

# Chapter 2

## Queueing Models

**Sabine Wittevrongel**  
*Ghent University, Belgium*

### **Contributors:**

Samuli Aalto (Helsinki University of Technology, Finland), Nail Akar (Bilkent University, Turkey), Carlos Belo (Telecommunications Institute, Portugal), Hans van den Berg (TNO Telecom, Netherlands), Markus Fiedler (Blekinge Institute of Technology, Sweden), Dieter Fiems (Ghent University, Belgium), Peixia Gao (Ghent University, Belgium), Veronique Inghelbrecht (Ghent University, Belgium), Robert Janowski (Warsaw University of Technology, Poland), Udo Krieger (Otto Friedrich University Bamberg, Germany), Koenraad Laevens (Ghent University, Belgium), Tom Maertens (Ghent University, Belgium), Michel Mandjes (Center for Mathematics and Computer Science, Netherlands), Ilkka Norros (VTT Information Technology, Finland), Olav Østerbø (Telenor Research & Development, Norway), Detlef Sass (University of Stuttgart, Germany), Ana da Silva Soares (Université Libre de Bruxelles, Belgium), Kathleen Spaey (University of Antwerp, Belgium), Hung Tran (Telecommunications Research Center Vienna, Austria), Jorma Virtamo (Helsinki University of Technology, Finland), Stijn De Vuyst (Ghent University, Belgium), Joris Walraevens (Ghent University, Belgium), Sabine Wittevrongel (Ghent University, Belgium)

## **2.1 Introduction**

This chapter presents an overview of queueing models studied within COST Action 279. Such models are important tools to investigate the behavior of the buffers used in various subsystems of telecommunication networks, and hence to evaluate the quality of service, in terms of loss and delay, experienced by the users of a communication network. In Section 2.2 a number of discrete-time queueing models are discussed. Section 2.3 addresses some new developments with respect to fluid flow analysis. Work on Gaussian storages is reported in Section 2.4. In Section 2.5 some new results on processor sharing models are

presented. Section 2.6 discusses recent work on multilevel processor sharing models. Section 2.7 is devoted to the analysis of a variety of other continuous-time queueing models. Some techniques to study end-to-end delays in networks of queues are described in Section 2.8. Finally, Section 2.9 overviews some specific models and analysis techniques for performance evaluation in the context of optical networks.

## 2.2 Discrete-Time Queueing Models

In a discrete-time queueing model the time axis is assumed to be divided into fixed-length intervals, usually referred to as *slots*. This section provides an overview of a number of specific discrete-time queueing models studied within COST 279, as well as results obtained for these models. For the analysis of the models, analytical techniques mainly based on an extensive use of Probability Generating Functions (PGF) have been developed.

### 2.2.1 Queues with Static Priority Scheduling

Priority scheduling is a hot topic in multimedia networks. For real-time applications, it is important that the mean packet delay and delay jitter are not too large. Therefore, this type of traffic is given priority over non-real-time traffic in the switches/routers of the network, i.e., delay-insensitive traffic is serviced in a switch only when no delay-sensitive traffic is present.

In [105] is considered a discrete-time single-server queueing system with infinite buffer space, a Head-of-Line (HOL) priority scheduling discipline, and a general number of priority classes. All types of packet arrivals are assumed to be independent and identically distributed (iid) from slot to slot, but within one slot the numbers of packet arrivals from different classes can be correlated. The system has one server that provides the transmission of packets at a rate of one packet per slot, i.e., the service times of the packets are deterministically equal to one slot. First, an expression for the joint PGF of the system contents of all priority classes is derived. From this joint PGF, the marginal PGFs of the system contents of each priority class separately and of the total system content are found. Furthermore, the PGFs of the packet delays of each class are calculated. The analysis of the latter is largely based on the concept of *sub-busy periods*. From the generating functions obtained, performance measures, such as the moments and approximate tail probabilities of system content and packet delay, are derived. Especially the analysis of the tail behavior is an important result of [105]. It is shown that the tail behavior is not necessarily exponential. Figure 2.1 illustrates the effect of HOL priority scheduling in a  $16 \times 16$  output

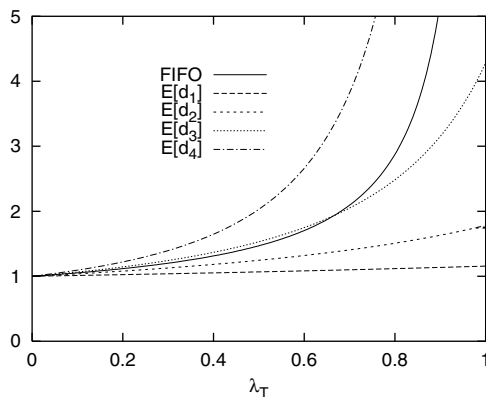


Figure 2.1: Mean delay for each priority class versus the total arrival rate  $\lambda_T$  in case of HOL priority and FIFO

queueing switch with independent and uniform routing. Packet arrivals on each inlet are independent and generated by a Bernoulli process. There are 4 priority classes and each class corresponds to a fraction of 0.25 in the overall traffic mix. The figure shows the mean packet delay in an output queue for each priority class versus the total arrival rate, together with the mean packet delay in the case of First-In-First-Out (FIFO) scheduling. For two priority classes, the analysis of [105] has also been extended to the case of general packet service times. Specifically, in [106, 107], general packet service times and a preemptive priority discipline are considered, whereas [108] considers general service times and non-preemptive priorities.

Another way to study queues with priority scheduling is discussed in [109]. In this paper, a discrete-time single-server queue subjected to *server interruptions* is investigated. Server interruptions are modeled as an on/off process with geometrically distributed on-periods and generally distributed off-periods. The messages that arrive in the system possibly require more than one slot of service time, implying that a service interruption can occur while a message is in service. Therefore, different operation modes are considered, depending on whether service of an interrupted message continues, partially restarts, or completely restarts after an interruption. For each alternative, expressions for the steady-state PGFs of the buffer contents at message departure times and at random slot boundaries, of the unfinished work at random slot boundaries, of the message delay, and of the lengths of the idle and busy periods are established. From these results, closed-form expressions for various performance measures, such as means and variances of the buffer occupancy and of the message delay, are derived. Numerical results show the deterioration of sys-

tem performance caused by service repetitions. In particular, it is observed that the mean length of the server availability periods crucially determines the system stability for the partial repeat after interruption mode.

The model considered in [109] can be used to assess the performance of a multi-class preemptive priority scheduling system. In this case, the system interrupts service of lower class messages to serve higher class messages. Assume that class  $i$  has a higher priority than class  $j$ , for  $i < j$ . Then, class 2 messages receive service during the idle periods of class 1 messages. Class 3 messages are served during the idle periods of class 2 messages (the busy periods include the interruptions), and so on. The continue after interruption (CAI) mode and the repeat after interruption (RAI) mode then correspond to preemptive resume and preemptive repeat priority scheduling, respectively. The partial repeat after interruption (p-RAI) operation mode can be considered as an intermediate case between both types of priority scheduling and allows the study of the influence of packetizing in preemptive priority systems.

### 2.2.2 Queues with Dynamic Priority Scheduling

In a static priority scheme, as discussed above, priority is *always* given to the delay-sensitive class, and thus packets of this class are always scheduled for service before delay-insensitive packets. It has been shown that this scheme provides relatively low delays for the delay-sensitive class. However, if a large portion of the network traffic consists of high-priority traffic, the performance for the low-priority traffic can be severely degraded. Specifically, HOL priority scheduling can cause excessive delays for the low-priority class, especially if the network is highly loaded. This drawback is also known as the *starvation problem*. In order to find a solution for this problem, several dynamic priority schemes have been proposed in the literature. These schemes are mostly obtained by alternately serving high-priority traffic and low-priority traffic, depending on a certain threshold, or by allowing priority jumps. In the latter type, referred to as head-of-line with priority jumps (HOL-PJ), when high- and low-priority packets arrive at the respective queues, packets of the low-priority queue can jump to the high-priority one, as illustrated in Figure 2.2. Many criteria can be used to decide if and when low-priority packets jump: a maximum queueing delay in the low-priority queue, a queue-length threshold of the low-priority queue, or a random jumping probability per slot. Further, the jumping process is also characterized by the number of packets that jump at the same time and by the specific moments when these packets jump, e.g., at the beginning of a slot, or at the end of a time slot.

