Dynamic Traffic Profiling for Efficient Link Bandwidth Utilization in QoS Routing

K.Kalapriya[1], B.R.Raghucharan[1], Abhijit.M.Lele[2], S.K.Nandy[1]

[1] CAD Lab, Indian Institute of Science.
[2] ST-IISc Research Lab
{kalapriya, raghu, abhijit, nandy @ cadl.iisc.ernet.in}

Abstract. Traffic Engineering has been the prime concern for Internet Service Providers (ISP's), with the main focus being minimization of over-utilization of network capacity even though additional capacity is available which is under-utilized. Furthermore, requirements of timely delivery of digitized audio-visual information raises a new challenge of finding a path meeting these requirements. This paper addresses the issue of (a) distributing load to achieve global efficiency in resource utilization. (b) Finding a path satisfying the real time requirements of delay and bandwidth requested by the applications. In this paper we do a critical study of the link utilization that varies over time and determine the time interval during which the link occupancy remains constant across days. This information helps in pre-determining link utilization that is useful in balancing load in the network. Finally, we run simulations that use a dynamic time interval for profiling traffic and show improvement in terms number of calls admitted/blocked.

1.Introduction

Streaming media applications like distance education, teleconferencing, telemedicine, teleshopping and ubiquitous computing impose stringent real time performance guarantees. Typically such performance guarantees are specified in terms of delay, jitter, bandwidth and cost. These requirements often have conflicting effects on overall application performance. Quality of Service (QoS) is a set of parameters that qualify the application requirements in terms of these performance guarantees on the network. A typical network environment comprises nodes and links. Nodes represent routers while the links define connectivity between routers. These routers are broadly classified into edge routers and core routers. Edge routers are those that route packets between a self-contained network and other outside networks along a network backbone. A core router is a router that forwards packets to computer hosts within a network, but not between networks. The real bottleneck for performance comes at these core routers where traffic arrival rates are much higher than the service rates.

This work is partially supported by a research grant from STMicroelectronics.

Traffic engineering is concerned with distribution of traffic across the network to avoid such bottlenecks and the use of network resources efficiently.

Traffic engineering along with guaranteed service is the prime concern for the Internet Service Providers. Traffic Engineering is done by the routing algorithm that uses the current network statistics to efficiently route the traffic on under utilized links so as to avoid over utilization of some other links. Additionally, routing algorithm has to select a path that satisfies QoS demands of applications. The QoS requirements can be specified by multiple criterias'. Multi constraint QoS routing is defined as finding a qualified path, that meets multiple criteria of QoS requirement. Finding a Multi-constrained path to satisfy bandwidth and delay requirement is a NP hard problem [1]. Many heuristic algorithms have been proposed for solving this multi-constrained problem [4] [5] [6]. Though all of these algorithms solve the multi-constrained problem in polynomial time, they do not consider current network statistics such as current link utilization. We believe that, if *a priori* knowledge of link utilization forms an input to the path selection, network resource utilization can be increased.

In this paper, we propose a novel scheme for QoS aware multi-constrained routing which is an improvement over profile based routing algorithm [3], by finding optimal time interval(s) for traffic profiling and satisfying the bandwidth-delay constraints. This uses a priori knowledge of the link utilization for efficiently allocating resources for future demands on that link. The routing algorithm has two phases. The first phase comprises traffic profiling to determine link occupancy. This link occupancy information is an input to the multi-commodity flow problem for resource reservation. The second phase is the online phase, which uses this resource reservation and determines routes satisfying QoS demands of the applications.

The rest of the paper is organized as follows. Section 2 gives the related work. Section 3 states the problem and section 4 discusses traffic profiling for link occupancy prediction. Section 5 discusses routing algorithm and performance analysis. Section 6 presents the results and Section 7 concludes the paper.

0-7803-8114-9/03/$17.00 ©2003 IEEE.

