

# Modelling Transient States in Queueing Models of Computer Networks: A Few Practical Issues

Tadeusz Czachórski<sup>1</sup>(✉), Monika Nycz<sup>2</sup>, and Tomasz Nycz<sup>2</sup>

<sup>1</sup> Institute of Theoretical and Applied Informatics of Polish Academy of Sciences,  
Gliwice, Poland

tadek@iitis.pl

<sup>2</sup> Institute of Informatics, Silesian Technical University, Gliwice, Poland  
{Monika.Nycz, Tomasz.Nycz}@polsl.pl

**Abstract.** The article summarises author's experience in two problems related to the use of queueing models in performance evaluation of computer networks: modelling transient states of queues and computations for queueing network models having large number of nodes. Both issues are not well represented in classical queueing theory, yet important to applications, because the observed traffic is time dependant and network topologies that should be considered become larger and larger. The article discusses two approaches: diffusion approximation and fluid-flow approximation that can cope with much larger models that are attainable with the use of Markov chains.

## 1 Introduction

Queueing models are frequently used in modelling and evaluation of computer networks. Queueing theory, introduced a century ago to model telephone exchanges was successfully adapted to the needs of computer science but new problems arise following the constant development of computer networks. The problems are mainly related to computational aspects of queueing models and more precisely to the need of taking into account very large topologies, corresponding to real ones encountered in computer networks and the necessity to analyse transient behaviour of queues, as the intensity of traffic flows generated by users, e.g. internet applications is permanently changing. The quality of transmission services depends on current load of links and not on its average value. Also modelling and understanding the performance of traffic control mechanisms, control stability and its impact on quality of service needs transient state analysis, e.g. [27].

The use of analytical models known in classical queueing theory is limited to single M/M/1 and M/M/1/N stations and even there the transient state solutions are quite complex. Moreover, the results refer to transient states but it is assumed that the model parameters, the input rate in particular, are constant. Therefore, they do not fit well the problem of modelling IP routers, where the incoming streams of packets are not Poisson and the size of packets is not

exponentially distributed. We need models describing constantly changing non-Poisson flows and considering general distributions of service times. We need also the possibility to include in these models the description of self-similarity of flows. The models should also be scalable to meet very large topologies characteristic to the Internet.

In sections that follow we describe our experience and present simple numerical examples referring diffusion approximation, and fluid-flow approximation – the approaches we think are most suitable to this purposes.

## 2 Diffusion Approximation of a Single Queue

This approach is merging states of the considered queueing system and thus needs much less computations than the Markov models. We present here the principles of the method following [9] where steady-state solution of a single G/G/1/N model was given and then extended to the network of queues in [10]. We supplemented these results with semi-analytical, semi-numerical transient state solution [3] given for constant model parameters but it could be applied also in case of time-dependent parameters if we only make them constant within small intervals.

Let  $A(x)$ ,  $B(x)$  denote the interarrival and service time distributions at a service station and  $a(x)$  and  $b(x)$  be their density functions. The distributions are general but not specified, the method requires only the knowledge of their two first moments. The means are denoted as  $E[A] = 1/\lambda$ ,  $E[B] = 1/\mu$  and variances are  $\text{Var}[A] = \sigma_A^2$ ,  $\text{Var}[B] = \sigma_B^2$ . Denote also squared coefficients of variation  $C_A^2 = \sigma_A^2 \lambda^2$ ,  $C_B^2 = \sigma_B^2 \mu^2$ .  $N(t)$  represents the number of customers present in the system at time  $t$ .

Diffusion approximation, replaces the process  $N(t)$  by a continuous diffusion process  $X(t)$ , the incremental changes  $dX(t) = X(t + dt) - X(t)$  of which are normally distributed with the mean  $\beta dt$  and variance  $\alpha dt$ , where  $\beta$ ,  $\alpha$  are coefficients of the diffusion equation

$$\frac{\partial f(x, t; x_0)}{\partial t} = \frac{\alpha}{2} \frac{\partial^2 f(x, t; x_0)}{\partial x^2} - \beta \frac{\partial f(x, t; x_0)}{\partial x}. \quad (1)$$

This equation defines the conditional pdf of  $X(t)$ :

$$f(x, t; x_0) dx = P[x \leq X(t) < x + dx \mid X(0) = x_0].$$

The both processes  $X(t)$  and  $N(t)$  have normally distributed changes; the choice  $\beta = \lambda - \mu$ ,  $\alpha = \sigma_A^2 \lambda^3 + \sigma_B^2 \mu^3 = C_A^2 \lambda + C_B^2 \mu$  ensures that the parameters of these distributions increase at the same rate with the length of the observation period. In the case of G/G/1/N station, the process evolves between barriers placed at  $x = 0$  and  $x = N$  where barriers *with instantaneous jumps* are placed, [9]. When the diffusion process comes to  $x = 0$ , it remains there for a time exponentially distributed with a parameter  $\lambda_0$  and then it returns to  $x = 1$ . The time when the process is at  $x = 0$  corresponds to the idle time of the system.

