QUALITY OF SERVICE (QoS) ROUTING FOR CONNECTION BUNDLES USING STOCHASTIC KNAPSACK FRAMEWORK

By

SURENDRAN RAMASAMY

Bachelor of Science in Computer Science

Bharathiar University

Coimbatore, Tamil Nadu

2000

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2006

QUALITY OF SERVICE (QoS) ROUTING FOR

CONNECTION BUNDLES USING STOCHASTIC

KNAPSACK FRAMEWORK

Thesis Approved:

Dr. VENKATESH SARANGAN

Thesis Adviser

Dr. G. E. HEDRICK

Dr. DEBAO CHEN

Dr. A. Gordon Emslie

Dean of the Graduate College

ACKNOWLEDGEMENTS

I would like to convey my sincere gratitude to my thesis advisor Dr. Venkatesh Sarangan for the guidance and encouragement that he gave me throughout my thesis work.

I would also extend my appreciation and gratefulness to my committee members Dr. G. E. Hedrick and Dr. Debao Chen for their support and advice.

I would like to thank my mother Mrs. Thangamani Ramasamy and my father Mr. Ramasamy Subramaniam and my brother Mr. Sudhakar for their love and support throughout the years.

Finally I would like to thank all my friends who stood beside me with their unfailing and indispensable support.

TABLE OF CONTENTS

Chapter **Page**

LIST OF TABLES

Table **Page**

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Today's network provides only best effort traffic where the traffic is processed quickly. There is no guarantee as timely delivery of packets and selecting the best path between source and destination. By using modern technologies we can increase the bandwidth that automatically delivers QoS. But the number of users and applications are increasing drastically every day. So the necessity of QoS routing has become a major factor to provide secured and guaranteed service that must also improve the efficiency of the working of the network routing protocols. The Internet Engineering Task force has proposed many service models to meet the demand for QoS. The most significant among them are as follows: [1]

- Integrated
- Differentiated
- Multi-Protocol Label Switching (MPLS)
- Constraint Based Routing

In the rest of this section we shall discuss in detail about each of the service models.

1.1 Integrated Service:

In this service model before data transmission the service path must be found and resources between the source and destination must be reserved. The RSVP protocol is the signaling protocol used to set the path and reserve the resources along the path from the source to the destination [1].

In Integrated Service the sender sends the path message to the receiver by specifying the available QoS (bandwidth, delay, etc) along the path. When the receiver acquires the path message the receiver sends the RESV message back to sender to reserve resources along the path. Any intermediate router can either accept or reject the RESV message. If any intermediate router rejected the request then the router will send an error message to the receiver, and the signaling process will terminate. If the request is accept then the bandwidth and buffer space is allocated for the flow and data transmission is done from the sender to the receiver. The RSVP signaling protocol has been modified to reserve resources for aggregation of flows which is still a hotly debated issue in the IETF.

The integrated service is implemented by the four components: the signaling protocol (RSVP), admission control routine, packet classifier and packet scheduler where each component plays an important role in deciding a path from the sender to the receiver.