2 Related Work

An extensive survey on QoS routing can be found in [8]. Among the proposed schemes, multi-constrained routing schemes are relevant to this paper. The authors in [9] propose distributed routing algorithm to find paths that satisfy end-to-end delay constraint while minimizing the cost. Although the scheme considers two constraints delay and bandwidth, it does not provide complete solution for the problem because the cost metric is not bounded.

Various QoS routing algorithms [10][11] are proposed that are based on weighted fair scheduling schemes. Ma[10] showed that delay, delay-jitter and buffer space can be expressed in terms of bandwidth if weighted fair queue scheduling is used. However, in high-speed networks propagation delay plays a significant role and cannot be mapped to bandwidth and therefore the scheme in [10] fails when propagation delay is considered. In this paper we overcome this by computing the latency between two nodes as a summation of propagation delay and Queuing delay as discussed in section 5.

Jaffe [1] proposed a distributed algorithm that solves 2 - constrained problems with a time complexity of $O(|N|^5 b \log(N b))$, where b is the largest number of the weights. This algorithm is pseudo-polynomial where the execution time depends on the value of the weights and the size of the network. Widyono [5] proposed exhaustive search on the QoS guaranteeing paths in exponential time. Yuan [5] studied the limited granularity heuristic and limited path heuristic for 2-constrained problems. Chen [4] proposed a heuristic algorithm that effectively solves 2- constrained routing problems. All these algorithms do not take advantage of the link utilization for effectively managing the network resources.

In [7], Kodailam and Lakshman proposed Minimum Interference Routing Algorithm (MIRA), which takes ingress-egress node pairs and identifies the link that is bottleneck for future demands as critical links. Though traffic engineering is well addressed here, the scheme fails to provide a feasible path under conditions expressed in [3]. Waldvogel *et al* [3] proposed an improvement over MIRA called *profile based routing algorithm.* In their algorithm, they use traffic profile to solve the multi-commodity flow problem which pre-allocates link bandwidth for each source destination pair based on the aggregate of 24 hours traffic as a good estimate for future demands. In this paper we argue that the optimal time to profile traffic as an estimate for future demands varies largely with time and thus allocation is dependent on the link occupancy over time. We estimate this time interval for which traffic profiling will provide a good estimate of the future demands.

3 Problem Statement

The network is modeled as a directed graph. We assume that nodes within the network know the complete topology. Consider the network as G=(V, E), where V is the set of nodes and E is the set of directed links among the nodes in V. We use the notation (D, B) to denote QoS parameters Delay and Bandwidth on each E. A path from V_0 to V_n is denoted by (V_0->V_1->V_2...... V_n) where the directed link $(V_i, V_{i+1}) \in E$ for $0<i<n$. Let the link (V_i, V_j) be represented by $E_{i->j}$ and delay–bandwidth on this link be denoted by $(D, B)_{i->j}$, $Di_{->j}$, $B_{i->j}$ being the delay and bandwidth of (V_i,V_j) respectively. The delay of the path from V_0 to V_n is an additive constraint and is expressed as $\sum(D_{i->i+1})$. The Bandwidth is a concave constraint and is expressed as minimum $\{B_{i->i+1}\} \forall i$ on the path V_o to V_n. A subset of nodes are assumed to be ingress-egress nodes and is denoted by (IG, EG) such that IG, EG ε V. A request is identified by (id, IG_i, EG_i, BW_i, DE_i), where id is the request identifier, IG_i and EG_i denotes the Ingress and the Egress nodes, BW_i is the Bandwidth and DE_i is the Delay requirement for that request. Traffic profile is a rough indication of the amount of traffic that can be expected between a Ingress-Egress pair. Traffic profile is defined by (Cid, IG_i, EG_i, BA_i) where Cid is the traffic class, IG_i, EG_i are ingress and Egress nodes, and BA_i is the aggregate bandwidth for the Ingress-Egress pair. We assign $B_{i->j} = BA_i \ \forall \ E_{i->j}$ on the path from IG_i to EG_i. The problem of Multi-constraint path finding for a given (id, IG_i, EG_i, BW_i, DE_i) request can be stated as determining the path that satisfy the conditions: $DE_i < \sum D_{i->j} \ \forall \ E_{i->j}$ along that path and $BW_i < B_{i->j} \ \forall \ E_{i->j}$ along that path.

