TCP fluid modeling with a variable capacity bottleneck link

Andrea Baiocchi and Francesco Vacirca
INFOCOM Dept., University of Roma "La Sapienza"
Via Eudossiana 18 - 00184 Roma, Italy, Phone: +39-0644585654, Fax: +39-064744481
Email: andrea.baiocchi@uniroma1.it, vacirca@infocom.uniroma1.it

***Abstract*—A single TCP connection with a time varying capacity bottleneck link is considered. The bottleneck capacity varies arbitrarily and independently of the TCP connection traffic; it aims at reproducing typical situations where capacity is modulated by exogenous process, such as wireless channel with link adaptation or bursty channels.**

The key point is to understand the interplay of the TCP congestion control evolution and the time constants of the bottleneck link capacity time variation. Fluid modeling is used to describe the time evolution of the congestion window size and of the bottleneck buffer content with a completely general capacity time function. ns-2 based simulations are used as mean to assess the fluid model accuracy. Numerical results show the existence of a "resonance" phenomena, that can significantly degrade the TCP long term performance (even more than halved). The degradation depends on the ratio between the fundamental time constant of the link capacity variation and the TCP average round trip time.

I. Introduction

TCP congestion control follows the so called Additive Increase Multiplicative Decrease (AIMD) principle: as TCP does not observe network congestion, it increases linearly its permitted sending rate, by opening its congestion window; as soon as TCP detects network congestion, it halves its congestion window in order to reduce its own sending rate. Standard TCP uses packet losses as an indication of network congestion. The sending rate is based on a self-clocking mechanism, which is a fundamental part of the classic Van Jacobson TCP congestion control [1]. For this reason, it is also said that the TCP dynamics is ack-clocked. An important consequence of the ack-clocking is that a TCP flow can be heavily affected by time-varying capacity of the forward and reverse links crossed by the TCP connection, since the sending rate is based on a delayed version of the time-varying capacity.

The analysis of a TCP connection congestion control with a single bottleneck link is a classic well known topic. In [2] the steady state throughput of a single TCP connection is computed by means of a simple analytical model that captures not only the fast retransmit events of TCP, but also the retransmission timeout events. In [3], TCP performance is derived by means of analytical models when the crossed links are shared with multimedia traffic, and random losses can be provoked by transient fluctuations in real time traffic which may cause irregularly spaced losses for data traffic.

This work has been partially supported by the Italian Research Ministry under the PRIN FAMOUS grant.

In the recent literature, the interaction between TCP congestion control loop and variable capacity links has been extensively analyzed in the context of TCP interaction with available bit rate (ABR) service in ATM networks. In this scenario ABR sessions utilize the remaining capacity left over by CBR and VBR services. In [4], by means of analytical tools and simulation results, the interaction between the end-to-end TCP congestion control and the explicit rate-based congestion control of ABR is studied. The proposed model is able to capture the effect of slow bottleneck rate variations (with respect to TCP end-to-end round trip time - RTT) but not the effect of comparable and fast variations. These two cases are analyzed by means of simulation experiments. In [5], the authors explore the performance of a modified version of TCP congestion control that exploits the explicit notification of bottleneck rate. Through an analytical model, it is shown that in a variable capacity scenario the proposed congestion control mechanism outperforms standard TCP congestion control.

We aim at extending fluid models to the consideration of a variable capacity bottleneck link. Time variability of the bottleneck link capacity is arbitrary and assumed to be exogenous and unaffected by the TCP connection traffic. A typical application scenario is the interaction of TCP congestion with a wireless channel that employs a link adaptation scheme that modifies the TCP available capacity independently of the TCP behavior and depending on the measured link quality. The proposed model can be exploited to optimize the link layer configuration to enhance TCP performance, e.g. adaptive modulation refresh rate. A variable capacity can also mimic bursty transmission channels like in a wireless access, where each user radio session is granted a given amount of capacity by cycling among many users; this can be modeled as an interrupted server, resulting in a time varying modulated capacity link. A time-varying link capacity model also enables the study of transient behavior of TCP connection, e.g. the effect of a temporary capacity shut down during handoff. Another application scenario includes the capacity sharing of network links between TCP elastic traffic and higher priority traffic, e.g. delay sensitive traffic or prioritized data traffic.

In this paper, we characterize the TCP congestion control behavior by means of a fluid model approach that describes the dynamics of the TCP congestion control window and of the bottleneck buffer content as a function of the bottleneck link time varying capacity. Fluid models have recently raised a large interest (e.g. [6], [7], [8], [9]) due to the relatively simple and compact description they allow, especially for general time varying behavior of closed loop systems. Except of special simple cases, they seldom yield to analytic closed solution; rather they provide a powerful way to define quite accurate dynamic evolution equations of a system, that can be used as a "simulation" tool in place of usually much slower and cumbersome discrete event, packet aware simulation software.

By means of the developed fluid model, we are able to show the performance of TCP obtained when the time scale of rate variability is lower, comparable and higher than the TCP connection round trip time, highlighting the existence of a "resonance" phenomena, that can significantly degrade the TCP long term performance. The analysis shows that TCP performance degrades when capacity variation time constant is of the order of magnitude of the end-to-end round trip delay. In this case, in fact, the TCP sending rate is based on a wrong picture of the network condition. Moreover the analysis is able to highlight the effect of bottleneck buffer size in the perceived connection performance as done in [10] in case of costant bottleneck capacity.

The paper is organized as follows. In Section II the basic TCP congestion control mechanisms are described. Section III introduces the fluid model of a single bottleneck with a general time-varying capacity. Section IV is devoted to assessing the fluid model accuracy as compared to ns-2 [11] based simulations and to show the average link utilization versus the variation rate of the bottleneck capacity; we also apply the fluid model analysis in a case study example of the time varying capacity function. Finally, in Section V, the main conclusion are drawn.

II. Background on TCP congestion control

The basic TCP congestion control is essentially made of a probing phase and a decreasing phase. The probing phase of standard TCP consists of an exponential phase (i.e. the "Slow Start" phase) and a linear increasing phase (i.e. the "Congestion Avoidance" phase). The probing phase stops when congestion is experienced in the form of timeout or reception of $DupThresh$ duplicate acknowledgments (DUPACKs); the default value of $DupThresh$ is 3. The TCP dynamic behavior in "steady state" condition can be considered with good approximation a sequence of congestion avoidance phases followed by reception of $DupThresh$ DUPACKs. When $DupThresh$ DUPACKs are received, the TCP implements a multiplicative decrease behavior. The generalization of the classic additive increase multiplicative decrease TCP settings can be made as follows:

a) On ACK reception
$cwnd \longleftarrow cwnd + a(cwnd)$
b) When $DupThresh$ DUPACKs are received
$cwnd \longleftarrow cwnd - b \cdot cwnd$

where $a(cwnd)$ is $1/cwnd$ and b is $1/2$ for standard TCP. When the packet loss is detected, TCP enters a recovery phase that can differs according to different TCP versions (NewReno TCP [12] and SACK [13]). In particular, NewReno TCP recovery phase is based only on the cumulative ACK information whereas SACK TCP receiver exploits the TCP Selective Acknowledge option [14] to advertise the sender about out-of-order received blocks. This information is employed by the sender to recover from multiple losses more efficiently than NewReno TCP.

