The Equivalent Queuing Model by a Partition Algorithm for Tree Connected Servers

CHUNG-PING CHEN*, YING-WEN BAI**, HSIANG-HSIU PENG**, YING-YU CHEN**
Graduate Institute of Applied Science and Engineering, Fu Jen Catholic University*
Department of Electronic Engineering, National Taipei University of Technology*
Department of Electrical Engineering, Fu Jen Catholic University**
510 Chung Cheng Rd. Hsinchuang, New Taipei City 24205
TAIWAN, R.O.C.
491598038@mail.fju.edu.tw, bai@ee.fju.edu.tw, 051307@mail.fju.edu.tw,
499216210@ mail.fju.edu.tw

Abstract: - This paper aims at analysis efficiency in estimating the performance of tree connected servers. We use a queuing model to represent their equivalent performance and service quality. The queue types of the connection servers can be classified as serial, parallel and tree connections. We design an algorithm to simplify the equivalent serial-parallel queues. According to the equivalent queues we compute the system response time of tree connected servers. We use a network simulation and an analytical software tool to represent the equivalent performance of the queue. Our simulation uses various different service rates and arrival rates of the queue models and finds the system response time. We also measure the average system response time in comparison with the simulation result to find out the service rates of the actual servers and evaluate the accuracy of the algorithm. We will find that the error margin of measurement, simulation and computing ranges from 1.37%-19.27%.

Key-Words: - Serial-parallel Network, Web Servers, Service Rate, Tree Connected Servers, System Response Time, Equivalent Queuing Network.

1 Introduction

When the network structure becomes more and more complicated, and Web service requests become bigger and bigger, this generates a high blocking probability of the Web service and lengthens the system response time, with unacceptable results. In the designs of server architecture many improved methods have been put forth, as stated in the following examples. Upgrading the equipment using a multiple connection method increases the service rate by making use of a load balance mechanism in order to strengthen the Web service capability [1]. A Gigabit Ethernet between the network interface controller and the corresponding CPU raises the network flow and reduces the network transmission time [2]. Also, the use of a preemption mode improves the performance of the network and handles the high priority class information in advance. Then the low priority class makes use of the blank in the idle period of flow, thus reducing the whole blocking probability of the network [3].

This paper uses a serial-parallel method to improve the service capacity and to reduce the problem of a high blocking probability, and, by making use of the results of experiments, predicts the performance and efficiency of the tree connected servers [4, 5].

The server performance can be enhanced by a multi-layer with tree connection. This method analyzes the error margin of the system response time between the simulation and the physical measurement of the tree connected servers [4, 5]. We use a serial-parallel queue to imitate the system response time of Web requests of both the simulation and the measurement. We also compare the error margin between the measurement and the simulation.

As for the network system response times, their factors of influence which induce sorting are as follows [6]:

Network system response time = Network Time + Web Site Time + Time of DNS + Web Page Size + Multiple clicks

Network Time = Node Latency + Transmission Time

Web Site Time = Queuing Time + Service Time

Service Time = Function (CPU performance + disk driver performance + network processing performance)

Time of DNS = Time required for DNS look-up

Multiple clicks = Total click for all users. (Each click starts a Web page.)

The purpose of 'Fuzzy Classification Analysis of Rules Usage on Probability Reasoning Test with Multiple Raw Rule Score' is to analyze the rule usage of probability reasoning items by fuzzy partition with multiple rule score [7]. Another paper applies the fuzzy partition algorithm in the data analysis. In a computational fluid dynamics (CFD) flow simulations study, parallel computing and grid adaptation techniques are employed to achieve high efficiency and accuracy in a hybrid unstructured flow solver. The modified Recursive Coordinate Bisection (RCB) partition algorithm is exploited to repartition the computational domain due to its simplicity and efficiency once the load imbalance is detected [8].

Rohan presents a new method of improving the flow control mechanism in 10 Gigabit Ethernet (10GbE) WANs. The On/Off control method prescribed by the standard leads to fluctuations of flow rates and the queue levels of the switches. The approach converts the fair share calculated by the switch to an equivalent pause time. As a result, the algorithm complies with the standard [9]. Vasiliadis analyzes a network model in the context of providing trusted information between a central database and one or more servers in the network. In the proposed network scheme a server manager is coupled between the central database and the servers. The server manager provides trusted communication by transmitting configuration information between the central database and the servers in single communication channels [10].

Annop proposes a heuristic design algorithm called M-MENTOR that IP networks to support traffic engineering for both unicast and multicast traffic [11]. The goal of P. M. Papazoglou is to propose an alternative novel real time scheduling mechanism based on a synthesis of multitasking theory and queuing theory techniques, which could be involved in generating and investigating a new generation of event scheduling algorithms suitable for simulation models of cellular networks bandwidth management [12].

In wireless ad hoc networks, Osamah Badarneh proposes two algorithms for multilayered video multicast over heterogeneous wireless ad hoc networks [13]. Tzay-Farn Shih proposes a distributed cluster-based QoS multicast routing algorithm which only requires maintaining a local state at each node [14].

A simple and effective algorithm performs distance queries between a large number of points stored in quadtrees and octrees. The algorithm is developed and tested for the construction of diffusion-limited aggregates. The structure of the trees is the only feature used for the determination of approximate distances at any stage [15]. M. Vijayakumar's proposed system is designed to improve the K-means clustering algorithm with efficient centroid estimation models. These are a random selection with distance management, mean distance model and inter-cluster distance model [16].

The grid-based clustering algorithm is efficient, but its effect is seriously influenced by the size of the cells. The main idea of the deflected grid-based algorithm (DGD) is to deflect the original grid structure in each dimension of the data space after the clusters generated from this original structure have been obtained [17].

To increase the efficiency of data mining, through the establishment of transaction-item association matrix, Wei-Qing Sun's paper changes the process of association rule mining to elementary matrix operation, which makes the process of data mining clear and simple. He proposes an FP-network model which compresses the data needed in association rule mining in an FP-network [18].

Boutkhil Sidaoui introduces and investigates the performance of a simple framework for multiclass problems of a support vector machine (SVM) and presents a new architecture named EBTSVM (Efficient Binary Tree Multiclass SVM) to achieve high classification efficiency for multiclass problems. He builds a binary tree for multiclass SVM by genetic algorithms with the aim of obtaining optimal partitions for the optimal tree [19].

Cornel Balint focuses on the problem of performance evaluation in Global System for Mobile communications (GSM)/General Packet Radio Service (GPRS) networks, establishing dimensioning rules based on traffic evaluation and quality of service level for GSM/GPRS users [20]. For all optical networks, K. Ramesh Kumar an offers offline RWA (Routing and Wavelength Assignment) algorithms Scheme called WpDp-MaMiQ which is presented with dedicated path protection consideration that mitigates physical layer impairments (PLI) [21]. Due to high demand of optical virtual private network (OVPN) connection setups with guaranteed quality of service (QoS) requirement, Santos Kumar Das proposes a QoS-based OVPN connection setup mechanism over a WDM (Wavelength Division Multiplexing)

network to the end customer, which also maintains the minimum blocking probability [22].