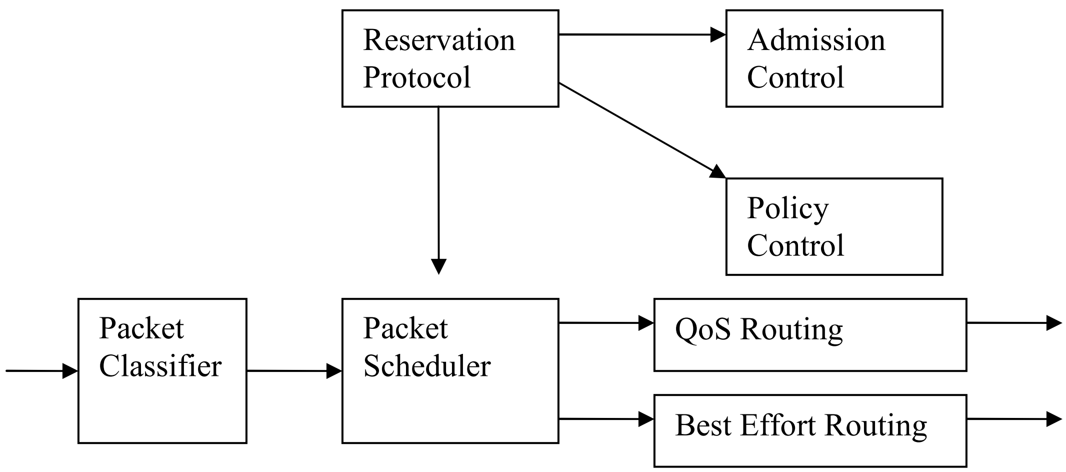

Figure 1.1 Integrated Service Components

In integrated service the flow has to set up a path before it can reserve resource along the path of flow of traffic which is set up by the signaling RSVP protocol. To reserve a resource the RSVP protocol communicates with the admission control and policy control. Admission control determines whether there are available resources in the node to satisfy the requested QoS. The policy control determines whether the particular user has permission to make reservations. If any one of them fails then an error message is sent back to the receiver and the signaling process is terminated. If there are no error messages then the reservation protocols set the parameters in the packet classifier and the packet scheduler to obtain the desired QoS. The packet classifier determines the QoS for each packet while the scheduler schedules the packets accordingly to the meet the QoS requirements. Though Integrated Service is simple in concepts it has some major drawbacks as:

1. As the state information is stored for each flow of traffic the amount of information stored is proportional to the number of flows.
2. All routers must implement RSVP, admission control, Multi-Field Classification and packet Scheduling.
3. Due to Storage overhead the scalability of Integrated Service is minimized.

1.2 Differentiated Service:

The drawbacks of Integrated Service had led to the conception of Differentiated Service. Here the traffic is classified into groups and the state information is stored for groups. Thus it is now sufficient that the routers store the routing information for a group rather than store the information for each flow which makes it more scalable than Integrated Service. In Differentiated Service the packet is classified, policed and possibly shaped in the ingress/egress router of the domain rather than have this done in each router. The global state information is maintained in the border routers of the domain and the core routers maintain the detailed local state information. Thus much of the routing table complexity is pushed to the border router from the core routers. So the routing in the core routers are done much faster as their routing table is kept simple. By using hierarchical routing the border routers store the aggregated state information about other group of nodes in the network. The border routers still store detailed information about its domain.