When the process comes to the barrier at  $x = N$ , it stays there for a time which is exponentially distributed with a parameter  $\mu_0$  which corresponds to the time when the system is full and do not accept new customers (the completion time of current service from the moment when the queue becomes full). The assumption on exponential sojourn times in barriers will be dropped below where transient model is presented. Diffusion equation becomes and is supplemented by balance equations for probabilities  $p_0(t)$  and  $p_N(t)$  of being at the barriers

$$\begin{aligned} \frac{\partial f(x, t; x_0)}{\partial t} &= \frac{\alpha}{2} \frac{\partial^2 f(x, t; x_0)}{\partial x^2} - \beta \frac{\partial f(x, t; x_0)}{\partial x} + \\ &\quad + \lambda_0 p_0(t) \delta(x - 1) + \lambda_N p_N(t) \delta(x - N + 1), \\ \frac{dp_0(t)}{dt} &= \lim_{x \rightarrow 0} \left[ \frac{\alpha}{2} \frac{\partial f(x, t; x_0)}{\partial x} - \beta f(x, t; x_0) \right] - \lambda_0 p_0(t), \\ \frac{dp_N(t)}{dt} &= \lim_{x \rightarrow N} \left[ -\frac{\alpha}{2} \frac{\partial f(x, t; x_0)}{\partial x} + \beta f(x, t; x_0) \right] - \lambda_N p_N(t), \end{aligned} \tag{2}$$

where  $\delta(x)$  is Dirac delta function.

Our solution of these equations is based on the representation of the density function  $f(x, t; x_0)$  of the diffusion process with barriers with jumps by a superposition of the density functions  $\phi(x, t; x_0)$  of diffusion processes with absorbing barriers at  $x = 0$  and  $x = N$ , which has the following form

$$\phi(x, t; x_0) = \begin{cases} \delta(x - x_0) & \text{for } t = 0 \\ \frac{1}{\sqrt{2\pi\alpha t}} \sum_{n=-\infty}^{\infty} \left\{ \exp \left[ \frac{\beta x'_n}{\alpha} - \frac{(x-x_0-x'_n-\beta t)^2}{2\alpha t} \right] \right. \\ \quad \left. - \exp \left[ \frac{\beta x''_n}{\alpha} - \frac{(x-x_0-x''_n-\beta t)^2}{2\alpha t} \right] \right\} & \text{for } t > 0, \end{cases} \tag{3}$$

where  $x'_n = 2nN$ ,  $x''_n = -2x_0 - x'_n$ . If the initial condition is defined by a function  $\psi(x)$ ,  $x \in (0, N)$ ,  $\lim_{x \rightarrow 0} \psi(x) = \lim_{x \rightarrow N} \psi(x) = 0$ , then the pdf of the process has the form  $\phi(x, t; \psi) = \int_0^N \phi(x, t; \xi) \psi(\xi) d\xi$ .

The probability density function  $f(x, t; \psi)$  of the diffusion process with elementary returns is composed of the function  $\phi(x, t; \psi)$  which represents the influence of the initial conditions and of a spectrum of functions  $\phi(x, t - \tau; 1)$ ,  $\phi(x, t - \tau; N - 1)$  which are pd functions of diffusion processes with absorbing barriers at  $x = 0$  and  $x = N$ , started at time  $\tau < t$  at points  $x = 1$  and  $x = N - 1$  with densities  $g_1(\tau)$  and  $g_{N-1}(\tau)$ :

$$f(x, t; \psi) = \phi(x, t; \psi) + \int_0^t g_1(\tau) \phi(x, t - \tau; 1) d\tau + \int_0^t g_{N-1}(\tau) \phi(x, t - \tau; N - 1) d\tau. \tag{4}$$

Densities  $\gamma_0(t)$ ,  $\gamma_N(t)$  of probability that at time  $t$  the process enters to  $x = 0$  or  $x = N$  are

$$\begin{aligned}
\gamma_0(t) &= p_0(0)\delta(t) + [1 - p_0(0) - p_N(0)]\gamma_{\psi,0}(t) + \int_0^t g_1(\tau)\gamma_{1,0}(t - \tau)d\tau \\
&\quad + \int_0^t g_{N-1}(\tau)\gamma_{N-1,0}(t - \tau)d\tau, \\
\gamma_N(t) &= p_N(0)\delta(t) + [1 - p_0(0) - p_N(0)]\gamma_{\psi,N}(t) + \int_0^t g_1(\tau)\gamma_{1,N}(t - \tau)d\tau \\
&\quad + \int_0^t g_{N-1}(\tau)\gamma_{N-1,N}(t - \tau)d\tau, \tag{5}
\end{aligned}$$

where  $\gamma_{1,0}(t)$ ,  $\gamma_{1,N}(t)$ ,  $\gamma_{N-1,0}(t)$ ,  $\gamma_{N-1,N}(t)$  are densities of the first passage time between corresponding points, e.g.

$$\gamma_{1,0}(t) = \lim_{x \rightarrow 0} \left[ \frac{\alpha}{2} \frac{\partial \phi(x, t; 1)}{\partial x} - \beta \phi(x, t; 1) \right]. \tag{6}$$

For absorbing barriers

$$\lim_{x \rightarrow 0} \phi(x, t; x_0) = \lim_{x \rightarrow N} \phi(x, t; x_0) = 0,$$

hence  $\gamma_{1,0}(t) = \lim_{x \rightarrow 0} \frac{\alpha}{2} \frac{\partial \phi(x, t; 1)}{\partial x}$ . The functions  $\gamma_{\psi,0}(t)$ ,  $\gamma_{\psi,N}(t)$  denote densities of probabilities that the initial process, started at  $t = 0$  at the point  $\xi$  with density  $\psi(\xi)$  will end at time  $t$  by entering respectively  $x = 0$  or  $x = N$ .

Finally, we may express  $g_1(t)$  and  $g_N(t)$  with the use of functions  $\gamma_0(t)$  and  $\gamma_N(t)$ :

$$g_1(\tau) = \int_0^\tau \gamma_0(t)l_0(\tau - t)dt, \quad g_{N-1}(\tau) = \int_0^\tau \gamma_N(t)l_N(\tau - t)dt, \tag{7}$$

where  $l_0(x)$ ,  $l_N(x)$  are the densities of sojourn times in  $x = 0$  and  $x = N$ ; the distributions of these times are not restricted to exponential ones as it is in Eq. (2).

