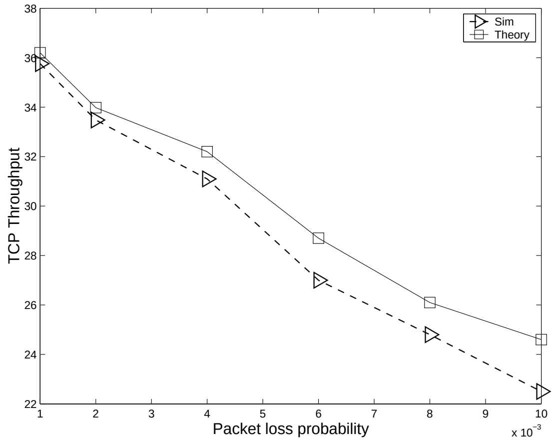

Fig. 5. Throughput for UMTS in pkts/sec

In [4] we have compared our results with those in [14], [23] and some other approximations and have found $M/GI/1$ approximation to work better. Due to lack of space we do not present those results here.

VI. UMTS THROUGHPUT

In this section we use the $M/GI/1$ approximation of Section of V to compute the throughput of a TCP connection passing through a UMTS network. The scenario considered is explained in Section III. The wireline network will be replaced by a fixed propagation delay of Δ. The queue at node B is assumed to be the bottleneck where PDUs are being stored for transmission on the wireless link (Fig. 1). Thus the difference from Section V is that each TCP packet is being segmented into PDUs and ARQ is being used at the wireless link.

For throughput computation we again use the $M/GI/1$ queue approximation. Thus, the throughput is obtained from equations (23) and (24) but now the mean $E[s/C]$ and the second moment $E[(s/C)^2]$ of transmission time of the TCP packets are replaced by $E[S]$ and $E[S^2]$ obtained in (6) and (7) in Section II.

We compare the TCP throughput obtained in a UMTS network via the $M/GI/1$ approximate model with that obtained from an $ns2$ based UMTS simulator [22]. We consider a single TCP Reno connection with a packet size of 1000 Bytes and $W_{max} = 10$, where a mobile node is the TCP receiver with the TCP sender in the wired part of the network as shown in Fig. 1.

Various parameters we consider in our simulations are: PDU size $= 40$ bytes, $N = 25$, $N_1 = 4$ and the propagation delay in the wired part of the network is 250 msec. Wireless link bandwidth$= 1.92$ Mbps. The throughput obtained via simulations and via the $M/GI/1$ approximation is provided in Fig 5 for different values of TCP packet loss probability p. One sees a reasonably good match between the theory and simulations especially for small values of p.

VII. CONCLUSIONS

In this paper we obtain an approximate model for computation of throughput of a TCP connection in a UMTS network. Due to the segmentation of the TCP packets into PDUs and retransmission of PDUs at link layer, there can be significant random queueing/transmission delays at the wireless links. On the otherhand the link may not be fully utilized. For such a scenario the existing TCP models do not provide accurate throughput. Therefore, we develop new models to compute the mean TCP window size and the throughput. We have shown that our theoretical model provides a good approximation to the TCP throughput in a UMTS network.

REFERENCES

[1] A.A. Abouzeid, S. Roy and M. Azizoglu, "Comprehensive Performance Analysis of a TCP Session Over a Wireless Fading Link With Queueing", *IEEE Trans. Wireless Communications*, Vol. 2, NO. 2, March 2003.

[2] E. Altman, K. Avrachenkov and C. Barakat, "A Stochastic Model of TCP/IP With Stationary Random Losses", *IEEE/ACM Trans. Networking*, Vol. 13, NO. 2, April 2005.

[3] K. Anand, "Providing QoS to TCP and Real-time connections in a UMTS Network", MSc(Engg) Thesis under preperation, ECE Dept, Indian Institute of Science.

[4] K. Anand, V. Sharma, "Simple Models for Performance Analysis of TCP Connections", Submitted.

