

On the validity of fluid-dynamic models for data networks

Ciro D'Apice and Rosanna Manzo

Department of Electronic and Computer Engineering, University of Salerno, Fisciano (SA), Italy

Email: {cdapice, rmanzo}@unisa.it

Benedetto Piccoli

Department of Mathematical Sciences, Rutgers University, Camden, NJ, US

Email: piccoli@camden.rutgers.edu

Abstract—Simulators have been acknowledged as one of the most flexible tool in studying and evaluating network performance. Various approaches have been followed in data networks modeling. The present paper explores the range of applicability of a fluid-dynamic model, proposed in [5], able to describe the load evolution of a data network. Looking at intermediate time scale, it is assumed that packets are conserved, and hence the packets density obey to a conservation law. The assumption underlying the model is the behaviour of the information loss probability from which the flux is derived. Here the validity of the above hypothesis and the rules introduced to solve dynamics at nodes are discussed. In particular packet loss estimations using queueing models have been compared with the assumed loss probability. Moreover a TCP throughput simulation is proposed to evaluate the feasibility of the “tent” flux.

Index Terms—fluid-dynamic model, packet loss probability estimations, TCP simulation

I. INTRODUCTION

There are a considerable number of different approaches to telecommunication and data networks and their load evolution. Focusing the attention on Internet and the control congestion algorithms, as TCP/IP, see for instance [2, 3, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]. The layered structure of Internet renders it a difficult task to model in a complete way the dynamics, or even just to measure it as showed in [1].

A fluid-dynamic model for telecommunication networks, similar to that introduced for car traffic (see [4, 7]), was proposed in [5] and then sources and destinations have been considered in the model, thus taking care of packets paths inside the network (see [6]). The idea is to look at the network at an intermediate time scale so that the packets transmission happens at a faster level but the equilibria of the whole network are reached only as asymptotic. This permits to construct a model relying on macroscopic description.

Focus on a network formed by a finite collection of transmission lines and nodes (or routers) which receive and send information encoded in packets. Having in mind Internet as key model, it is assumed that:

1. Each packet travels on the network with a fixed speed and with assigned final destination;
2. Routers receive, process and then forward packets, which may be lost with a probability increasing with the number of packets to be processed. Each lost packet is sent again.

The behavior of a single straight transmission line with consecutive routers is modelled assuming that each router sends packets to the successive one a first time, then packets that are lost in this process are sent a second time and so on until they reach next router. Considering “backbones” of the network, which are formed by many consecutive routers, and looking at larger space scale, the packets are conserved and the density obey to the conservation law:

$$\rho_t + f(\rho)_x = 0, \quad (1)$$

where ρ is the packet density, v is the velocity and $f(\rho) = v\rho$ is the flux. An average transmission speed among routers can be derived assigning a loss probability as function of the density. It is then possible to pass to the limit in the (re)sending procedure getting a velocity function and thus a flux function.

In order to deal with complex networks, one has to assign the dynamics at nodes in which many lines (backbones) intersect. Two different “routing” algorithms have been introduced in [5]:

RA1 Packets from an incoming line are sent to an outgoing one according to their final destination. Packets are processed ordered by arrival time.

RA2 Packets are processed by arrival time and are sent to outgoing lines in order to maximize the flux.

Moreover a further refinement of the model was given in [6], where packets were directed taking into account original sources and final destinations (nonlocal information).

The key point of the model is the loss probability, used to define the flux function. Indeed the choice of a non

reasonable loss probability function could invalidate the model.

The aim of this paper is to verify the conceptual validity of the model, determining if the underlying assumptions are correct and the representation of the problem is adequate for modeling purposes. To achieve this goal, the loss probability function to be validated is compared with the behaviour of the packet loss derived from known models used in literature to infer network performance and the shape of the velocity and flux functions is discussed. A tool able to simulate TCP mechanism has been used to compare the “tent” flux function with the simulated flux. All the comparisons confirm the validity of the assumptions underlying the fluid-dynamic model. Finally the RA1 and RA2 rules have been interpreted in terms of routing protocols according to which packets are guided in real data networks.

The paper is organized as follows. Section 2 provides a description of the fluid dynamic model. Then, in Section 3, we discuss the validity of the assumption on the loss probability function, the velocity and flux. Section 4 is devoted to the analysis of the range of applicability of the fluid-dynamic model. A TCP throughput simulation is presented, with a detailed discussion on the simulated flux function, compared to the “tent” flux. Finally in Section 5 we try to give an interpretation of rules RA1 and RA2 in terms of global and decentralized algorithms.

II. MODEL DESCRIPTION

A telecommunication network is a finite collection of transmission lines connected together by nodes. Formally we introduce the definition:

Definition 1 A telecommunication network is given by a 4-tuple $(N, \mathcal{I}, \mathcal{F}, \mathcal{J})$ where

Cardinality N is the cardinality of the network, i.e. the number of lines in the network;

Lines \mathcal{I} is the collection of lines, modelled by intervals $I_i = [a_i, b_i] \subseteq \mathbb{R}, i = 1, \dots, N$;

Fluxes \mathcal{F} is the collection of flux functions $f_i : [0, \rho_i^{\max}] \mapsto \mathbb{R}, i = 1, \dots, N$;

Nodes \mathcal{J} is a collection of subsets of $\{\pm 1, \dots, \pm N\}$ representing nodes. If $j \in J \in \mathcal{J}$, then the transmission line $I_{|j|}$ is crossing at J as incoming line (i.e. at point b_i) if $j > 0$ and as outgoing line (i.e. at point a_i) if $j < 0$. For each junction $J \in \mathcal{J}$, we indicate by $\text{Inc}(J)$ the set of incoming lines, that are I_i 's such that $i \in J$, while by $\text{Out}(J)$ the set of outgoing lines, that are I_i 's such that $-i \in J$. We assume that each line is incoming for (at most) one node and outgoing for (at most) one node.