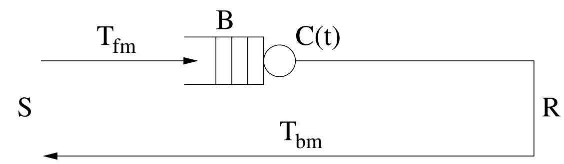

Fig. 1. Schematic of a TCP connection.

In Figure 1, the TCP connection scheme in a single bottleneck scenario is depicted. When the bottleneck capacity does not vary, it is easy to show by means of a fluid window approximation and by considering only the Congestion Avoidance phase, that the TCP link utilization is given by:

$$\rho = \frac{3}{4} \cdot \frac{(1+\beta)^2}{1+\beta+\beta^2}$$

with $\beta = \min\{1, B/(C \cdot (T_{fm} + T_{bm}))\}$, where B is the bottleneck buffer size (in packets) and $C \cdot (T_{fm} + T_{bm})$ is the bandwidth-delay product (in packets).

III. Fluid model

Let us consider an ever lasting TCP connection delivering equal size segments, say L bytes long. The TCP connection has a single bottleneck link of capacity C (packets/s) equipped with a buffer of size B (packets) (see Figure 1). The bottleneck link capacity available to the TCP connection is time dependent, denoted with $C(t)$, with $0 \leq C(t) \leq C$.

Further, we make the following assumptions.

- The Slow Start phase is neglected; all other main TCP procedures are considered, namely congestion avoidance, fast retransmit, fast recovery, time-outs.
- Internal TCP buffers have infinite size.
- The advertised window is always larger than the congestion window.
- Packet loss is only due to overflows of the bottleneck link buffer.
- Upon packet loss detection, via multiple duplicated acknowledgment receptions, the congestion window is scaled down by a factor $1-b$ with $0 < b < 1$.

A deterministic fluid model approach is used, i.e. we neglect the discrete description of packetized information flows and deal with TCP related quantities as taking values in an interval of the real axis, with arbitrarily small information chunks being transferred in the TCP pipe and stored in the bottleneck buffer.

We denote with $W(t)$ the congestion window at time t. The bottleneck buffer content at time t is denoted by $Q(t)$. Time is aligned with TCP sender and the base Round Trip Time (RTT) T_m is split into the forward base RTT, T_{fm}, from the

TCP sender to the bottleneck buffer input, and the backward base RTT, T_{bm}, from the output of the bottleneck buffer to the TCP receiver and then back to the TCP sender[1].

Let $R(t)$ be the sending rate of the TCP sender at time t. The input rate at the bottleneck buffer is $\lambda(t) = R(t-T_{fm})$. It must always be $0 \le Q(t) \le B$, where B is the buffer size. For $0 < Q(t) < B$ the buffer content satisfies the differential equation $\dot{Q}(t) = \lambda(t) - C(t)$. The buffer content fluctuates between two reflecting barriers, namely 0 and B. Once the buffer gets empty at time t, which can only occur if $\lambda(t) < C(t)$, it will stay empty for all $\tau \ge t$ such that $\lambda(\tau) \le C(\tau)$ and in the meanwhile, the output rate is $\lambda(\tau)$. As long as the buffer is non-empty, the output rate is $C(t)$. When the buffer gets full at time t, which implies that $\lambda(t) > C(t)$, it will stick to the upper limit B for all $\tau \ge t$ such that $\lambda(\tau) \ge C(\tau)$. These are the boundary conditions of the buffer content differential equation.

The output rate of the buffer link is denoted by $\mu(t)$. It is

$$\mu(t) = \begin{cases} \lambda(t) & Q(t) = 0 \\ C(t) & Q(t) > 0 \end{cases}$$

TCP sender receives ACK packets at rate $R_{ACK}(t) = \mu(t - T_{bm})$. The congestion window is updated as a function of R_{ACK}. In case of TCP Reno, the variation of the congestion window is proportional to R_{ACK}, i.e. it is increased by $1/W(t)$ for each received ACK in sequence. The differential equation is

$$\dot{W}(t) = \frac{\mu(t-T_{bm})}{W(t)} = \begin{cases} \dfrac{R(t-T_m)}{W(t)} & Q(t-T_{bm}) = 0 \\ \dfrac{C(t-T_{bm})}{W(t)} & Q(t-T_{bm}) > 0 \end{cases} \tag{1}$$

The connection between the sending rate $R(t)$ and the congestion window size can be derived by expressing the number of packets in flight at time t and equating this number with $W(t)$. The amount of information emitted by the TCP sender and still waiting to be acked amounts to all that sent in a RTT, i.e. in the time interval elapsing since a piece of information is emitted by the TCP sender, say at time t_0, until the relevant ACK is received back at the sender, say at time t. The starting time t_0 can be found from $t_0 + T_{fm} + D(t_0 + T_{fm}) + T_{bm} = t$, which is an implicit relation between t and t_0. Here $D(t)$ denotes the waiting time of a piece of information entering the bottleneck buffer at time t. In case of FIFO queue, $D(t)$ can be found as the minimum non negative solution to

$$Q(t) = \int_t^{t+D(t)} C(\tau)d\tau \tag{2}$$

This comes from the assumption of a work conserving server (buffer output link) and the FIFO discipline. Taking derivatives of both sides of eq. (2), we find

$$\dot{Q}(t) = \lambda(t) - C(t) = C(t + D(t))[1 + \dot{D}(t)] - C(t) \quad \Rightarrow$$

$$\dot{D}(t) = \frac{\lambda(t)}{C(t+D(t))} - 1 \tag{3}$$

The derivative of $D(t)$ is used to derive the relation between the congestion window and the sending rate. To this end, the equality of the information in flight at time t and the congestion window size $W(t)$ yields:

$$W(t) = \int_{t_0}^t R(\tau)d\tau$$

From this we get

$$\dot{W}(t) = R(t) - R(t_0)\frac{dt_0}{dt} \tag{4}$$

By taking derivatives of both sides of the implicit definition of t_0 as a function of t, we get

$$\frac{dt_0}{dt} = \begin{cases} 1 & Q(t-T_{bm}) = 0 \\ \dfrac{1}{1+\dot{D}(t_0+T_{fm})} & Q(t-T_{bm}) > 0 \end{cases} =$$

$$= \begin{cases} 1 & Q(t-T_{bm}) = 0 \\ \dfrac{C(t-T_{bm})}{R(t_0)} & Q(t-T_{bm}) > 0 \end{cases} \tag{5}$$

where we used eq. (3) at time $t_0 + T_{fm}$ and exploited the identities $\lambda(t) = R(t-T_{fm})$ and $t_0 + T_{fm} + D(t_0 + T_{fm}) = t - T_{bm}$. In case $Q(t - T_{bm}) = 0$, we have $t_0 = t - T_m$.