4 Traffic Profiling

In this section we carry out an analysis of the traffic[1] traces to study the optimal time period over which the link utilization remains approximately same across days for a particular link. Taking into account the dynamic nature of Internet traffic, it is our belief that time duration of 24 hours for profiling traffic as reported in [3] is not optimal..

4.1 Data Analysis

The traffic analysis is carried out for traffic traces[1] collected over 30 consecutive days at the edge link between the local gateway and the backbone at time intervals of thirty minutes. Fig 1 plots the link utilization for Day-1 through Day-3. Table 1 gives the Average

[1] Data Traces were collected at the VSNL-IISc link for 30 days, but the results are limited for 3 days.

Link Utilization (ALU) over the 24 hours across 3 days. Although the average link utilizations shows similar pattern across all 3 days, a close observation of Fig 1 indicates that the link utilization varies over time. Tables 2, 3 and 4 give the average link utilization and variance (VAR) for time intervals varying from 3 hours to 12 hours for Day-1 to Day-3. Analysis of the data traces from Day-4 to Day-30 shows a similar trend. The variances are calculated using the expression, Variance= [Σ ((val)$_i$ - (Avg)) / N] where (Val)$_i$ is value of link occupancy at i[th] time and Avg= [Σ (val)$_i$ / N] where N= Number of samples taken in that time interval. From Tables 2, 3 and 4 it is observed that there is a considerable variance in the average link utilization across different time durations of the day. For example, for Day-1, the variance for average link utilization for first 3 hours is 24.17% and for first 6 hours is 19.02% and for first 12 hours, it is 28.77%. The same pattern repeats, with some exceptions across all the 6 days. Taking into account the variance in the average link utilization across different times of the day and comparing Table 1 with Tables 2, 3 and 4, it is observed that using the average link utilization computed over a duration of 24 hours as claimed in [3] for Day-1 is not a correct metric to estimate the link utilization for Day-2. Without loss of generality the same argument can be extended for other days.

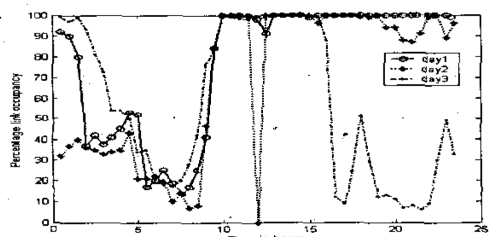


Fig. 1. Link Occupancy for 3 continuous days.

From Table 2 assuming an optimal difference in the variances across consecutive days and comparing the variance across these days we find that the average difference in variance that falls within the optimal value is about (2.8%) for 9 out of 12 time intervals. From Table 3, comparing the variance across consecutive days, the average difference in variance that falls within the optimum value is about 3.95 for 3 out of 8 time intervals. From Table 3, we find that the difference in variance across consecutive days is 4.89 for 3 out of 4 time intervals. Comparing these results we find that 3 and 12 hour duration is an optimal time interval for traffic profiling that gives good estimate of future demands. In order to keep the overhead of the routing algorithm minimum we choose the maximum time interval over which the difference in variance is minimum. Hence we choose 12 hours to be a optimal time interval for traffic profile for the above data traces analyzed.