In [110] and [111], a discrete-time single-server two-class queueing system

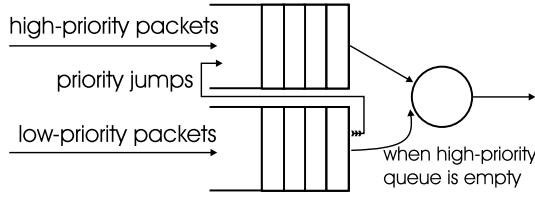


Figure 2.2: Two-class single-server queue with HOL-PJ priority scheduling

with infinite buffer size and HOL-PJ priority scheduling is considered. Two types of packets thus arrive in the system, and these two classes are assumed to arrive in separate, logical queues, i.e., a high- and a low-priority queue. The numbers of arrivals of both classes are assumed to be iid from slot to slot. However, within one slot, the numbers of arrivals from both classes can be correlated. The service times of the packets are equal to one slot. Whenever there are packets in the high-priority queue, they have service priority, and only when this queue is empty can packets in the low-priority queue be served. Within a queue, the service discipline is FIFO.

The difference between the models studied in [110] and [111] lies in the jumping process. In [110], the total content of the low-priority queue jumps with a constant probability  $\beta$  to the high-priority queue in each slot, while in [111], only the packet at the HOL-position of the low-priority queue can jump to the high-priority queue. The possible jump in [111] depends on the content of the high-priority queue, i.e., only when that queue is not empty, the packet jumps with probability one. When the high-priority queue is empty, the low-priority packet at the HOL-position is immediately served. For both models, an expression for the joint PGF of the system contents of the high- and low-priority queues is obtained. From this joint PGF, the marginal PGFs of the system contents of each queue separately and of the total system content are derived. Also, the PGFs of the packet delays of both classes are calculated. From the obtained PGFs, performance measures, such as mean values and variances, are found. Numerical results show the impact and significance of both investigated dynamic priority scheduling disciplines in an output queueing switch.

### 2.2.3 Queues with Place Reservation

In [112], a new kind of priority discipline is studied that provides a better Quality of Service (QoS) to packets that are delay-sensitive at the cost of allowing higher delays for best-effort packets. The idea, first suggested in [113], is to

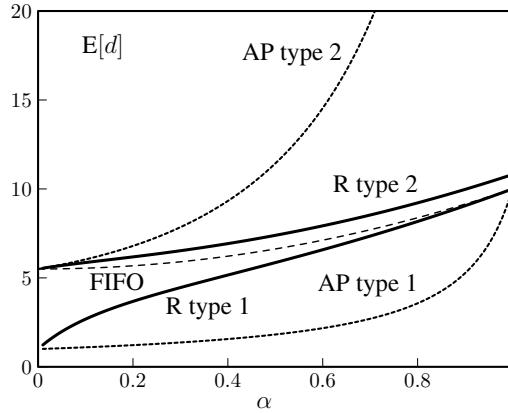


Figure 2.3: Mean delay for both types of packets versus the fraction  $\alpha$  of type 1 traffic in case of place reservation (R), absolute priority (AP) and FIFO

introduce a reserved space in the queue that acts as a place holder for future arriving high-priority packets. A discrete-time queue is considered with arrivals of type 1 (delay-sensitive) and type 2 (best-effort). Whenever a packet of type 1 enters the queue, it takes the position of the reservation that was created by a previous arrival of that type and creates a new reservation at the end of the queue. On the other hand, type 2 arrivals always take place at the end of the queue in the usual way. In this way, a packet of type 1 may jump over one or more packets of type 2 when being stored in the queue, thus lowering its delay compared to packets of type 2. In [112], the exact PGFs of the delays experienced by both types of packets are obtained and, from these PGFs, mean values, variances and tail probabilities are calculated.

Figure 2.3 illustrates the delay differentiation between the two types of packets under the considered reservation discipline. The numbers of arrivals per slot of type 1 and type 2 are independent and have a geometric and a Poisson distribution, respectively. The total arrival rate is  $\lambda_T = 0.9$ , and the mean delays for packets of type 1 and type 2 are plotted versus the fraction  $\alpha$  of type 1 packets in the overall traffic mix. The values obtained for FIFO and both types of packets under absolute priority (AP), also known as HOL priority, are shown as well. Note that, in the case of FIFO, it is considered the delay of an arbitrary packet *regardless* of its type. It is observed that under the reservation discipline, type 2 packets are less likely to experience an extremely large delay than in the case of absolute priority.

### 2.2.4 Multiserver Queues

In most of the existing literature on discrete-time multiserver queueing models, the service times of customers are assumed to be constant, equal to one slot or multiple slots. In [114], a discrete-time multiserver queue with geometric service times, an infinite buffer size, a First-Come-First-Served (FCFS) service discipline, and general independent packet arrivals is considered. The behavior of the queueing system is studied analytically by means of a generating-functions approach. This results in closed-form expressions for the PGFs of the system content and the packet delay. From these PGFs, expressions are obtained for performance measures such as the mean values, variances, and tail probabilities of the system content and the delay. In [115], the analysis is further extended from the case of an uncorrelated packet arrival process to the case of a two-state correlated arrival process. The delay analysis is based on a general relationship between system content and packet delay established in [116], valid for any discrete-time multiserver system with geometric service times, regardless of the exact nature of the arrival process.

In [117], a discrete-time multiserver queueing system with preemptive resume priority scheduling is investigated. Two classes of traffic are considered; the first class has preemptive resume priority over the second one. The service times are again assumed to be geometrically distributed, but now with a rate dependent on the traffic type. An expression for the steady-state joint PGF of the system contents of the high- and the low-priority traffic is derived. From this, closed-form expressions for the PGFs and the moments of the system contents, both for the high and low priority traffic, can be obtained. By means of Little's law, the mean delay for the two types of traffic can then also be found.

### 2.2.5 Queues with Server Vacations

Queueing systems with server vacations have proven to be a useful abstraction of systems where several classes of customers share a common resource, such as priority systems and polling systems, or of systems where this resource is unreliable, such as maintenance models and Automatic Repeat Request (ARQ) systems. A discrete-time gated vacation system is considered in [118]. The classical gated vacation system can be seen as one with two queues separated by a gate. Arrivals are routed to the queue before the gate, whereas customers in the queue after the gate are served on a FIFO basis. When there are no more customers in the latter queue, the server takes a vacation and opens the gate—all customers move instantaneously from the queue before the gate to the queue after it—upon returning from this vacation. In [118], the classical gated vacation queueing system is extended in the sense that customer arrivals are also

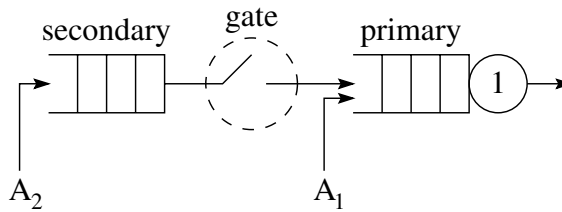


Figure 2.4: Gated-exhaustive vacation system

allowed to be immediately routed to the queue after the gate, as illustrated in Figure 2.4. The model under investigation allows to capture the performance of, among others, the exhaustive (only arrivals in primary queue) and the gated (only arrivals in secondary queue) queueing systems with multiple or single vacations. Using a generating-functions approach and the method of the supplementary variable, expressions are obtained for performance measures such as the moments of the system content at various epochs in equilibrium and of the customer delay. The results depend on a constant that has to be determined numerically.

### 2.2.6 Queues in ARQ Systems

ARQ protocols are used to obtain reliable transfer of packets from a sender to a receiver communicating over an unreliable channel, where packet corruption or loss is possible. In [119], an analytical approach for studying the queue length and the packet delay in the transmitter buffer of a system working under a stop-and-wait retransmission protocol is presented. The operation of the stop-and-wait ARQ protocol is illustrated in Figure 2.5. The transmitter sends a packet available in its queue and then waits for  $s$  slots (the feedback delay) until it receives the corresponding feedback message. If the packet was transmitted correctly (positive acknowledgement or ACK), the next packet waiting in the queue is transmitted. Otherwise, if an error occurred (negative acknowledgement or NACK), the packet is retransmitted. The buffer at the transmitter side is modeled as a discrete-time queue with an infinite storage capacity. The numbers of packets entering the buffer during consecutive slots are assumed to be iid random variables. The information packets are sent through an unreliable and non-stationary channel, modeled by means of a two-state Markov chain. An explicit formula is derived for the PGF of the system content. This PGF is then used to derive several queue-length characteristics as well as the mean packet delay.