Hence, from eqs. (4) and (5) we get

$$\begin{aligned} \dot{W}(t) &= \begin{cases} R(t) - R(t-T_m) & Q(t-T_{bm}) = 0 \\ R(t) - C(t-T_{bm}) & Q(t-T_{bm}) > 0 \end{cases} \\ &= R(t) - \mu(t-T_{bm}) \end{aligned} \tag{6}$$

Along with eq. (1), this last identity gives an expression for the rate $R(t)$ as a function of the link capacity $C(t)$ and of the buffer content $Q(t)$, i.e. $R(t) = \dot{W}(t) + \mu(t - T_{bm})$.

The evolution equation (1) for the congestion window in case of TCP Reno yields to a formally explicit solution:

$$W(t) = \sqrt{W^2(t_0) + 2\int_{t_0}^t \mu(\tau - T_{bm})d\tau} \qquad \text{for } t \ge t_0 \tag{7}$$

The buffer output rate $\mu(t)$ is equal to $\lambda(t)$ if $Q(t) = 0$, to $C(t)$ if $Q(t) > 0$, hence $\dot{Q}(t) = \lambda(t) - \mu(t)$ and this identity holds also when the buffer is empty. Moreover, $\lambda(t) = R(t-T_{fm}) = \dot{W}(t-T_{fm}) + \mu(t-T_m)$. So, $\dot{Q}(t) - \dot{\tilde{W}}(t) = \mu(t-T_m) - \mu(t)$, where we have introduced the quantity $\tilde{W}(t) \equiv W(t - T_{fm})$ for ease of notation. By integrating both sides between t_0 and t and assuming no packet loss occurs in the buffer in this time interval we get

$$\begin{aligned} & Q(t) - \tilde{W}(t) = \\ & = Q(t_0) - \tilde{W}(t_0) + \textstyle\int_{t_0}^t [\mu(\tau - T_m) - \mu(\tau)]d\tau \\ & = Q(t_0) + \textstyle\int_{t_0-T_m}^{t_0} \mu(\tau)d\tau - \tilde{W}(t_0) - \int_{t-T_m}^t \mu(\tau)d\tau \end{aligned}$$

holding for all $t \ge t_0$ until a packet loss occurs in the bottleneck buffer. By rearranging terms, it follows

$$Q(t) + \int_{t-T_m}^t \mu(\tau)d\tau - \tilde{W}(t) = constant = 0 \tag{8}$$

[1]We are assuming no bottleneck is in the way of the TCP ACK segments; this is not the case of half duplex links, e.g. IEEE 802.11 WLAN exploiting the Distributed Coordination Function access scheme

the last equality being consistent with the initialization of the congestion window when getting out of a loss recovery period, a time out or at TCP connection start up. As a matter of fact, in all those cases the congestion windows equals the flight-size, i.e. the amount of information stored in the TCP pipe.

To sum up, the evolution of $W(t)$ and $Q(t)$ starting from an initial time t_0 is given by eqs. (7) and (8) respectively. These equations hold for $t \geq t_0$ until a buffer overflow occurs, t_0 being the initial time of a congestion avoidance phase. The time varying bottleneck link capacity $C(t)$ appears indirectly through the output rate of the buffer, $\mu(t)$. To start the equations, an initial value $W(t_0)$ is needed along with the specification of $\mu(\tau)$ for $\tau \in (t_0 - T_m, t_0]$.

As for performance, the average TCP throughput Λ can be obtained as

$$\Lambda = \lim_{\Delta t \to \infty} \frac{1}{\Delta t} \int_{t_0}^{t_0+\Delta t} \mu(\tau) d\tau$$

The flight-size $Y(t)$, i.e. the amount of fluid in the pipe plus buffer at time t, can be evaluated as $Y(t) = \int_{t-T_{fm}}^{t} R(\tau)d\tau + Q(t) + \int_{t-T_{bm}}^{t} \mu(\tau)d\tau$; by exploiting eq. (6), we find $Y(t) = W(t) - \tilde{W}(t) + Q(t) + \int_{t-T_m}^{t} \mu(\tau)d\tau$. This reduces to $Y(t) = W(t)$ as long as there is no loss in the bottleneck buffer; instead, it is $Y(t) < W(t)$ after loss has occurred, until full recovery is done. From the time average S of $Y(t)$ we can compute the average packet delay of the TCP connection as S/Λ, according to Little's law.

Initial conditions to start the fluid evolution of the TCP connection equations from the initial time $t = 0$ are as follows: at the start the congestion window is set to one packet, $R(0)$ is set equal to a Dirac pulse of area one packet, the buffer is empty and the pipe is empty too, which means $\mu(\tau) = 0$ for $\tau \in (-T_m, 0]$.

There remains to define what happens as soon as a packet loss occurs. After initialization at time t_0, let t_B be the first time such that $Q(t) = B$: more precisely, $t_B = \inf_{t \geq t_0}\{t : Q(t) > B\}$. The TCP sender becomes aware of the packet loss after the whole content of the buffer has been sent out and up to $DupThresh$ duplicated ACKs requesting the lost packet are received, i.e. after a delay $D_B \equiv D_e + T_{bm}$, where D_e is given by

$$B + DupThresh = \int_{t_B}^{t_B+D_e} C(\tau)d\tau$$

The additional term $DupThresh$ in the left hand side is put there to account for the Duplicate ACK mechanism used by the fast retransmit procedure of TCP.