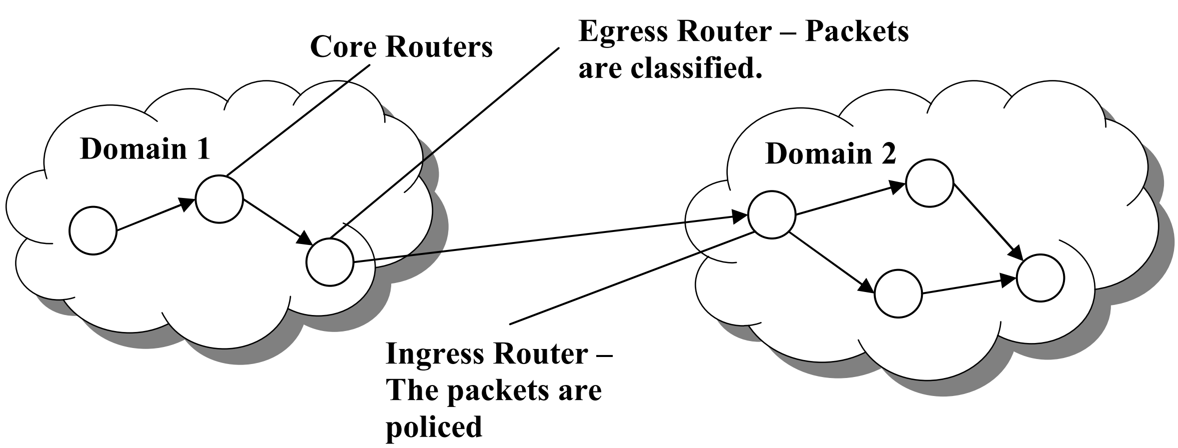

Figure 1.2 Differentiated Service

From the above figure, if the total traffic from Domain 1 to Domain 2 is greater than the quantity specified in the SLA (Service Level Agreement) between both domains then it shapes the traffic to the confirmed parameters. The first hop in the domain does the policing that is, it checks for traffic parameters and sets the code points for the packets from the host. The interior routers (core routers) have queues that have different QoS parameters. When a packet enters one domain from another its differentiated Service (DS) field is remarked as determined by the SLA between the two domains. Using the different policies and classifications different services can be provided as Assured, Premium and Olympic services. [1]

1.2.1 Assured Service:

In assured service the classification and policing are done at the ingress router. If the assured service traffic does not exceed the bit-rate specified by the SLA then they are considered as IN profile, otherwise they are considered as OUT of profile. All packets of the assured service are put into the assured queue and the queue is managed by the queue management scheme called Random Early Detection RED with In and Out or RIO.

Random Early Detection will randomly drop packets to avoid traffic congestion and therefore avoid tail drop traffic. There are two thresholds for each queue. When the queue is below the first threshold no packets are dropped. When the queue size is in between the two thresholds then the packets with out profile are randomly dropped. If the queue size exceeds the second threshold then the queue will drop both the In and Out packets but the Out packets are aggressively dropped. Thus the network is better utilized by avoiding congestion.

1.2.2 Premium service:

In such kind of service the SLA will specifies the desired peak bit rate and it is the responsibility of the user not to exceed the peak rate else the extra packet will be dropped. Premium service is more expensive than the Assured service so it is desirable for Internet Service Providers (ISP) to support static and dynamic SLA. Dynamic SLA allows the user to request premium service on demand without subscribing for it. The premium service works as follows:

- By admission control the amount of premium traffic can be limited to a small percentage of the bandwidth of input links.
- Excess packets are dropped at the ingress router of the networks.
- Premium packets are forwarded before other classes of packets, thus they can potentially use 100 % of the bandwidth of the output links.

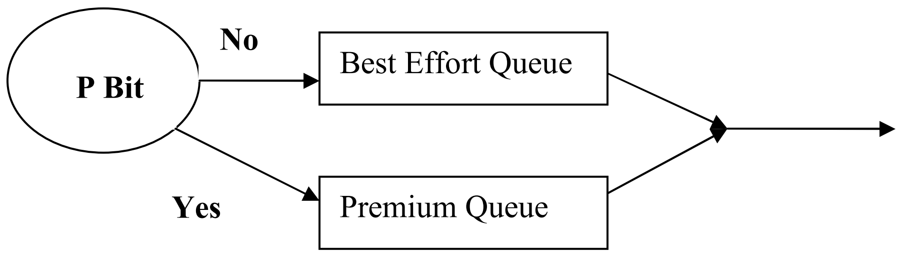

Figure 1.3 Queues in Differentiated Service

From the above diagram the router checks if the p bit of the packet is set. If the bit is set then it is sent to the premium queue which has higher priority else it is sent to the best effort queue. There are certain drawbacks in Premium Service:

1. Consider many border routers trying to communicate to the core routers and all links have the same bandwidth. With premium service the arrival rate of the premium traffic is far below the service rate. Thus Differentiate service alone cannot solve such problems so Traffic Engineering/Constraint Based routing is used to avoid congestion.
2. Premium service might starve Best effort and Assured Service traffic. To avoid such problems we have to limit the amount of bandwidth for premium service which might limit the extensibility of Premium service.

1.3 Multi Protocol Label Switching (MPLS) and Traffic Engineering:

In traditional IP routing the packet is forwarded by checking the packet header with those in the routing table in each router. In such cases the router has to check for the longest match of the IP address of the packet with that of the IP address in the routing table. The prefix matching is done for packets that belong to the same subnet or if they follow the same route and interface. This task becomes very tedious and time consuming in a large network.