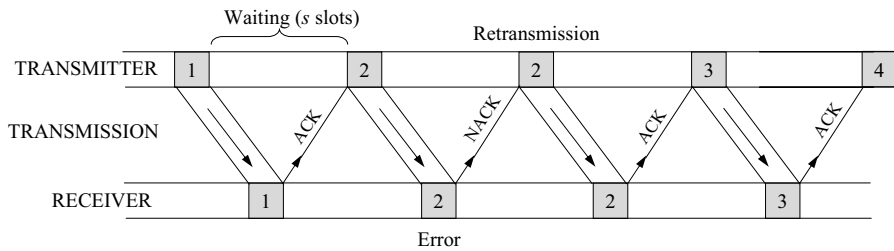


Figure 2.5: Stop-and-wait ARQ protocol

The model was studied further in [120], where not only the mean value but the whole distribution of the packet delay is obtained, as well as an expression for the maximum throughput of the system. For both, the analysis is based on the *conditional effective service times*, i.e., the time that elapses from a packet's first transmission until the transmitter is notified of the correct receipt of that packet.

### 2.2.7 Queues with Bursty Traffic

Two discrete-time queueing models are compared in [121]: a *packet model*, where two timescales, a burst and a packet timescale, are present in the input traffic, and a *fluid model*, where only an (identical) burst timescale is present. The time axis is assumed to be divided in fixed-length time units, called frame times, and every frame time is further divided into fixed-length units, called packet times. A two-state discrete-time Markov source is considered, which has a frame time as underlying time unit and thus can only change state at frame time boundaries. During a frame time in which the source is in the first, respectively second state, it generates a certain amount of bytes. In the packet model, these bytes are divided into fixed-length packets that are sent all together in the beginning of the frame time. In the fluid model the bytes are sent at a constant rate over the whole duration of the frame time. A queueing model with an infinite buffer capacity is then considered in which the traffic of either  $M$  identical packet sources, or  $M$  identical fluid sources with the same parameters, is multiplexed. For both systems, the distribution of the amount of unfinished work is derived, and the impact of approximating packetized flows by fluid flows on the complementary cumulative distribution (CCD) of the unfinished work is investigated. The main conclusions are described in Section 3.4.5.

## 2.3 Fluid Flow Models

In a fluid flow model (FFM) the amount of work delivered to a queue or processed by a server is modeled as a continuous-time flow. A fluid queue is generally solved by first finding the eigenvalues and eigenvectors of the underlying differential system and then obtaining the coefficients of the associated spectral expansion by solving a linear matrix equation. This section presents some new COST 279 developments with respect to fluid flow analysis.

### 2.3.1 Algorithmic Approach

Consider a Markov modulated fluid queue, i.e., a two-dimensional continuous-time Markov process  $\{(X(t), \varphi(t)) : t \in \mathbb{R}^+\}$  where  $X(t)$  takes values in  $\mathbb{R}^+$  and  $\varphi(t)$  in  $\mathcal{S}$ , a finite set. The component  $X(\cdot)$  is called the *level* and  $\varphi(\cdot)$  is called the *phase*. The level is subordinated to the phase in the following way. The phase process  $\{\varphi(t) : t \in \mathbb{R}^+\}$  is an irreducible Markovian process. During those intervals of time in which the phase is constant, say equal to  $i$ , the level increases or decreases at a constant rate dependent on  $i$ , or it remains constant. If  $X(t) = 0$  and if the rate at time  $t$  is negative, then the level remains at 0.

Ramaswami [122] shows that  $\{X(t)\}$  has a phase-type stationary distribution using the dual process of  $\{(X(t), \varphi(t))\}$ . Most importantly, he also constructs a very efficient computational procedure, based on the logarithmic-reduction algorithm of Latouche and Ramaswami [123] for discrete-level Quasi-Birth-Death (QBD) processes: he thereby reduces a complex continuous time, continuous state space problem to a familiar, simple discrete time, discrete state space system. The use of the dual process in [122] is motivated by a property that relates the stationary distribution of the original  $\{(X(t), \varphi(t))\}$  process to first passage probabilities at level 0 in the dual process. In [124], the similarities with QBDs are reinforced by showing that one may actually *directly* use these first passage probabilities in the original process. Also, another probabilistic interpretation of Ramaswami's computational procedure is given.

### 2.3.2 Large-Scale Finite Fluid Queues

Except for some structured models, e.g., the Anick-Mitra-Sondhi (AMS) fluid flow model [125], it is in general hard to find the eigensystem in a computationally efficient and stable manner. Moreover, the linear matrix equation to solve for the coefficients in the finite fluid queue case is known to be ill-conditioned, especially in the case of large buffer sizes.

In [126], a numerically efficient and stable method for solving large-scale finite Markov fluid queues is developed. No special structure is imposed on the underlying continuous-time Markov chain, i.e., the eigenvalues and eigenvectors need not be determined in closed form. The authors propose an alternative method that relies on decomposing the differential system into its stable (forward) and anti-stable (backward) subsystems, as opposed to finding eigenvalues, using a method based on the additive decomposition of a matrix pair with respect to the imaginary axis. There are a variety of numerical linear algebra techniques, with publicly available codes, that can be used for such an additive decomposition, including the generalized Newton iterations, the generalized Schur decomposition, and the spectral divide and conquer methods. Using the generalized Newton iterations, which have quadratic convergence rates, it is shown in [126] that the accuracy of the proposed method does not depend on the buffer sizes and that in the limit the finite fluid queue solution converges to that of the infinite fluid queue. Moreover, it is demonstrated that fluid queues with thousands of states are efficiently solvable using the proposed algorithm.

### 2.3.3 Voice and Multi-Fractal Data Traffic

The FFM has shown being able to incorporate many types of traffic, i.e., superpose them for analysis in a unified model. In [127], it is augmented by a model displaying multi-fractal behavior, described in Section 3.4.6. This model can be matched to the characteristics of real traffic by choosing the appropriate parameters for the sub-processes. The paper investigates the interaction between multi-fractal data traffic and voice traffic with suppressed silence phases and consideration of the on-hook-state of the Internet phone, the model for the latter being a 3-state ON-OFF model. The fluid flow calculations can be used in a straightforward manner, with the only exception that the pseudo-rates of the sub-processes contributing to the data traffic process are multiplied instead of added. As a result, formulas expressing queuing delay and loss as experienced individually by voice and data are obtained. A case study is carried out, investigating the maximal load under given delay quantiles. As expected, this load level depends heavily on the variance of the data traffic. In general, the voice traffic yields better performance in terms of loss and delay and helps to increase the maximal load, while still meeting the target performance values.

### 2.3.4 Superposition of General ON-OFF Sources

The effect of a superposition of general ON-OFF sources on a multiplexer is studied in [128]. Sources are statistically identical and independent. During the ON period a source emits at a constant rate, either in a fluid flow fashion, or

by periodically emitting fixed size packets. During the OFF period the source remains silent. Both the ON and OFF periods are random variables with general distributions but finite mean values. The distributions considered include the Pareto type, known to lead to traffic having the long range dependence (LRD) property.

The superposition of a number of such general ON-OFF sources results in a stochastic process called semi-birth-and-death (semi-BD). The state of the semi-BD process is the number of active sources at a given time; the random variable of interest is the holding time in that state. In this case, the traffic generated simply equals the number of active sources multiplied by the individual rate. For the case of the semi-BD, the stationary distribution of the holding time is given in [128] in explicit form as a function of the state considered, the number of sources, and the distribution of the ON and OFF periods. It is further argued that, for the semi-BD, the distribution of the holding time would tend to an exponential even with a moderate number of sources. The above result strongly suggests that the now classical AMS solution for the probability of buffer overflow in an infinite buffer multiplexer with a superposition of exponential ON-OFF sources as input could be applied to the case of general ON-OFF distributions. The remaining of [128] is devoted to evaluating, theoretically and by way of simulation, the circumstances under which the AMS solution is a good approximation as a function of the number of sources, the distributions of the ON and OFF periods, and the desired level of probability of overflow.