Let $W_f \equiv W(t_B+D_B)$. We model in a fluid approximation the loss recovery procedure of TCP SACK, with a window reduction factor $1 - b$. First, the lost packet is sent and immediately after that the sending rate is set to zero, until an amount bW_f of "flying" information is delivered, which is detected at the sender by means of duplicated ACKs. Then, the sending rate is set equal to the ACK'ed information rate, i.e. so much information is pushed through the loop as it gets out at the receiver. This recovery phase goes on for an RTT starting from time $t_B + D_B$, until the cumulative ACK of the W_f worth of information already delivered and not acknowledged is finally received[2]. At that time the normal congestion avoidance operation is resumed starting from a congestion window of size $(1 - b)W_f$.

The details of the fluid model of the recovery phase is as follows. Packet loss is detected at time $t_B + D_B$ at the TCP sender. Then $R(t) = 0$ for $t \in [t_B + D_B, t_B + D_B + t_S)$, where t_S is such that

$$bW_f = \int_{t_B+D_B}^{t_B+D_B+t_S} \mu(\tau - T_{bm})d\tau$$

After that, we set $R(t) = \mu(t - T_{bm})$ for $t \in [t_B + D_B + t_S, t_B + D_B + t_S + t_R)$, where t_R is such that

$$(1-b)W_f = \int_{t_B+D_B+t_S}^{t_B+D_B+t_S+t_R} \mu(\tau - T_{bm})d\tau$$

At time $t_{res} = t_B + D_B + t_S + t_R$, congestion avoidance operation is resumed with a congestion window $W(t_0) \equiv W(t_{res}) = (1 - b)W_f$.

The recovery model works provided there is enough "fluid" in the pipe as the packet loss is detected at the sender at time $t_B + D_B$ and the sending rate is set to 0. In fact, the sender waits for an amount of fluid bW_f to be delivered. If so many packets have been lost that the overall fluid in the pipe at $t_B + D_B$ is less than bW_f, it follows that the outlined recovery procedure will be stalled with the congestion window set to bW_f and the sending rate reset to 0, waiting for a never coming event. Real TCP is not trapped this way when duplicated ACK recovery is too slow or impossible thanks to the Retransmission Time Out (RTO) mechanism.

To take into account the time-out in a simple way, we assume that a timer TO is set to a fixed value when the sender detects the packet loss and sends the last piece of information before shutting down the sending rate to 0, i.e. at time $t_B + D_B$. The value of TO is set to $5T_m$. At time-out, the congestion window is set to 1 (measured in packets) and $R(t)$ is re-initialized with a Dirac pulse of area 1. Note that at time-out the pipe (hence the buffer) must be empty, if the value of TO is dimensioned correctly: in this model either there is more than bW_f fluid in the pipe when loss is detected, so that duplicated ACK based recovery will work, or after the entire pipe has been drained empty, a time out will occur.

Finally, the RTT of a packet sent at time t is this fluid model is $RTT(t) = T_m + D(t + T_{fm})$, where the buffer delay $D(t)$ is defined in eq. (2).

A. Discretization of the evolution equations

To solve numerically the differential equations governing the congestion window and the buffer content, we sample time with a time step of δ, i.e. at the times $t_k = t_0 + k\delta$, $k \geq 0$. We denote the samples of instantaneous functions (*cwnd*, buffer

[2]This is a simplification neglecting multiple packet losses

content) as $F(k) \equiv F(t_k)$; as for rate functions $x(t)$, we let $X(k) \equiv \int_{t_k}^{t_{k+1}} x(\tau)d\tau$.

Let $X(k)$, $U(k)$ and $L(k)$ denote the arrivals at the bottleneck buffer, the departures from the buffer out on the bottleneck link and the lost packets, respectively. Then

$$\begin{aligned} Q(k+1) &= Q(k) + X(k) - U(k) - L(k) \\ U(k) &= \min\{C(k), Q(k) + X(k)\} \\ L(k) &= \max\{0, Q(k) + X(k) - C(k) - B\} \end{aligned}$$

Let us define the normalized fixed delays $K_m \equiv T_m/\delta$, $K_{fm} \equiv T_{fm}/\delta$ and $K_{bm} \equiv T_{bm}/\delta$; we assume these numbers are integer valued.

For TCP Reno the congestion window discrete evolution equation, during the congestion avoidance phase, is

$$W(k+1) = W(k) + \frac{U(k-K_{bm})}{W(k)}$$

The sending rate is

$$R(k) = U(k-K_{bm}) + \frac{U(k-K_{bm})}{W(k)}$$

Since the input rate at the bottleneck buffer is a delayed copy of the sending rate, we have $X(k) = R(k - K_{fm})$ and therefore:

$$X(k) = \left[1 + \frac{1}{W(k-K_{fm})}\right] U(k-K_m)$$

A formal simplification is obtained if we introduce the congestion window function delayed by K_{fm}, i.e. shifted to the buffer input time axis: let $\tilde{W}(k) \equiv W(k - K_{fm})$.

To sum up, the complete discretization of the key quantities is as follows:

$$U(k) = \min\left\{C(k), Q(k) + \left[1 + \frac{1}{\tilde{W}(k)}\right] U(k-K_m)\right\}$$

$$\tilde{W}(k+1) = \tilde{W}(k) + \frac{U(k-K_m)}{\tilde{W}(k)}$$

$$Q(k+1) = Q(k) + \frac{U(k-K_m)}{\tilde{W}(k)} + U(k-K_m) - U(k)$$

This is carried out for $k \geq k_0$, where k_0 is a time where a previous packet loss has been completely recovered and a congestion avoidance phase starts, until a time k_{fin} such that it first occurs that $L(k_{fin}) > 0$. At that time the modeling of the recovery phase starts.

As for the loss recovery phase, its discrete version is quite straightforward. It is just to mention that the buffer delay can be computed, by discretizing the differential equation governing $D(t)$ for a non empty buffer, i.e. $\dot{D}(t) = \lambda(t)/C(t + D(t)) - 1$. Therefore, we have

$$D(k+1) = \begin{cases} 0 & Q(k+1) = 0 \\ D(k) + \delta\left[\dfrac{X(k)}{C(k+D(k))} - 1\right] & Q(k+1) > 0 \end{cases}$$

Then, we are able to compute the discretization of D_e, and of all the times involved in the loss recovery phase.

B. Approximate analysis for periodic capacity

The equations governing the congestion window and the buffer evolution can be given a rather simple analytical form useful to gain insight in case we can assume the buffer stays not empty and the capacity function has an analytically convenient form, i.e. it is sinusoidal. This is useful to gain some insight into the effect of the time scale of variation of the capacity, in this case just the period $1/F$.