5 Routing Algorithm

This section describes the routing algorithm in detail. The routing algorithm consists of two phases, offline and online phase. Traffic profile information records the expected flow between pairs of ingress-egress routers, and represents aggregated demand profile between ingress-egress pairs and is denoted by (Cid, IG_i, EG_i, BA_i). In the Offline phase we estimate the window (time interval) $\triangle T$ for traffic profile Cid (identified by a Ingress-Egress pair), duration of which is used for profiling traffic that provides a good estimate of the future demands (BA_i). We use the previous history of the traffic flow between each ingress-egress pair to calculate $\triangle T$. When no *a priori* profile information is available, it can be assumed that current demand is an indication of future demands. The profile collected during the time $\triangle T$ between the (IG_i, EG_i) pair is used to calculate the aggregate bandwidth BA_i, required for that traffic class. This aggregate bandwidth forms the input to the multi-commodity flow problem to pre-allocate bandwidth on all the links that are on the route between IG_i EG_i. In the online phase we determine the route for requests identified by (id, IG_i, EG_i, BW_i, DE_i). Each request can be uniquely mapped to a traffic profile. The request arrive online, one at a time, and the algorithm does not know anything about individual future requests, their bandwidth requirements or their time of arrival.

5.1 Offline phase

The offline phase uses the network G=(V, E) with bandwidth $B_{i\text{-}>j}$ for each edge ε $E_{i\text{-}>j}$. We estimate $\triangle T$ as given in section 4.We treat each traffic class given by (Cid, IG_i, EG_i, BA_i) as a separate commodity. Suppose there are N commodities, the goal is to find out routes in the network to send as much of each commodity as possible from its ingress to the egress node. Between every ingress - egress pairs we have an excess edge with infinite capacity and high cost (∞). Otherwise cost ($E_{i\text{-}>j}$) =1. This allows as much of the feasible flow as possible to go through original network edges. Let $x_k(e)$ denote the amount of commodity k that is routed through edge $E_{i\text{-}>j}$. Then the multi - commodity problem can be solved for graph G' obtained after the addition of excess edge for each ingress - egress pair to G by linear constraint model and can be expressed as minimize ($\sum$ (cost ($E_{i\text{-}>j}$) $\sum$ x_k(e)) $\forall$ i,j = 0 to n satisfying the following constraints

Day	Day-1	Day-2	Day-3	Day-4	Day-5	Day-6
Link Utilization (%)	69.09	96.70	80.85	57.32	68.28	77.00

Table 1. Average Link Utilization over 24 hours across 6 days.

Time (hrs)	ALU-Day-1	VAR-Day-1	ALU-Day-2	VAR-Day-2	ALU-Day-3	VAR-Day-3
0-3	63.17	24.17	35.5	2.0	90.33	8.89
3-6	38.00	13.10	29.33	8.0	41.0	11.67
6-9	23.50	6.83	17.33	10.11	34.0	17.00
9-12	97.00	4.33	80.67	27.11	96.33	4.45
12-15	98.33	2.45	100	0.0	99.5	0.67
15-18	100.00	0.00	99.16	1.11	47.17	32.17
18-21	100.00	0.00	93.67	4.11	13.3	5.2
21-24	99.80	0.32	95	4.0	25.4	14.32

Table 3. Average Link Utilization and variance for 3 hours across 3 days.

1. Capacity constraints are satisfied for all edges E i.e., $\sum x_k(E_{i\text{->}j}) < B_{i\text{->}j}$
2. The flow for each commodity is conserved at all nodes, except at the corresponding ingress and egress nodes, and
3. The amount of commodity k reaching its destination EG_i is BA_i. Output of this algorithm will be link capacities that will be used for pre-allocation for various traffic classes and used by the online phase

5.2 Online phase

The input phase of this algorithm takes the input network G=(V,E), and the set of requests identified by the tuple $(id, I_i, E_i, BW_i, DE_i)$. The residual bandwidth $res_{i\text{->}j}$ for each link is maintained after allowing each request. Initially the residual bandwidth is set to the pre-allocation capacities obtained for each commodity on the links as the result of Offline phase. After each request $res_{i\text{->}j} = B_{i\text{->}j} - BW_i$. The input to the algorithm is graph representing the network G=(V, E) and pre-allocated link bandwidths for each commodity on the links. The algorithm returns a path from IG to EG such that edges along this path satisfy the capacity constraint $res(E_{i\text{->}j}) > BW$ and $\sum D_i \ \forall i$ on the physical path is less than DE_i

Time (hrs)	ALU-Day-1	VAR-Day-1	ALU-Day-2	VAR-Day-2	ALU-Day-3	VAR-Day-3
0-6	50.59	19.02	32.42	5.61	65.67	24.67
6-12	60.25	36.75	49.0	39.83	65.17	33.14
12-18	99.17	1.39	99.58	0.70	73.34	32.89
18-24	99.91	0.16	94.27	4.11	18.8	11.95

Table 2. Average Link Utilization and variance for 6 hours across 3 days.