To define the dynamics of packet densities along lines, we make the following hypothesis:

(H1) Lines are composed of consecutive processors N_k , which receive and send packets. The number of packets at N_k is indicated by $R_k \in [0, R_{\max}]$;

(H2) There are two time-scales: Δt_0 , which represents the physical travel time of a single packet from node to node (assumed to be independent of the node for simplicity); T representing the processing time, during which each processor tries to operate the transmission of a given packet;

(H3) Each processor N_k tries to send all packets R_k at the same time. Packets are lost according to a loss probability function $p : [0, R_{\max}] \rightarrow [0, 1]$, computed at R_{k+1} , and lost packets are sent again for a time slot of length T ;

(H4) The number of packets not transmitted for a whole processing time slot is negligible.

Since the packet transmission velocity on the line is assumed constant, it is possible to compute an average velocity function and thus an average flux function as follows.

Let us focus on two consecutive nodes N_k and N_{k+1} , assume a static situation, i.e. R_k and R_{k+1} are constant, and call δ the distance between the nodes. During a processing time slot of length T the following happens. All packets R_k are sent a first time: $(1 - p(R_{k+1}))R_k$ are sent successfully and $p(R_{k+1})R_k$ are lost. At the second attempt, of the lost packets $p(R_{k+1})R_k$, $(1 - p(R_{k+1}))p(R_{k+1})R_k$ are sent successfully and $p^2(R_{k+1})R_k$ are lost and so on.

Let us indicate by Δt_{av} the average transmission time of packets, by $\bar{v} = \frac{\delta}{\Delta t_0}$ the packet velocity without losses

and $v = \frac{\delta}{\Delta t_{av}}$ the average packets velocity. Then, we can compute:

$$\Delta t_{av} = \sum_{n=1}^M n \Delta t_0 (1 - p(R_{k+1})) p^{n-1}(R_{k+1})$$

where $M = [T/\Delta t_0]$ (here $[\cdot]$ indicates the floor function) represents the number of attempts of sending a packet.

The hypothesis (H4) corresponds to assume $\Delta t_0 \ll T$ or, equivalently, $M \sim +\infty$. Making the identification, $M = +\infty$, we get:

$$\Delta t_{av} = \frac{\Delta t_0}{1 - p(R_{k+1})},$$

and

$$v = \bar{v}(1 - p(R_{k+1})). \quad (2)$$

Let us call now ρ the locally averaged density and ρ_{\max} its maximum. Using assumption (H4), we can interpret

the function p as a function of the locally averaged density ρ and, using (2), determine the corresponding flux function, given by the averaged density times the average velocity.

The loss probability function should reflect the expected behavior of routers versus increasing packets density. More precisely, it is natural to assume that:

1. The loss probability function p is increasing as a function of the locally averaged density ρ .
2. The function p is equal to zero for low densities.
3. The function p tends to 1 when the locally averaged density tends to its maximal value.

The modelling interpretation of such rules are as follows. Assumption 1. corresponds to the fact that higher number of packets will slow down internal router processes and render packets losses more probable. Assumption 2. follows from the hypothesis that routers are able to manage low number of packets producing no loss. Finally 3. should be interpreted as the possibility of the system to reach a complete stuck. Notice that all three assumptions have some limited range of validity for real data networks, but at the same time they allow a consistent modeling framework.

To achieve rules 1., 2. and 3., a possible choice of the loss probability function is:

$$p(\rho) = \begin{cases} 0, & 0 \leq \rho \leq \sigma, \\ \frac{\rho_{max}(\rho - \sigma)}{\rho(\rho_{max} - \sigma)}, & \sigma \leq \rho \leq \rho_{max}. \end{cases} \quad (3)$$

In this case, the averaged transmission velocity is equal to

$$v(\rho) = \begin{cases} \bar{v}, & 0 \leq \rho \leq \sigma, \\ \bar{v} \frac{\sigma(\rho_{max} - \rho)}{\rho(\rho_{max} - \sigma)}, & \sigma \leq \rho \leq \rho_{max}. \end{cases}$$

Since $f(\rho) = v(\rho)\rho$, it follows that

$$f(\rho) = \begin{cases} \bar{v}\rho, & 0 \leq \rho \leq \sigma, \\ \bar{v}\sigma \frac{\rho(\rho_{max} - \rho)}{\rho_{max} - \sigma}, & \sigma \leq \rho \leq \rho_{max}. \end{cases}$$

Setting, for simplicity $\rho_{max} = 1$ and $\sigma = \frac{1}{2}$, we get the loss function and simple “tent” function of Fig. 1.

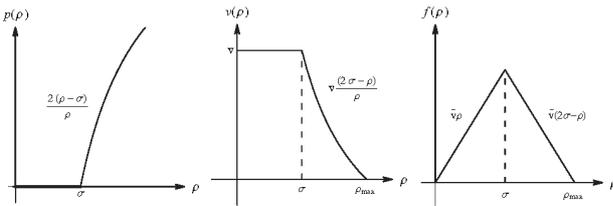


Figure 1. Loss probability, average velocity and flux behaviours.

To simplify the treatment of the corresponding conservation laws, we will assume the following:

(F) Setting $\rho_{max} = 1$, on each line the flux $f : [0,1] \rightarrow \mathbb{R}$ is concave, $f(0) = f(1) = 0$ and there exists a unique maximum point $\sigma \in]0,1[$.

Notice that the flux of Fig. 1 satisfies the assumption (F).

Remark 2 Other possible loss probability functions whose corresponding flux satisfy assumption (F) are:

$$p(\rho) = \begin{cases} 0, & 0 \leq \rho \leq \sigma, \\ \frac{\rho - \sigma}{\sigma}, & \sigma \leq \rho \leq \rho_{max}, \end{cases} \quad (4)$$

$$p(\rho) = \begin{cases} 0, & 0 \leq \rho \leq \sigma, \\ \frac{(\rho - \sigma)^2}{\sigma^2}, & \sigma \leq \rho \leq \rho_{max}, \end{cases} \quad (5)$$

see Fig. 2.

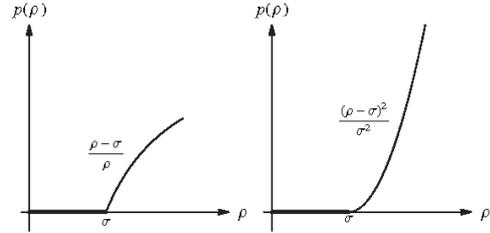


Figure 2. Loss probability functions: (4), left and (5), right.

