TASK QUARTERLY 11 No 3, 237–252

ANALYTICAL INVESTIGATION OF CONGESTION-AVOIDANCE STRATEGIES IN CLOSED-TYPE QUEUING MODELS OF COMPUTER NETWORKS WITH PRIORITY SCHEDULING

WALENTY ONISZCZUK

Bialystok University of Technology, Faculty of Computer Science,
Wiejska 45A, 15-351 Bialystok, Poland
walenty@ii.pb.bialystok.pl

(Received 11 September 2006; revised manuscript received 28 December 2006)

Abstract: A new approach is presented to modelling intelligent admission control and congestion avoiding mechanism, without rejecting new requests, embedded into a priority closed computer network. Most Call Admission Control (CAC) algorithms treat every request uniformly and hence optimize network performance by maximizing the number of admitted and served requests. In practice, requests have various levels of importance to the network, for example priority classes. Here, the investigated closed network with priority scheduling has been reduced to two service centres, which allows for decomposition of a larger network into a chain of individual queues, where each queue can be studied in isolation. A new algorithm (approach) of this special type of closed priority queuing systems is presented, including a node consisting of several priority sources generating tasks, designated as an Infinite Server (IS), and a service centre with a single service line. This model type is frequently described as a finite source, pre-emptive-resume priority queue (with general distribution of service time). The pre-emptive service discipline allows a task of lower priority to be returned to the head of a queue when a new task of higher priority arrives. A mathematical model of provisioning and admission control mechanism is also described. The idea behind this mechanism has been derived from the Hidden Markov Model (HMM) theory. It is crucial in the CAC process that the network manager obtains correct information about the traffic characteristics declared by the user. Otherwise, the quality of service (QoS) may be dramatically reduced by accepting tasks based on erroneous traffic descriptors. Numerical results illustrate the strategy's effectiveness in avoiding congestion problems.

Keywords: pre-emptive-resume queuing model, mean value analysis (MVA), congestion problem, call admission control (CAC), hidden Markov models (HMM's)

1. Introduction

Overload of computer networks is observed when there is long-term demand for service resources which exhausts or exceeds their operational capacity. In order to avoid such problems, special mechanisms of resource management and provisioning are used, classified as Topology, Capacity and Flow Assignment (TCFA) methods.

Their solutions are often rough estimates or are based on inexact approximations. To counteract network congestion, restrictive procedures are introduced that regulate the intensity of information flow from the source, referred to as admission controls (CAC). It is important to remember that the activity of sources varies in time and the ability to predicate the consequences of the current flow of information from a source is directly related to the overall network's health. It is also required that each of the control mechanisms operates in real time and maximizes the network's resources. Therefore, call admission control is introduced as a preventive measure against overload in order to enforce a pre-negotiated level of quality of service (QoS) parameters.

Various admission control algorithms have been proposed in the literature [1]. The deterministic approach derives a formula of the maximum number of admitted requests (tasks) under the worst-case load, since the admission control policy is based on worst-case scenarios. The approach is based on predication from measurements of the resource usage status. The statistical approach assumes that the average data access time does not change significantly and it admits new tasks as long as the network server can meet the statistical estimation of the total data rate. Adaptive CAC admits new tasks on the basis of an extrapolation from past measurements of the storage server's performance.

The above-mentioned research does not consider different priorities of client tasks and attempts to admit as many tasks as possible without considering the importance of each task. A majority of priority service schemes proposed and studied in the past can be classified either as "time-priority" or "space-priority" ones. Recently, attempts have been made to incorporate both space- and time-priority policies to deliver a diversified service [2].

Generally, call admission control algorithms are responsible for determining if a new request can be accepted when server load does not exceed the available node capacity or rejected otherwise. In this paper, a new approach for modelling an intelligent admission control mechanism is proposed based on the Hidden Markov Model (HMM) theory embedded into pre-emptive priority computer networks. The proposed analytical models aim at finding the best partition of input streams, optimizing the network's performance by avoiding congestion, without rejecting new requests (tasks). In the process of call admission control and congestion avoidance, it is crucial that the network manager obtains correct information about the traffic characteristics declared by the group of priority users. The pre-emptive priority scheduling described here, allows tasks of higher priority to temporarily interrupt lower-priority services and resume their execution after they have been completed.

The performance evaluation of an extensive computer network with multiple nodes is extremely difficult and is usually an approximation obtained by decomposing such network into a sequence (or set) of individual queues, so that each of them can be investigated in isolation [3–8]. A mathematical study that measures the effectiveness of a closed (finite task population) computer network described by the queuing theory as the finite source model with pre-emptive priority task scheduling is presented in the next section. The analysis is based on a network containing embedded intelligent admission control mechanisms for avoiding sustained network overload.