### 2.3.5 Feedback Fluid Queues

In [129] is considered a single point in an access network where several bursty users are multiplexed. The users adapt their sending rates based on feedback from the access multiplexer. Important parameters are the user's peak transmission rate  $p$ , which is the access line speed, the user's guaranteed minimum rate  $r$ , and the bound  $\epsilon$  on the fraction of lost data. Two feedback schemes are proposed and studied. In both schemes the users are allowed to send at rate  $p$  if the system is relatively lightly loaded, at rate  $r$  during periods of congestion, and at a rate between  $r$  and  $p$ , in an intermediate region. For both feedback schemes an exact analysis is presented, under the assumption that the users' file sizes and think times have exponential distributions. The techniques are used to design the schemes jointly with admission control, i.e., the selection of the number of admissible users, to maximize throughput for given  $p$ ,  $r$ , and  $\epsilon$ . Next is considered the case where the number of users is large. Under a specific (many-sources) scaling, explicit large deviations asymptotics for both models

are derived. The extension to general distributions of user data and think times is discussed.

The model is also extended to a “buffer-dependent” Markov fluid queue, defined as follows. A Markov fluid source is defined as a continuous-time Markov chain with transition rate matrix  $Q$  of dimension  $d$ , and a traffic rate vector  $r$ , describing the generation of traffic at a constant fluid rate  $r(i)$  when the Markov chain is in state  $i$ , for  $i = 1, \dots, d$ . Now consider  $N$  of these Markov fluid sources of dimension  $d$ , characterized by the pairs  $(Q_n, r_n)$ , for  $n = 1, \dots, N$ , and suppose traffic is generated according to the  $n$ th Markov fluid source when the buffer level is between  $B_{n-1}$  and  $B_n$ , where the  $B_n$  are increasing, with  $B_0 = 0$  and  $B_N$  finite or infinite. For this model, the complete buffer content distribution is derived in terms of the solution of an eigensystem.

### 2.3.6 Fair Queueing Systems

In [130], an FFM for a fair queueing system with unidirectional coupling for several different classes is considered, where each class has a predefined minimum bandwidth guaranteed. These minimum bandwidths form a decomposition of the total bandwidth and avoid starvation of a class caused by another class. A multiplexing gain is achieved by passing down the residual bandwidth of a class to the lower adjacent class. Therefore, this system is called *unidirectionally coupled*. The case of a unidirectionally coupled fair queueing system consisting of two classes is shown in Figure 2.6. Also, each class has its own FIFO buffer for exclusive usage. The system is described by an FFM, and hence sources and server are assumed to be Markov modulated fluid processes (MMFP). The key observation concerning the residual bandwidth is that a class lending its residual bandwidth is oblivious to this. Also receiving residual bandwidth is, from the receiving class point of view, just an additional server process. This process is interpreted to be, again, an MMFP, and certain states and the buffer’s mean busy period of the lending adjacent class are used to model this additional process. The states mentioned are the underload states, because state transition among these states reflects—assuming the buffer to be empty—the dynamics of the residual bandwidth. The transition matrix of the additional server process can be built with it. The distribution of the buffer content is found by applying the FFM. To determine the distribution for a specific buffer, the distribution of the previous adjacent buffer has to be calculated, apart from that of the first buffer. Also, two estimates for the overflow probability of a buffer are obtained. First, a more accurate estimate is calculated by applying the full FFM, i.e., all eigenvalues and eigenvectors are used. Second, a fast and robust estimate is found by using only the domi-

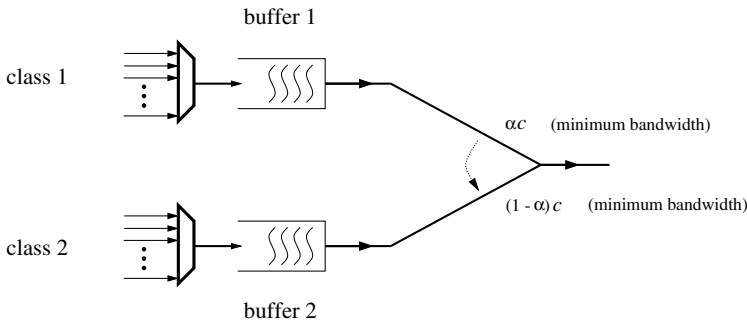


Figure 2.6: Example of a unidirectionally coupled fair queueing system

nant eigenvalue and the Chernov large deviation bound. This is a conservative estimate with larger deviation than the other estimate.

### 2.3.7 Bottleneck Identification and Classification

The stochastic FFM [125] has also shown to be capable of revealing the impact of bottlenecks, i.e., shortages in capacity, on packet streams by comparing bit rate histograms at the output with those at the input of the bottleneck [131]. From standard fluid flow analysis for Markov Modulated Rate Processes (MMRP), the joint probabilities that the buffer is empty, or non-empty, in each state are calculated. As shown in [131], these probabilities are the key for obtaining the output bit rate distribution both for individual and total traffic streams. From comparisons with input bit rate distributions, it can be seen whether there is interfering traffic in the bottleneck or whether the bottleneck has a buffer of significant size. This allows not only for an identification, but also for a classification of the bottleneck. While the maximal capacity of the bottleneck is revealed in the output bit rate histogram of the total stream, it may under certain conditions also become visible in the corresponding histogram of an individual stream.

## 2.4 Gaussian Storages

This section overviews some new developments with respect to the most probable path technique to derive estimates of the queueing performance for queues with Gaussian input traffic. Also, a method to determine delay quantiles of a multiplexer with Gaussian input, involving a fitting procedure to Ornstein-Uhlenbeck processes, is discussed.

### 2.4.1 Most Probable Path Technique

By the theory of large deviations of Gaussian processes, the probability that a simple queue with Gaussian input exceeds a level  $x$  can be approximated by

$$\Pr[Q \geq x] \approx \exp\left(-\frac{1}{2}\|f_x\|_R^2\right), \quad (2.1)$$

where  $f_x$  is the path of the input process that creates a queue of size  $x$  at time 0 and has the smallest reproducing kernel Hilbert space (RKHS) norm  $\|\cdot\|_R^2$  among all paths creating such a queue.

The framework was generalized in a straightforward way to a two-class priority system in [132] as follows. Assume that the two arrival processes are independent continuous Gaussian processes with stationary increments. Consider the most probable *path pair* that creates a total queue (both classes together) of size  $x$ . If this path of the higher priority input does not create a positive queue at time 0, then this path pair is also the most probable one to create a lower class queue of size  $x$ .

The remaining case is studied in [133]. The idea is to compute an easily characterized heuristic approximation to the most probable path pair, where the higher priority traffic essentially fills the link while the lower class traffic is accumulating in the queue. The same principles can be applied to a generalized processor sharing (GPS) system with two classes by replacing link filling by filling the quota guaranteed for a traffic class. Simulations show that these approximations are sufficiently accurate for many practical purposes, such as studying the effects of setting GPS weights.

In [134], another kind of modification of the basic Gaussian queue is studied. The service capacity is not any more constant, but continuously varied according to the traffic rate observed, with a constant delay  $\Delta$ . The allocated capacity is  $1+\epsilon$  times the observed rate. That is, the cumulative service process is defined as

$$C_t = (1 + \epsilon)(A_{t-\Delta} - A_{-\Delta}), \quad (2.2)$$

where  $A$  is the cumulative input process. The queue length process is defined as a supremum of the net input process:

$$Q_t = \sup_{s \leq t} ((A_t - A_s) - (C_t - C_s)). \quad (2.3)$$

Since the net input process is Gaussian, the basic estimates of the queue length distribution and of the most probable paths are directly available.

Figure 2.7 shows, in case of fractional Brownian motion (fBm) input, the most probable path that creates a queue of size 4 at time 0. Note how the input

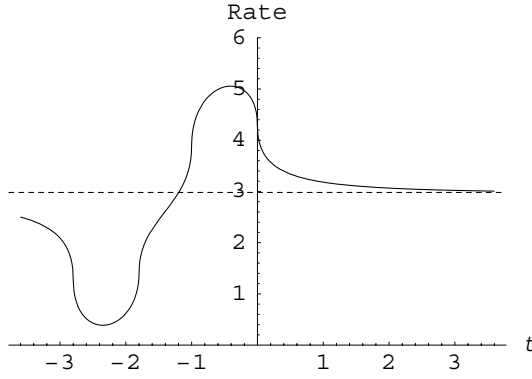


Figure 2.7: The input rate of the most probable way to obtain a queue of size 4, when service capacity is varying according to traffic prediction. The input process is a fBm with Hurst parameter 0.75

process “fools” the prediction by making the input first very slow and then, when the control cannot react any more, suddenly speeding up.