Choosing a loss probability function of type (4) and (5) we get velocity and flux behaviour as in Fig. 3-4 :

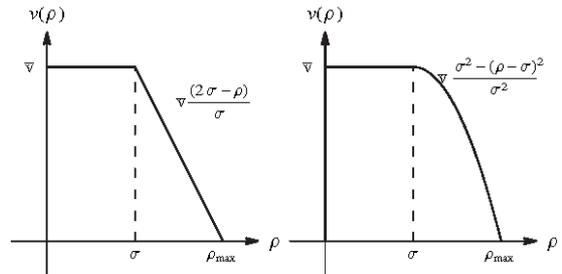


Figure 3. Velocity functions: assumption (4), left and (5), right.

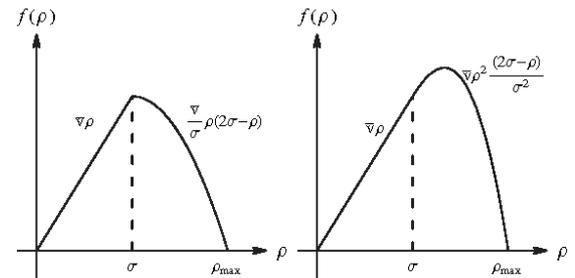


Figure 4. Flux functions: assumption (4), left and (5), right.

On each transmission line I_i we consider the evolution equation

$$\partial_t \rho_i + \partial_x f_i(\rho_i) = 0,$$

where we use the assumption (F) and, for simplicity, assume $f_i = f$, $\forall i = 1, \dots, N$.

In order to model networks composed by junctions, we have to solve Riemann Problems at nodes (Cauchy problems corresponding to an initial data which are constant on each transmission line) using two different “routing” algorithms RA1 and RA2.

Remark 3 In the model with sources and destinations ([6]) a telecommunication network is given by a 7-tuple $(N, \mathcal{I}, \mathcal{F}, \mathcal{J}, \mathcal{S}, \mathcal{D}, \mathcal{R})$ where:

Sources \mathcal{S} is the subset of $\{1, \dots, N\}$ representing lines starting from traffic sources. Thus, $j \in \mathcal{S}$ if and only if j is not outgoing for any node. We assume that $\mathcal{S} \neq \emptyset$;

Destinations \mathcal{D} is the subset of $\{1, \dots, N\}$ representing lines leading to traffic destinations. Thus, $j \in \mathcal{D}$ if and only if j is not incoming for any node. We assume that $\mathcal{D} \neq \emptyset$;

Traffic distribution functions \mathcal{R} is a finite collection of functions $r_j : \text{Inc}(J) \times \mathcal{S} \times \mathcal{D} \rightarrow \text{Out}(J)$. For every J , $r_j(i, s, d)$ indicates the outgoing direction of traffic that started at source s , has d as final destination and reached J from the incoming road i .

The load dynamics is described by the conservation law for the packet density and a semilinear equation

$$\partial_t \pi_i(t, x, s, d) + \partial_x \pi_i(t, x, s, d) \cdot v_i(\rho_i(t, x)) = 0,$$

on each I_i , where $\pi_i(t, x, s, d)$ specifies the fraction of the density $\rho_i(t, x)$ that started from source s and is moving towards the final destination d .

III. MODEL ASSUMPTIONS

The aim of this Section is to verify that the assumptions underlying the data networks fluid-dynamic model (shortly FD model) are correct. Different modeling approaches are used to describe TCP, and they can be characterized in the following five categories: renewal theory models, fluid models, processor sharing models, control theoretic models, and fixed-point models. Here we focus on the last models category, and considering various set-ups with TCP traffic in a single bottleneck topology, we investigate queuing models for estimating packet loss rate.

A. Loss probability function

The choice of a non adequate loss probability function could invalidate the model. It is reasonable to assume that this function is null for some interval, which is a right neighborhood of zero. This means that at low densities no

packet is lost. Then p should be increasing, reaching the value 1 at the maximal density, the situation of complete stuck. From now on, we focus on the loss probability function given in (3) with $\rho_{max} = 1$ and $\sigma = \frac{1}{2}$, which can be written as:

$$p(\rho) = \begin{cases} 0, & 0 \leq \rho \leq 1/2, \\ \frac{2\rho-1}{\rho}, & 1/2 \leq \rho \leq 1. \end{cases} \quad (6)$$

We analyze some models used in literature to evaluate the packets loss rate with the aim to compare its behaviour with the function depicted in Fig. 1.

The proportional – excess model

Let us consider the transmission of two consecutive routers. The node that transmits packets is called *sender*, while the receiving one is said *receiver*. Among the nodes, there is a link or channel, with limited capacity. Assume that the sender and the receiver are synchronized each other, i.e. the receiver is able to process in real time all packets, sent by the sender. In few words, no packets are lost. The packets loss can occur only on the link, due to its finite capacity. Under the zero buffer hypotheses the loss rate is defined as the proportional excess of offered traffic over the available capacity. If R is the sender bit rate and C is the link capacity, we have a loss if $R > C$. The model is said *proportional-excess* or briefly *P/E* and suppose deterministic arrivals. The packets bit rate is:

$$p = \begin{cases} 0, & R < C, \\ \frac{R-C}{C}, & R > C. \end{cases} \quad (7)$$

In Fig. 5, loss probability of the *P/E* model (continuous curve) and FD model (dashed curve) are shown, assuming $C = \sigma = 1/2$. For values $C < \rho < 2C$, the FD model overestimates the loss probability.

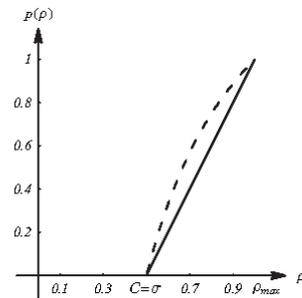


Figure 5. Loss probability. Dashed line: FD model. Continuous line: P/E model.