The remainder of the paper is organized as follows. An exact analysis of the priority network is given in Section 2. The algorithm of calculating the main measures of effectiveness is developed in Section 3. An analytical model of the intelligent CAC mechanism based on the theory of Hidden Markov Models is described in Section 4. The numerical experiment results obtained from the analytical model of the priority network and the CAC mechanism are presented in Section 5. Final conclusions are drawn in Section 6.

2. A closed two-centre network's model and its exact analysis

Let us consider a two-node closed network (sources limited calling population) with an absolute (pre-emptive) priority task selection policy as shown in Figure 1. Unlike single queues, there is no easy notation to specify the type of a queuing network. The simplest classification of a queuing network is as either open or closed: an open queuing network has external arrivals and departures, a closed network has none. As shown in Figure 1, the tasks circulate in the system from one station to another, the total number of tasks in the system remaining constant. Tasks exiting the system immediately re-enter it. Users generate requests (tasks, jobs) at the terminals (Source Centre) serviced at the Service Centre. After a job is done, it waits at the user terminal for a random "think-time" interval before cycling again. The station-to-station flow of tasks defines the closed model's throughput. Here, the source centre is designated as an infinite server (IS), while the service centre consists of a single service line. In such a network with absolute priority, any task of higher priority is allowed to enter service immediately even if another lowers priority task is already present at the service station (node). When such a task arrives at the service node, the current one, with lower priority, is returned to the head of the queue [9–13].

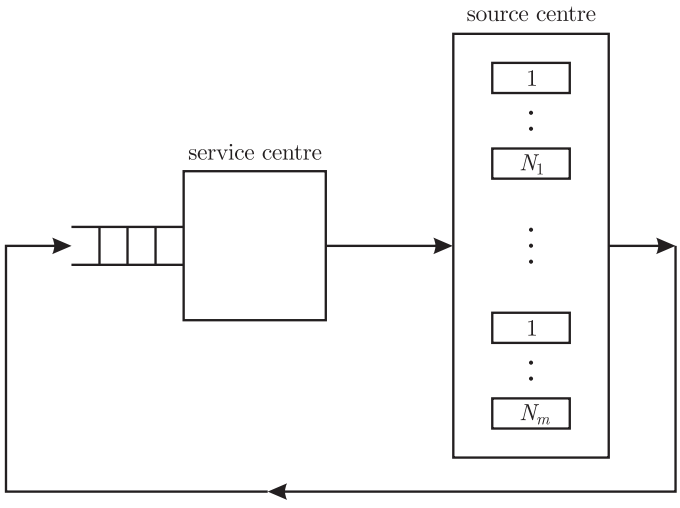

Figure 1. Closed two-centre network with pre-emptive resume priority

The general assumptions of the presented model with pre-emptive priority are as follows:

1. there are m task priority sources;
2. the source of class k (for $k=1,\ldots,m$) is finite, say of size N_k;
3. any source unit of class k generates tasks independently (with exponential distributed inter-arrival time a_k) and the arrival process is depicted with parameter $\lambda_k = 1/a_k$;
4. each service time is generally distributed (type G) with the first two moments given, $s_k^{(1)} = s_k$ and $s_k^{(2)}$ $(k=1,\ldots,m)$.

The mean response time of a service centre (waiting + service times), q_k, for a k priority task is as follows:

$$q_k = w_k + s_k + u_k, \tag{1}$$

where:

w_k – mean waiting time for the task to be serviced,
s_k – mean service time (without interruption periods),
u_k – mean interruption time (when a task of class k is interrupted by a higher priority task and returned to the head of the k-class queue).

The interruption time, u_k, is not applicable to the highest class of tasks ($k=1$), since they are serviced as FIFO. Generally, the value of mean interruption time is determined by stream intensity of the higher priority task and equals:

$$u_k = \frac{\sum\limits_{i=1}^{k-1} l_i \cdot \sum\limits_{i=1}^{k-1} \Lambda_i}{(1-\sum\limits_{i=1}^{k-1} l_i) \cdot \sum\limits_{i=1}^{k-1} (N_i \cdot \lambda_i)} \cdot s_i, \tag{2}$$

where:

l_k – the utilization factor for k priority tasks,
Λ_k – the mean rate of k-class task arrivals at the service station.