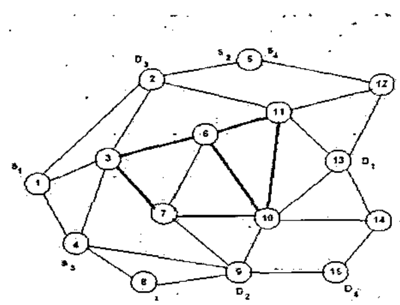
Time hrs	ALU-Day-1	VAR-Day-1	ALU-Day-2	VAR-Day-2	ALU-Day-3	VAR-Day-3
0-12	55.42	28.77	40.71	24.0	65.42	28.89
12-24	99.52	0.83	97.04	3.59	47.26	35.77

Table 4. Average Link Utilization and variance for 12 hours across 3 days.

Operation

1. For all requests (i) BW=BWi, DE=DEi
2. Delete all edges from G that doesn't satisfy the constraint BW > res(e), these edges don't satisfy the bandwidth constraints.
3. From the sub-graph obtained after deleting the edges that do not satisfy the bandwidth constraint, find a path P that has maximum number of hops and satisfying the delay constraint $D_{i\text{->}j} < DE \ \ \forall \ i,j$ on the path by modifying Dijkstra's algorithm.
4. For each edge in the path P, decrease the residual capacity by BW. If the B of that edge becomes negative, the link is marked as unusable until some application frees the resource.
5. Route the request i along the path P.
6. If no route is found that satisfy the constraints, add BW to the excess edge of the IG_i, EGi for the path returned by the Dijkstra's algorithm to keep track of the extra Bandwidth resource required.

Step-3 preserves the shortest path for more sensitive to delay applications. Since the bandwidth is pre-allocated there is no loss of network resources by taking a longer path.



Network Characteristics:
Core links (dark lines) represents OC-48 (4800 mbps)
Edge links (gray lines) represents OC-12 (1200 mbps)

Fig 2a. Sample Network 1 used in Simulation

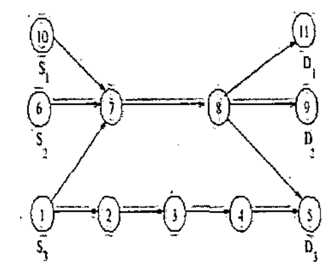


Fig2b. Sample Network 2 used Simulation

Total Latency between the two nodes is given by $L_{i->j}$ = Pd +Td + Qd, where Pd is propagation delay =Link distance /speed of light in fiber, Td is transmission delay=Number of bits in packet (worst case)/link BW ,Qd is (queue/process) delay = number of nodes * processing delay +sum of queuing delays (QD).

For obtaining QD we model the network as single server queuing system where the packets arrive according to the standard distributions (with the arrival rate λ) that simulates the arrival of appropriate applications and service time for each application follow a general distribution. The waiting time for such a model is given by $w= \lambda * \mu / 2(1-p)$; where w is the expected packet waiting time in queue , μ is the arrival rate and p = rate of arrival * Average service time. When the service times are identical for all packets we have w=p/2(service rate)(1-p).