Let us assume that the link capacity is given by $C(t) = C_0 + C_1 + C_1\cos(2\pi Ft + \phi_0)$ for $t \geq t_0$ and that the buffer is non empty for all $t \geq t_0 - T_m$. Then, we have from eq. (7) by exploiting the special analytical form of the capacity:

$$W(t) = \sqrt{W^2(t_0) + 2\frac{t-t_0}{T_m} f(t)} \tag{9}$$

with

$$f(t) = (C_0 + C_1)T_m + \tag{10}$$
$$+ C_1 T_m \frac{\sin[\pi F(t-t_0)]}{\pi F(t-t_0)} \cos[2\pi F(t+t_0) + \phi_0]$$

Also, again under the hypothesis that the buffer never empties starting from $t_0 - T_m$, the pipe balance equation yields for $t \geq t_0$

$$\tilde{W}(t) = Q(t) + \int_{t-T_m}^{t} C(\tau)d\tau = Q(t) + g(t)$$

with

$$g(t) = (C_0 + C_1)T_m + \tag{11}$$
$$+ C_1 T_m \frac{\sin(\pi F T_m)}{\pi F T_m} \cos\left[2\pi F\left(t - \frac{T_m}{2}\right) + \phi_0\right]$$

From these expression, we can gain some insight on the effect of F for two extreme cases.

As $F \to \infty$, we have $f(t), g(t) \to (C_0 + C_1)T_m$, hence the congestion window and the buffer content evolve as if the capacity were constant to the average value of the actually available link capacity. In other words, for very fast time variation of the link capacity ($FT_m \gg 1$) the TCP congestion control averages out the variation.

In the limit $F \to 0$, we have $f(t), g(t) \to (C_0 + C_1)T_m + C_1 T_m \cos\phi_0$, i.e. TCP sees a constant capacity whose value is the one holding at the initial time. This too matches with intuition, since for $FT_m \ll 1$, it is almost as if the link capacity were constant.

An interesting interplay between T_m and F arises when $F = n/T_m$, for some integer n. Then, $g(t) = (C_0 + C_1)T_m$, so $Q(t) = \tilde{W}(t) - (C_0 + C_1)T_m$; in view of eqs. (9) and (10), we can write:

$$Q(t) \leq$$
$$\sqrt{\tilde{W}^2(t_0) + 2(C_0 + C_1)(t - t_0) + \frac{2C_1 T_m}{n\pi}} - (C_0 + C_1)T_m$$

This inequality shows that the buffer increase during congestion avoidance is upper bounded by a time function very similar to the one holding for a constant capacity, equal to the average value of the actually available capacity. This results holds by virtue of the special relation between F and T_m, i.e. $FT_m = integer$ and suggest that better performance are to be expected for these special values with respect to non null values of F that yield relative maxima of the oscillation factor $\sin(\pi FT_m)/(\pi FT_m)$ appearing in eq. (11).

IV. Numerical and simulation results

This section has a twofold aim. First, it should assess the accuracy of the fluid model of TCP connection evolution as compared to ns-2 based simulations, where all details of TCP and packetized flows are considered; simulated bottleneck link capacity variation is obtained by means of higher priority interfering traffic flows, not by an instantaneously varying function $C(t)$. A second aim is to highlight the effect of the capacity variation time scale with a given fixed capacity marginal distribution. This way, the average capacity left over to TCP and its probability distribution are the same while the time scale over which a given variation is seen can be adjusted with respect to the base round trip time of the TCP connection, T_m. In this section we consider three different capacity variation patterns: a) periodical sinusoidal function, b) random stationary process, c) UMTS application case study.

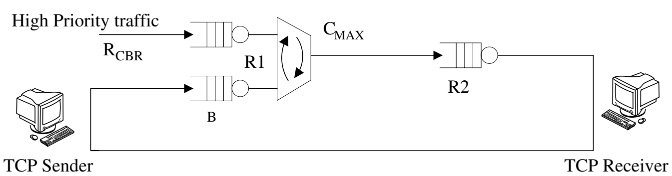

Fig. 2. ns-2 simulation scenario.

Fluid model results are compared with ns-2 simulation results. Figure 2 depicts the ns-2 simulation scenario. The simulation scenario is composed of a TCP sender (S) that has always data to send to the TCP receiver (R). The bottleneck link is the link between R1 and R2 and it is provided with a capacity C_{max}, whereas the links between S and R1, between R2 and R and between R and S back are provided with a capacity much greater than C_{max}. Capacity variations are simulated by a CBR source with a time-varying sending rate $R_{CBR}(k\Delta)$ that is kept constant during a sampling period of duration Δ and depends on the discretized version of the target time-varying capacity function $C(t)$. During the k-th sampling interval, the CBR transmission rate is $R_{CBR}(k\Delta)$=$C_{max} - C(k\Delta)$ bps. If L_{CBR} is the CBR packet size in bits (in all simulations L_{CBR} is 400 bits), the CBR interarrival time is L_{CBR}/R_{CBR}. The sampling time Δ has been chosen much smaller of the typical time-constant of the theoretical target capacity function. The scheduler at the bottleneck link always serves CBR data packets with priority. As far as regards the TCP settings, we used the NewReno TCP variant but the same results (not shown herein) have been obtained with the SACK version of TCP. In accordance with [15], the TCP receiver does not pose any limitations to the value of the TCP sender congestion window (i.e. the TCP advertised window is supposed infinite). TCP packet size is constant and equal to 1500 bytes. All simulations last 1000s.

A. Sinusoidal capacity function

A very simple way to accomplish this is to reduce to a deterministic capacity profile, defined as a periodic function of time, with a fundamental period of $1/F$. We therefore use a sinusoidal function, i.e.

$$C(t) = \frac{C_{max} + C_{min}}{2} + \frac{C_{max} - C_{min}}{2} \sin(2\pi Ft)$$

In the numerical example, C_{min}=10Mbps and C_{max}=100Mbps.