The above equations are transformed by the Laplace transform, and the transform of  $f(x, t, x_0)$  is obtained analytically and then its original is computed numerically using e.g. Stehfest algorithm [22].

In case of unlimited queue of G/G/1 type we just remove the barrier at  $x = N$  and related to it terms and equations.

### 3 Open Network of G/G/1, G/G/1/N Queues, One Class, Steady State and Transient Solution

The steady-state open networks models of G/G/1 queues were studied in [10]. Let  $M$  be the number of stations and suppose at the beginning that there is one class of customers. The throughput of station  $i$  is, as usual, obtained from traffic equations

$$\lambda_i = \lambda_{0i} + \sum_{j=1}^M \lambda_j r_{ji}, \quad i = 1, \dots, M, \tag{8}$$

where  $r_{ji}$  is routing probability between station  $j$  and station  $i$ ;  $\lambda_{0i}$  is external flow of customers coming from outside of network.

Second moment of interarrival time distribution is obtained from two systems of equations; the first defines  $C_{Di}^2$  as a function of  $C_{Ai}^2$  and  $C_{Bi}^2$ ; the second defines  $C_{Aj}^2$  as another function of  $C_{D1}^2, \dots, C_{DM}^2$ :

(1) The formula (9) is exact for  $M/G/1$ ,  $M/G/1/N$  stations and is approximate in the case of non-Poisson input [1]

$$d_i(t) = \varrho_i b_i(t) + (1 - \varrho_i) a_i(t) * b_i(t), \quad i = 1, \dots, M, \quad (9)$$

where  $*$  denotes the convolution operation. From (9) we get

$$C_{Di}^2 = \varrho_i^2 C_{Bi}^2 + C_{Ai}^2(1 - \varrho_i) + \varrho_i(1 - \varrho_i). \quad (10)$$

(2) Customers leaving station  $i$  according to the distribution  $D_i(x)$  choose station  $j$  with probability  $r_{ij}$ : intervals between customers passing this way has pdf  $d_{ij}(x)$

$$d_{ij}(x) = d_i(x)r_{ij} + d_i(x) * d_i(x)(1 - r_{ij})r_{ij} + d_i(x) * d_i(x) * d_i(x)(1 - r_{ij})^2 r_{ij} + \dots \quad (11)$$

hence

$$E[D_{ij}] = \frac{1}{\lambda_i r_{ij}}, \quad C_{Dij}^2 = r_{ij}(C_{Di}^2 - 1) + 1. \quad (12)$$

$E[D_{ij}]$ ,  $C_{Dij}^2$  refer to interdeparture times; the number of customers passing from station  $i$  to  $j$  in a time interval  $t$  has approximately normal distribution with mean  $\lambda_i r_{ij} t$  and variation  $C_{Dij}^2 \lambda_i r_{ij} t$ . The sum of streams entering station  $j$  has normal distribution with mean

$$\lambda_j t = \left[ \sum_{i=1}^M \lambda_i r_{ij} + \lambda_{0j} \right] t \quad \text{and variance} \quad \sigma_{Aj}^2 t = \left\{ \sum_{i=1}^M C_{Dij}^2 \lambda_i r_{ij} + C_{0j}^2 \lambda_{0j} \right\} t,$$

hence

$$C_{Aj}^2 = \frac{1}{\lambda_j} \sum_{i=1}^M r_{ij} \lambda_i [(C_{Di}^2 - 1)r_{ij} + 1] + \frac{C_{0j}^2 \lambda_{0j}}{\lambda_j}. \quad (13)$$

Parameters  $\lambda_{0j}$ ,  $C_{0j}^2$  represent the external stream of customers. For  $K$  classes of customers with routing probabilities  $r_{ij}^{(k)}$  (let us assume for simplicity that the customers do not change their classes) we have

$$\lambda_i^{(k)} = \lambda_{0i}^{(k)} + \sum_{j=1}^M \lambda_j^{(k)} r_{ji}^{(k)}, \quad i = 1, \dots, M; \quad k = 1, \dots, K, \quad (14)$$

and

$$C_{Di}^2 = \lambda_i \sum_{k=1}^K \frac{\lambda_i^{(k)}}{\mu_i^{(k)2}} [C_{Bi}^{(k)2} + 1] + 2\varrho_i(1 - \varrho_i) + (C_{Ai}^2 + 1)(1 - \varrho_i) - 1. \quad (15)$$

A customer in the stream leaving station  $i$  belongs to class  $k$  with probability  $\lambda_i^{(k)}/\lambda_i$  and we can determine  $C_{Di}^{(k)2}$  in the similar way as it has been done in Eqs. (11–12), replacing  $r_{ij}$  by  $\lambda_i^{(k)}/\lambda_i$ :

$$C_{Di}^{(k)2} = \frac{\lambda_i^{(k)}}{\lambda_i} (C_{Di}^2 - 1) + 1; \tag{16}$$

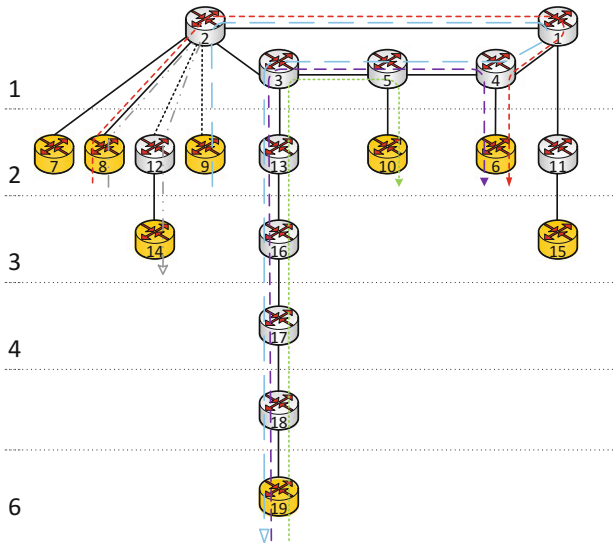
then

$$C_{Aj}^2 = \frac{1}{\lambda_j} \sum_{l=1}^K \sum_{k=1}^K r_{ij}^{(k)} \lambda_i \left[ \left( \frac{\lambda_i^{(k)}}{\lambda_i} (C_{Di}^2 - 1) \right) r_{ij}^{(k)} + 1 \right] + \sum_{k=1}^K \frac{C_{0j}^{(k)2} \lambda_{0j}^{(k)}}{\lambda_j}. \tag{17}$$