The w_k parameter of Equation (1) can be divided into three components:

(a) the mean time of waiting for a task of higher priority, or that of equal priority already present at the service station, to be serviced:

$$t_{1_k} = t_{11_k} + t_{12_k} = \sum_{i=1}^{k-1} (l_i \cdot \Delta_i) + l_k \cdot \Delta_k, \tag{3}$$

where

$$\Delta_i = \frac{s_i^{(2)}}{2 \cdot s_i},$$

(b) the mean time of waiting for pre-empted and resumed tasks from 2 to k classes to be serviced:

$$t_{2_k} = t_{21_k} + t_{22_k} = \sum_{i=2}^{k-1} (u_i \cdot \Lambda_i \cdot \Delta_i) + u_k \cdot \Lambda_k \cdot \Delta_k, \tag{4}$$

(c) the mean time of waiting for tasks of priorities from 1 to k already present in the queue to be serviced (v_i being the average queue length with i priority):

$$t_{3_k} = t_{31_k} + t_{32_k} = \sum_{i=1}^{k-1} (v_i \cdot s_i) + v_k \cdot s_k, \tag{5}$$

(d) the average time required to service tasks of higher priority which have joined the queue during the waiting time, w_k, of task k (Λ_i being the total arrival rate for i-priority tasks):

$$t_{4_k} = w_k \cdot \sum_{i=1}^{k-1} (\Lambda_i \cdot s_i). \tag{6}$$

The analysis of such systems becomes complicated as the Λ_k parameter (for $k = 1,\ldots, m$) and the above-described constraints are directly dependent on the number of k-priority tasks present at the service node, which is often unknown. Parameter Λ_k can be calculated from the following equation:

$$\Lambda_k = (N_k - n_k) \cdot \lambda_k, \tag{7}$$

where n_k is the average number of k-class tasks present at the service node.

Algorithms for calculating the effectiveness of a two-node closed network are presented in the next section.

3. MVA recursive algorithm

The presented recursive algorithm for analysis of a network with two service centres belongs to Mean Value Analysis (MVA) methods [14–18]. Basing on this approach, we can start network analysis for each priority class from higher priorities, *i.e.* from $N_k = 1$ with step 1 and continue until $N_{k_{\max}}$ by incrementing N_k by 1 (for $k = 1,\ldots, m$). As the model is pre-emptive, k-class tasks are unaffected in any way by the existence of classes $k+1$, $k+2$,.... In particular, class 1 tasks behave as they would in a single-class queuing system. Thus, we can start computing the effectiveness of a two-node closed network basing solely on the highest priority tasks ($k = 1$).

ALGORITHM. Calculate the mean waiting and response times (w_k, q_k), throughputs (l_k), the mean queue lengths (v_k) and the common mean arrival rate, Λ_k ($k = 1,\ldots, m$), for all priority classes. Here, parameters with an additional $(N-1)$ symbol (*e.g.* $l_{k(N-1)}$) belong to the net of source size reduced by 1.

(a) Part 1. Let $k = 1$ (higher priority class)

Step 1. Let $N_1 = 1$, which means $v_1 = 0.0$, $w_1 = 0.0$,

$$\Lambda_1 = \frac{1}{(a_1 + s_1)},$$

$$l_1 = \Lambda_1 \cdot s_1 .$$

Step 2. Let $N_1 = 2$

$$w_1 = l_{1(N-1)} \cdot \frac{s_1^{(2)}}{2 \cdot s_1},$$

$$q_1 = w_1 + s_1 ,$$

$$\Lambda_1 = \frac{N_1}{a_1 + q_1},$$

$$v_1 = w_1 \cdot \Lambda_1 ,$$

$$l_1 = \Lambda_1 \cdot s_1 .$$

Step 3. Let $N_1 = 3$

$$w_1 = l_{1(N-1)} \cdot \frac{s_1^{(2)}}{2 \cdot s_1} + v_{1(N-1)} \cdot s_1 .$$

Calculations for parameters q_1, Λ_1, v_1, l_1, are the same as in step 2.

Step 4. Let $N_1 = 4$ – identical calculation as in step 3 and so forth.

Last step. Let $N_1 = N_{1_{\max}}$ – identical calculation as in step 3.

Stop.

For lower priority classes ($k > 1$), the calculation algorithm requires a modification to allow lower priority tasks to be omitted, interrupted or even returned to the queue if currently served. As above, our analysis starts from the source size equal to 1, which means that $N_k = 1$.