The organization of this paper is as follows. In Section 2 we explain the process of the partition algorithm of an equivalent serial-parallel queue model to represent a local area network. In Section 3 we use the queue model for tree connected servers. In Section 4 we use multiple serial-parallel Website servers, make some physical measurements of the experiment and obtain the simulation results from the software tool. We also discern their error margin, if any. In Section 5 we draw our conclusions according to the results of the experiment.

2 The Queue Representation of the Tree Connected Server

Fig. 1 shows the campus network connection. The current research in this area, many examples are similar, and we use the campus network to illustrate this. Each department links the teachers' and the students' PCs by the hub or the switch, and the hub of the department links to each switch of the building. The switch of each building links to the switch of each college. The switch of each college then links to a computer center high-speed switch. Fig. 2 shows the queue representation that configures the campus network connection.

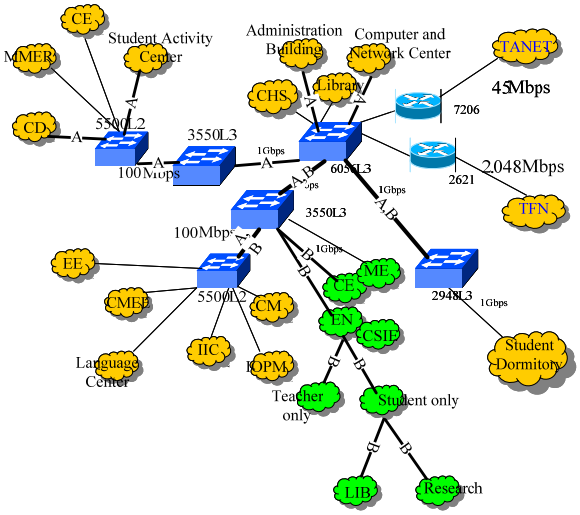

Fig. 1 The campus network connection.

The connection type of the campus network can be a tree-structured network. The LAN connection has many switches from the user to the servers. The server is a root, the switch is a tree branch, and the user is a leaf. We want to know the system response time and the number of users that connect in order to establish their particular relationships. The user's waiting time for Web page service is related to the size of the Web page, and we can estimate the amount of the exchange by the size of the Web page and the number of servers and thus predict the system response time of the network and the service rate of the servers.

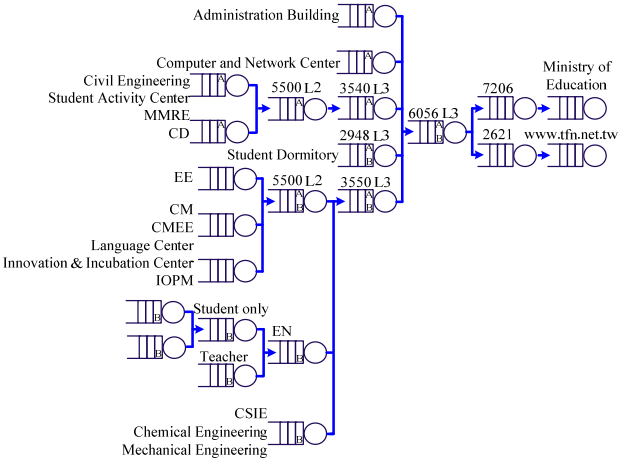

Fig. 2 The campus network connection.

2.1 The Tree Structure of the Equivalent Queuing Network

We use the tree structure as the foundation of the equivalent queuing network in this paper. The tree is constituted by the node and branch elements. The node-related number of branches is the branch degree of this node. When a branch connects to a node, it provides a branch degree. When the direction of this branch is restricted by a node, the direction of each branch is outward from the respective node. The root of the inward branch degree is a null. In addition to a root, all nodes of the tree must just have an inward branch degree. The leaf called external node has a zero outward branch degree. Neither a root nor a leaf can call an internal node. As the father node's outward branch degree is null, the leaf has an inward branch degree. Nodes which have the same father are called brother nodes. The ancestry of any node in the path that goes to this node comes from the root, and the posterity is any node below this father node. The path consists of a continuous sequence of nodes. Among them, each node tightly answers the next node. The stratum of the node is the distance between the node and the root. The height of the tree is the stratum from the leaf to the root. Any conjoint structure is under the root of the subtree. The binary tree has all leaves without two subtrees.

2.2 The Extrovert Tree Conversion to a Binary Tree

We simplify a local area network into a single queue model to investigate the performance of the whole local area network. Because the tree for a large area network may have an uncertain diversity in a hierarchical binary tree, we need to transfer a general tree into a binary tree by means of the follow steps.

Step 1: In addition to the branch of the left-most subtree delete the rest of the branch.

Step 2: Link all brothers nodes' usage branches.

Step 3: Turn the brother nodes into a right subtree, and turn the father and son nodes into a left subtree, thus forming a binary tree.

From the tree structures of a local area network the left subtree will become a serial server node, and the right subtree will become a parallel server node.

2.3 The Algorithm of the Equivalent Queues

After the binary tree transformation we simplify the root direction by the leaf. The node of the right subtree is a parallel node and that of the left subtree is a serial node. The father and son in the right subtree are two nodes which gradually move into a queue with a parallel connection. The father and son in the left subtree also are two nodes which gradually move into a queue with a serial connection. When the father node merges with the first subtree with two leaves in the left node, we are computing a right node. Based on this procedure, we keep the simplification till the last whole large network is equalized into a single equivalent queue. Fig. 3 shows the flow chart of the algorithm.