In [135] is made a substantial contribution to the most probable path approach described above by the establishment of a technique for identifying most probable paths that are truly infinite-dimensional combinations of covariance functions. This paper focuses on the simplest of this kind of problems. Consider a simple Gaussian queue with centered input process  $Z$  and unit service rate. What path  $\beta^*$  is the most probable one, in the sense of smallest RKHS norm, among input paths  $f$  that produce a busy period starting at 0 and straddling the interval  $[0, 1]$ , that is, satisfying  $f(t) \geq t$  for all  $t \in [0, 1]$ ? This event is an intersection of (infinitely many!) half-spaces and thus convex. Assume that it is also non-empty, very mild conditions being sufficient for this. Then,  $\beta^*$  exists and is unique. Denote  $S^* = \{s \in [0, 1] \mid \beta^*(s) = s\}$ . The crucial observation is that

$$\beta^* \in \bigcap_{\epsilon > 0} \overline{\text{sp}} \{Z_u \mid u \in [0, 1], d(u, S^*) < \epsilon\}, \quad (2.4)$$

where  $d(u, S^*) = \inf \{|u - s| \mid s \in S^*\}$  and  $\overline{\text{sp}}$  denotes the closure of linear span. This result makes it often possible to identify a most probable path numerically and in some cases even analytically. When the process  $Z$  is non-smooth, for example fBm, we have  $\beta^* \in \overline{\text{sp}} \{Z_u \mid u \in S^*\}$ , and  $\beta^*(t) = \mathbb{E}[Z_t \mid Z_s = s \forall s \in S^*]$ . For Brownian motion, it is well known that  $S^* = \{1\}$ . In the case of fBm, it turns out that  $S^*$  has the form  $[0, s^*] \cup \{1\}$  (resp.  $[s^*, 1]$ ) if the self-similarity parameter  $H$  is larger (resp. smaller) than  $1/2$ .



When  $Z$  is smooth, a characterization on the path usually contains values of the derivative of the path at some boundary points of  $S^*$ .

In the slightly general form of events  $\{Z_s \geq \zeta(s) \forall s \in S\}$ , where  $\zeta \in R$  is any function belonging to the RKHS, the results of [135] on infinite intersections make it possible to determine the most probable paths to high buffer levels in priority queues exactly, instead of the heuristic approximations used in [133]. (Since the differences are numerically negligible, this is also an argument in favor of these heuristics.) However, the simple definition of the lower priority queue given above must be changed to the following. Let the service rate be  $c$ , and define first the cumulative capacity available for the lower priority traffic as  $C_t^2 = \sup_{s \leq t} (cs - A_s^1)$ , and then the lower priority queue as

$$Q_t^2 = \sup_{s \leq t} (A_t^2 - A_s^2 - (C_t^2 - C_s^2)) \quad (2.5)$$

( $A^i$  denotes the cumulative input of class  $i$ ). This definition agrees with the old one when the input processes are non-decreasing, but it is better in the Gaussian case since it makes the queue length process non-negative. The event of a large lower priority queue can be written as

$$\{Q_0^2 \geq x\} = \bigcup_{s \leq 0} \bigcup_{t \leq s} \bigcup_{a \geq x} (B_{s,t,a}^1 \cap B_{s,a}^2), \quad (2.6)$$

where  $B_{s,t,a}^1 = \{A_u^1 - A_t^1 \geq c(u-t) + x - a \forall u \in [s, 0]\}$ , which is an infinite intersection event depending on  $A^1$  only, and  $B_{s,a}^2 = \{-A_s^2 = a\}$ , which is a one-dimensional condition on  $A^2$  only. This determines the possible shapes of the most probable path pairs, and it remains to optimize over the quantities  $s$ ,  $t$  and  $a$ .

Delays can be analyzed with the same technique as queue lengths. The natural definition of the queueing delay in the lower priority class is

$$V_t^2 \doteq \inf \{u \geq 0 \mid Q_t^1 + Q_t^2 + (A_{t+u}^1 - A_t^1) < cu\}. \quad (2.7)$$

The structure “union of infinite intersections” is obtained once again:

$$\begin{aligned} \{V_0^2 \geq t\} = & \bigcup_{v \leq s \leq 0} \bigcup_{a \in \mathbb{R}} \left( \bigcap_{u \in [0, t]} \{A^1(v, u) \geq -a + c(u-v)\} \right. \\ & \left. \cap \{A^2(s, 0) = a\} \right). \end{aligned}$$

Finally, a tandem queue allows a very similar analysis, see [136].

### 2.4.2 Delay Quantiles

Delay quantiles for the Gaussian voice traffic model are derived in [137]. Delay quantiles  $\gamma_k$  with  $\Pr[\text{delay} > \gamma_k] = 10^{-k}$  play an important role when it comes to dimensioning and performance evaluation. Especially large links should be dimensioned such that delay quantiles remain relatively small. However, as related methods often suffer from time and memory limitations together with numerical instabilities, it is not necessarily simple to obtain numerical results for systems incorporating many streams. The superposition of a large number of homogeneous Markovian ON-OFF sources asymptotically approaches an Ornstein-Uhlenbeck Process (OUP) representing a Gaussian process with exponential autocorrelation function. For such an OUP/D/1 model, a closed-form expression is derived in [137] for delay quantiles as

$$\gamma_k \simeq \frac{\sigma_R}{\omega_R C m} \left( k \log(10) - \frac{7}{4}m - \left( \frac{k}{30} + \frac{1}{4} \right) m^2 \right), \quad (2.8)$$

where  $m = \frac{C - \mu_R}{\sigma_R}$ . The parameter  $\mu_R$  represents the mean rate,  $\sigma_R$  is the corresponding standard deviation,  $\omega_R$  denotes the reciprocal time constant of the autocorrelation function of the rate, and  $C$  stands for the capacity of the link. By comparing, in the sense of the model, with exact results for a non-finite number of fluid flow on-off sources, we find that for the relevant parameter region defined by  $\Pr[\text{source on}] \simeq 0.4$ ,  $k \in [3, 6]$ ,  $m \in [0, 1.6]$  and the number of sources  $N > 100$ , the related deviations of the approximated delay quantiles do not exceed 10%, which makes (2.8) a well-working approximation formula.

## 2.5 Processor Sharing Models

Processor Sharing (PS) models are widely applicable to situations in which different users receive a share of a scarce common system resource. In particular, over the past few decades PS models have found many applications in the field of the performance evaluation of computer-communication systems. The standard PS model consists of a single server assigning each customer a fraction  $1/n$  of the service rate when there are  $n$  customers in the system. Cohen [48] generalizes the PS model to the so-called GPS model, where each customer receives a fraction  $f(n)$  of the service speed when there are  $n$  customers at a node, where  $f(\cdot)$  is, except for weak assumptions, an arbitrary function. The GPS model significantly enhances the modelling capabilities of the PS model. Interestingly, over the past few years the GPS model studied by Cohen in [48] in 1979 has received a renewed interest in the literature on performance of computer-communication networks (see, e.g., [138, 139, 140]).

A particularly attractive feature of (G)PS models is that in many applications they cover the main factors determining performance, and on the other hand, are still simple enough to be analytically tractable (see, e.g., the analysis in [141, 142, 48]). The present section gives a comprehensive overview of recent work on PS models within COST Action 279. Aside from the more generic analyses included in this section, we refer to [143, 144, 145, 146], discussed in chapter 4 on wireless networks, for specific applications of different PS models to analyze the performance of different mobile access technologies.

### 2.5.1 Sojourn Times for PS Models with Multiple Servers and Priority Queueing

In [61] is studied a PS model with multiple servers and two priority classes (without loss of generality). When the number of high-priority customers does not exceed the number of servers,  $C$ , each high-priority customer occupies a single server and is served at unit rate. When the number of high-priority customers is larger than  $C$ , the system switches to a processor sharing mode and the total service capacity  $C$  is equally shared among the high-priority customers. The service process of low-priority customers proceeds in a similar way, but with two specific restrictions: (i) high-priority customers have strict priority, in a preemptive resume fashion, over low-priority customers, and (ii) at any moment in time the only servers available to low-priority customers are those servers that are not used by high-priority customers at that moment.