Let $N_k = 1$. If there are no other k-class tasks, the waiting time, w_k, is determined only by higher-priority classes. In this case Equation (6) must be modified to reflect the change (w_{k-1} is related to a network with source size reduced by 1) and has the following form:

$$w_k = t_{11_k} + t_{21_k} + t_{31_k} + w_{k-1} \cdot \sum_{i=1}^{k-1} (\Lambda_i \cdot s_i). \tag{8}$$

As the mean interruption time, Equation (2), is dependent on the highest-priority tasks, we can calculate the response time according to the following formula:

$$q_k = w_k + u_k + s_k, \tag{9}$$

while the mean task arrival rates areas follows:

$$\Lambda_k = \frac{1}{a_k + q_k}. \tag{10}$$

The remaining measurements can be calculated in the following way (l_k being throughputs, v_k – mean queue lengths):

$$l_k = \Lambda_k \cdot s_k, \tag{11}$$

$$v_k = w_k \cdot \Lambda_k. \tag{12}$$

Let $N_k > 1$, up to $N_{k_{\max}}$. In this case the waiting time for k-class tasks will be calculated from Equations (3)–(6) (the diminishing source size is factored in the equation, as all arriving tasks recursively see its related source reduced to $N_k - 1$):

$$w_k = t_{1_k} + t_{2_k} + t_{3_k} + t_{4_k}, \tag{13}$$

or more precisely:

$$\begin{aligned} w_k = {} & t_{11_k} + l_{k(N-1)} \cdot \Delta_k + t_{21_k} + u_k \cdot \Lambda_{k(N-1)} \cdot \Delta_k + \\ & + t_{31_k} + v_{k(N-1)} \cdot s_k + w_{k(N-1)} \cdot \sum_{i=1}^{k-1} (\Lambda_i \cdot s_i). \end{aligned} \tag{14}$$

Therefore, the mean response time is obtained as:

$$q_k = w_k + u_k + s_k, \tag{15}$$

and the mean arrival rate is:

$$\Lambda_k = \frac{N_k}{a_k + q_k}. \tag{16}$$

The other parameters can be calculated as follows:

$$l_k = \Lambda_k \cdot s_k, \tag{17}$$

$$v_k = w_k \cdot \Lambda_k. \tag{18}$$

ALGORITHM (continued):

(b) Part 2. Let $k > 1$ (for lower priority classes):

Step 1. Let $N_k = 1$.
Parameters q_k, w_k, Λ_k, l_k, v_k are calculated from Equations (8)–(12).

Step 2. Let $N_k = 2$.
Use Equations (14)–(18) to calculate parameters w_k, q_k, Λ_k, l_k, v_k.

Step 3. Let $N_k = 3$ – identical calculations as in step 2, *etc.*

Last step. Let $N_k = N_{k_{\max}}$ – identical calculations as in step 2.

Stop.

A set of experiments was performed for the above-presented algorithm, with selected distribution of service time (type G) in each priority class. The analytical results were compared with experimental simulation results. Mean waiting and response times, mean queue lengths and a server throughput were chosen in this validation for all priority classes. The simulation models were written in the SIMSCRIPT II.5 language. The warm-up period was rejected during the simulation experiments and their duration was chosen so as to warrant high reliability of the obtained results. A comparison of the mathematically calculated and simulation results (95% confidence interval) exhibit good resemblance in a wide spectrum of various input parameters (from small to high server utilization).

The recursive algorithm presented in this section is used for evaluating a set of possible combination variants of input streams, sub-optimally allocating the service resources to several classes of users. When performing call admission control (CAC) in priority networks, users are requested to declare their traffic descriptors, on the basis of which the aggregated load is estimated. Obviously, source groups cannot demand the maximum input stream at the same time, as that would lead to an instant network overload. Based on the assumption related to the Quality of Service (QoS) principle, the maximum permissible input stream must be chosen from the set of all possible combinations of arriving input streams. Consequently, CAC must be viewed as deciding whether to accept or reject a certain traffic configuration. Of course, in this type of computer network, priority classes may be assigned to groups of users according to a regular rule (*i.e.* permanently) or may be regulated dynamically.

4. Intelligent Call Admission Control (CAC) mechanisms

An approach to modelling provisioning and call admission control mechanisms in closed-type computer networks based on the theory of Hidden Markov Models

(HMM's) [19–24] is proposed in this paper. Hidden Markov Models are a new class of Markov chains which can be described as an extension of the classical chain, where observation is a probabilistic function of state. In HMM, the resulting model is a doubly embedded stochastic process with an underlying stochastic process that is not observable or hidden. The nature of the hidden process can be deduced from another process, producing a sequence of observations.