6 Simulation Results

Figure 2 gives the networks used for the purpose of the simulation. We have used the same network (Figure 2a.) that was used in Minimum Interference Routing Algorithm for comparisons (Network 1). Network shown in Figure 2b is chosen to depict the Concentrator-Distributor network (Network 2). The Source-Destination pairs are considered as Ingress-Egress pairs. For brevity and simplicity, we model the application arrival to follow the poisson model. Without loss of generality this can be extended to any other distribution. The simulation results are plotted in terms of number of admitted (blocked) calls. The results are given for 24 hour traffic. Fig 3 shows a plot of percentage of calls accepted when unlimited bandwidth was used for routing. We observe that although 99% of calls are admitted into the network due to unlimited bandwidth, it falls down to nearly 40% when delay constraints are imposed by the applications. We estimate the window size (time interval) for which the aggregate profile of previous day traffic is used as a good predictor of future demands. Simulations were carried over an interval of 24 hours with the window size varying from 30 min to 24 hours. Table 5 gives the result in terms of percentage of calls accepted when link capacity were pre-allocated using the link occupancy aggregate collected during the previous day for the time interval $\triangle$T and satisfying the bandwidth-delay constraints requested by the applications.

In Table 5, a window size of 24 hrs represents the time used for finding the aggregate bandwidth in profile based routing algorithm [3]. We find that using window sizes less than 24 hours generally gives better performance in terms of calls admitted for both networks. We also find that the percentage of calls admitted varies for different window sizes. The values clearly indicate that Windows Sizes of 30 mins, 1hr, 1hr 30 mins, 5 hrs perform better than the other window sizes for network 1 and Window Sizes 30 mins, 1hr, 1hr 30 mins, 2 hrs and 7 hrs perform better than other window sizes for network 2. This window size is dependent on the link statistics. This window size has to be estimated for links dynamically. We run simulations using values of the optimal window sizes obtained from the Table 5 (varying time slots) and compare the performance in terms of calls accepted, with the profile-based algorithm. Fig 4a and Fig 4b gives a plot of number of calls accepted versus time, satisfying bandwidth and delay constraints for network 1 and network 2 respectively. The results indicate an improvement in performance over profile based routing algorithm, in terms of number of calls accepted when time intervals are 30 minutes and 5 hours for network 1 and 2 hours and 7 hours for network 2.

From the results, we conclude that optimal window size estimation plays an important role in improving the number of calls admitted into the network. The window size is dependent on the network used and should be dynamically calculated for every network..

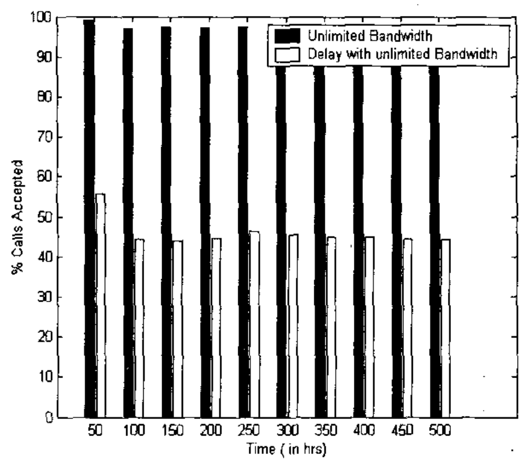


Fig. 3. Percentage of Calls Accepted Vs Time (Network 1)

ΔT (hrs : mins)	Network 1 % Calls Accepted	Network 2 % Calls Accepted
00:30	62.4	68.4
1:00	60.4	67.4
1:30	58.56	56.63
2:00	50.85	57.85
3:00	56.56	55.56
4:00	50.85	54.5
5:00	56.6	53.22
6:00	54.71	53.71
7:00	52.79	54.16
8:00	50.85	53.05
9:00	48.93	52.93
10:00	47.01	52.08
12:00	45.08	51.68
14:00	43.5	51.5
24:00	**40.1**	**50.2**

Table 5. Percentage of calls accepted for various window sizes and satisfying the Bandwidth-Delay criteria.