Equations (10), (13) or (15), (17) form a linear system of equations and allow us to determine  $C_{Aj}^2$  and, in consequence, parameters  $\beta_i, \alpha_i$  for each station.

In our approach to transient analysis, the time axis is divided into small intervals (equal e.g. to the smallest mean service time) and at the beginning of each interval the Eqs. (8), (10), (13) are used to determine the input parameters of each station based on the values of  $\varrho_i(t)$  obtained at the end of the precedent interval. As the values of parameters are changed at each interval, also external flows  $\lambda_{0j}^{(k)}(t)$  may be modelled following any, possibly self-similar process.

**Numerical example 1.** The considered exemplary network topology is presented in Fig. 1. It was generated by aSHIIP generator [29] allowing generation of hierarchical networks, typical for Internet - this sample network consists of 6 levels. The same topology was used in an example referring to the fluid flow



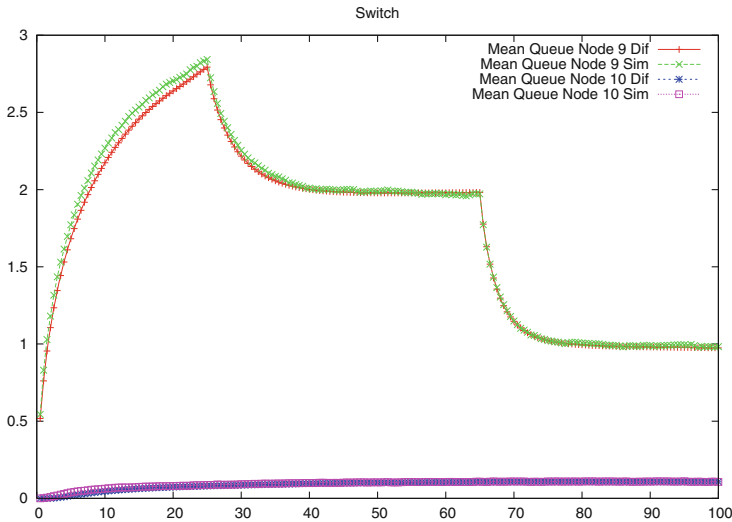
**Fig. 1.** Topology of the network considered in numerical examples.

**Table 1.** Generated flows  $\lambda_i$ ,  $i = 8, 9, 19$ , as a function of time

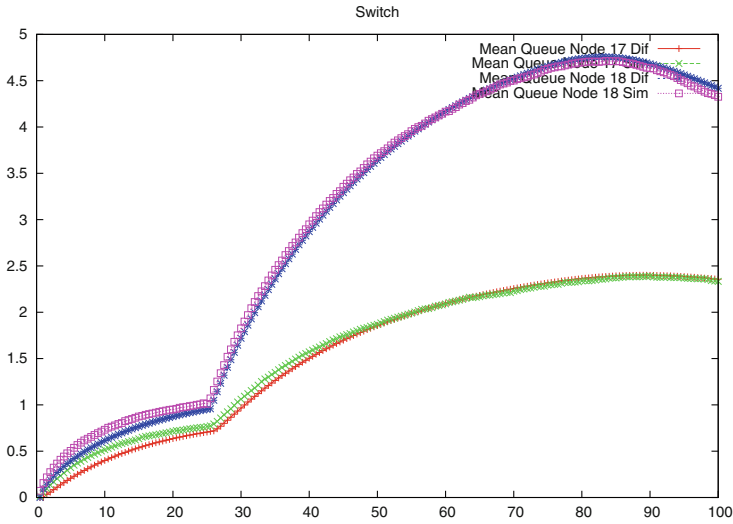
time units	0–20	20–25	25–65	65–70	70–75	75–100
node 8	0.8	1.7	1.7	1.7	1.7	1.3
node 9	1.8	1.8	1.5	1.0	1.0	1.0
node 19	1.0	1.0	2.5	2.5	2.5	1.5

approximation presented in the next section. We do not compare the results, as diffusion model does not incorporate the TCP congestion window mechanism (although it is possible) and the loads of networks are different in both examples. The examples demonstrate rather the possibilities of both approaches. Here, flows are generated by nodes 8, 9, 19 and their intensity is piecewise, given in Table 1, the routing of flows is indicated in the figure. All nodes have the same service intensity  $\mu = 3$ , the queue capacity is  $N = 20$ , and the squared coefficient of variation for all flaws and stations are:  $C_A^2 = C_B^2 = 1$ . The propagation time between nodes is null (it is easy to compute knowing the length of the links and the speed of light in nodes).