Any HMM is characterized by the following set of parameters:

1) N – the number of states in a Markov chain (The states are hidden and are denoted by $X=\{X_1, X_2,\ldots, X_N\}$, a state in time t is denoted as q_t.);
2) M – the number of events per state, denoted as $V=\{v_1, v_2,\ldots, v_M\}$;
3) the state transition probability distribution matrix, $A=\{a_{ij}\}$, where:

$$a_{ij}=P(q_{t+1}=X_j|q_t=X_i),\ \text{for } i,j=1,\ldots,N; \tag{19}$$

4) matrix B of event (observation symbol) probability distribution for each state (As an example, for a state with index j it is $B=\{b_j(k)\}$, where:

$$b_j(k)=P(v_k \text{ at } t|q_t=X_j),\ \text{for } j=1,\ldots,N \text{ and } k=1,\ldots,M; \tag{20}$$

5) the initial state distribution, $\pi=\{\pi_i\}$, where:

$$\pi_i=P(q_1=X_i),\ \text{for } i=1,\ldots,N. \tag{21}$$

The given values of N, M, A, B, π are used to generate an output sequence of observation symbols from the given alphabet, as follows:

$$O=O_1\,O_2\,O_3\ldots O_T, \tag{22}$$

where T is the number of observations in this sequence.

This means that the full specification of an HMM requires detailed information about N and M as the model dimension parameters, the probability distributions related to the A, B and π matrices, collectively described as $\lambda=(A, B, \pi)$, and generating a set of events (observation symbols). Additionally, in order to create an intelligent admission control mechanism based on the HMM theory it is necessary to realize a training procedure. During the training process, the $\lambda=(A, B, \pi)$ model parameter is adjusted so as to enable optimization of $P(O|\lambda)$ (the estimation problem). The training process optimally adopts the model parameters to an observed output sequence producing the best model for real phenomena [21, 23, 25, 26].

Prior to adjustment, we compute for any given observation sequence $O=O_1\,O_2\ldots O_T$ and model $\lambda=(A, B, \pi)$ the $P(O|\lambda)$ function, the probability that the observed sequence produced by the given model represented by N (number of states) and set of parameters in matrices A, B, π.

Let us consider a fixed state sequence:

$$Q=q_1\,q_2\ldots q_T, \tag{23}$$

where q_1 is an initial state.

The probability of this observation sequence can be calculated by using the forward procedure presented in [19] and [23]. A definition of forward variable $\alpha_t(i)$,

the probability of the partial observation sequence $O_1\, O_2 \ldots O_T$ until time t and state X_i at time t, can be obtained from the model:

$$\alpha_t(i) = P(O_1\, O_2 \ldots O_t,\, q_t = X_i|\lambda). \tag{24}$$

Calculating $P(O|\lambda)$ requires the following steps:

1. initialization of forward probabilities as joint probabilities of state X_i and initial observation O_1:

$$\alpha_1(i) = \pi_i b_i(O_1),\ \text{for } i = 1, \ldots, N; \tag{25}$$

2. iterative calculation of $\alpha_{t+1}(j)$ by multiplication of a quantity. This characterizes all possibilities of reaching state X_j at time $t+1$ from N possible states, X_i $(i = 1, \ldots, N)$, at time t and probability of observation O_{t+1} in state j:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}),\ \text{for } t = 1, \ldots, T-1 \text{ and } j = 1, \ldots, N; \tag{26}$$

3. termination with the following given definition:

$$\alpha_T(i) = P(O_1\, O_2 \ldots O_T,\, q_T = X_i|\lambda) \tag{27}$$

summing all of the forward variables $\alpha_T(i)$:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i). \tag{28}$$

The second part of the forward procedure is the backward procedure, as explained in [23, 24]. Both procedures are used in the training processes.

Let a backward variable be defined as follows:

$$\beta_t(i) = P(O_{t+1}, O_{t+2} \ldots O_T|q_t = X_i, \lambda), \tag{29}$$

the probability of the partial observation sequence from $t+1$ till the end, given state X_i at time t and the model λ.

Calculation of $\beta_t(i)$ consists of two steps only:

1. initialization:

$$\beta_T(i) = 1,\ \text{for } i = 1, \ldots, N,\ \text{and} \tag{30}$$

2. looping:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j),\ \text{for } t = T-1, T-2, \ldots, 1 \text{ and } i = 1, \ldots, N. \tag{31}$$

The later equation describes how HMM can be in the state X_i at time t, from all possible states at time $t+1$.

Another question related to the adjustment procedure is the computation of variable $\gamma_t(i)$, where the probability of being in state X_i at time t, given the

observation sequence O and the model λ [23] can be derived from the following formula:

$$\gamma_t(i) = P(q_t = X_i | O, \lambda). \tag{32}$$

Let us define variable $\xi_t(i,j)$, the probability of being in state X_i at time t, and in state X_j at time $t+1$, given the model and the observation sequence:

$$\xi_t(i,j) = P(q_t = X_i, q_{t+1} = X_j | O, \lambda) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum\limits_{i=1}^{N} \sum\limits_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \tag{33}$$

and the previously defined value will be equal to:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j). \tag{34}$$

The summation of $\gamma_t(i)$ over time index t, can be interpreted as:

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{expected number of times that state } X_i \text{ is visited.} \tag{35}$$