Figures 3 and 4 depict the mean utilization of the average bottleneck link capacity ($(C_{max}+C_{min})/2$=55Mbps) obtained by means of the fluid model and of ns-2 simulations respectively when T_{fm} is 10ms and T_{bm} is 90ms (T_m=100ms). Figures 5 and 6 depict the scenario where T_{fm} is 30ms and T_{bm} is 270ms (T_m=300ms).
All figures show the link utilization varying the frequency of the sinusoid F, for different values of the bottleneck buffer size B. The buffer size B is expressed in terms of the ratio between the maximum bandwidth delay product and the queue limit (e.g. $B = 0.1 \cdot C_{max} \cdot T_m$). Focusing our attention on fluid model results obtained with T_m=100ms (Figure 3), it is worth to pinpoint that the link utilization achieved by the TCP connection strongly depends on the frequency of the bottleneck capacity variation. When the rate of variation is very low, the link utilization is very high and TCP congestion window is able to follow the capacity variation. Obtained results reflect the ones achievable in the case of constant capacity. Similar results are obtained when the rate of variation is very high. In this case TCP behaves as it would see the average capacity. In case of intermediate values of F (between 0.1 and 10 Hz), the TCP performance are highly degraded; the smaller is the buffer size the higher is the degradation. E.g. when $B = 0.1 C_{max} T_m$=84 packets, the utilization falls below the 30% of the available average capacity, whereas when the

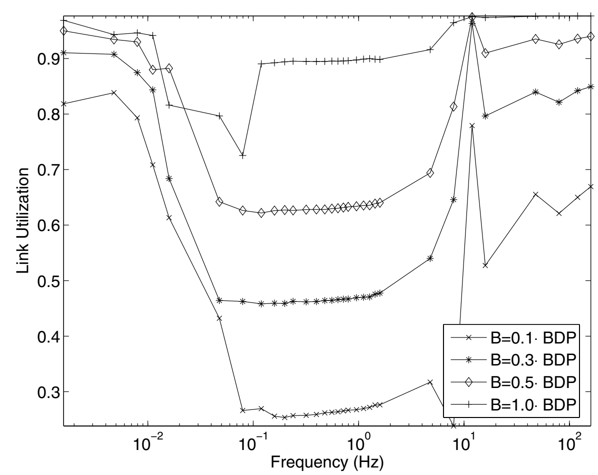

Fig. 3. Fluid model results: sinusoidal capacity, T_m=100ms.

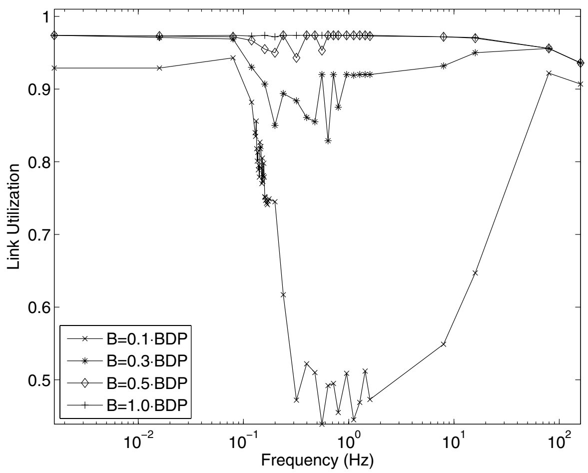

Fig. 4. ns-2 simulation results: sinusoidal capacity, T_m=100ms.

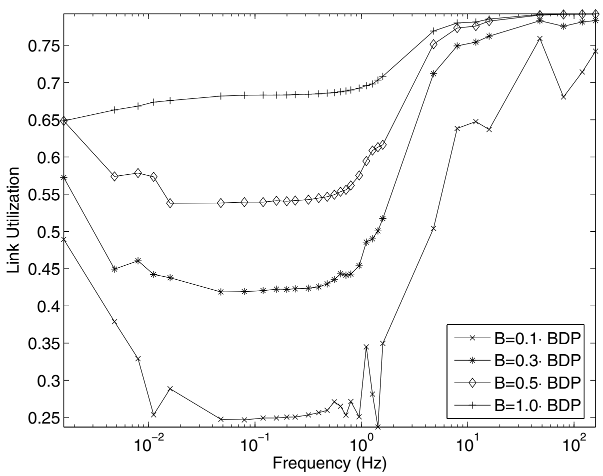

Fig. 5. Fluid model results: sinusoidal capacity, T_m=300ms.

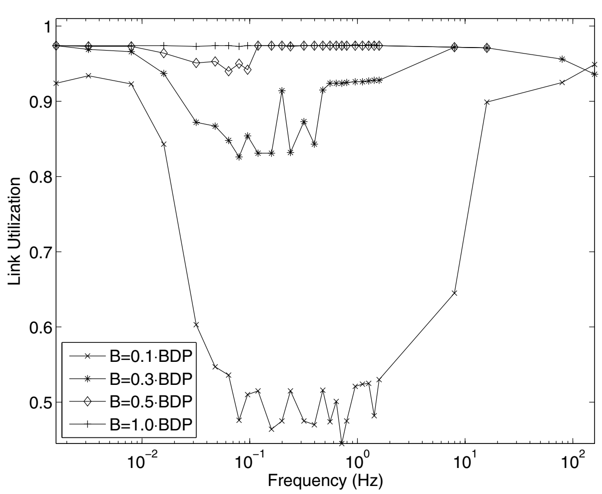

Fig. 6. ns-2 simulation results: sinusoidal capacity, T_m=300ms.

buffer is big, the effect is marginal since the buffer absorbs the capacity variation avoiding useless losses.

Comparing simulation results with fluid model results, we notice that results obtained for low and high value of F are the same of the ones obtained with the fluid model, whereas for intermediate values of F the degradation region is enclosed in a smaller range of frequencies. Moreover it is worth to observe that simulation results are more stable and do not exhibit the large oscillation behavior present in the fluid model cases, due to the deterministic nature of the capacity variation function.

Figures 5 and 6 depict results obtained with T_m equal to 300ms. Also in this scenario, the general behavior is similar, however, the degradation zone is shifted on the left side of the frequency scale since the ratio between the period of the sinusoid ($1/F$) and T_m is lower. An insight into the

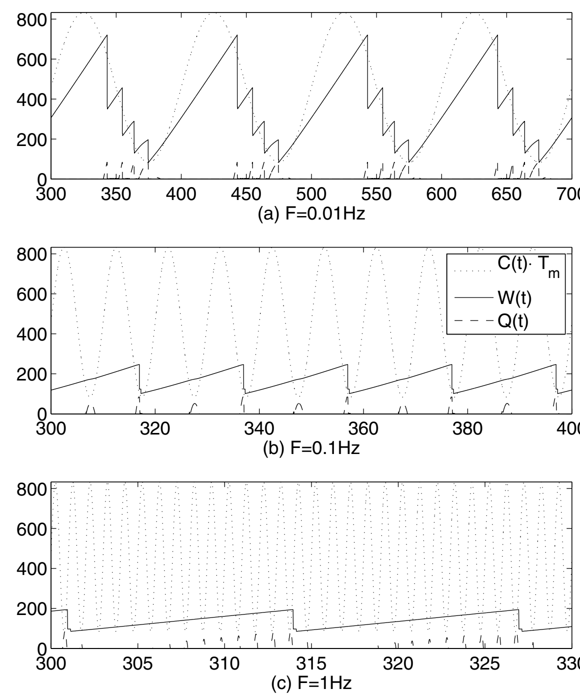

Fig. 7. $Q(t)$, $C(t) \cdot T_m$ and $W(t)$ evolution for T_m=100ms and $B = 0.1 \cdot C_{max}T_m$.

fluid simulation evolution is given by Figure 7 where the congestion window $W(t)$, the instantaneous bandwidth-delay product $C(t) \cdot T_m$ and the buffer content $Q(t)$ are depicted for three different values of F and $B = 0.1C_{max}T_m$. When the frequency is low ((a) - F=0.01Hz), the TCP congestion control is able to follows the slow capacity variation and to inflate the congestion window towards high values, leading to a high value of the link utilization. When F is higher ((b) - F=0.1Hz and (c) - F=1Hz), TCP congestion window synchronizes with capacity period and packet losses occur at small values of the congestion window. Observing the figure, it possible to note that the congestion window oscillates approximately between $C_{min}T_m + B$ and $(C_{min}T_m + B)/2$, proving that TCP is not able to exploit the average capacity but just the minimum one.