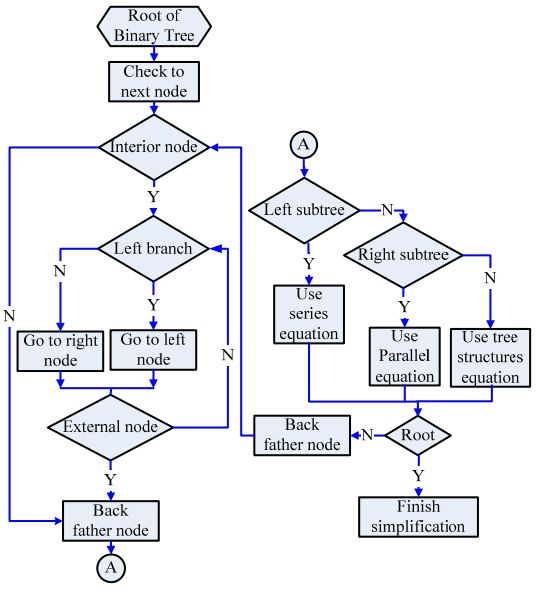

Fig. 3 The flow chart of the partition algorithm

Fig. 1 shows the original connection network. Part (a) of Fig. 4 shows the simplification steps of the equivalent queue. Step 2 shows the branch of the subtree farthest to the left, deletes the branch of the rest, and then links all brother node usage branches, as shown in part (b) of Fig. 4. Step 3 shows the transformation of the binary tree, as shown in part (c) of Fig. 4. Step 4 shows the GKL and the PRS, both of which have three nodes, with a tree-like serial-parallel link, and use an equation to merge them as the G' and the P'. The MNO has three nodes with a parallel connection, and we therefore use a parallel equivalent queue to merge them as M', as shown in part (d) of Fig. 4. Step 5 establishes DG' and HM' by an equivalent queue to merge them as D' and H', and the three nodes of the JP'Q are tree-like serial-parallel links and use the serial-parallel equivalent queue to merge as J', as shown in part (e) of Fig. 4. Step 6 shows the parallel H'IJ', and we use a parallel equivalent queue to merge them as F' with F again, as shown in part (f) of Fig. 4. Step 7 shows the parallel BCD'EF', and we use this parallel equivalent queue to merge B', as shown in part (g) of Fig. 4. Finally, this subtree serial merges with A and then gets the equivalent queue for these tree connected servers.

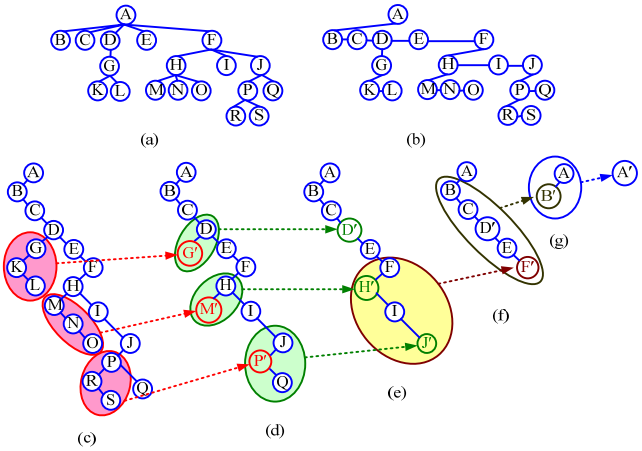

Fig. 4 The simplification steps for the equivalent queue.

Fig. 5 shows the pseudo code that simplifies a queueing network to an equivalent queue.

```
(Root of Binary Tree)
(Check to next Node)
IF (Interior node) -> "Y" THEN
IF (Left branch) -> "Y" THEN
```

```
        (Go to Left node)
ELSE
        (Go to Right node)
ELSE
                (Back father node)
I F (External node) -> "Y" THEN
        (Back father node)
ELSE
        (Interior node)
IF (Left subtree) -> "Y" THEN
        (Use series equation)
ELSE
        IF (Right subtree) -> "Y" THEN
        (Use Parallel equation)
        ELSE
                (Use tree structures equati
IF (Root) -> "Y" THEN
        (Finish simplification)
ELSE
(Back father node)
```

Fig. 5 The pseudo code for the simplification of a queueing network.

3 The Queueing Model for the Tree Connected Servers

To perform the simulation and measurement for the performance of the tree connected servers, we install one physical server computer network environment and use our algorithm to simplify this queueing network to obtain a single equivalent queue. We use the network software (QNAT) simulation system response time gained by the equivalent model for the measurement. The system definition and model parameters are shown in Table 1. The unit of measurement here is msec.

Table 1 System Definition and Parameters of the Model

Parameter	Description	Definition
λ	Web request arrival rate	Requests/sec
λ_{Pn}	Arrival rate of the nth parallel connection server	Requests/sec
μ_{Pn}	Service rate of the nth parallel connection server	Requests/sec
μ_{Peq_n}	Equivalent service rate of the nth parallel connection queue	Requests/sec
P_n	Dispatch probability of the nth parallel queue	None
μ_{S_n}	Service rate of the nth serial connection queue	Requests/sec
μ_{Seq_n}	Equivalent service rate of the nth serial connection queue	Requests/sec
μ_{eq}	Service rate of the equivalent queue	Requests/sec
$E_{S_n}(T)$	System response time of the nth serial connection queue	msec
$E_{P_n}(T)$	System response time of the nth parallel connection queue	msec
$E_{Peq_n}(T)$	Equivalent system response time of the nth parallel connection servers	msec
$E_{Seq_n}(T)$	Equivalent system response time of the nth serial connection servers	msec
$E_{eq}(T)$	Equivalent system response time of servers	msec

We use the idea of a serial-parallel equivalent electric circuit as our analytical foundation and verify the performance by measuring, and we use approximate equations for the multiple stages of the serial-parallel network. At the beginning we use queueing networks and Markov Chains [23] to analyze serial-parallel networks with the following assumptions.

- ✧ All requests are first in first out of the system.
- ✧ The total number of requests in the system is unlimited.
- ✧ The requests can leave the system from another node.
- ✧ All service times are exponentially distributed.

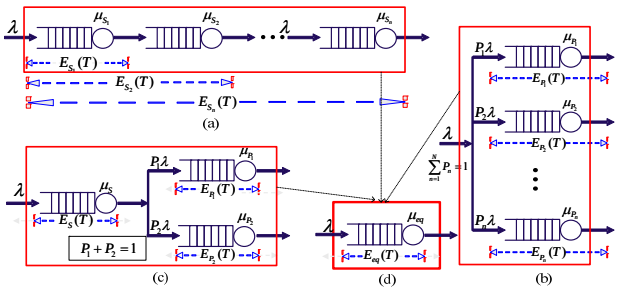

Fig. 6 Equivalent model of serial-parallel connection servers.

If a customer demands a packet, we set the arrival rate at λ. As the packet arrives at the server set, it is assigned into each serial-parallel node one after another. Because there is no need to detect the network status or operation, the required service time is so short that we can neglect it. In Figs. 6 (a) serial connected queues, (b) parallel connected queues, (c) tree connected queues, (d) equivalent queues, the Web service, after connecting with the node, has one small segment of waiting time in the buffer, and then this service is handled by the processor before it leaves the queue node.

The multiple serial queues under the same service rate are shown in Fig. 6 (a).