Similarly, summation $\xi_t(i,j)$ over time t can be interpreted as:

$$\sum_{t=1}^{T-1} \xi_t(i,j) = \text{expected number of transitions from state } X_i \text{ to state } X_j. \tag{36}$$

Formulae (35)–(36) allow us to re-estimate the HMM parameters as follows:

(a) the expected frequency of being in state X_i for time $t = 1$:

$$\overline{\pi_i} = \gamma_1(i), \text{ for } I = 1, \ldots, N; \tag{37}$$

(b) the state transition probability distribution:

$$\overline{a_{ij}} = \frac{\sum\limits_{t=1}^{T-1} \xi_t(i,j)}{\sum\limits_{t=1}^{T-1} \gamma_t(i)}, \text{ for } i, j = 1, \ldots, N; \tag{38}$$

(c) the observation symbol probability distribution in state j:

$$\overline{b_j}(k) = \frac{\sum\limits_{t=1, s.t. Q_t = v_k}^{T} \gamma_t(j)}{\sum\limits_{t=1}^{T} \gamma_t(j)}. \tag{39}$$

In conclusion, on the basis of current model parameters, $\lambda = (A, B, \pi)$, values from formulae (37)–(39) can be computed and a re-estimated model can be defined, described as $\overline{\lambda} = (\overline{A}, \overline{B}, \overline{\pi})$. The new model $\overline{\lambda}$ is more likely than the λ model, $P(O|\overline{\lambda}) > P(O|\lambda)$, which means that a new model has been found from which a better observation sequence can be obtained. The final result of this iterative re-estimation procedure is called the maximum likelihood estimate of HMM.

The process of modelling and implementing the dynamic resource management and provisioning algorithms embedded in the computer network QoS manager, based on the main principles of Hidden Markov Models, requires the following steps:

1. establishing and building a list (set) of input streams from each priority source acceptable by a QoS manager in the service station. The list is related to the number of distinct observation symbols per state, M, from Hidden Markov Models. Another important HMM parameter will be the number of states, N, related to the number of priority classes in a closed network;
2. using the theory and algorithms presented in Sections 2 and 3, a set (table) of possible variants (combinations) of input streams must be obtained, sub-optimally allocating the service resources to several classes of users. The table must be located in the network QoS manager, where requests of input streams coming from each priority class relate to the $O = O_1\, O_2\, O_3 \ldots\, O_T$ observation sequence described in the HMM theory;
3. on the basis of statistical investigation of input stream intensities from each source and their modifications, we can create a matrix B from HMM's – a sequence of probability distribution in each state;
4. we have to choose a type of HMM suitable for this kind of application (for example a left-right model or its modification) and establish coefficients of matrix A and an initial state of distribution π;
5. for each variant of the input stream from the selected set of possible variants, we must build a Hidden Markov Model where its parameters will be adjusted to create an optimal function of $P(O|\lambda)$;
6. each intelligent admission control mechanism (QoS manager) in the network should regularly collect and detect the demands of input streams coming from priority sources. The QoS manager must recognize each input steam set by computing probabilities $P(O|\lambda)$ for each trained HMM and choosing the most adequate variant with the highest likelihood.

The workload characteristics of such a network are changeable so that a static admission control algorithm is not feasible and unable to adapt to the changes in run time. The server of such a network requires dynamic admission control policies to guarantee the delivery of on-demand priority input streams with QoS requirements. The QoS manager estimates the aggregated traffic based on the traffic descriptors declared by priority users and compares this estimation with the node's capacity.

5. Numerical example

In this section, numerical results of investigating a pre-emptive priority two-centre network (configured as shown in Figure 1) with embedded intelligent admission control mechanisms are presented to illustrate the potential effectiveness of the studied strategy in avoiding congestion problems.

For example, an engineering firm running several different laboratories provides each of its analysts with a personal computer, all of which are hooked up to a database server over an LAN. In addition, there is an expensive, standalone graphics workstation used for special-purpose design task. Engineers complain to their manager that the waiting time to use the workstation is too long. The manager is surprised, as the

utilization of the workstation is far from full, say 6/8 (6 hours out of 8). To convince the manager, one of the engineers performs a queuing analysis in which it is desirable to use priorities to resolve this problem. Priorities may be assigned in a variety of ways. For example, priorities may be assigned on the basis of traffic type. An important case is priority being assigned on the basis of the average service time. Often, tasks with shorter expected times are given priority over tasks with longer service times, *etc.*

The following configuration parameters are chosen:

- 4 priority classes of tasks,
- size of each source capacity – from 1 to 10 (ten acceptable values of input streams, $M = 10$),
- mean source time $a_k = 1/\lambda_k = 35.2$, for $k = 1, 2, 3, 4$,
- service parameters: $s_1 = 1.0$; $s_1^{(2)} = 2.00$ (exponential distribution),
  $s_2 = 1.2$; $s_2^{(2)} = 2.44$ (normal distribution),
  $s_3 = 2.0$; $s_3^{(2)} = 4.00$ (constant distribution),
  $s_4 = 1.6$; $s_4^{(2)} = 3.41$ (uniform distribution).