To overcome this problem MPLS was designed such that with the addition of a label over the IP header, it can speed up the process to route the packet. This also made it easier as the packet is forwarded based on fixed labels. The MPLS has the Forwarding Equivalent Class (FEC) which specifies the type of service and the destination. So packets with the same Forwarding Equivalence Class are being treated the same.

The control component of the MPLS creates a binding for a label with the FEC and informs other Label Switch Router (LSR) about the binding and utilized the label to maintain the forwarding table.

Label Switched Path (LSP) defines a route from the ingress to egress router for a FEC which is unidirectional and by using traffic engineering the path can be decided. Selection of the LSP for a particular FEC can be hop by hop or explicit. In hop by hop the LSR can independently select the next hop thus not supporting traffic engineering while in explicit the LSP is usually mentioned in the ingress/egress router.

Traffic Engineering too plays a vital role to improve the utilization of the network resources and QoS routing. Due to uneven traffic or lack of resources, possibilities of network congestion may occur. This is because many routing protocols select the shortest path between two nodes which causes heavy traffic in the link. Traffic engineering is the process of arranging traffic flows through the network so that congestion caused by uneven network utilization can be avoided. To achieve traffic engineering the traffic must be measured, characterized and modeled [6].

1.4 Constraint Based Routing:

QoS routing finds a route from the source to the destination based only on the route that satisfy the QoS requirement. Constraint Based Routing is an extension of QoS routing where the selected route meet the QoS requirement and increase the utilization of the network. Constraint based routing considers the network policies, availability of resources and the requirements of the flow. In order to compute routes using Constraint Based Routing the routers need to advertise the link state information and resource availability information. To avoid frequent flooding of link state information, the link state information must be advertised only in the change in the topology or at regular longer time intervals. Route metrics that are used in Constraint Based Routing are hop-count, bandwidth, delay and jitter. For *d (i, j)* be as the metric for link *(i, j)* in any path say *P = (i, j, k … l, m), metric d* is [1] the routing can be divided into:

- *Additive* if *d (p) = d (i, j) + d (j, k) + ….. + d(l, m)*
- *Multiplicative* if *d (p) = d (i, j) * d (j, k) * ….. * d(l, m)*
- *Concave* if *d (p) = min {d (i, j), d (j, k)… d(l, m)}*

Metrics such as delay, hop count, jitter and cost are additives, reliability (loss rate) is multiplicative and bandwidth is concave. Using Bellman Ford and Dijkstra's algorithm it is easy to compute routes with bandwidth and hop count constraints. Delay and jitter can be mapped from bandwidth and hop count information. Constraint Based Routing also causes the increase in size of the routing table in each router which can be reduced by using coarse routing granularity and hop quantization. That is, the hop count values can be divided into classes and the routing table only maintains the information about the best effort traffic and compute flows with QoS demand [1].

CHAPTER II

BACKGROUND

Future integrated service will carry both QoS traffic and best effort traffic. This will affect the throughput of best effort traffic if the traffic distribution is misjudged. The network state changes dynamically and it is difficult to gather updated state information due to the growing network size and non-negotiable time delay. QoS routing can be degraded if the state information is outdated. QoS routing depends on the state information and is classified as Source, Distributed and Hierarchical routing. Each of these routing strategies is unique and have their own pros and cons as mentioned below.

2.1 Source Routing:

Each node maintains the complete global state information on which a feasible path is computed in the source node. This kind of routing is simple, easy to implement, evaluate, debug, upgrade and loop-free routes [2]. Though this routing is simple it has some major drawbacks.

- Each node has to maintain the global state information so every node has to update the state information thus leading to communication overhead.
- Source routing can provide only approximate global state due to the non-negligible propagation delay of state messages.
- The computation overhead at the source is very high in the case of multicast routing or when multiple QoS constraints are involved. Thus Source routing was not scalable in large networks.