Eq. (1) shows the total system response time.

$$E_{Seq_n}(T) = E_{S_1} + E_{S_2} + \ldots + E_{S_n} = \frac{1}{\mu_{Seq} - \lambda} \tag{1}$$

When the service rates of servers are different, the equivalent service rate of a multiple serial queue is as shown in Eq. (2).

$$\mu_{Seq_n} = \frac{1 + E_{Seq}\lambda}{E_{Seq}} = \frac{1 + E_{Seq}\lambda}{E_{Seq}} = \frac{1 + \lambda(E_{S_1} + E_{S_2} + \ldots + E_{S_n})}{E_{S_1} + E_{S_2} + \ldots + E_{S_n}} \tag{2}$$

If the serial servers have the same service rate, then the system response time is as shown in Eq. (3).

$$E_{Seq_n}(T) = \frac{n}{\mu_{S_1} - \lambda} = \frac{1}{\mu_{Seq} - \lambda} = nE_{S_1} \tag{3}$$

If the serial servers have the same service rate, then the equivalent service rate is as shown in Eq. (4).

$$\mu_{Seq} = \frac{\mu_{S_1} + (n-1)\lambda}{n} = \frac{1 + n\lambda E_{S_1}}{nE_{S_1}} \tag{4}$$

To understand the impact of the service rate of the queuing model, as we can't change the performance of the CPU, so our design adjusts the load on the CPU. We use a multiplication loop execution ASP (Active Server Pages) program to increase the CPU load. When we increase the number of multiplication loops, the CPU load increases, and the service rate is reduced. The variation of the service rate shall verify the model. Fig. 7 shows a CPU loading of ASP=1, 1 K, 10 K, 20 K, … 100 K, 110 K, …, 200 K, 300 K, …, 900 K, 1 M, a total of 29 different stages, and the number of serial servers is 1-10. When CPU loads are increased or the number of serial servers is increased, the curves all present a linear rise.

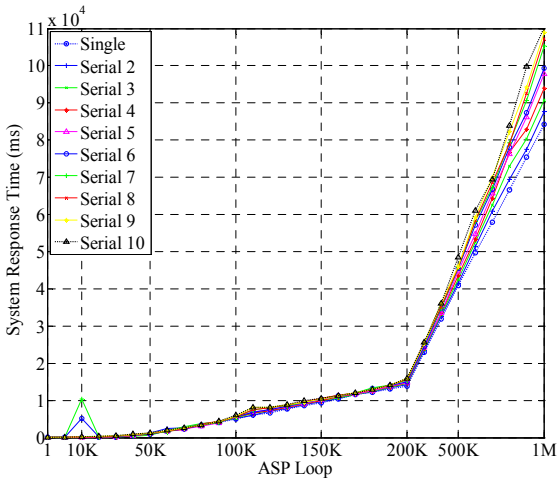

Fig. 7 System response time in various regions of the service rates.

The service rates of the multiple parallels are the same. The system response time is as shown in Fig. 6 (b).

$$E_{Peq_n}(T) = \frac{1}{n}(E_{P_1} + E_{P_2} + \ldots + E_{P_n}) = \frac{1}{\mu_{Peq} - \lambda} \tag{5}$$

Arrival rate

$$\lambda = \lambda_{P_1} + \lambda_{P_2} + \ldots + \lambda_{P_n}$$

Dispatch probability

$$P_1 + P_2 + \ldots + P_n = 1$$

System response time

$$E_{Peq_n}(T) = \frac{1}{n}\left(\frac{1}{\mu_{P_1} - P_1\lambda} + \frac{1}{\mu_{P_2} - P_2\lambda} + \ldots + \frac{1}{\mu_{P_n} - P_n\lambda}\right) = \frac{1}{\mu_{Peq} - \lambda} \tag{6}$$

The servers have different service rates; their parallel equivalent service rate is as shown in Eq. (7).

$$\mu_{Peq_n} = \frac{n + \lambda(E_{P_1} + E_{P_2} + \ldots + E_{P_n})}{E_{P_1} + E_{P_2} + \ldots + E_{P_n}} \tag{7}$$

If the servers have the same service rate, then the system response time is as shown in Eq. (8).

$$E_{Peq_n}(T) = \frac{1}{n}\left(\frac{n}{\mu_{P_1} - \lambda_{P_1}}\right) = \frac{n}{n\mu_{P_1} - n\lambda_{P_1}} = \frac{1}{\mu_{Peq} - \lambda} \tag{8}$$

If the servers have the same service rate, then the equivalent service rate is as shown in Eq. (9).

$$\mu_{Peq} = \frac{n\mu_{P_1} - \lambda}{n} + \lambda = \frac{n\mu_{P_1} - (n-1)\lambda}{n} \tag{9}$$

Fig. 8 shows the measurement data. The system response time uses the Y axis logarithm to express the characteristics of the response time. In Fig. 8, by utilizing a light loading, the parallel connection can't reduce the system response time by much.

With a middle loading the parallel connection reduces the system response time by up to four times. If the number of parallel connections increases, the system response time can be reduced to 1/2n - 1/n [2]. With a heavy loading the system response time is decreased by about 1/n of a single server [24].

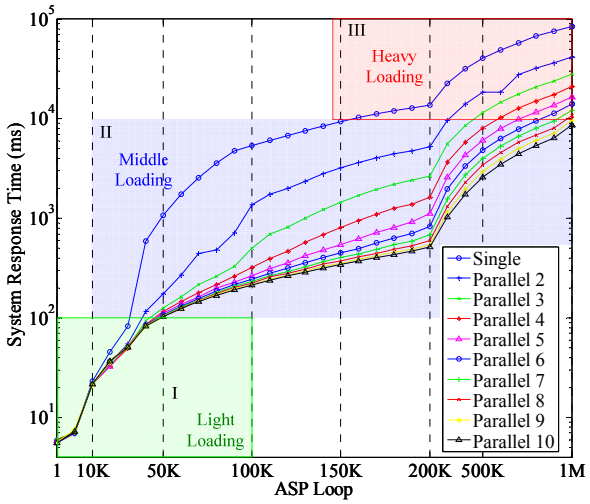

Fig. 8 System response time in various regions of the service rates.

The tree connection has the same service rate, as shown in Fig. 6 (c).

The system response time of tree connected servers is shown in Eq. (10).

$$E_{Teq}(T) = E_S + \frac{1}{2}(E_{P_1} + E_{P_2}) = \frac{1}{\mu_{Teq} - \lambda} \tag{10}$$

The service rates of servers are different in tree connected servers.

$$\mu_{Teq} = \frac{2 + 2\lambda E_S + \lambda(E_{P_1} + E_{P_2})}{2E_S + E_{P_1} + E_{P_2}} \tag{11}$$

If the service rates are the same, and the dispatch probabilities are the same, then the system response time is as shown in Eq. (12).

$$E_{Teq}(T) = \frac{1}{\mu_S - \lambda} + \frac{1}{2}\left(\frac{2}{\mu_S - \frac{1}{2}\lambda}\right) = \frac{1}{\mu_{Teq} - \lambda} \tag{12}$$

The tree connected servers have the same service rate, then the equivalent service rate is as shown in Eq. (13).

$$\mu_{Teq} = \frac{(\mu_S - \lambda)(2\mu_S - \lambda)}{4\mu_S - 3\lambda} + \lambda \tag{13}$$

4 Comparison of Simulation and Measurement

To verify the system performance of the tree connected servers of a real network we use a local area network as the measuring environment. We use the WebServer Stress Tool measurement software to obtain the serial-parallel service rate of the ASP network. The service time is made up of the system response time of the tree connected servers. We set up the servers' IP addresses as shown in Fig. 9.