In this kind of network, tasks of varying priority classes enter the service centre via the QoS manager. The main responsibility of the QoS manager is admission control and dynamic assignment of suitable intensities of the input stream to priority sources with changing workload. One way to control call admission intelligently is to identify the possible workload conditions before the server is up for service. The sub-optimal partition of the priority stream intensities is maintained in a table such that the QoS manager is capable of watching workload change. The limitation of this approach is the contents of the table. At the same time, CAC algorithms are capable of finding a sub-optimal solution in such networks as workload changes in real time.

Analysis: step 1

A set of possible input stream variants is chosen (a special table created for the QoS manager), which properly (sub-optimally) allocate the service resources to several user classes on the basis of the acceptable server utilization parameter, $l = l_1 + l_2 + l_3 + l_4 = [0.85 - 0.90]$, and the permissible relation for the lowest priority class of the mean delay time to mean service time < 12. The results are presented in Table 1.

Selected streams intensities, being the $O = O_1\, O_2\, O_3 \ldots O_T$ sequence processes by the HMM.

Analysis: step 2

Initial parameters of the Hidden Markov Model are established.

(a) HMM type (see Figure 2):

For applications discussed in this paper, other than ergodic (fully connected) types of HMM have been found with the state transmission probability distribution, A, equal to:

$$A = \{a_{ij}\} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

and the initial state distribution $\pi = \{0, 0, 1, 0\}$.

Table 1. Variants of possible input streams

Variant	Stream intensities for each priority				Utilization	Delay time (max)
	1	2	3	4		
1	10	3	5	8	0.89	11.988
2	4	3	6	10	0.88	9.416
3	3	10	4	7	0.85	10.288
4	8	2	10	4	0.90	16.280
5	6	9	4	7	0.88	12.491
6	9	7	8	2	0.90	18.411
7	2	4	7	9	0.87	9.633
8	5	8	9	1	0.86	14.245
9	1	6	10	5	0.89	14.154
10	9	5	3	9	0.87	10.143

The HMM type presented here is a slight modification of the well-known left-right model (without the ability to transit to a previously visited state). The values of parameters of matrix A and vector π can be treated only as examples demonstrating the main idea of the intelligent admission control mechanism.

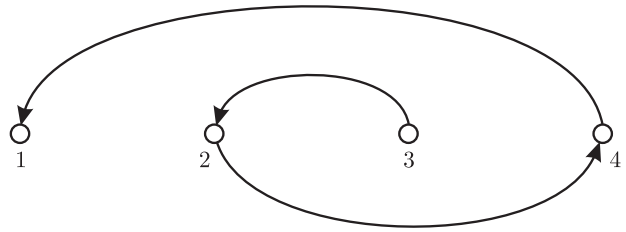

Figure 2. Illustration of a 4-state model without repeated state transition

(b) Coefficients of matrix B:

Table 2. Network stream intensities

Priority	Stream intensities variants									
	1	2	3	4	5	6	7	8	9	10
1	0.10	0.05	0.05	0.20	0.05	0.30	0.05	0.10	0.05	0.05
2	0.05	0.05	0.20	0.05	0.05	0.05	0.05	0.40	0.05	0.05
3	0.10	0.05	0.10	0.30	0.05	0.10	0.10	0.05	0.05	0.10
4	0.10	0.05	0.20	0.05	0.10	0.10	0.05	0.05	0.20	0.10

The values shown present the stream intensities and their relationship to priority classes.

Analysis: step 3

The probability of the $O = O_1\, O_2\, O_3 \ldots O_T$ observation sequence is calculated given the Hidden Markov Model, which means computing the $P(O|\lambda)$ function using the algorithm presented in Section 4.

Table 3. Probabilities of the observation sequence

Variant	Probability $P(O\|\lambda)$
1	0.0002000000
2	0.0003000000
3	0.0000125000
4	0.0000500000
5	0.0000125000
6	0.0000062500
7	0.0000062500
8	0.0004000000
9	0.0000250000
10	0.0000250000

Analysis: step 4

The model parameters are adjusted to maximize the $P(O|\lambda)$ function by training procedure with the algorithm presented in Section 4. Each variant of input stream intensities from Table 1 is adjusted separately to guarantee optimal selection all coefficients for matrices A, B and vector π for each input stream variant.