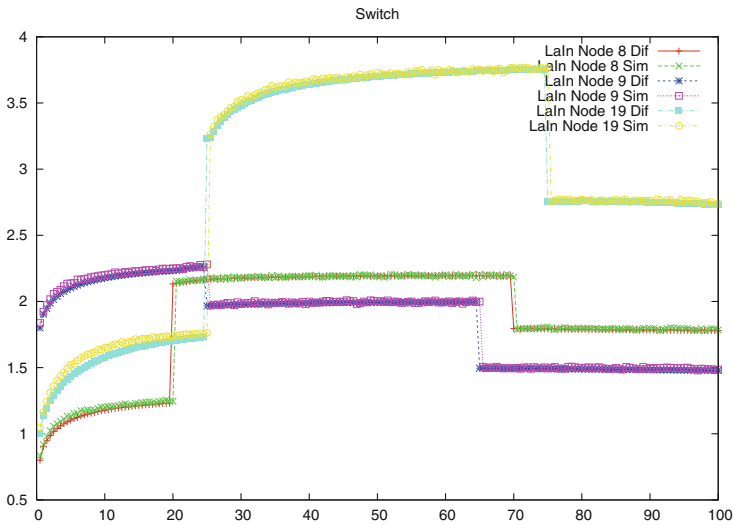
In Figs. 2, 3 the time evolution of mean queues in a few chosen nodes, predicted by diffusion approximation and simulation (we treat simulation results as almost exact) are compared, giving the idea of the errors of the approach. The model naturally gives not only these mean values but also the distributions of queues. Figure 4 compares the time-dependant flow intensities at certain nodes obtained via diffusion approximation and simulation. With our software we may generate and solve numerically much larger models, having hundreds of thousands or millions nodes and flows. It seems that our solver concerning transient diffusion models is the unique existing one.



**Fig. 2.** Mean queue lengths at nodes 9 and 10, diffusion approximation and simulation results



**Fig. 3.** Mean queue lengths at nodes 9 and 10, diffusion approximation and simulation results



**Fig. 4.** Flow intensities at nodes 8, 9, and 19, diffusion approximation and simulation results

### 4 Fluid-Flow Approximation

Fluid-flow approximation is a well-known approach of modelling transient behaviour where only mean values of traffic intensity and service times are considered. Compared to the diffusion approximation, mathematical side of the model is



simple: instead of partial differential equations of second order, the first-order ordinary linear differential equations are used. Due to its simplicity, it gained much interest in the analysis of transient states in Internet and in investigation of TCP-IP connections stability [4, 31]. Some solvers already exist [6, 7] but we have developed our own to have a better inside to its functionalities and to optimise its performance.

Fluid approximation gives larger *methodological errors* than the diffusion approximation which is a second-order approximation and considers not only the mean values but also the variance of flow changes. We have also observed it our tests [5].

Here we present the method in a form proposed in [11, 14], already adapted to TCP congestion window mechanism and RED algorithm in routers, hence in an easy way incorporating some essential details of Internet transmissions.

Let the modelled network  $V$  be composed of routers. The fluid approximation computes the average values of the queues at the routers while the implemented RED mechanism requires the instantaneous values of them to estimate discard probability. Hence, the instantaneous and average queue lengths in the network are noted by the vectors  $\mathbf{q}$  and  $\mathbf{x}$ . The values of the routers' discard probabilities depend on the instantaneous queue length and are recorded in vector  $\mathbf{p}(\mathbf{x})$  which depends for the router  $v \in V$  only on  $x_v$ . The network structure is represented by binary matrix  $\mathbf{A}$ . Its rows correspond to TCP flows and the columns represent network nodes. If a flow  $i$  traverses a node  $k$ , the value of the element  $a_{ik}$  is determined as "one", otherwise the element is set to "zero". The matrix  $\mathbf{A}$  and vector  $\mathbf{p}(\mathbf{x})$  are used to define a new matrix  $\mathbf{B}$ : the rows of the matrix  $\mathbf{B}$  are formed by multiplying the corresponding element of  $\mathbf{p}$  and a row of  $\mathbf{A}$ , such that  $B_{ij} = A_{ij} \cdot p_j(x_j)$ . The  $\mathbf{B}$  matrix is used to calculate the total packet loss probability on the path of the entire flow. Each row of the matrix stores the probabilities for routers on the route. To determine the total loss probability, it is necessary to calculate all possible combinations of packet drop probabilities on the path from source to destination. The way to do it is to calculate the success probability (the successful packet arrival traverse through all nodes on the path). The dynamics of the congestion window  $W_i(t)$  at a connection  $i$  is:

$$\frac{dW_i(t)}{dt} = \frac{1}{R_i(\mathbf{q}(t))} - \frac{W_i(t)}{2} \cdot \frac{W_i(t - \tau)}{R_i(\mathbf{q}(t - \tau))} \cdot \left( 1 - \prod_{j \in V} (1 - B_{ij}) \right). \quad (18)$$

The other equations concern the mean queue length  $q_v$  of router  $v$  at this instant, the average queue  $x_v$  and delays  $R_i$ .

The router's AQM packet discard probability  $p(x(t))$ , Eq. 20, included in the above formula is determined with the use of the weighted average queue length:

$$x_v(k\delta) = (1 - \gamma_v) \cdot x_v((k - 1)\delta) + \gamma_v q_v(k\delta) \quad (19)$$

where:

- $\delta$  – queue sampling parameter,
- $\gamma$  – weight parameter, specifying the percentage of current queue  $q$  taken in the moving average,
- $k$  – iteration step,

$$p_v(x_v) = \begin{cases} 0, & 0 \leq x_v < t_{min_v} \\ \frac{x_v - t_{min_v}}{t_{max_v} - t_{min_v}} p_{max_v}, & t_{min_v} \leq x_v < t_{max_v} \\ 1, & t_{max_v} \leq x_v \leq B_v \end{cases} \quad (20)$$

$x_v(t), q_v(t)$  are respectively the average and instantaneous queue lengths in router  $v$ . With the transmission capacity  $C_v$  the time change of  $q_v$  is

$$\frac{dq_v(t)}{dt} = \sum_{i=1}^K \frac{W_i(t)}{R_i(\mathbf{q}(t))} - \mathbf{1}(q_v(t) > 0) \cdot C_v, \quad (21)$$