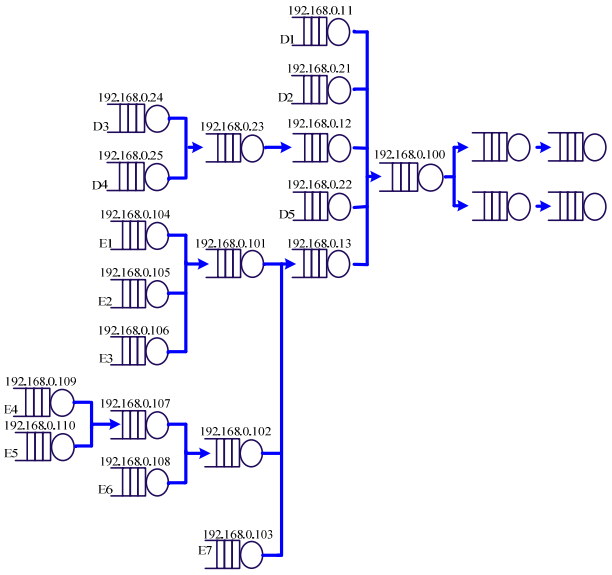

Fig. 9 IP addresses of the serial-parallel connection servers.

The client port requests the network to run a Web page service. We can neglect the Web page access time and only look at the tree connected server performance, so the Web page provides just simple data in mathematical operation.

In the actual measurement, to both adjust the service rate and transfer the simulation data of the server, we use the ASP Web page execution multiplication loop. The ASP Web page controls the service time of the server CPU through the execution multiplication loop. The multiplication loop has 1, 1 K, 10 K, 20 K, …, 190 K, 200 K times and both calculates and compares the system response time for every case.

The operating system uses the Windows Advanced Server for the measurement, the network is LAN (100Mbs/sec), the client uses the measurement software of the setup as a Webserver Stress Tool, and the server uses the software IIS 6.0. In the actual measurement we use 19 pcs to be carried as server system and a low rank server into the customer's port. We use the Webserver Stress Tool as a measurement tool in the client and make use of an ASP network in the server.

4.1 Analytical Process and Measurement Results

Fig. 10 shows the D networks, which use a total of 8 servers and have 5 sets of serial stages. D1 of the IP of the server is 192.168.0.100 with 192.168.0.11, and the two servers are serial, D2 of the IP of the server is 192.168.0.100 with 192.168.0.21, and the two servers are serial, D3 of the IP of the server is 192.168.0.100 with 192.168.0.12, 192.168.0.23, and 192.168.0.24, and the four servers are serial. D1-D5 have altogether 5 sets of serial stages. We carry out a parallel measurement with 5 sets of stages, with an ASP of 1 K, 10 K, 20 K, … , 190 K, 200 K. Each time the ASP includes 1 user, 5 users, 100 users, total with a measure of 441 times, at the rate of 20 minutes every time.

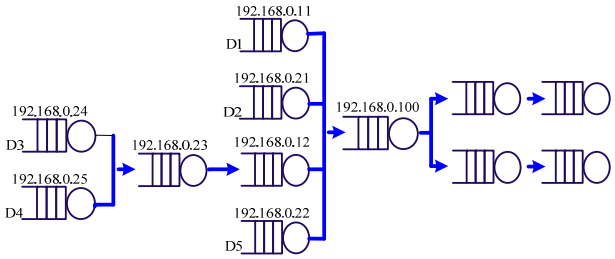

Fig. 10 IP addresses of the serial-parallel connection servers of D networks.

We use the partition algorithm. Step 1 transfers the queues of Fig. 10 into Fig. 11. By Steps 2 and 3 we transfer the original tree into the binary tree, then we use the serial and the parallel calculation method. Step 4 lists all serial servers while being also a physical measurement of the tree connection. Because the Webstress Tool sends out the service demands to the server, it measures the response time of the servers. Fig. 11 shows 4 steps, the load in the CPU being ASP=1 K, and the average response time of D1-D5 ranges over 12.04762ms, 13.38095ms, 29.42857ms, 29.7619ms, and 13.52381ms. From the diagram we know that D3 and D4 have four serial servers, consequently the response time is longer, while the CPU of the other S11 servers and the specifications of the memory are better than the specifications of S21 and S22, and therefore their response time is shorter. We measure the response time and find the average of the system. Figs. 12 and 13 show the computation results. We find the error margin between the simulation and the measurement to be 12.69%.

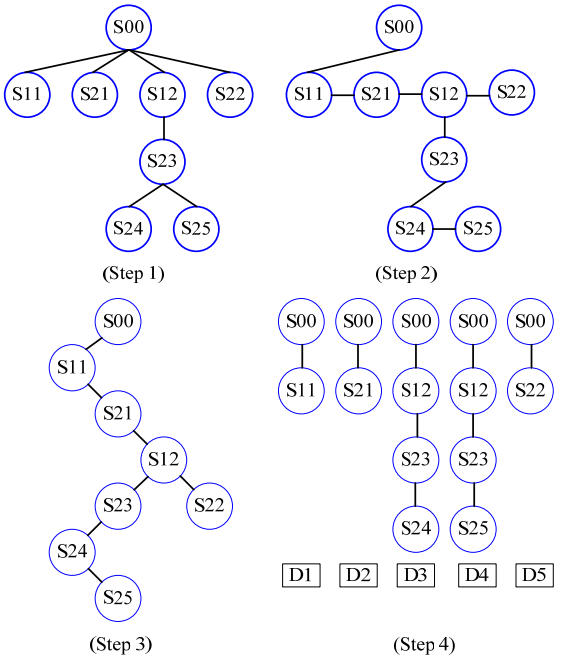

Fig. 11 Partition algorithm of D network.

We collect the measurement data and compare them with the computation results of the system response time from Eqs. (1) to (5). The computation value of the system response time of each serial connection is from 1-100 users and takes the average value of the serial connection. The measurement value of the system response time is from the measurement of 1-100 users' system response time. The error margin between computation by equation and by measurement is smaller when CPU loads are low; this is due to the basic amount of the service rate and the arrival rate. The Web packets do not stay in the buffer long, and the system response time is short. The error margin is bigger when CPU loads are high; this is because the service rate approaches an arrival rate and has already had more Web packets staying around the buffer, so the system response time increases. Fig. 12 shows the comparison of the system response times. Fig. 12 shows the D network, on a different CPU load, the results of measurement, calculation and simulation. With the values of Fig. 12 we found that the error between measurement, calculation, and simulation. Fig. 12 shows that we change the ASP multiplication loop increments by 10K loop. As for the error percentage of Fig. 12 shown in Fig. 13, we found that the simulated and the measured or calculated error is up to 30% of the difference. Comparing the calculation and measurement errors, we get a maximum of about 6%. We can see that the calculated and actual values are closer. Fig. 13 shows the comparison of the errors.