In both cases, in fact, when the capacity is in the decreasing phase, its variation is faster than the one that can be absorbed by the bottleneck buffer.

Similar results, not shown herein, have been obtained in case of a symmetric and periodic square wave capacity function.

B. Random capacity function

In this section we model the time varying bottleneck link capacity as a stationary random process. To keep things simple, we assume a first order, autoregressive model with uniformly distributed innovation in the interval $[C_{min}, C_{max}]$.

The discrete time model of the capacity is stated as follows:

$$C(k+1) = \alpha C(k) + (1-\alpha)I(k)$$

where $I(k)$ is a sequence of identically distributed, independent random variables with uniform distribution between C_{min} and C_{max}. The weighting coefficient α is chosen such that the autocorrelation coefficient of the sequence $C(k)$ at a time lag of T_c equals a conveniently low value: we choose 0.1, so we require $\alpha^{T_c/\delta} = 0.1$. The meaning of T_c is a coherence time of the time varying capacity function, i.e. a time span over which the capacity function de-correlates. In the numerical results, we use the ratio $1/x = T_m/T_c$ as an independent variable. For each value of the ratio, the corresponding α is obtained as $\alpha = 0.1^{\frac{\delta}{xT_m}}$. The initial value of the capacity is set to the average capacity, i.e. $C(0) = (C_{max} + C_{min})/2$. In the numerical examples we set $C_{max} = 100$ Mbps and $C_{min} = 10$ Mbps.

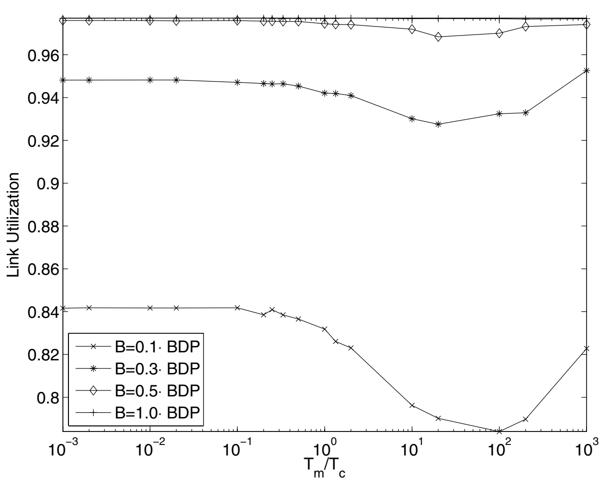

Fig. 8. Fluid model results: random capacity, T_m=100ms.

In Figures 8 and 9 the bottleneck link utilization is plotted against the ratio $x = T_m/T_c$ for $T_m = 100$ ms and $T_m = 300$ ms respectively and for four values of the buffer size (normalized as before with respect to the bandwidth-delay product $C_{max}T_m$).

Two main facts are apparent as compared to the results in case of deterministic periodic capacity function. First, the qualitative effect of the capacity function time scale variation is analogous to the case of deterministic sinusoidal capacity

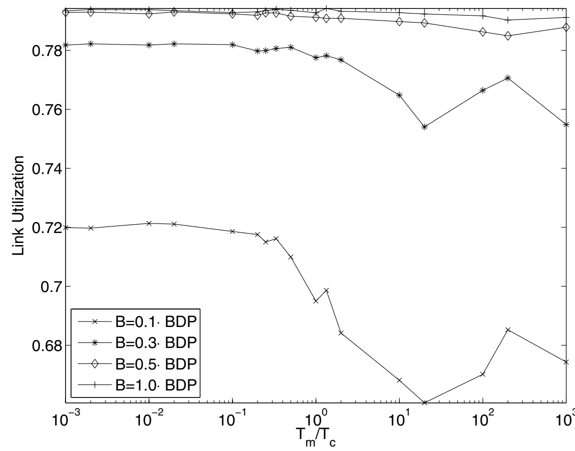

Fig. 9. Fluid model results: random capacity, T_m=300ms.

function: values of T_c much larger and much smaller than T_m yield better performance as compared to intermediate values ($\frac{T_m}{T_c}$ in the range $10 \div 100$ in both figures.) Secondly, the quantitative effect is quite reduced with respect to the deterministic case, since the range of values of each utilization curve for a given buffer size value never entails a throughput penalty larger than 33% in the worst case with respect to the largest achieved throughput. Also, the curve behavior is much smoother than in case of sinusoidal capacity, due to the randomization of the variable capacity.

C. Case Study

As an application example, we consider a wireless link with adaptive modulation and coding and a link adaptation mechanism that adjust current transmission configuration to follow the radio channel time variation. Numbers are chosen so as to adhere to the evolved Universal Terrestrial Radio Access (E-UTRA) suggested test cases and to the High-Speed Downlink Packet Access service capacities [16]. Forward peak bit rate is 14.4 Mbps; the reverse link is assumed to always have at least the capacity required to deliver the TCP ACKs. The radio channel is modeled by means of distance dependent deterministic attenuation according to a power law with exponent equal to 4, and of log-normal shadowing with 0 dB mean and 8 dB standard deviation. The time evolution of the radio channel is sampled with a constant period θ. The k-th sample $S(k)$ is obtained as $S(k) = 10^{-S_{dB}(k)/10}$, where $S_{dB}(k) = \beta S_{dB}(k-1) + (1-\beta)U(k), k \geq 0$, with $U(k)$ a zero mean white Gaussian sequence and β a correlation coefficient chosen so that the coherence distance of the shadowing is $d = 50$ m; then β is found by requiring that the correlation at a time lag is equal to the time the mobile user takes to travel 50 m at a constant speed v is 0.1, i.e. $\beta^{d/(v\theta)} = 0.1$. Worst case interference is considered from the first tier neighboring cells, with a cell radius of R (1 km in the numerical example). Then, the Signal-to-Noise Ratio (SNR) of the downlink wireless link during time interval k is $SNR(k) = S(k) \cdot SNR_0$, where SNR_0 is the SNR for shadowing equal to 1, which is assumed to be constant. We associate a capacity of $C(k) = \min\{C_0, C_a \cdot log_2(1 + SNR(k))\}$ to the wireless link, where $C_0 = 14.4$ Mbps and C_a is chosen so that the obtained average available capacity is 2/3 of C_0, so as to make performance evaluation easy.