Observe that the *P/E* model is not realistic. In fact, the sender and the receiver are never synchronized each other and whatever transmission protocol is used by the transport layer, the receiver has a finite length buffer, where the packets wait to be processed and eventually

sent to the next node. Thus queueing models are needed, to infer about network performance.

Models with finite capacity

Queueing models are good at predicting loss in a network with many independent users, probably using different applications. Consider the traffic from TCP sources that send packets through a bottleneck link. The traffic is aggregated and used as an arrival process for the link. The aggregated throughput is computed as the sum of the throughput from all the TCP sources that send packets through the bottleneck link. The arrival process, being the aggregation of independent sources, is approximated as a Poisson process, and the aggregated throughput is used as the rate of the Poisson process (see [17]). These considerations justify the assumption that the times between the packets arrivals are exponentially distributed. Depending on the hypothesis on the length of packets arriving to the queue the data transmission can be modelled with different queueing models, as $M/D/1/B$ and $M/M/1/B$, characterized by deterministic and exponentially distributed lengths, respectively, and a buffer with capacity $B-1$. From the queue length distribution, known in closed formulas or iteratively in the finite buffer case, expected time in queue and in the system, as well as packet loss rate can be derived. In what follows we denote the arrival intensity by λ , the service intensity by μ and define the load as $\rho = \lambda/\mu$.

Fixed packets dimensions

In a scenario where all senders use the same data packets size, the queueing model $M/D/1/B$ is the most natural choice. The probability that the buffer is full gives the loss rate:

$$p(\rho) = \frac{1 + (\rho - 1)\alpha_B(\rho)}{1 + \rho\alpha_B(\rho)}, \quad (8)$$

where

$$\alpha_B(\rho) = \sum_{k=0}^{B-2} \frac{e^{\rho(B-k-1)} (-1)^k (B-k-1)^k \rho^k}{k!}, \quad B \geq 2.$$

Fig. 6 shows a comparison among the loss rate (6) and (8), assuming $B = 10$.

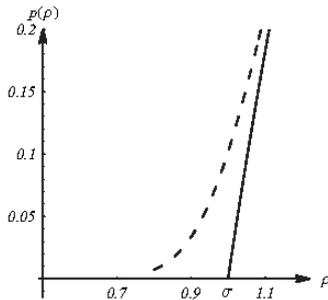


Figure 6. Loss rates. Dashed line: $M/D/1/B$ model. Continuous line: FD model.

However, an $M/D/1/B$ queue predicts lower loss rate and higher throughput than those seen in a true network. This is due to fact that in real routers packet sizes are not always fixed to the maximum segment size, therefore packet sizes are more variable than a deterministic distribution.

Exponentially distributed packet size

Assume the packet size is exponentially distributed. This assumption is true if we consider the total amount of traffic as the superposition of traffic fluxes, coming from different TCP sources, each configured to use its own packet size. The $M/M/1/B$ queue is a good approximation of the simulated bottleneck link shared among TCP sources under any traffic load ([17]). The loss rate for the $M/M/1/B$ queueing model is:

$$p(\rho) = \frac{\rho^B (1 - \rho)}{1 - \rho^{B+1}}. \quad (9)$$

In Fig. 7 the loss bit rate for different values of the buffer ($B = 10, 20, 30$) is reported. Notice that, increasing B values, dashed lines tend to the continuous one.

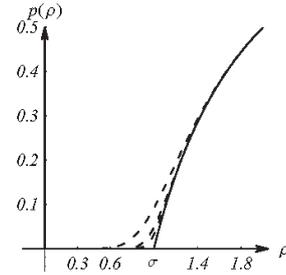


Figure 7. Loss bit rate for different values of the buffer. Dashed lines: $M/M/1/B$. Continuous line: P/E model.

In fact, the loss probability of the FD model represents for $\sigma = 1$ (up to a scale factor 2) a limit case of (9):

$$\lim_{B \rightarrow \infty} \frac{\rho^B (1 - \rho)}{1 - \rho^{B+1}} = \begin{cases} 0, & 0 < \rho \leq 1, \\ \frac{\rho - 1}{\rho}, & \rho > 1. \end{cases}$$

The loss probability for the queueing model (dashed line) and the P/E one (continuous line) is shown in Fig. 8.

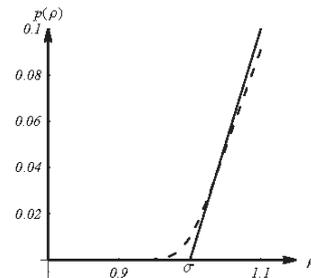


Figure 8. $M/M/1/B$ queueing model and P/E model.

The two curves almost match for small bit rate values, i.e. in the load range $0.9\sigma < \rho < 1.1\sigma$. For greater loads values, the P/E model overestimates the loss probability.

Theoretical and simulative studies pointed out that $M/D/1/B$ and $M/M/1/B$ queueing models give good prediction of the loss rate in network with many independent users performing short file transfers (shorts FTP).

In literature other queueing models have been considered to describe different scenarios, as batch arrivals. For a comparison among different models see Fig. 9, where the packet loss rate for $M/D/1/B$, $M/M/1/B$, $M^2/M/1/B$, $M^5/M/1/B$ and the P/E models are reported for the case $B=100$ and loads in the interval $0.8 < \rho < 1.1$, in such way to appreciate the different behaviour. Observe that $M^r/M/1/B$ denotes a queue with Poisson batch arrivals of size r and describes the fact that TCP traffic is likely to be quite bursty due to synchronized loss events that are experienced by multiple users.

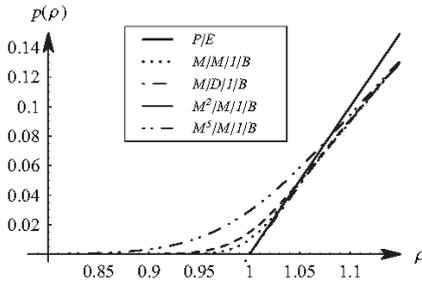


Figure 9. Comparison of different queueing models.