[5] D. Bertsekas and R. Gallager, "Data Networks", Prentice Hall, 1992.

[6] T. Bu and D. Towsley, "Fixed Point Approximations for TCP Behaviour in an AQM Network", *ACM SIGMETRICS Performance Evaluation Review*, Vol. 29, June 2001.

[7] A-F. Canton and T. Chahed, "End-to-end Reliability in UMTS: TCP over ARQ", *in Proc. IEEE Globecom*, November 2001.

[8] T.Chahed, A-F.Canton, S-E. Elayoubi, "End-to-end TCP Performance in W-CDMA / UMTS", *in Proc. IEEE ICC*, 2003.

[9] S. Floyd, "Random Early Detection Gateways for Congestion Avoidance", *IEEE/ACM Trans. Networking*, Vol. 1, NO. 4, August 1993.

[10] A. Gupta and V. Sharma, " A Unified Approach for Analyzing Persistent, Non-persistent and ON-OFF TCP Sessions in the Internet", *Performance Evaluation*, Vol. 63, NO. 2, February 2006.

[11] T.V. Lakshman and U. Madhow, "The Performance of TCP/IP for networks with high bandwidth-delay products and random loss", *IEEE/ACM Trans. Networking*, Vol. 5, June 1997.

[12] J. R. Magnus and H. Neudecker, "Matrix Differential Calculus with Applications in Statistics and Econometrics", John Wiley & Sons, 1988.

[13] M. Mathis, J. Semke and J. Mahdavi, "The Macroscopic behaviour of TCP Congestion Avoidance Algorithm", *in Proc. ACM SIGCOMM*, 1997.

[14] V. Misra, W.-B. Gong and D. Towsley, "Stochastic Differential Equation Modeling and Analysis of TCP-Windowsize Behaviour", *in Proc. Performance'99 Conf*, Istanbul, Turkey, 1999.

[15] T.J. Ott, J.H.B. Kemperman and M. Mathis, "The Stationary Behaviour of Ideal TCP Congestion Avoidance", *available at*: ftp://ftp.telecordia.com/pub/tjo/TCPwindow.ps, Aug 1996.

[16] J. Padhye, V. Firouiu and D. Towsley, " Modeling TCP Reno Performance: A Simple Model and Its Empirical Validation", *IEEE/ACM Trans. Networking*, Vol. 8, NO. 2, April 2000.

[17] V.V. Reddy, V. Sharma and M.B. Suma, "Providing QoS to TCP and Real Time Connections in the Internet", *Queueing Systems*, pp.461-480, 2004.

[18] W. Rudin, "Principles of Mathematical Analysis", McGraw-Hill, 1976.

[19] V. Sharma and P. Punyaslok, "Performance analysis of TCP connections with RED control and exogenous traffic", *Queueing Systems*, Vol.48, pp. 193-235, 2004.

[20] H. Shetiya and V. Sharma, "Algorithms for Routing and Centralized Scheduling to Provide QoS in IEEE 802.16 Mesh Networks", *in Proc. WMuNeP*, Montreal, Canada, Oct' 2005.

[21] B. Sikdar, S. Kalyanaraman and K.S. Vastola, "Analytic Models for the Latency and Steady-State Throughput of TCP Tahoe, Reno and Sack", *IEEE/ACM Trans. Networking*,Vol. 11, NO. 6, Dec'2003.

[22] A. Todini and F. Vacirca, "UMTS module for NS", *available at*: http://net.infocom.uniroma1.it

[23] F. Vacirca, A.D. Vendictis and A. Baiocchi, "Optimal Design of Hybrid FEC/ARQ Schemes for TCP over Wireless Links with Rayleigh Fading", *IEEE Trans. Mobile Computing*, Vol. 5, NO. 4, April 2006.

[24] J. Zhu and S. Roy, "Modeling TCP over Selective Repeat ARQ in Wireless Networks with Non-Negligible Propagation Delay", *in Proc. IEEE ICC*, 2003.