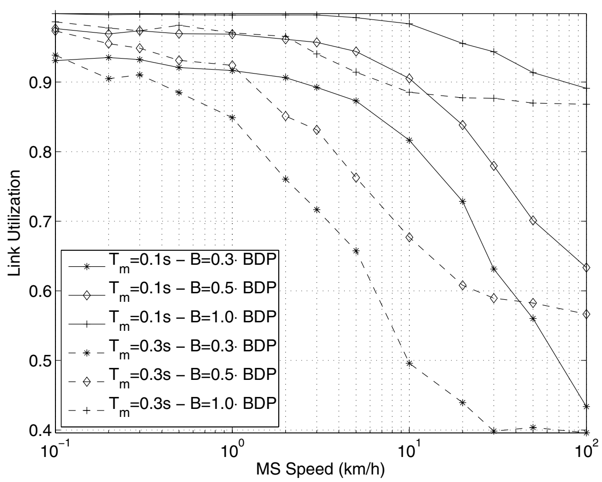

Fig. 10. Fluid model results: UMTS Application case study.

Figure 10 depicts the link utilization (goodput divided by C_a) varying the mobile user speed, for different values of the bottleneck buffer size (normalized with respect to the bandwidth-delay product C_0T_m), for T_m equals to 100 ms (solid lines) and 300 ms (dashed lines). The figure shows the effect of capacity variations due to mobile user movements on TCP congestion control mechanisms. Obtained performance substantially degrade when the mobile user speed increases. Moreover, the relevance of the buffer size provisioning is highlighted by fluid model results.

V. Final remarks

A basic model of a TCP connection with a single bottleneck with time varying capacity is developed in detail. Basic as it might be, to the best of our knowledge, such a model has not been fully reported and analyzed in the literature. Our model is stated in term of (discretized) state evolution equation, allowing a very fast and efficient fluid simulation of the TCP connection with any time varying capacity profile. This opens the way to easy evaluation of the impact of various kind of transient behaviors.

We exploit the fluid model, along with ns-2 based simulation to assess the accuracy of the fluid approximation, to highlight a "resonance" phenomenon between TCP performance and the fundamental time constant of the time varying capacity. This can be shown both with a simple yet somewhat artificial capacity model defined as a periodic function (sinusoidal, square wave) and, as an application example, we consider a wireless link with adaptive modulation and coding and a link adaptation mechanism that adjust current transmission configuration to follow the radio channel time variation.

It turns out that TCP performance can be highly affected, depending also on the bottleneck buffer size. This should be taken into account when designing wireless link adaptive modulation schemes that modify TCP available capacity according to the perceived link quality or when designing priority scheme for sharing the capacity of network links among real time flows and low priority best effort, TCP based traffic. As shown by the results, it could end up with a poor utilization of the possibly highly variable available capacity.

Further work aims at refining the model and studying its solution for special cases, so as to gain a deep insight into the very intricate interplay among TCP parameters, link capacity variation characteristics and obtained performance. Also the extension of the approach to multiple TCP connections deserves being pursued.

References

[1] V. Jacobson, *Congestion Avoidance and Control*, ACM Computer Communications Review, 18(4): 314–329, Aug. 1988.
[2] J. Padhye et al., *Modeling TCP Reno Performance: A Simple Model and Its Empirical Validation*, IEEE/ACM Transactions on Networking, Vol. 8(2), pp. 133–145, Apr. 2000.
[3] T. V. Lakshman and U. Madhow, *The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss*, IEEE/ACM Transactions on Networking, Vol. 5(3), pp. 336–350, Jun. 1997.
[4] S. Shakkottai et al., *TCP Performance over End-to-End Rate Control and Stochastic Available Capacity*, IEEE/ACM Transactions on Networking, Vol. 9(4), pp. 377–391, Aug. 2001.
[5] A. Karnik, A. Kumar, *Performance of TCP congestion control with explicit rate feedback*, IEEE/ACM Transactions on Networking, Vol. 13(1), pp. 108–120, Feb. 2005.
[6] F. Baccelli, D. Hong, *Interaction of TCP Flows as Billiards*, IEEE/ACM Transactions on Networking, Vol. 13(4), pp. 841–853, Aug. 2005.
[7] C. V. Hollot, Y. Liu, V. Mishra, and D. Towsley, *Unresponsive flows and AQM performance*, In Proc. of INFOCOM, vol. 1, pp. 85–95, San Francisco, USA, Apr. 2003.
[8] M. Ajmone Marsan at al., *Using Partial Differential Equations to Model TCP Mice and Elephants in Large IP Networks*, IEEE/ACM Transactions on Networking, Vol. 13(6), pp. 1289–1301, Dec. 2005.
[9] Y. Liu et al., *Fluid models and solutions for large-scale IP networks*, Proc. of ACM SIGMETRICS, San Diego, CA, USA, Jun. 2003.
[10] C. Villamizar and C. Song, *High Performance TCP in ANSNET*, ACM Computer Communication Review, vol. 24, no. 5, pp. 45–60, 1995.
[11] The network simulator. URL: www.isi.edu/nsnam/ns/
[12] M. Allman, V. Paxson and W. Stevens, *TCP Congestion Control*, IETF RFC 2581, Apr. 1999.
[13] E. Blanton, M. Allman, K. Fall, L. Wang, *A Conservative Selective Acknowledgment (SACK)-based Loss Recovery Algorithm per TCP*, RFC 3517, Apr. 2003.
[14] M. Mathis, J. Mahdavi, S. Floyd, A. Romanow, *TCP Selective Acknowledgment Options*, RFC 2018, Oct. 1996.
[15] M. Allman, A. Falk, *On the Effective Evaluation of TCP*, ACM Computer Communication Review, 29(5), Oct. 1999.
[16] 3GPP TR 25.814, *Technical Specification Group Radio Access Network, Physical layer aspects for evolved Universal Terrestrial Radio Access (UTRA) (Release 7)*, Sep. 2006.