In [61], the mean sojourn times in the multiserver queueing model with PS service discipline and two priority classes described above are studied. For the high-priority class, closed-form expressions for the mean sojourn times are presented in a general parameter setting, based on known results for the GPS model (see [48]). For low-priority customers, closed-form expressions are derived for several special cases: the single-server case where the service times of the low-priority customers are exponentially distributed, and the multiple-server case with exponential service times with the same means. In all other cases, exact explicit expressions for the mean sojourn times of the low-priority customers cannot be obtained. Therefore, a simple and explicit approximation is proposed and tested. Numerical results demonstrate that the approximation is accurate for a broad range of parameter settings. As a by-product, it is observed that the mean sojourn times of the low-priority customers tend to decrease when the variability of the service times of the low-priority customers increases.

Application of these results to the setting with elastic Transmission Control

Protocol (TCP) traffic is given in [59], discussed in Section 1.5.3 of the chapter on IP-based networks of this report.

## 2.5.2 Throughput Measures for PS Models

In [147] are specified, derived, and compared a set of throughput measures in PS queueing systems modeling a network link carrying elastic TCP data calls, e.g., file downloads. The available service capacity is either fixed, corresponding to a stand-alone dedicated General Packet Radio Service (GPRS) network, or randomly varying, corresponding to an integrated services network where the elastic calls utilize the capacity left idle by prioritized stream traffic such as speech.

While from the customer's perspective the *call-average throughput* is the most relevant throughput measure, in PS systems this quantity may be hard to determine analytically, and this is an important reason to assess the closeness of a number of other throughput measures. Alternatives applied to approximate the call-average throughput are the *time-average throughput* [148, 149, 150], defined as the expected throughput the "server" provides to an elastic call at an arbitrary (non-idle) time instant, and the *ratio* of the expected transfer volume and the expected sojourn time [58, 151, 152, 62].

In [147] is introduced a new throughput measure that can be analyzed relatively easily, the *expected instantaneous throughput*, i.e., the throughput an admitted call experiences immediately upon admission to the system. The experiments demonstrate that the newly proposed expected instantaneous throughput measure is the *only* one among these throughput measures that excellently approximates the call-average throughput for each of the investigated PS models over the entire range of elastic traffic loads. In particular for the model integrating speech and data traffic, the other throughput measures, such as the time-average throughput or the ratio of the expected call size and the expected sojourn time, significantly underestimate the call-average throughput. An intuitive reasoning for the generally near-perfect fit of the expected instantaneous throughput is that, apparently, the throughput an elastic call experiences immediately upon arrival is an excellent predictor of what the call is likely to experience throughout its lifetime. Moreover, among the considered throughput measures, the expected instantaneous throughput is the *only* approximate measure that is truly *call-centric*.

The numerical experiments further reveal that the expected call-average throughput of elastic calls in the considered PS models is to a considerable degree *insensitive* to both the variability of the available capacity and the call duration distribution. This insensitivity does not hold if the data performance is measured by the expected sojourn time.

### 2.5.3 PS Models with State Dependent Blocking Probability and Capacity

In [56], PS models with state dependent blocking probability and capacity are investigated. It is shown that if the blocking probability and capacity of a PS system are functions of the system state, the state probabilities are insensitive to the detailed distributions of thinking time and file size. This insensitivity property allows the evaluation of performance through the offered traffic only. From here, formulas to calculate state probabilities, blocking probabilities, and the conditional mean sojourn times of users are derived by using the so-called A-formula. By proper transformation of the state probabilities to GPS rates it is possible to calculate the resources obtained by every source via the use of the convolution algorithm. This technique allows the evaluation of the performance of sources having individual bandwidth (multi-rate traffic) under the GPS strategy.

## 2.6 Multilevel Processor Sharing Models

Multilevel Processor Sharing (MLPS) scheduling disciplines, introduced in [49] permit to model a wide variety of non-anticipating scheduling disciplines. A discipline is *non-anticipating* when the scheduler does not know the remaining service time of jobs. Such disciplines have recently attracted attention in the context of the Internet as an appropriate flow-level model for the bandwidth sharing obtained when priority is given to short TCP connections [153, 154, 155, 156].

An MLPS scheduling discipline is defined by a finite set of level thresholds  $a_1 < \dots < a_N$ . A job belongs to level  $n$  if its attained service is at least  $a_{n-1}$  but less than  $a_n$ . Between these levels, a strict priority discipline is applied, with the lowest level having the highest priority. Within each level  $n$ , an internal discipline is applied, belonging to the set  $\{\text{FB}, \text{PS}, \text{FCFS}\}$ . The foreground-background (FB) discipline is also known as least-attained-service (LAS), giving priority to the job with the least attained service.

In [157] and [158], the mean delay of those MLPS disciplines whose internal disciplines belong to the set  $\{\text{FB}, \text{PS}\}$  is compared to that of the ordinary PS discipline. In [157], it is proved that such MLPS disciplines with just *two* levels are better than PS with respect to the mean delay, whenever the service time distribution is of type Decreasing Hazard Rate (DHR), e.g., hyperexponential. In [158], a similar result is derived for such MLPS disciplines with *any* number of levels.

In [159], the mean delay is compared among all MLPS disciplines. The

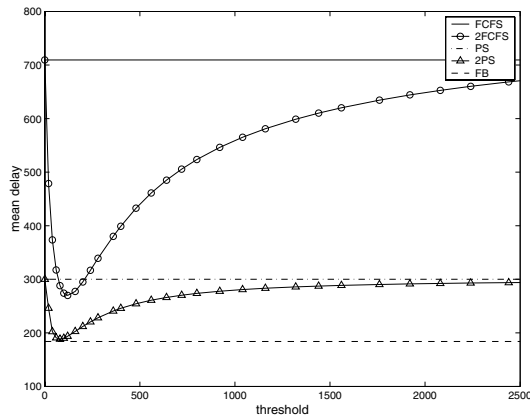


Figure 2.8: Mean delay as a function of the level threshold  $a$  for disciplines  $2FCFS(a)$  and  $2PS(a)$ . The three horizontal lines correspond to the mean delay of disciplines FCFS, PS, and FB

main result states that given an MLPS discipline, the mean delay is reduced, under the DHR condition, if a level is added by splitting an existing one. Furthermore, it is shown that given an MLPS discipline, the mean delay is reduced, under the DHR condition, if an internal discipline is changed from FCFS to PS, or from PS to FB. These two results define a natural partial order among the MLPS disciplines: if one MLPS discipline is derived from another by splitting levels and/or changing internal disciplines from FCFS to PS, or from PS to FB, then the mean delay is reduced under the DHR condition. This reduction is illustrated in Figure 2.8, where the mean delay is depicted as a function of the threshold  $a$  for two-level disciplines  $2FCFS(a)$  and  $2PS(a)$  and a Pareto service time distribution. As stated, the mean delay for any  $2PS(a)$  discipline is less than that of the corresponding  $2FCFS(a)$  discipline or the ordinary PS discipline. The mean delay of an MLPS discipline with just a few levels gets close to the minimum feasible delay, achieved by FB. Since flow size distributions in the Internet typically satisfy the DHR condition, these results are interesting in view of recent work that proposes to provide differential treatment to flows on the Internet based in just two classes: mice and elephants.

## 2.7 Other Continuous-Time Queueing Models

This section discusses a variety of continuous-time queueing models and associated analysis techniques studied in COST Action 279 .

### 2.7.1 Instantaneous and Averaged Queue Length in an M/M/1/K Queue

In [160] is studied the dynamics of the joint process of the instantaneous queue length  $L(t)$  of an M/M/1/K system together with the exponentially averaged queue length  $S(t) = \int_0^\infty L(t-u)\alpha e^{-\alpha u} du$ , where  $\alpha$  is a weighing parameter. The arrival and service rates are denoted by  $\lambda$  and  $\mu$ . The state of the system is specified by the pair  $(L(t), S(t))$ , i.e., one discrete and one continuous variable. The setting is very similar to that of a fluid queue driven by an MMRP.

The evolution of the joint distribution of the state variables is governed by a system of coupled ordinary differential equations (ODE) for the partial cumulative distribution functions  $F_i(t, x) = \Pr[L(t) = i, S(t) \leq x]$ ,

$$\begin{aligned} \frac{\partial}{\partial t} F_i(t, x) - \alpha(x-i) \frac{\partial}{\partial x} F_i(t, x) &= (\lambda F_{i-1}(t, x) - \mu F_i(t, x)) 1_{i>0} \\ &+ (\mu F_{i+1}(t, x) - \lambda F_i(t, x)) 1_{i<K}, \quad i = 0, \dots, K. \end{aligned} \quad (2.9)$$

An analytical stationary solution to these equations is found in a few special cases. A general stationary solution is not known and is believed not to have a simple form. Therefore, different alternative approximate ways for obtaining the stationary distribution are developed in [160].