2.2 Distributed Routing:

Control messages are exchanges among intermediate routers and the state information in each node is collectively used to compute a path. Distribute routing algorithms uses distance vector protocols to maintain the global state. Based on the distance vectors the routing is done on hop by hop basis. Though distributed routing is more scalable than source routing it has some drawbacks:

- Due to inconsistent state information it might cause loop which can be detected if the routing message reaches the source node the second time. But the distance vector protocols do not provide alternative routing paths.
- It is difficult to implement distributed routing in multicast routing, because there is no detailed topology and link state information available [2].

2.3 Hierarchical Routing:

In hierarchical routing, nodes are clustered into groups which are further clustered recursively, creating different levels of hierarchy. The hierarchical routing scales well because each node maintain detailed state information about the nodes in the same group but aggregated state information about other groups. So the flooding of the network state information is reduced. The complexity of routing table is pushed towards the border routers thus creating simplicity in routing with in the domain.

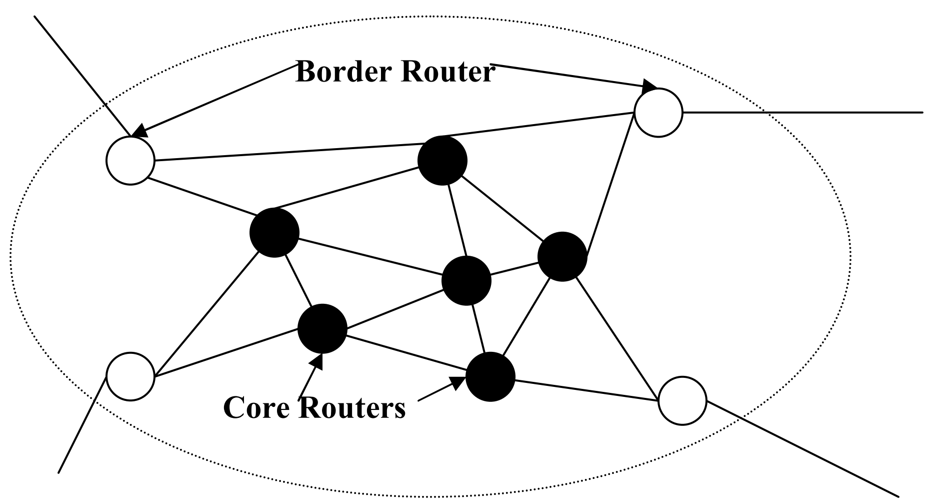

Figure 2.1 Routing Architecture in Hierarchical Routing

In order to maintain the updated information at each node, every node periodically advertises its domain topology information to every other node in the network. The hierarchical routing shares the advantages of source routing if it needs to compute a route within the same hierarchical level. It also shares the advantage of distributed routing because the routing computation is shared by many nodes. Though hierarchical routing is scalable and looks like a better approach than Source and Distributed routing, they too incur some major drawbacks:

- As the network state information is aggregated the accuracy of individual link state information is lost. It is difficult to get the exact end to end delay from a particular source node to its destination node.
- When the routing involves in two or more QoS constraint, the routing gets more complicated.

Though hierarchical routing has a few draw back it is relatively better than source and distributed routing. So in our thesis hierarchical routing is used due to its scalability over Source and Distributed routing.

CHAPTER III

LITERARTURE REVIEW

QoS routing is used to find a feasible path that satisfies the QoS constraints. To do so the network must maintain precise network state information and must be scalable. Different approaches were done to maintain accurate state information and reduce the flooding of network state information.

The network had to send update message to other nodes in the network, since the routing is based on availability of resources. Advertising resource availability to all other routers consumes significant amount of network bandwidth. In [3], the author mentions Quantity and Frequency reduction as two main techniques to reduce the overhead and improve scalability. The goal of Quantity reduction is to reduce the number and size of the routing messages while maintaining the routing performance. This is done by restricting the messages and sending the accurate message where it is needed the most. The goal of frequency reduction is to send the update messages as infrequent as possible. Based on these concepts the author proposed two aggregation models: full mesh and star.