$C_v$  is transmission capacity of the router  $v$ . A router allows reception of traffic from  $K$  TCP flows ( $K \leq N$ ). Each flow  $i (i = 1, \dots, N)$  is determined by time varying congestion window size  $W_i$ , measured in packets and constant propagation delay  $Tp_i$  throughout flow route. Total packet delay for flow  $i$  (Round Trip Time) consists of queue delay and propagation delay.

$$R_i(\mathbf{q}(t)) = \sum_{j=1}^M \frac{q_j(t)}{C_j} + Tp_i. \quad (22)$$

**Numerical example 2.** The topology of the considered network is the same as in Example 1 and generated by aSHIIP. The flows, propagation times, and buffer sizes at routers were also generated randomly. Table 2 illustrates the network parameters:  $B$  - maximum buffer capacity;  $C$  - service intensity;  $Q_0$  - initial queue length;  $\gamma, t_{min}, t_{max}, p_{max}$  - weight, thresholds and maximum probability for RED algorithm. On the whole structure, we ran our algorithm for selecting the pair of border routers as a flow endpoints and searched for the shortest paths from the sender to the receiver using the Dijkstra algorithm. Table 3 shows the flow parameters:  $W_0$  - starting window size;  $Tp$  - total propagation delay that is the sum of the link delays on the route.

**Table 2.** Router parameters

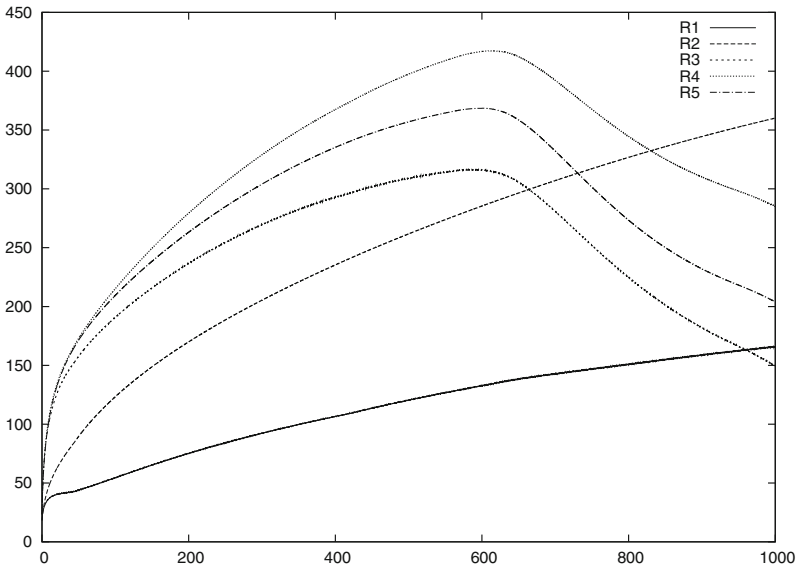
parameter \ node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
B	91	370	90	114	353	246	169	152	160	271	162	141	34	43	80	81	18	17	15	[pack]
C	0.15	0.225	0.225	0.225	0.225	0.15	0.15	0.15	0.075	0.075	0.15	0.075	0.225	0.075	0.15	0.225	0.225	0.225	0.225	[pack/t.u.]
$Q_0$	0																			[pack]
$\gamma$	0.05																			[-]
$t_{min}$	15	61	15	19	58	41	28	25	26	45	27	23	5	7	13	13	3	2	5	[pack]
$t_{max}$	45	185	45	57	176	123	84	76	80	135	81	70	17	21	40	40	9	8	10	[pack]
$p_{max}$	0.1																			[-]

**Table 3.** Flow parameters

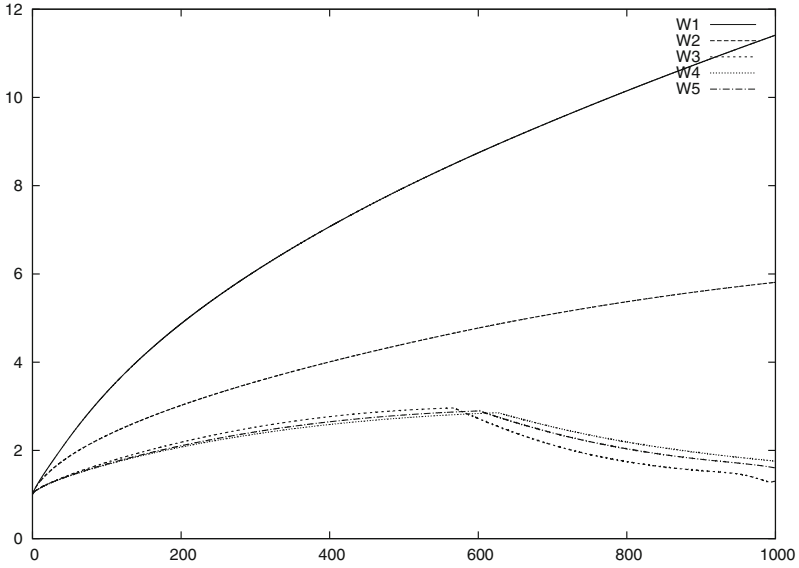
parameter \ flow	1	2	3	4	5	
$Tp$	16.61	14.98	53.38	42.01	30.72	[t.u.]
$W_0$	1					[pack]

Few results are displayed, e.g. the RTT as a function of time for flow classes, Fig. 5. The value of RTT specifies the time needed to propagate an information through the network after which a sender may react on losses. The first overload occurs roughly at  $t = 300$  time units (t.u.) and the RTT time of classes 3–5 at that moment ranges between [260, 325], therefore the expected moment when the sender reduces the transmitted traffic is after about 300 t.u. It is visible in Fig. 6 – the window sizes of class 3–5 are reduced at a time close to  $t = 600$  t.u. For other classes the losses are so small that the continuous increase of window sizes is observed.