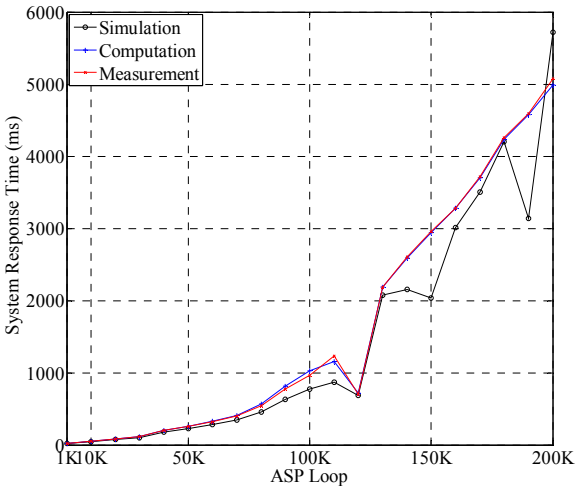

Fig. 12 System response times of simulation, computation and measurement of D networks.

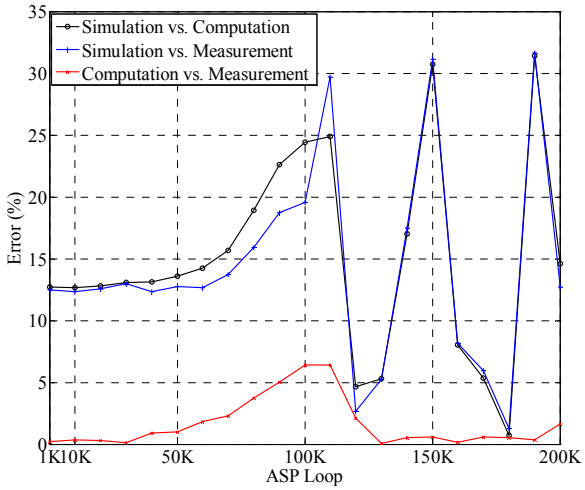

Fig. 13 Errors of simulation, computation and measurement of D networks.

Fig. 14 shows E networks, which use 12 servers and have 7 sets of serial loops. E1 of the IP of the server is 192.168.0.100 with 192.168.0.13, 192.168.0.101, 192.168.0.104, and the four servers are serial, E2 of the IP of the server is 192.168.0.100 with 192.168.0.13, 192.168.0.101, and 192.168.0.105, and the four servers are serial, E3 of the IP of the server is 192.168.0.100 with 192.168.0.13, 192.168.0.101, and 192.168.0.106, and the four servers are serial. E1-E7 have altogether 7 sets of serial stages. We carry out a parallel measurement with 7 sets of loops, and an ASP of 1 K, 10 K, 20 K, … , 190 K, 200 K. Each time the ASP includes 1 user, 5 users, … , 100 users, total with a measure of 441 times, at the rate of 20 minutes every time.

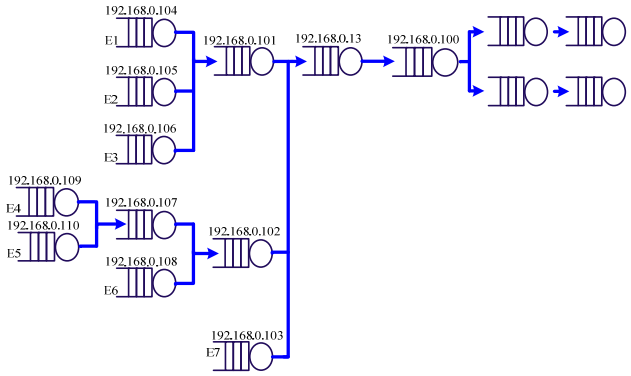

Fig. 14 IP addresses of the serial-parallel connection servers of E network.

Fig. 15 shows the 4 steps how the E networks are simplified by the partition algorithm. The CPU loading uses ASP=1K, the E1-E7 have a serial average response time of 24.95238ms, 24.80952ms, 25.28571ms, 31.38095ms, 31.33333ms, 26.14286ms, and 17.52381ms. From the diagram we know that E1-E3 and E6 have 4 serial servers, E4 and E5 have 5 serial servers, while E7 only has 3 serial servers. Therefore the response time of each is according to its serial amount, and S01-S10 all having the same specification, the response time is in a direct proportion to the serial response time. We take the serial time to find the average of the response time of the system. Figs. 16 and 17 show the medium computations for using this algorithm. The error margins between the simulation and the measurement are 8.32% and 0.04%.

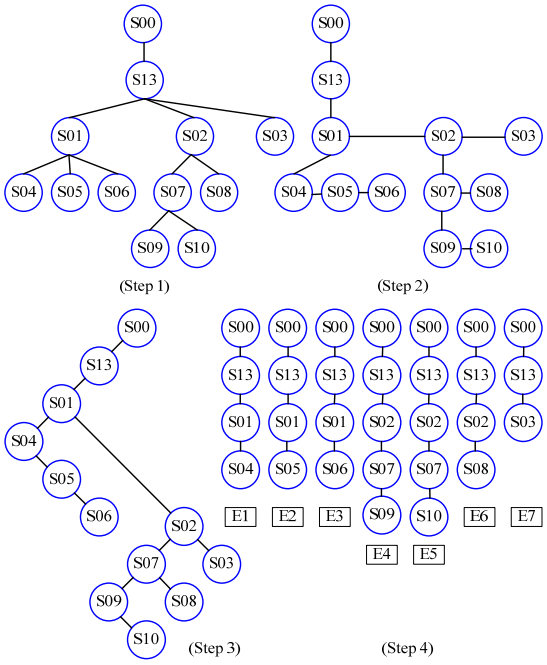

Fig. 15 E network partition.

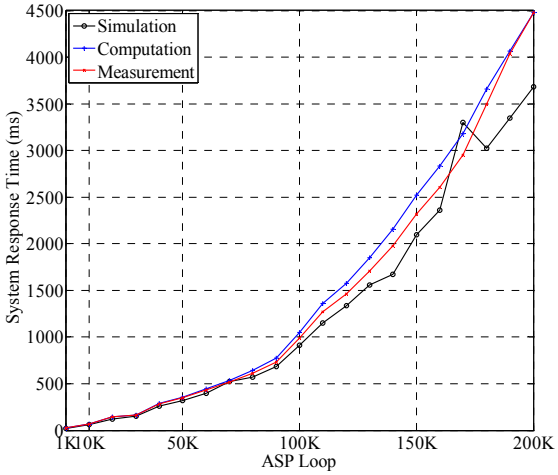

Fig. 16 The comparison of the system response times.