Significant differences are restricted to the range $0.9\sigma < \rho < 1.1\sigma$. As the load increases above 1.1 the loss estimates become very close in the different queueing models. Each of these models predict the loss rate equally well. However, under low loss environments, the best queueing model depends on the type of transfers by TCP sources, i.e. persistent or transient. It is shown in [13] that $M/D/1/B$ queues estimations of the loss rate can be used for transient sources. However, for sources with a slightly longer on and off periods, $M/M/1/B$ queues best predict the loss rate, and for (homogeneous) persistent sources, $M^r/M/1/B$ queues give better performance inferences, due to the traffic burstiness stemming from the TCP slow-start and source synchronization effect. Even if some models are more appropriate in situations of low load, others when the load is heavy, Fig. 9 shows that the assumption on the loss probability function of the FD model is valid.

B. Velocity

The loss probability, influencing the average transmission time, has effects on the average velocity of packets:

$$v(\rho) = \bar{v}(1 - p(\rho)).$$

From (6) we get

$$v(\rho) = \begin{cases} \bar{v}, & 0 \leq \rho \leq 1/2, \\ \bar{v} \frac{1-\rho}{\rho}, & 1/2 \leq \rho \leq 1, \end{cases} \quad (10)$$

whose behaviour is depicted in Fig. 1. Notice that the velocity is constant if the system is free (no losses). Over the threshold, losses occur, and the average travelling time increasing reduces the velocity. The average packet velocity for the P/E model and the $M/M/1/B$ model is plotted in Fig. 10.

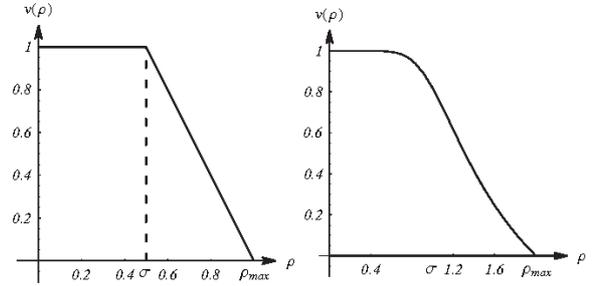


Figure 10. Average velocity. Left: P/E model. Right: $M/M/1/B$ model.

Such two curves fit the curve of the FD model, confirming the goodness of its assumptions.

C. Flux

Once the velocity function is known, the flux is given by:

$$f(\rho) = v(\rho)\rho.$$

In case of the FD model

$$f(\rho) = \begin{cases} \bar{v}\rho, & 0 \leq \rho \leq 1/2, \\ \bar{v}(1-\rho), & 1/2 \leq \rho \leq 1, \end{cases} \quad (11)$$

see Fig. 1. For the P/E model, we get

$$f(\rho) = \begin{cases} \bar{v}\rho, & 0 \leq \rho \leq \sigma, \\ \frac{\bar{v}\rho(2\sigma - \rho)}{\sigma}, & \sigma \leq \rho \leq \rho_{\max}, \end{cases} \quad (12)$$

The flux in the P/E model and $M/M/1/B$ model is depicted in Fig. 11.

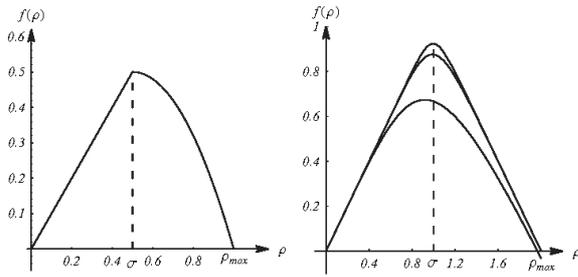


Figure 11. Flux. Left: P/E Model. Right: $M/M/1/B$ (for $B = 5$, $B = 15$, $B = 25$).

Note the effects of a finite buffer on the maximal value of the flux. If B tends to infinity, the flux best approximates the FD model flux. For small B values, the maximal flux decreases and the load value in which the maximum is attained is shifted on the right due to the fact that packets are lost for load values smaller than the threshold.

A more realistic behaviour can be obtained with the following procedure. The average time necessary to transmit a packet (as function of the load) in a queueing system $M/M/1/B$ is given by:

$$T(\rho) = \frac{1 - (B+1)\rho^B + B\rho^{B+1}}{(1-\rho)(1-\rho^B)}.$$

Once the transmission time T is known, the average transmission velocity of packets can be computed as $1/T$. Fig. 12 illustrates the transmission time and the velocity versus load for different values of the buffer capacity.

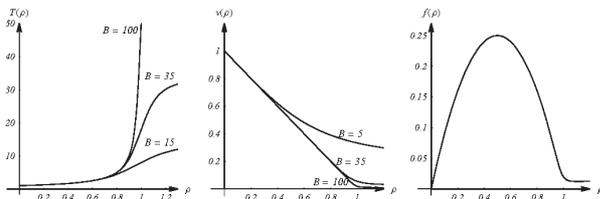


Figure 12. Transmission time and average velocity as function of the load network for different values of B . Flux function for $B = 100$.

Consider now a binary channel. Assume the loss probability is given by:

$$p(\rho) = \frac{\rho + \bar{v} - 1}{\bar{v}}.$$

Unlike other cases, for which losses did not occur for low load values, we suppose the loss probability is always not zero and it increases linearly when load increases. The left curve of Fig. 13 shows the corresponding flux, while the right curve reports the flux obtained simulating the transmission on a binary channel characterized by 100.000 packets.

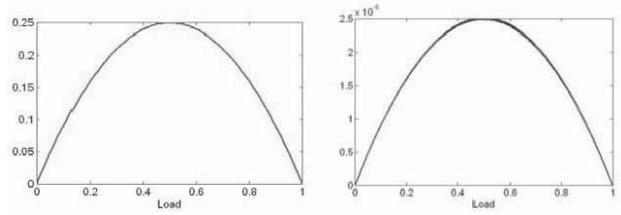


Figure 13. Flux functions for a binary channel.