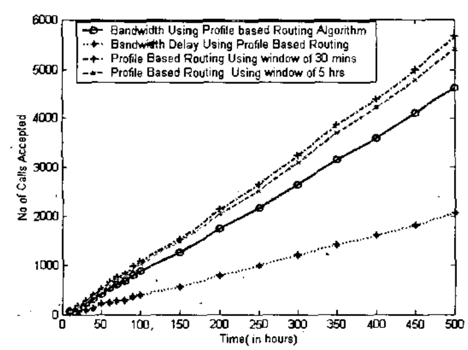


Fig 4a. Network 1- Number of calls accepted vs. time

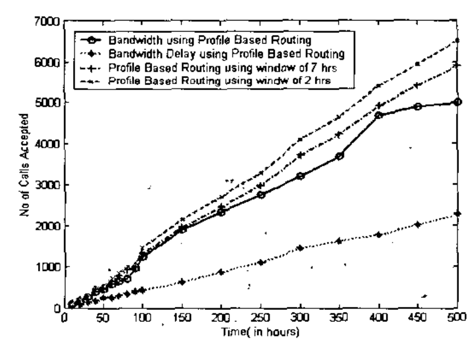


Fig 4b. Network 2- Number of calls accepted vs. time

Fig 5a and Fig 5b gives a measure of number of calls rejected with window size equal to 5 hours and 7 hours for network 1 and network 2 respectively. The results show a 10-16% reduction in call rejections. Based on the simulations carried out in this paper, we infer that the window size for allocating bandwidth plays a very important role in improving the network bandwidth utilization and it is not equal to 24 hours.

The link behavior has to be dynamically studied for every network to determine its aggregate occupancy and use it for pre-allocation to enhance network resource utilization. This point has been emphasized by the simulation results (Table 5) from the fact that optimal window sizes for the two networks are not the same. It varies with time, network topology and utilization pattern and therefore the utilization of the network bandwidth has to be studied for each link and based on the usage pattern the optimal window size has to be estimated.

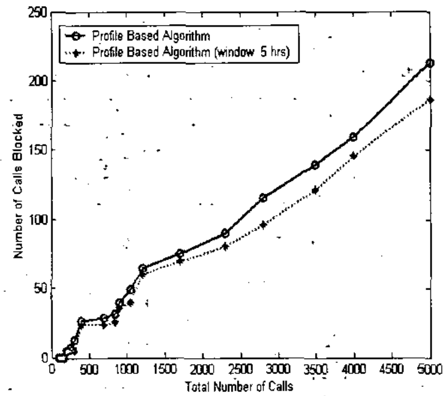


Fig 5 a. Network 1-Number of Calls Blocked for Window size of 5 hours

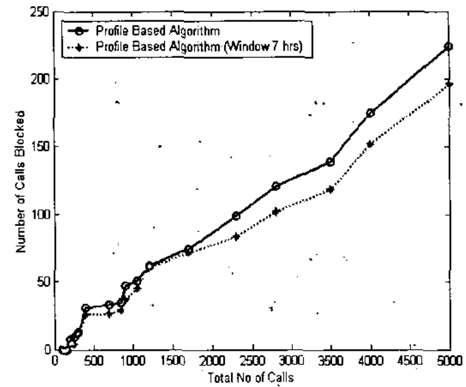


Fig 5 a. Network 2-Number of Calls Blocked for Window size of 7 hours

For the sake of completion, we derive a mathematical model to determine the optimum time interval for estimation of future demands. The problem can be modeled as two random variable(x = Time, y= Bandwidth) approximately equated to two variable normalized Gaussian distributions.

$$P(x) = A\ \exp[\ -1/2\ \sigma^2 (x^2 + y^2)\ ] \qquad \textbf{(1)}$$

Where $A = -1/2\ \sigma^2$
σ = Variance of the sample data

Given 'n' days samples (t_i , b_i) where ti is the time of the i^{th} day and b_i is the bandwidth i^{th} day in the time t_i (discrete domain), the problem scales down to determine the maximum duration of the interval during which the pattern (aggregate bandwidth) remains constant, i.e., bandwidth during this time interval remains as a good estimate of the future demands.