The three curves in the diagram show the computation, simulation and measurement results respectively. Fig. 16 shows the E network in the measurement with different CPU load, calculation and simulation of the comparison result. With a low blocking probability three curve error margins are smaller, but with a high blocking probability the error margin gradually increases. Fig. 16 shows the comparison of the system response times. Fig. 17 shows the comparison of the errors.

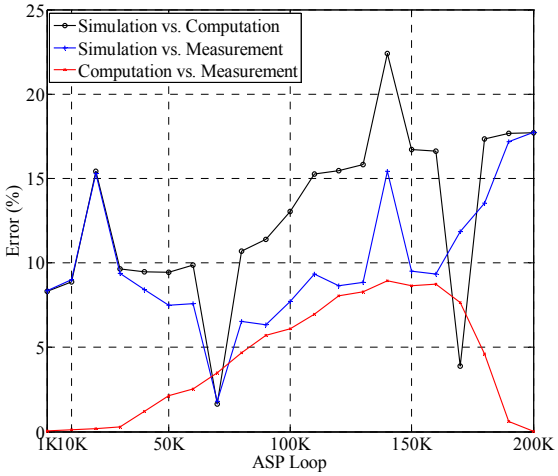

Fig. 17 Erors between simulation, computation and measurement of D networks when ASP=1K to 200K.

D1-D5 network and E1-E7 network merge and become an F network. Fig. 18 shows the system response time of the F networks, which use a total of 19 servers, and there are 12 sets of serial stages. F1 of the IP of the server is 192.168.0.100 with 192.168.0.11, and the two servers are serial, F2 of the IP of the server is 192.168.0.100, with 192.168.0.21, and the two servers are serial. F3 of the IP of the server is 192.168.0.100 with 192.168.0.12, 192.168.0.23, and 192.168.0.24, and the four servers are serial. F1-F12 have altogether 12 sets of serial stages. We carry out a parallel measurement with 12 sets of stages, the ASP being 1 K, 10 K, 20 K, … , 190 K, 200 K. Each time the ASP includes 1 user, 5 users, … , 100 users, total with a measure of 441 times, at the rate of 20 minutes every time.

We collect measurement data to compare with the computation results of the value of the system response time from Eqs. (1) to (5). The computed value of the system response time of each serial connection is from 1-100 users and takes the average value of the serial connection again. The value of the system response time of the measurement is from the system response time of the formula of 1-100 users. The error margin between computing by equation and by measurement is smaller when CPU loads are low; this is due to the service rate and the arrival rate. The Web packet will not stay in the buffer, and the system response time is short. The error margin is bigger when CPU loads are high; this is because the service rate approaches an arrival rate and has already had a Web packet staying around the buffer, so the system response time increases. Fig. 18 shows the comparison, Fig. 19 shows the comparison of the errors.

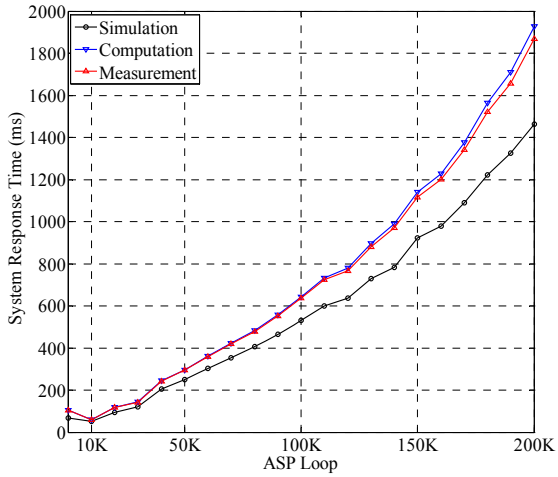

Fig. 18 System response times of simulation, computation and measurement when ASP=1K to 200K.

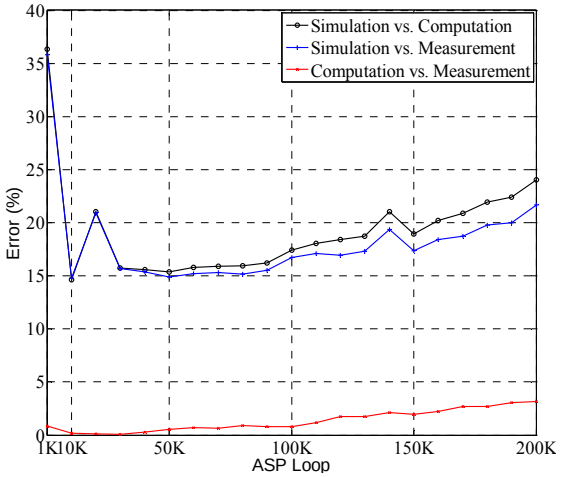

Fig. 19 System response times of simulation, computation and measurement when ASP=1K to 200K.

Our experiment uses 19 servers; in addition there is a server that imitates 1-100 users, so that we have at the same time altogether 12 sets of serial stages that are parallel. Fig. 18 shows three curves, respectively from the equation, the simulation and the measurement. When the parallel number increases, the system response time is ASP=200 K, a high blocking probability, obviously from the 1463.196ms of the F network. In Fig. 19, three curves show the simulation, computation and measurement system response times. Their average error margins are 19.27%, 18.20% and 1.37% respectively. The difference between simulation, computation, measurement of each subnetwork is shown in Table 2.

Table 2 Difference between simulation, computation and measurement of each subnetwork

	D Network	E Network	F Network
Difference between simulation and computation	15.08%	12.70%	19.27%
Difference between simulation and measurement	14.38%	9.97%	18.20%
Difference between computation and measurement	1.68%	4.24%	1.37%

5 Conclusion

We establish a parallel tree-like simple equation of equivalent queues to calculate the response time and the service rate of the system. Because of the tree characteristics of the Intranet, the network connectivity method can simply be classified as serial, parallel and tree connected. These characteristics make use of the algorithm of the binary tree in the Graph Theory and create the equivalent queuing model. The left subtree creates the equivalent serial equation, and the right subtree creates an equivalent parallel equation. The father and son are two groups. If we visit first the left subtree and then the right subtree and use a serial-parallel equivalent tree equation, we make a simplification. The sequence of the simplification is as follows. First, the leaves of the left side have the initialization, and the leaves of the right side go one after another toward the direction of the root. Second, we create the simplification procedure of the equivalent model.

With a low blocking probability, the D network performance analysis shows that the error margin of its service rate is limited to 6.43%. The campus F network is analyzed as having a low blocking probability; the error margin of its service rate is limited to 3.16%.

When the Web service is busy we use the parallel connection server to reduce the system response time. When user demand is greater than the system capability we have a medium high blocking probability. By using parallel connection servers we significantly reduce the system response time.