IV. A CONSERVATIVE PROTOCOL: TCP

In this Section we discuss the range of applicability of the FD model in data networks, showing that it can be considered reasonable for some scenarios. Many applications on the Internet require a certain Quality of Service (QoS), which can be evaluated through performance measures such as the expected packet loss and the expected delay. The greatest part of the transport protocols used in computer networks are of best effort type, or they do the best to guarantee a minimum QoS. The Transmission Control Protocol (TCP) ensures guaranteed services while User Datagram Protocol (UDP) provides best effort delivery. In fact TCP verifies that all information transmitted is received fully on the other end. UDP makes its best to deliver packets to the destination, but takes no steps to recover packets that are lost or misdirected. Time-sensitive applications often use UDP because dropping packets is preferable to waiting for delayed packets, which may not be an option in a real-time system. If error correction facilities are needed at the network interface level, an application may use TCP.

Each endpoint of a TCP connection has a buffer for storing data that is transmitted over the network before the application is ready to read it. TCP must recover from data that is damaged, lost, duplicated, or delivered out of order by the Internet communication system. TCP assigns a sequence number to each byte transmitted, and expects a positive acknowledgment (ACK) from the receiving TCP. If the ACK is not received within a timeout interval, the data is retransmitted. The receiving TCP uses the sequence numbers to rearrange the segments when they arrive out of order, and to eliminate duplicate segments. It follows that TCP can be considered a “conservative protocol”, in fact in a wide temporal scale the information sent by the sender is equal to the information received by the receiver, as acknowledgment mechanisms allow to retransmit lost or corrupted packets. Therefore in case of applications using TCP as transport protocol, the fluid-dynamic model can be used to describe the load dynamics of the data network.

A TCP source adjusts its sending rate (window congestion size) based on the observation of events such as packet loss and delay of ACKs. By reducing or increasing window size, the server and client each ensure that the other device sends data just as fast as the recipient can deal with. With the absence of loss packets, TCP increases gradually the sending rate (additive increment). Upon experiencing packet loss, through the reception of a NACK (negative acknowledge character),

TCP adjusts the sending rate and slash window size in half (multiplicative decrement).

Different analytical models have been proposed in literature to describe TCP connections and to calculate the expected throughput of a TCP source, given known packet loss rate. We implemented a “friendly” TCP simulator (briefly TCPSim), which gives qualitative suggestions about TCP throughput dependence on the loss probability and, hence the density, with the aim to test the applicability of the FD model to describe TCP load dynamics. Assigning the number of packets to transmit and the loss probability, the tool gives as output the transmission time (measured in time-slots), the packets velocity and the flux function. The TCPSim has been realized according to the following assumptions:

- independent packets losses;
- absence of ACK losses;
- retransmission of duplicated ACKs;
- no timeout;
- steady state.

The transmission link is modelled as a binary channel. We assume p is the probability that a packet is lost or corrupted and $1-p$ the probability of a successful retransmission. To evaluate the TCPSim goodness in reproducing TCP connections, we checked if the simulated throughput (the number of packets sent per unit of time, regardless of their eventual fate, i.e. whether they are received or not) exhibits the known relationship, which turns up in many modeling situations of TCP:

$$Th(p) \sim \frac{1}{\sqrt{p}}.$$

The simulated throughput (fluctuating in Fig. 14) is in accordance with the above theoretical result (which is more realistic for little p values), hence we deduce the TCPSim correctness.

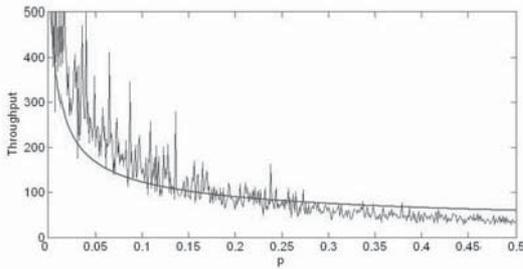


Figure 14. TCP Friendly – Test.

As we have seen, TCP, when losses are absent, regulates its own congestion window (with additive increments) sending more packets inside a time-slot and decreasing the transmitted packets when a loss occurs. The congestion control algorithm acts on the window size W as follows:

- $W \leftarrow W + 1$ if ACK is received;
- $W \leftarrow W/2$ if NACK is received.

With a low loss probability, the congestion window size can grow up linearly (starting by 1) until all packets have been transmitted (see Fig. 15, left).

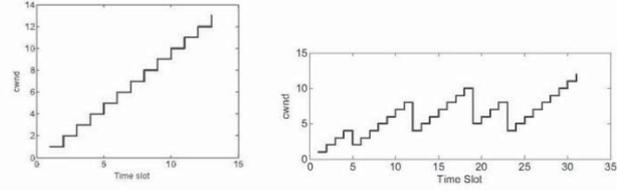


Figure 15. Congestion window in absence (left) and presence (right) of losses.

In order to transmit N packets, $T+1$ time slots are required, where T is such that $\frac{T(T+1)}{2} < N$. If the loss increases, with a given rate (on the average p), the congestion window can be reduced of about 50% as in Fig. 15, right. The number of time-slots necessary to transmit a single packet versus the loss probability is plotted in Fig. 16, on the left.

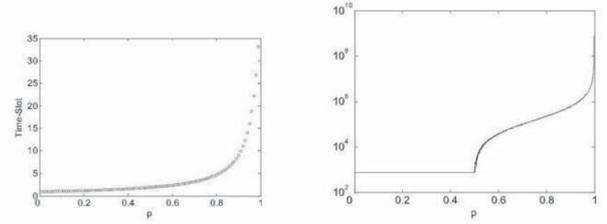


Figure 16. Transmission time of a packet (in time slots) as function of p .

If $p = 0$, exactly one time-slot is needed for the packet transmission. As p approaches to one, the needed time-slots start to increase in a meaningful way. For $0 < p < 1$, the average transmission time depends on the dynamics of the congestion window. To analyze flux function generated by TCP sources, a simple simulation is made, with $N = 300.000$ packets to transmit and a given load coefficient $\rho \in]0,1[$ which characterizes the loss probability of the FD model (i.e $\sigma = 0.5$). The algorithm consists of the following steps:

1. ρ varies in an iterative way among 0 and 1 with step 10^{-4} ;
2. $p(\rho)$ is computed;
3. the time required to retransmit N packets is evaluated (assuming TCP reduces of about 50% its congestion window pN times in the case of N transmissions attempts);
4. the transmission velocity is calculated;
5. the flux is computed.