We begin to find out "$\triangle$T" the time interval starting with largest value less than 24 (primarily a factor of 24, assuming the bandwidth for that duration for all days follows the Gaussian distribution) and satisfying the following condition:

$$\sigma^2 = \frac{\Sigma(X - \mu)^2}{N} \leq \epsilon \qquad \textbf{(2)}$$

for given $\triangle$T over the whole day

Where '$\in$' is arbitrarily small value and μ is the mean and σ^2 is the variance. We repeat this process with varying values of $\triangle$T until we find a "$\triangle$T" that satisfies the following condition.

$$\text{Maximum}\ \{\triangle T\} \ni \sigma^2 \leq \lambda \qquad (3)$$

where λ is the pre-fixed threshold.

This fixes the time interval during which the bandwidth remains constant over days. We can derive a good estimate of the bandwidth given this time duration along with the sample data (ti, bi) by approximating the problem to the maximum-likelihood estimate. Suppose we treat B as containing n samples b1.....bn of bandwidth over days during that time interval. Then, we have

$$P(B|\theta) = \prod_{k=1}^{k=n} P(X_k | \theta) = F(\theta) \qquad \textbf{(4)}$$

Where 'θ' is the parameter vector. P (B|θ) is the likelihood of 'θ' with respect to the given sample. This maximum estimate is the arithmetic average of the sample data if it follows Gaussian distribution. The results obtained by this model are in compliance with the results obtained in section 4.

7 Conclusions

This paper considers the issue of QoS routing with respect to multiple constraints and traffic profiling for efficient utilization of link bandwidth. Efforts have been made to study the network behavior in predicting link occupancy. We believe that this factor, if used in the pre-processing phase of the algorithm will help achieve performance improvements in terms of Call Blocking Probability (CBP) thereby *promoting* optimal utilization of network resources. The results show an improvement over the other schemes mentioned in this paper in terms of number of calls accepted. We give an insight into critical study of the link utilization and prove that it has to be studied dynamically to obtain performance enhancement in terms of network resource utilization.

References

1. J.M.Jaffe, "Algorithms , for finding paths with multiple constraints", Networks; 14:95-116, 1984.

2. Z.Wand and Jon Crowcroft, "QoS routing for supporting Resource Reservation", IEEE Journal on Selected Areas in Communication, pp 1228-1234, September 1996.

3. Subhash Suri, Marcel Waldvogel, Daniel Bauer and Priyank Ramesh Warkhede, "profile based Routing: A new framework for MPLS traffic engineering and Traffic Engineering", In the proceedings of QoFIS 2001.

4. S.Chen and K. Nahrstedt, "On finding multi-constrained paths", International Conference on Communications (ICC'98), June 1998.

5. R.Widyono, "The design and evaluation of routing algorithms for real time channels", TR-94-024, International Computer Science Institute, UC Berkeley.

6. X. Yuan, "On the extended Bell-Ford algorithm to solve two-constrained Quality of Service routing problems", In the eighth International conference on Computer Communications and Networks (IC3N'99), Boston, Massachusetts, October 1999.

7. M. Kodialam and T.V.Lakshman. "Minimum interference routing with Applications to MPLS traffic Engineering", In the proceedings of IEEE INFOCOM, Tel-aviv, Israel, March 2000.

8. Shigang Chen and Clara Nahrstedt, "An overview of Quality of Service routing for the next generation high-speed networks: problems and solutions", IEEE Networks special issue on · transactions and distributions of digital video, Nov/Dec 1998.

9. H.F. Salama, D.S.Reeves and Y.Viniotis, "A distributed algorithm for delay constrained unicast routing", In the proceedings of IEEE INFOCOM'97.

10. Q.Ma and P.Steenkiste, "Quality of Service routing with performance guarantees"" In the proceedings of InternationalQoS"7, May 1997.

11. C.Pornavalai, G.Chakraborthy and N.Shiratori, "QoS based routing algorithm in Integrated Services packet networks", In the proceedings of ICNP'97, PP-167-174, IEEE1997.