References:

[1] Ying-Wen Bai, Chia-Yu Chen, and Yu-Nien Yang, A Two-Pass Web Document Allocation Method for Load Balance in Multiple Grouping of a Web Cluster System, *ICON'04, 12th IEEE International Conference on Networks,* Vol. 1, 2004, pp. 177-181.

[2] Nathan L. Binkert, Lisa R. Hsu, and Ali G. Saidi, Performance Analysis of System Overheads in TCP/IP Workloads, *Proceedings of the 14th International Conference on Parallel Architectures and compilation Techniques,* 2005, pp.218-228.

[3] Zhen Zhao, Bryan Willman, Steven Weber and Jaudelice C.de Oliveira, Performance analysis of a parallel link network with preemption, *Proceedings of International Conference on Information Sciences and Systems,* 2006, pp.271-276.

[4] Ying-Wen Bai and Yu-Nien Yang, An Approximate Performance Analysis and Measurement of the Equivalent Model of Parallel Queues for a Web Cluster with a Low Rejection, *Proceedings of the 14th IEEE International Conference,* 2006, pp.1-6.

[5] Chung-Ping Chen, Ying-Wen Bai and Yin-Sheng Lee, Approximate Analysis and Measurement of Equivalent Model for Tree Connection of Web Server Systems, *Proceedings of the 19th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2007),* 2007, pp.91-96.

[6] Keith W. Ross, and James F. Kurose, *Computer Networking: A Top-Down Approach Featuring the Internet,* Publishing Reading, MA. Addison-Wesley, 2001.

[7] Yuan-Horng Lin, Min-Ning Yu and Berlin Wu, Fuzzy Classification Analysis of Rules Usage on Probability Reasoning Test with Multiple Raw Rule Score, *Proceedings of the 2nd WSEAS/IASME International Conference on Educational Technologies,* Bucharest, Romania, 2006, pp.54-59.

[8] Xiang Zhao, Miling Talpallikar and Sijun Zhang, Dynamic Load Balancing for Parallel Mesh Adaptation, *Proceedings of the 10th WSEAS International Conference on Applied Mathematics*, 2006, pp. 51-56.

[9] Rohan De Silva, Performance Enhancement of Flow Control in 10GbE WANs, *Proceedings of WSEAS Transactions on Computers.* Vol. 3, no. 6, 2004, pp. 2011-2016.

[10] D. C. Vasiliadis, G. E. Rizos, E. Stergiou, and S. V. Margariti, A trusted Network Model using the Lightweight Directory Access Protocol, *Proceedings of the 7th WSEAS International Conference on Applied Informatics and Communications,* Athens, Greece, 2007, pp. 252-256.

[11] Annop Monsakul, Pisit Charnkeitkong, M-MENTOR: A Design Algorithm for IP Networks with Mixed Traffic, *Proceedings of WSEAS Transactions on Communications,* Issue 10, Vol. 8, 2009, pp.1086-1095.

[12] P. M. Papazoglou, D. A. Karras, R. C. Papademetriou, On a New Generation of Event Scheduling Algorithms and Evaluation Techniques for Efficient Simulation Modelling of Large Scale Cellular Networks Bandwidth Management Based on Multitasking Theory, *Proceedings of WSEAS Transactions on Communications*, Issue 10, Vol. 7, 2008, pp.1024-1034

[13] Osamah Badarneh, Michel Kadoch, Ahamed Elhakeem, QoS Multilayered Multicast Routing Protocol for Video Transmission in Heterogeneous Wireless Ad Hoc Networks, *Proceedings of WSEAS Transactions on Computers,* Issue 6, Vol. 7, 2008, pp.680-693

[14] Tzay-Farn Shih, Chao-Cheng Shih, Chin-Ling Chen, Location-Based Multicast Routing Protocol for Mobile Ad Hoc Networks, *Proceedings of WSEAS Transactions on Computers,* Issue 8, Vol. 7, 2008, pp.1270-1279

[15] Sosa-Herrera Antonio, Rodriguez-Romo Suemi, Chin-Ling Chen, Variable Precision Distance Search for Random Fractal Cluster Simulations, *Proceedings of WSEAS Transactions on Computers*, Issue 8, Vol. 8, 2009, pp.1245-1255.

[16] M. Vijayakumar, S. Prakash, R. M. S. Parvathi, Inter Cluster Distance Management Model with Optimal Centroid Estimation for K-Means Clustering Algorithm, *Proceedings of WSEAS Transactions on Communications*, Issue 6, Vol. 10, 2011, pp.182-191.

[17] Nancy P. Lin, Chung-I Chang, Hao-En Chueh, Hung-Jen Chen, Wei-Hua Hao, A Deflected Grid-based Algorithm for Clustering Analysis, *Proceedings of WSEAS Transactions on Computers,* Issue 3, Vol. 7, 2008, pp.125-132.

[18] Wei-Qing Sun, Cheng-Min Wang, Tie-Yan Zhang, Yan Zhang, Transaction-item Association Matrix-Based Frequent Pattern Network Mining Algorithm in Large-scale Transaction Database, *Proceedings of WSEAS Transactions on Computers,* Issue 8, Vol. 8, 2009, pp.1327-1336.

[19] Boutkhil Sidaoui, Kaddour Sadouni, Efficient Binary Tree Multiclass SVM Using Genetic Algorithms for Vowels Recognition, *Proceedings of WSEAS Transactions on Computers,* Issue 1, Vol. 11, 2012, pp.11-18.

[20] Cornel Balint, Georgeta Budura, Adrian Budura, Eugen Marza, Dimensioning Rules Regarding Radio Resources in GSM/GPRS Networks, *Proceedings of WSEAS Transactions on Communications*, Issue 8, Vol. 8, 2009, pp.822-832.

[21] K. Ramesh Kumar, R. S. D. Wahida Banu, A Novel QoS Aware RWA with Dedicated Path Protection Consideration for All Optical Networks, *Proceedings of WSEAS Transactions on Communications,* Issue 7, Vol. 11, 2012, pp.251-261.

[22] Santos Kumar Das, Dhanya V. V., Sarat Kumar Patra, QoS Based OVPN Connection Set Up and Performance Analysis, *Proceedings of WSEAS Transactions on Communications*, Issue 7, Vol. 11, 2012, pp.275-286.

[23] Gunter Bolch, Stefan Greiner, Hermann de Meer, and Kishor S. Trivedi, *Queueing*

Networks and Markov Chains, Publishing Whiley-Interscience, 1998.

[24] Chung-Ping Chen and Ying-Wen Bai, Performance Measurement and Queueing Analysis at Medium-High Blocking Probability of Parallel Connection Servers with Identical Service Rates, *Proceedings of WSEAS Transactions on Communications,* Issue 12, Vol. 8, 2009, pp.1253-1262.