Fig. 16 on the right plots the transmission time (in time slots) obtained after 10.000 iterations, in logarithmic scale for the vertical axes. Observe that T goes to infinity for p tending to one, while it is equal to 775 time-slots for $p < 0.5$. The velocity depicted in Fig. 17

seems similar to that proposed in the FD model, but is very different in the range beyond the threshold for reasons discussed in what follows.

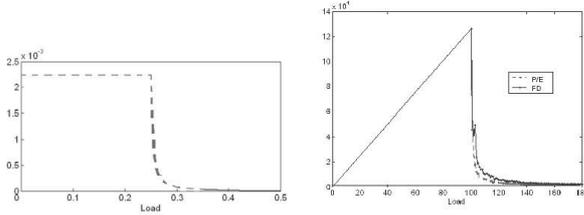


Figure 17. Velocity and TCP throughput (for FD and P/E models) as load functions.

The simulated flux function in Fig. 17 differs from the shape of Fig. 1, especially in the region on the right of the threshold. Such behaviour does not depend on the chosen loss probability. In fact, repeating the simulation with the loss probability of the P/E model, the result has no meaningful variations.

The “tent” flux is a good approximation of the real flux in TCP connections in low load regimes, and, in some cases, it is the most suitable to capture the throughput dependence on the load. However, we have to make some additional remarks. It is reasonable to assume a linear dependence for density values less than the threshold. The load is low and the loss probability is not meaningful: the congestion window increases linearly its own size and more packets are transmitted. As the load exceeds the threshold, the flux decreasing seems unrealistic. In some cases, among the increasing and decreasing phases, there is a third one, in which the flux is kept constant, as depicted in Fig. 18.

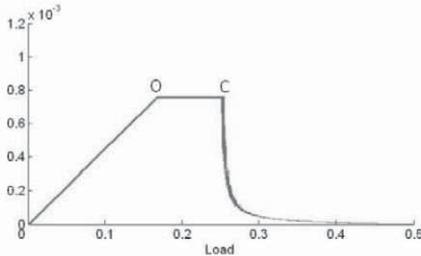


Figure 18. TCP throughput versus network load.

The point O represents the optimal “work” point (load minimum value in which the throughput reaches the maximum). The presence of a region in which the flux is constant (segment joining points O and C) is due to the finite capacity of the channel, which is responsible of packets losses mechanisms. If the transmission happens at a rate higher than the channel capacity, the loss probability is proportional to the excess of the capacity transmitting rate. A non zero probability should lead to a flux decrease while, in reality, crossing the point O , the function is kept constant. Such behaviour is justified by the buffer, which allows to move the threshold on the right and to use the whole channel potentialities (the channel can be used to the maximal capacity for more time). It is clear that it is better to use the network in load situations near to OC , nearest to O than C . After C the flux decrease is due to the fact that TCP throughput

behaves as the reciprocal of the square root of p . In fact high p values have negative effects on the throughput as they act on transmission times, reducing the congestion window. High transmission times imply low velocities and then low fluxes. Instead a great load increase implies greater losses, packets do not reach the destination but the receiver requires them, the sender transmits them, the load grows up and so the packets loss probability.

V. ROUTING ALGORITHMS

In a data network, layer protocols guide packets through the communication subset from sources to their final destinations. Given a group of routers, connected by various links, a routing algorithm finds the best path from source to destination. The *best route* (typically the *minimum cost* one) is chosen evaluating parameters like the number of hops (the trip a packet takes from one router or intermediate point to another in the network), time delay and communication cost of packet transmission. The efficiency of a routing algorithm depends on its performance, during network congestions.

Based on how routers gather information about the network and the type of its utilization, routing algorithms can be classified in global and decentralized ones. In global routing algorithms, known as LS (Link State) algorithms, every router must have complete information about other routers and the traffic status. LS algorithms compute the minimum cost path from a source to a destination, using a global knowledge of the network, in terms of connectivity among nodes and links costs. The most known of such algorithms is the Dijkstra algorithm. Instead, in decentralized routing algorithms, known as DV (Distance Vector) algorithms, each router has information about the routers that is directly connected to and not about all routers in network. According to them each router has a routing table that shows it the best route for any destination. Each router, in fact, counts the weight of the links directly connected to it and saves the information to its table. In a specific time period it sends the table to its neighbor routers (not to all routers) and receive the routing table of each of its neighbors. Based on the information collected in its neighbors' routing tables, it updates its own. Then, through an iterative process and information exchanges with neighbouring nodes, a node computes gradually the minimum costs path to a destination or a group of destinations

In the fluid dynamic model, in order to solve dynamics at node, rules RA1 and RA2 have been introduced. The algorithm RA1, already used for road traffic models, requires the definition of a traffic distribution matrix, whose coefficients describe the percentage of packets, forwarded from incoming lines to outgoing ones.

Using the algorithm RA2, not considered for urban traffic as redirections are not expected from modelling point of view (except in particular cases, as strong congestions or road closures), priority parameters, indicating priorities among flows of incoming lines, and distribution coefficients have to be assigned.

The main differences between the two algorithms are the following. The first one simply sends each packet to the outgoing line which is naturally chosen according to the final packet destination. The algorithm is blind to possible overloads of some outgoing lines and, by some abuse of notation, is similar to the behaviour of a “switch”. The second algorithm, on the contrary, sends packets to outgoing lines in order to maximize the flux both on incoming and outgoing lines, thus taking into account the loads and possibly redirecting packets. Again by some abuse of notation, this is similar to a “router” behaviour. Hence, RA1 forwards packets on outgoing lines without considering the congestion phenomena, unlike RA2.

Observe that a routing algorithm of RA1 type working through a routing table, according to which flows are sent with prefixed probabilities to the outgoing links, is of “distance vector” type.

Reverse, an algorithm of RA2 type can redirect packets on the basis of link congestions, so it works on the link states (hence on their congestions) and so it is of “link-state” type.

VI. CONCLUSIONS

In this paper we examined the validity of a fluid dynamic model for data networks. More precisely, the model is based on packets conservation at intermediate time scales, whose flux is determined via a loss probability function (at fast time scales).