Two of the methods consider the temporal behavior of the state distribution. By Kolmogorov's theorem, starting from any initial distribution the system will eventually approach an equilibrium, i.e., integrating the equations in time is inherently stable. The first of the methods considers the evolution of the system in continuous time, whereas in the second approach an embedded system in discrete time is studied. A disadvantage of these methods is that the equilibrium is only approached asymptotically, and with a long averaging time the convergence is slow. The third method focuses directly on the equilibrium distribution but using an approximation. The method applies the stochastic discretization approach, where the deterministic evolution of the continuous variable is replaced by small stochastic transitions, thus allowing the use of standard methods of Markovian systems.

### 2.7.2 M/D/1/K Vacation Queue

In [16], a queueing model is adopted where voice packets are fed into a finite buffer and served by a server representing the output link. The aggregate voice traffic is modelled by a Poisson process. Due to the presence of best-effort traffic and to the DiffServ-compliant non-preemptive priority scheduling, the operation of the server is considered in an exhaustive service and multiple

vacation scenario. That is, the server serves voice packets until the buffer becomes empty. At the finishing instant of the service, if the server finds the queue empty, it takes a vacation. If there are still no voice packets in the queue when the server returns from its vacation, it takes another vacation, and so on. The vacations of the server correspond to the situation where the output link is occupied by the best-effort traffic. The assumption of multiple vacations implies that the offered load of the best-effort traffic is sufficiently high to utilize immediately the link capacity whenever no voice packet is present. The vacation time is assumed to be the time needed for transmission of a best-effort packet with maximum transmission unit (MTU) size. In effect, a finite  $M/D/1/K$  queue with exhaustive service and multiple server vacations is obtained.

The steady-state solution of this queueing model is obtained, and useful quantities concerning packet loss probability and arbitrary percentile of delay are derived. The latter quantity is particularly valuable, because it stands for a statistical upper bound on the jitter, which is the main factor leveraging the perceived quality of voice connections.

### 2.7.3 BMAP/G/1 Queue with Feedback

Message transmission in wireless communication networks includes procedures for error correction at several layers of the protocol stack, e.g., at the data link layer (DLL) and transport layer. These procedures perform the repeated transmission of those protocol data units (PDU) that were transmitted with errors. An adequate performance model of such procedures is determined by specific feedback queues.

In [161], matrix-geometric modelling techniques are used to analyze and calculate the performance characteristics of such a telecommunication channel where the probability of a corrupted transmission is fluctuating. Such situations typically occur during information transmission in mobile networks when the users cross cell boundaries and the interference conditions change drastically.

First, the transmission process is modeled in terms of the BMAP/G/1 queue with feedback where the behavior of the input and the error probability depend on the state of a Markovian synchronous random environment. The latter describes the random changes of the interference conditions determining the success of a service completion. Here the probability of a repeated service, which can be interpreted as the error probability of a transmitted PDU, can change according to the state of that random environment. A general Batch Markovian Arrival Process (BMAP) describes the arrival stream of customers, modelling



the batch arrival of radio blocks of a segmented message at the DLL. Applying the machinery of matrix-geometric methods, the resulting model is characterized by a discrete-time Markov chain with quasi block-Toeplitz structure embedded upon service completions. Then necessary and sufficient conditions for the existence of the corresponding steady-state distribution of the queue length at these embedded epochs are determined. Furthermore, the latter is characterized as the unique solution of a generalized variant of the Pollaczek-Khintchine equation using a generating-function approach. Finally, the stationary queue-length distribution at arbitrary epochs is determined and an algorithm for its calculation is sketched.

The model of [161] includes as special cases both Takacs' single-server feedback and a BMAP/G/1 queue operating in a synchronous random environment without feedback.

### 2.7.4 Two-Class Non-Preemptive Priority Queue

A non-preemptive priority scheduler with two traffic classes and a separate buffer for each class is analyzed in [15]. The arrival streams of each class are assumed to be independent Poisson processes. The packet sizes are generally distributed. The aim of the analysis is to determine the maximum admissible low-priority traffic load under the assumption that the buffer size is finite and a constraint on the packet loss ratio is given. As a first step of the analysis, the system with two priority classes is translated into an equivalent system with a single queue and a single server where only low-priority traffic is present. For this system, the impact of the high-priority traffic is accounted by extending the service times experienced by low-priority packets. Furthermore, the system is analyzed using the diffusion approximation method, which only requires the two first moments of the packet arrival and departure processes. Since the input is Poisson, the crucial point is the departure process, characterized by the mean and the variance of the extended packet service times. These parameters are obtained from the busy period analysis of the high-priority traffic exploiting a functional equation approach. Finally, the probability distribution function of the low-priority queue size is determined and the admissible load is calculated. Numerical results illustrate the impact of the traffic load and the packet sizes of both priority classes on the admissible load. A comparison with the results obtained from the reduced service rate (RSR) method [162] is also included. The results indicate that when the high-priority traffic load is light or the high-priority packets are small compared to the low-priority packets, the impact of the high-priority traffic can be sufficiently captured by the first moment as in the RSR method. However, in the remaining cases the diffusion approximation method provides more accurate results than the RSR method.

## 2.8 Queueing Networks

The previous sections of this chapter deal with isolated queueing systems. In order to assess the performance of a communication network, however, it is also necessary to study networks of queues. This section is devoted to the COST 279 work on queueing networks. First, an approximate method to calculate end-to-end delay characteristics is presented. Next, a technique to determine the evolution of the characteristics of a traffic stream when it proceeds through a network is discussed. The latter can be useful to derive more accurate end-to-end performance characteristics.

### 2.8.1 End-to-End Delay Characteristics

The end-to-end delay is an important QoS parameter for real-time services. In [163], an analytical model to calculate end-to-end delays in packet networks is considered. The aim is to calculate the distribution of the end-to-end delay for a particular path consisting of a series of nodes. It is assumed that all the waiting times in the nodes in the end-to-end path are *statistically independent*; this is a key assumption to obtain the end-to-end delay by convolution. For queueing networks with FCFS queueing discipline this property only holds for the acyclic form of Jackson Networks, where a packet visits a node at most once. In [164], however, it is argued that if the load from a particular flow is only a small fraction of the total amount of traffic at a node and the input processes to the network are “smoother than Poisson,” i.e., with less variability, then the independence assumption will be quite reasonable and will represent a worst case scenario. Therefore, the M/G/1 queue is taken as the model to find the waiting time distribution in each node, and then the convolution is applied to obtain the end-to-end waiting time distribution.

If all nodes have identically distributed service times, the corresponding convolution may be substantially simplified, and closed-form expressions are obtained in terms of derivatives with respect to the load parameter. Special emphasis is put in [163] on the case with constant service times, since this is an important case for applications. Numerical results show that end-to-end delays in chains for up to 20 nodes may be analyzed without numerical difficulties. It is also possible to extend some of the results to cover convolutions between equally loaded groups of queues with different service time distributions in each group.

### 2.8.2 Evolution of Traffic Characteristics

The evolution of the characteristics of the interarrival and interdeparture times between voice packets as they proceed through a number of network nodes is

studied in [165]. Each network node is assumed to have an infinite-capacity buffer. The arrival process in a node is modelled as the superposition of a single tagged voice stream and an independent background process that aggregates the remaining traffic sources. Since the load of a single voice stream is very low compared to the load of the aggregate traffic, the tagged voice packets can be represented as *markers*, i.e., packets of size zero. At the entrance of each network node, one thus has the tagged marker stream and the background stream. The tagged marker stream is characterized by the interarrival times between successive markers, which are assumed to be identically distributed but may be dependent. The background arrival process is described on a slot-per-slot basis according to a general iid process, independent of the tagged marker stream.

In [165], first, an expression for the PGF of the interdeparture times of the voice packets after one stage is established. The PGF of this interdeparture time is then used as the PGF of the interarrival times of the voice packets in the next stage, in order to assess the evolution of the interarrival-time characteristics throughout the network. Following, the PGF of the interdeparture times between three successive voice packets (in case two successive interarrival times may be dependent of each other) is also calculated.