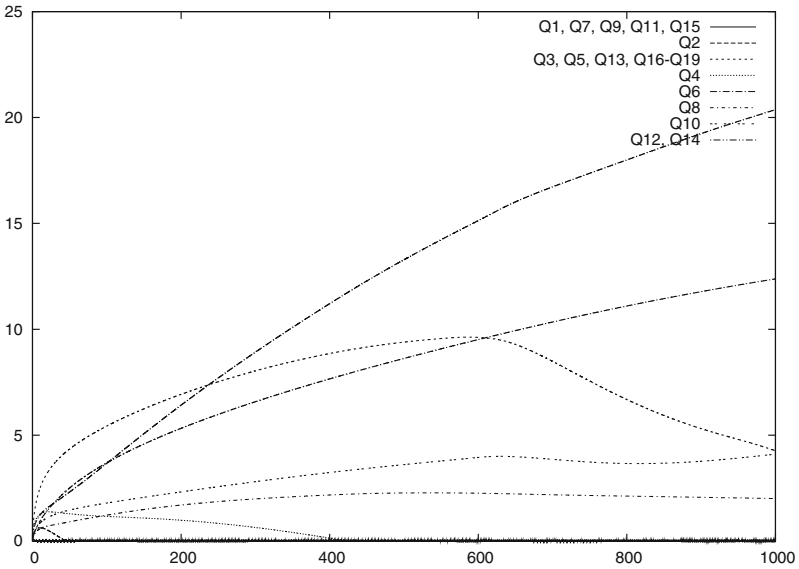
The reduction of window sizes contributed to an immediate decrease of queue length of congested nodes (Fig. 7). The queue in 17th and 18th node started to empty after 600th time unit because of exceeding the RED maximum threshold by average queue around 300th t.u. and rejecting new packets since then. However, the queues did not reach the maximum buffer sizes that were set respectively on 18 and 17 packets.



**Fig. 5.** The RTT time  $R_i$  in each flow class

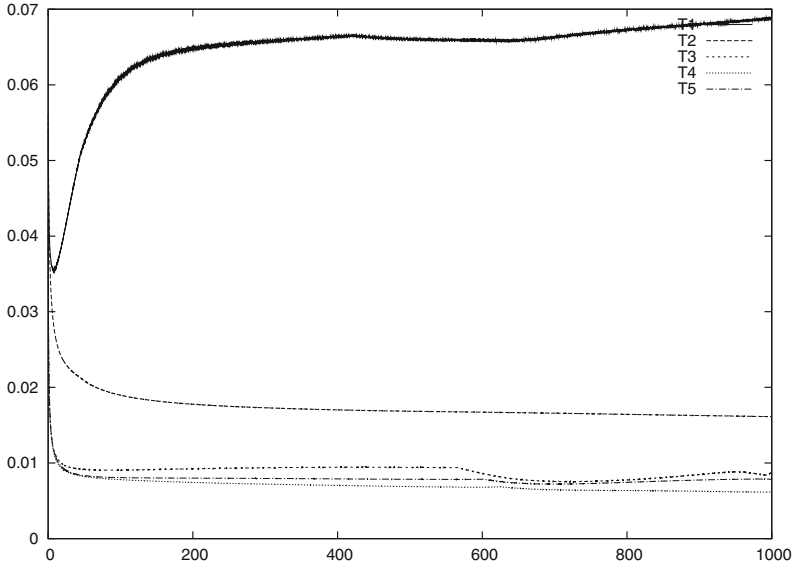


**Fig. 6.** The congestion window sizes  $W_i$  in each flow class



**Fig. 7.** The queue lengths  $Q_i$  in each node

Figure 8 displays the throughputs of flow classes. Each class has different characteristic, however the classes with a shared overloaded part of the route had similar behaviour. The flows that traversed through shared non-congested network fragments have different throughputs – as it is seen for class 1 and 2.



**Fig. 8.** The throughputs  $T_i$  in each flow class

## 5 Conclusions

We believe that diffusion and fluid approximations are useful approaches that are complementary to Markov models. The size and complexity of models which may be analysed by diffusion and fluid flow approximations are much larger than in case of traditional Markovian models. We have developed our own tools for the both methods and we are testing their possibilities. They are able to treat very large (up to millions of nodes) networks giving a software test bed to consider modifications of protocols or the choice of network topologies. The models based on Markov chains are still essential in performance evaluation and supporting the design of new communication protocols, mechanisms for regulation of the intensity of Internet transmissions and mechanisms to ensure the quality of transmission services. Their principal constraint is the number of states growing very rapidly with the complexity of an object being modelled; as each state of the Markov chain corresponds to one state of the system, it is necessary to construct and solve very large systems of equations linking the states probabilities. The existing solvers as e.g. XMARCA [28], PEPS [20], or PRISM [21] consider only steady state Markov chains.

We are developing our own Markovian package Olymp [19]. It is a library for generating transition matrices of continuous time Markov chains, and solve the resulting model. Olymp uses Java language to define network nodes and their interactions - that gives a lot of flexibility in defining a network structure and functions.

At the moment we are able to generate and solve Markov chains of the 150 million of states. The main method of solution is one of projection methods based on Krylov subspace with Arnoldi process, e.g. [23,26,28]. We plan to increase the size of tractable Markov chains by several orders through the use of a GPU-CPU (graphical processing unit) and a better design of computational algorithms for parallel computing and optimization of memory usage, [17].