The choice of the loss probability function is of paramount importance in order to achieve a feasible model. We compared the fluid dynamic model with those obtained using various queueing paradigms, from proportional/excess to models with finite capacity, including different distributions for packet sizes. The final result is that such models give rise to velocity profiles and flux functions which are quite similar to the fluid dynamic ones.

Then we considered the throughputs of TCP protocols: the result is again feasibility of the fluid dynamic model for low load regimes, while corrections are in order in congestion situations.

Last we focused on the routing “algorithms” used at nodes. Two different algorithms, used in the fluid dynamic approach, essentially corresponds to the policy used, respectively, by a “switch” and a “router”.

Finally, we can claim that the fluid dynamic model gives rise to network behaviors that are comparable to various different approaches well known in the literature and widely used. On the other side, there are some corrections in order especially at near congestion regimes. The developed analysis allows us to conclude that the fluid dynamic approach is feasible to model some phenomena of load evolution in data networks.

REFERENCES

[1] D. Alderson, H. Chang, M. Roughan, S. Uhlig and W. Willinger, “The many facets of internet topology and traffic,” *Netw. Heterog. Media*, vol. 1, Issue 4, pp. 569–600, 2007.

[2] F. Baccelli, A. Chaintreau, D. De Vleeschauwer and D. McDonald, “HTTP turbulence,” *Netw. Heterog. Media*, vol. 1, pp. 1–40, 2006.

[3] F. Baccelli, D. Hong and Z. Liu, “Fixed points methods for the simulation of the sharing of a local loop by large number of interacting TCP connections,” *Tech. Rep. RR-4154, INRIA*, 2001.

[4] G. Coclite, M. Garavello and B. Piccoli, “Traffic Flow on a Road Network,” *SIAM J. Math. Anal.*, vol. 36, pp. 1862–1886, 2005.

[5] C. D’Apice, R. Manzo and B. Piccoli, “Packets flow on telecommunication network,” *SIAM J. Math. Anal.*, vol. 38, no. 3, pp. 717–740, 2006.

[6] C. D’Apice, R. Manzo and B. Piccoli, “A fluid dynamic model for telecommunication networks with sources and destinations,” *SIAM J. Appl. Math.*, vol. 68, no. 4, pp. 981–1003, 2008.

[7] M. Garavello and B. Piccoli, *Traffic flow on networks*, AIMS book series on Applied Mathematics, 2006.

[8] T. Kelly, “Scalable TCP: improving performance in highspeed wide area networks,” *ACM SIGCOMM Computer Communication Reviews*, vol. 33, no. 2, pp. 83–91, 2003.

[9] F. Kelly, A.K. Maulloo and D.K.H. Tan, “Rate control in communication networks: shadow prices, proportional fairness and stability,” *J. Oper. Res. Soc.*, vol. 49, pp. 237–252, 1998.

[10] Y. T. Li, D. J. Leith, R. Shorten, “Experimental evaluation of TCP protocols for high-speed networks,” *IEEE/ACM Trans. Netw.*, 15(5), pp. 1109–1122, 2007.

[11] M. Mathis, J. Semke, and J. Mahdavi, “The macroscopic behavior of the TCP congestion avoidance algorithm,” *ACM Computer Communication Review*, vol. 27, no. 3, pp. 67–82, 1997.

[12] R. Mbarek, M. T. B. Othman, S. Nasri, “Fairness of High-Speed TCP Protocols with Different Flow Capacities,” *Journal of Networks*, vol 4, no. 3, pp. 163–169, 2009.

[13] J. Olsén, *On Packet Loss Rates used for TCP Network Modeling*, Technical Report, Uppsala University, 2003.

[14] L. Senthilkumar, V. Sankaranarayanan, “Integrated Proactive Admission Control Technique For both UDP And TCP Traffic Flows,” *Journal of Networks*, vol 2, no. 1 pp. 1–8, 2007.

[15] K. Tan, J. Song, Q. Zhang, and M. Sridharan, “A compound TCP approach for high-speed and long distance network,” *Proceeding of IEEE INFOCOM ’06*, Barcelona, Spain, 23–29 April 2006, pp. 1–12, 2006.

[16] A.S. Tanenbaum, *Computer Networks*, Prentice Hall, Upper Saddle River, 2003.

[17] A. Wierman, T. Osogami, J. Olsén, “A Unified Framework for Modeling TCP-Vegas, TCP-SACK, and TCP-Reno,” *Proc. IEEE/ACM International Symposium on modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2003.

[18] W. Willinger and V. Paxson, “Where Mathematics meets the Internet,” *Not. AMS*, vol. 45, pp. 961–970, 1998.



Ciro D’Apice was born in Castellammare di Stabia, Italy and obtained PhD degrees in Mathematics in 1996. He is full professor in Mathematical Analysis at the Department of Electronic and Computer Engineering of the University of Salerno. He is author of approximately 140 publications, with more than 100 journal papers about homogenization and

optimal control; conservation laws models for vehicular traffic, telecommunications and supply chains; spatial behaviour for dynamic problems; queueing systems and networks.

His e-mail address is cdapice@unisa.it.



Rosanna Manzo was born in Polla, Salerno, Italy. She graduated cum laude in Mathematics in 1996 and obtained PhD in Information Engineering in 2007 at the University of Salerno. She is a researcher in Mathematical Analysis at the Department of Electronic and Computer Engineering of the University of Salerno. Her research

areas include fluid – dynamic models for traffic flows on road, telecommunication and supply networks, optimal control, queueing theory, self – similar processes, computer aided learning. She is author of about 40 papers appeared on international journals and many publications on proceedings.

Her e-mail address is manzo@unisa.it.



Benedetto Piccoli is the Joseph and Loretta Lopez Chair in Mathematics at the Department of Mathematical Sciences, Rutgers University, Camden, NJ, US. He received the PhD in Mathematics from SISSA-SAS, Trieste Italy. His research interests include:

optimal control, hybrid control, systems biology, flows on networks, mathematical finance and applied math in general. He is author of 4 books and more than 150 papers. He is the Eic of Networks and Heterogeneous Media and recipient of the Fubini Prize 2009.

His e-mail address is piccoli@camden.rutgers.edu.