The results for variant 3 are given below for example. Table 4 contains coefficients of matrix B:

Table 4. Adjusted model parameters

Priority	Stream intensity variants									
	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	1	0	0	0
2	0	0	0	0	0	0	0	0	0	1
3	0	0	1	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0

All the coefficients of matrix A and vector π remained unchanged, with the value of $P_{\max}(O|\lambda)$ equal to 1.00.

Analysis: step 5 (a recognition mechanism for the stream demand vector)

A recognition and collection mechanism should be used periodically to digest all input streams. This can be achieved with trained Hidden Markov Models by calculating the functions of the largest probabilities and selecting the vector with the greatest value.

Let us choose variant 7 for example. In this instance, the $P(O|\lambda)$ sequence is calculated ten times by each trained HMM, as there are ten possible demand variants. The result is as follows: only for 7 variants the $P(O|\lambda)$ probability value is equal to 1.00, the remaining model probabilities are zero, which proves that the demand vector has been recognized properly.

6. Conclusions

A new approach to investigating a closed two-centre network with an intelligent control admission mechanism has been proposed. Starting from a theoretical study, a special type of queuing network with pre-emptive priority scheduling was proposed and efficient numerical procedures developed calculating the main measures of effectiveness in this type of network.

Following a formal verification, the analytical results of these values were compared with SIMSCRIPT II5 language simulation results for identical models and their initial parameters. The comparison (95% confidence interval) has shown good resemblance in a wide spectrum of various input parameters (from small to high server utilization).

A new approach for modelling an intelligent call admission control mechanism based on the theory of Hidden Markov Model (HMM) has also been presented. The mechanism is used for periodical collection and recognition of input stream demands from priority sources. It is also used to analyze and select optimal variants using the proposed HMM control mechanism.

Acknowledgements

This work was supported by the Bialystok University of Technology with grant no. W/WI/7/03.

References

[1] Cheng S-T, Chen C-M and Chen I-R 2003 *Perform. Eval.* **52** 1
[2] Stavrakakis I and Chen G 2000 *Perform. Eval.* **41** (1) 37
[3] Bruell S C and Balbo G 1980 *Computational Algorithms for Closed Queueing Networks*, North Holland, New York
[4] Bryant R M, Krzesinski A E, Lakshmi M S and Chandy K M 1984 *ACM Trans. Comput. Syst.* **2** (4) 335
[5] Gelenbe E and Mitrani I 1980 *Analysis and Synthesis of Computer Systems*, Academic Press, London
[6] Gelenbe E and Pujolle G 1987 *Introduction to Queueing Networks*, Wiley, New York
[7] Jaiswal N K 1968 *Priority Queues*, Academic Press, New York and London
[8] Morris R J T 1981 *The Bell System Tech. J.* **60** (8) 1745
[9] Berger A W and Whitt W 2000 *Perform. Eval.* **41** 249
[10] Choi B D, Choi D I, Lee Y and Sung D K 1998 *IEE Proc. Communications* **145** (5) 331
[11] Choi J S, Lee K S and Un Ch K 1997 *Perform. Eval.* **29** 177
[12] Laevens K and Bruneel H 1998 *Perform. Eval.* **33** (4) 249
[13] Li Z G, Yuan X J, Wen C Y and Soong B H 2001 *Computer Networks* **35** 203
[14] Lavenberg S S and Reiser M 1980 *J. Appl. Probability* **17** (4) 1048
[15] Reiser M and Kobayashi H 1975 *IBM J. Res. Dev.* **19** (3) 283
[16] Reiser M and Lavenberg S S 1980 *J. Assoc. Comput. Machinery* **27** (2) 313
[17] Reiser M 1981 *Perform. Eval.* **1** 7
[18] Wang H and Sevcik K C 2000 *Perform. Eval.* **39** 189
[19] Baum L E and Petrie T 1966 *Annals of Math. Statistics* **37** 1554
[20] Baum L E and Egon J A 1967 *Bull. Amer. Math. Soc.* **73** 360
[21] Bourlard H A and Morgan N 1994 *Connectionist Speech Recognition*, Kluwer Academic Publishers, Boston
[22] Morgan N and Bourlard H A 1995 *Proc. IEEE* **83** (5) 742
[23] Rabiner L R 1989 *Proc. IEEE* **77** (2) 257

[24] Rabiner L R and Jaung B H 1993 *Fundamentals of Speech Recognition*, Prentice-Hall, Englewood Cliffs, NJ
[25] Baum L E, Petrie T, Soules G and Weiss N A 1970 *Annals of Math. Statistics* **41** (1) 164
[26] Baum L E 1972 *Inequalities* **3** 1