## 2.9 Models for Optical Buffers and Networks

Optical packet switching (OPS) and optical burst switching (OBS) seem promising techniques to cope with the explosive growth of the Internet traffic. This section presents some new models and analysis techniques to evaluate the performance of burstification queues, optical cross-connects, and various types of fiber delay line buffers. We refer to chapter 5 on optical networks for a further discussion of the obtained results.

### 2.9.1 Burstification Queues

In the edge routers of an OBS network, IP packets are assembled into bursts. Core OBS routers forward these bursts in the optical domain through the OBS network. An OBS edge router can be decomposed into multiple burstification units (BU). Each BU consists of a set of separate output queues. In [166], the burstification of a single isolated output burst queue is investigated. First, a single-threshold burst assembly mechanism is studied, where bursts are released whenever they contain exactly  $S$  packets. In this case, especially for low throughputs, the packets may have long delays. Therefore, as a next step, also a two-threshold model is investigated, where besides a threshold on size,

a threshold on a burst's age is imposed. Thus, bursts are also released if, since the start of their assembly, a time  $T$  has expired. For both queueing models, some performance characteristics are calculated. Results include the Probability Mass Function (PMF) of the system content in the output queue of the OBS edge routers, the PMF of the delay of the bursts, defined as the interdeparture time between two bursts, and the PMF of the delay of the individual packets. Using these PMFs, the mean values, variances, and tail distributions of the system content and of the burst and packet delays are derived.

### 2.9.2 Optical Cross-Connects

An asynchronous bufferless optical cross-connect using a shared wavelength converter pool with sharing on a per-output-link basis is studied in [167]. In the model studied, an incoming optical burst, or optical packet, is blocked either because there is no available wavelength on the output link, or the incoming burst requires conversion but the converter pool is fully occupied. The goal is to exactly calculate the steady-state blocking probabilities as a function of the basic system parameters, e.g., mean arrival rate, arrival statistics, and converter pool size. Using the traditional model of Poisson burst arrivals, exponential burst lengths, and uniformly distributed burst colors, this problem is formulated in [167] as one of finding the steady-state solution of a finite Continuous-Time Markov Chain (CTMC) with a block tridiagonal infinitesimal generator or, equivalently, that of a finite non-homogeneous QBD process. The number of converters in use form the phase of the QBD process, whereas the level process is dictated by the number of wavelengths in use. Although matrix-geometric forms of solutions are available for finite and infinite QBDs with a homogeneous structure, i.e., where block rows repeat, numerical studies of non-homogeneous QBDs are rather rare. A stable and numerically efficient technique based on block tridiagonal LU factorizations is proposed for exactly calculating the steady-state probabilities of the non-homogeneous QBD. It is shown through numerical examples that blocking probabilities can exactly and efficiently be found even for very large systems and rare blocking probabilities. The formulation of the problem is also extended to phase-type (PH-type) burst arrivals by incorporating the phase of the arrival process in the phase process of the non-homogeneous QBD using Kronecker calculus. The results obtained using PH-type arrivals clearly demonstrate that the coefficient of variation of burst interarrival times is critical in burst blocking performance, and therefore burst shaping at the edge of the burst/packet switching domain can be used as a proactive congestion control mechanism in next-generation optical networks.

### 2.9.3 Fiber Delay Line Buffers

In the design of all-optical switches, the lack of optical Random Access Memory (RAM) poses a big challenge. Besides wavelength conversion and deflection routing, the use of fiber delay lines (FDL) can help alleviate the output port contention problem. These FDLs are passive components that can delay an optical packet or an optical data burst for a fixed time. Usually, an FDL buffer implements the delays  $0 \cdot D, 1 \cdot D, \dots, N \cdot D$ , where  $D$  is the so-called *granularity* and  $N \cdot D$  can be considered as the *capacity* of the FDL buffer. Note that not all delays can be thus obtained, typically leading to the creation of voids in the scheduling and to underutilization of the output channel. For this reason, FDL buffers are also sometimes called degenerate buffers.

The performance of a single-wavelength FDL buffer is analyzed in [168], for the synchronous case, and in [169], for the asynchronous case. The quantity of interest in the analysis is the so-called *scheduling horizon*. It is defined as the earliest time at which the channel will become available again, and can be considered the equivalent of the unfinished work in non-degenerate buffers. If one denotes by  $H_k$  this scheduling horizon as seen by the  $k$ -th arrival, one can easily establish, assuming an infinite FDL buffer, the following recursion:

$$H_{k+1} = \left[ B_k + D \left\lceil \frac{H_k}{D} \right\rceil - \tau_k \right]^+. \quad (2.10)$$

Here  $B_k$  denotes the size of the  $k$ -th burst, and  $\tau_k$  the interarrival time between the  $k$ -th and  $(k + 1)$ -th burst. One easily recognizes part of the evolution equation for non-degenerate buffers, involving the operation

$$[\dots - \tau_k]^+. \quad (2.11)$$

Under the usual assumptions of iid interarrival times and iid. burst sizes, the solution to this problem in the transform domain is well-known. The part

$$D \left\lceil \frac{H_k}{D} \right\rceil \quad (2.12)$$

reflects the finite granularity of the FDLs. In discrete time, as was done in [168], this operation on random variables can be translated into an operation on their PGFs, by using an identity involving the complex  $D$ -th roots of unity. By combining both partial solutions, one obtains in the end the PGF of the scheduling horizon  $H$  in equilibrium. From the latter, one can obtain several measures of interest, such as the maximum tolerable load, i.e., the load at which the infinite system becomes unstable. Due to the creation of voids, this load is typically less than unity; it also shows a slight dependency on the burst

size distribution. Further, a heuristic can be used to map the tail probabilities  $\Pr[H > M \cdot D]$  to loss probabilities in a finite system of capacity  $M \cdot D$ . An optimum granularity  $D_{opt}$  exists, establishing a compromise between increasing capacity ( $D \rightarrow \infty$ ) and small voids ( $D \rightarrow 0$ ). This optimal value not only depends on the burst size distribution, but also on the load offered to the system. Results for the asynchronous case were obtained in [169] by taking the appropriate limits for slot lengths going to zero. A more direct analysis, explicitly taking into account the continuous-time nature of asynchronous FDL buffers, leads to the same results. In the end, one obtains the Laplace-Stieltjes transform of the scheduling horizon in equilibrium, from which other results can be obtained, proceeding as in the discrete-time case. Here too, the optimal granularity is sensitive to both the load and the burst size distribution.

In [170], the performance of an optical packet switch is investigated. Specifically, an FDL-structure consisting of  $N$  delay lines with increasing lengths is considered. It is assumed that the optical packets have deterministic lengths and the  $i$ -th delay line ( $i = 1, \dots, N$ ) has a length of  $i$  times the packet length. Time is slotted, where one slot corresponds to the time needed to transmit a packet. The numbers of packet arrivals are assumed to be iid from slot to slot. The scheduling discipline is smallest FDL first: if  $i$  packets arrive at the same time, they are put in the  $i$  delay lines with the smallest lengths, with  $i - N$  packets lost if  $i > N$ . First, an expression for the steady-state PGF of the delay line content of the FDL-structure with increasing lengths is derived. From this PGF, the Packet Loss Ratio (PLR) in an output buffering optical packet switch is then calculated. Through some figures, the impact of the number of delay lines and the load on the PLR is shown. An important conclusion is that putting one or two delay lines at each output can reduce the PLR significantly. Adding even more delay lines per output does not reduce the PLR significantly, though. If a lower PLR has to be obtained, the scheduling discipline discussed in [170] is not sufficient, and more complex scheduling methods are necessary.

The performance of a two-stage optical buffer is investigated in [171]. Fixed-length packets enter the first stage according to the simple routing scheme investigated in [170]. This scheme routes incoming packets to delay lines such that there is no contention at the input of the delay lines. However, contention at the output is possible. The output traffic is then routed to the FDLs of the second stage, again according to the simple routing scheme. Using a generating-functions approach, the PLR of the buffer structure under consideration is obtained.

Analysis and Design of Advanced Multiservice Networks  
Supporting Mobility, Multimedia, and Internetworking  
COST Action 279 Final Report

Brazio, J.; Tran-Gia, P.; Akar, N.; Beben, A.; Burakowski,  
W.; Fiedler, M.; Karasan, E.; Menth, M.; Olivier, P.;  
Tutschku, K.; Wittevrongel, S. (Eds.)

2006, XVI, 252 p., Hardcover

ISBN: 978-0-387-28172-8