An alternative to analytical models is discrete event simulation – also used here to evaluate diffusion approximation results. We have developed an extension of OMNET++ (a popular simulation tool written in C++, [18]) allowing simulation of transient state models. In this case a simulation run should be repeated a sufficient number of times (e.g. 500 thousands in our examples) and the results for a fixed time should be averaged. That makes transient simulation models time-consuming.

**Acknowledgments.** This work was supported by Polish project NCN nr 4796/B/T02/2011/40 “*Models for transmissions dynamics, congestion control and quality of service in Internet*” and the European Union from the European Social Fund (grant agreement number: UDA-POKL.04.01.01-00-106/09).

## References

1. Burke, P.J.: The output of a queueing system. *Oper. Res.* **4**(6), 699–704 (1956)
2. Carrasco, J.A.: Transient analysis of some rewarded markov models using randomization with quasistationarity detection. *IEEE Trans. Comput.* **53**(9), 1106–1120 (2004)
3. Czachórski, T.: A method to solve diffusion equation with instantaneous return processes acting as boundary conditions. *Bull. Pol. Acad. Sci. Tech. Sci.* **41**(4) (1993)
4. Czachórski, T., Grochla, K., Pekergin, F.: Stability and dynamics of TCP-NCR(DCR) protocol in presence of UDP flows. In: García-Vidal, J., Cerdà-Alabern, L. (eds.) *Euro-NGI 2007*. LNCS, vol. 4396, pp. 241–254. Springer, Heidelberg (2007)
5. Czachórski, T., Pekergin, F.: Diffusion approximation as a modelling tool. In: Kouvatsos, D.D. (ed.) *Next Generation Internet: Performance Evaluation and Applications*. LNCS, vol. 5233, pp. 447–476. Springer, Heidelberg (2011)
6. Simulating large networks using Fluid Flow Models (FFM). <http://www-net.cs.umass.edu/fluid/fm.html>
7. FSIM. <http://www.ispl.jp/oosaki/software/fsim>
8. Genin, D., Marbukh, V.: Bursty fluid approximation of TCP for modeling internet congestion at the flow level. In: *Allerton’09: Proceedings of the 47th annual Allerton Conference on Communication, Control, and Computing*, IEEE Press (2009)
9. Gelenbe, E.: On approximate computer systems models. *J. ACM* **22**(2), 261–263 (1975)
10. Gelenbe, E., Pujolle, G.: The behaviour of a single queue in a general queueing network. *Acta Inf.* **7**(2), 123–136 (1976)
11. Hollot, K., Liu, Y., Misra, V., Towsley, D., Gong, W.B.: Fluid methods for modeling large heterogeneous networks. Technical report AFRL-IF-RS-TR-2005-282 (2005)

12. Kleinrock, L.: Queueing Systems. Volume I: Theory. Wiley, New York (1975)
13. Kleinrock, L.: Queueing Systems. Volume II: Computer Applications. Wiley, New York (1976)
14. Liu, Y., Lo Presti, F., Misra, V., Gu, Y.: Fluid models and solutions for large-scale IP networks. In: ACM/SigMetrics (2003)
15. Liu, J.: Packet-level integration of fluid TCP models in real-time network simulation. In: Proceedings of the 38th Conference on Winter Simulation, Monterey, California, 03–06 December , pp. 2162 - 2169 (2006)
16. Misra, V., Gong, W., Towsley, D.: A fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. In: Proceedings of the Conference on Applications, Technologies, Architectures and Protocols for Computer Communication (SIGCOMM 2000), pp. 151–160 (2000)
17. Numerical computation for Markov chains on GPU: building chains and bounds, algorithms and applications. Project POLONIUM 2012–2013, bilateral cooperation PRISM-Université de Versailles and IITiS PAN, Polish Academy of Sciences
18. OMNET++ Community Site. <http://www.omnetpp.org>
19. Pecka, P., Deorowicz, S., Nowak, M.: Efficient representation of transition matrix in the Markov process modeling of computer networks. In: Czachórski, T., Kozielski, S., Stańczyk, U. (eds.) Man-Machine Interactions 2. AISC, vol. 103, pp. 457–464. Springer, Heidelberg (2011)
20. PEPS. <http://www-id.imag.fr/Logiciels/peps/userguide.html>
21. PRISM - probabilistic model checker. <http://www.prismmodelchecker.org/>
22. Stehfest, H.: Algorithm 368: numeric inversion of Laplace transform. *Comm. ACM* **13**(1), 47–49 (1970)
23. Saad, Y.: Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.* **29**(1), 208 (1992)
24. Sakumoto, Y., Asai, R., Ohsaki, H., Imase, M.: Design and implementation of flow-level simulator for performance evaluation of large scale networks. In: 15th Annual Meeting of the IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS) (2007)
25. Sakumoto, Y., Ohsaki, H., Imase, M.: Accelerating flow-level network simulation with low-pass filtering of fluid models. In: SAINT 2012, Izmir (2012)
26. Scientifique, C., Philippe, B., Sidje, R.B.: Transient solutions of Markov processes by Krylov subspaces. In: 2nd International Workshop on the Numerical Solution of Markov Chains (1989)
27. Srikant, R.: The Mathematics of Internet Congestion Control. Springer, Heidelberg (2004)
28. Stewart, W.: Introduction to the Numerical Solution of Markov Chains. Princeton University Press, Chichester (1994)
29. Tomasiak, J., Weisser, M.A.: Internet topology on AS-level: model, generation methods and tool. In: 29th IEEE International Performance Computing and Communications Conference IPCCC 2010 (2010)
30. Weltzl, M.: Network Congestion Control: Managing Internet Traffic. Wiley, New York (2005)
31. Gu, Y., Liu, Y., Towsley, D.: On integrating fluid models with packet simulation. In: Proceedings of IEEE INFOCOM (2004)