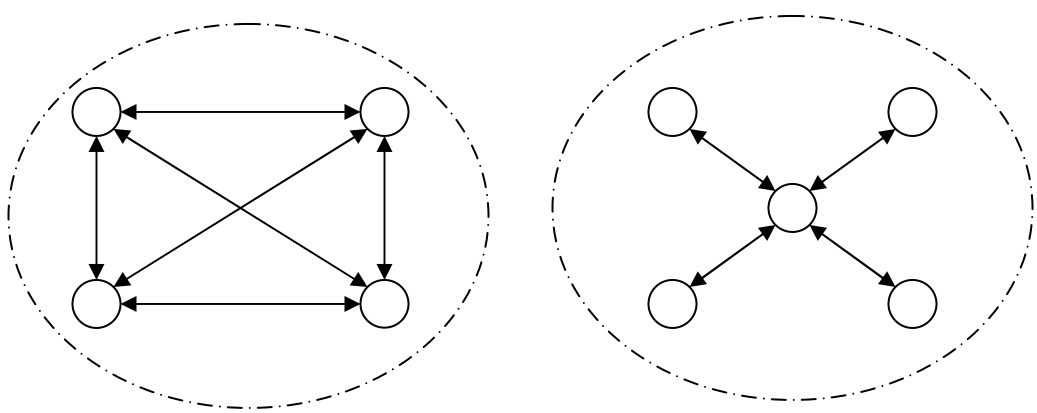

Figure 3.1 Mesh and Star Aggregation

3.1 Mesh Aggregation:

The full mesh approach focuses on the accuracy of the aggregated information. In the full mesh the port to port distances is represented as a matrix with one entry per port. Thus a domain with n ports has a matrix of size n^2. Though this kind of topology aggregation is simple and gives accurate information, it is not scalable in a large network.

3.2 Star Aggregation:

In the Star aggregation the topology is aggregated such that it has one central virtual node called the nucleus and n border nodes. This central node is explicitly connected to all of the border nodes via *spokes*. The spokes may be given weights different from the radius. The aggregation model allows the aggregated representation to specify logical links directly between borders; such links are called *bypasses.* The information in the star kind of representation is often inaccurate assuming the distances between two ports are the same but is scalable as it reduces the routing information size complexity to O (1).

3.3 Service Curve:

The availability and accuracy of state information plays an important role in QoS routing, but it is impractical to maintain the complete state information of a large network at a single location. In [4], a new approach was proposed to represent the service a domain can support. Figure 3.2 represents a domain with bandwidth and delay of each physical path from A to E.

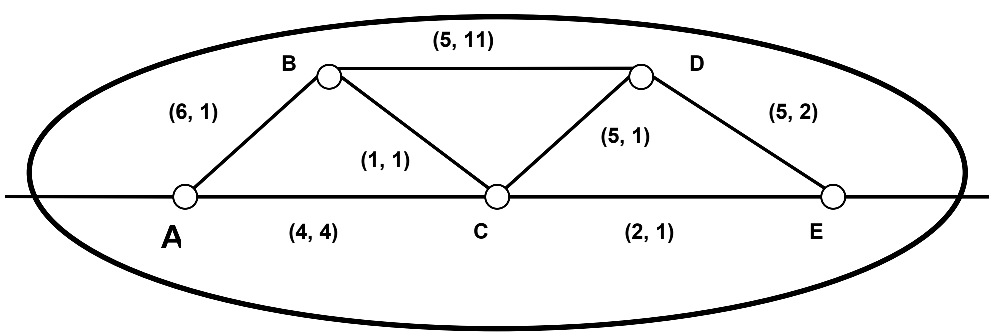

Figure 3.2 Domains with Bandwidth and Delay

For the above domain the paths between the border routers A and E with bandwidth and delay are:

Number	Path	(Bandwidth, Delay)
P1:	A→C→E	(2,5)
P2:	A→C→D→E	(4,7)
P3:	A→C→B→D→E	(1,8)
P4:	A→B→D→E	(5,14)
P5:	A→B→C→E	(1,3)
P6:	A→B→C→D→E	(1,5)
P7:	A→B→D→C→E	(2,14)

Table 1 Physical path between the border routers

The points of all the physical paths from A to E can represent the service staircase on the bandwidth/delay plane.

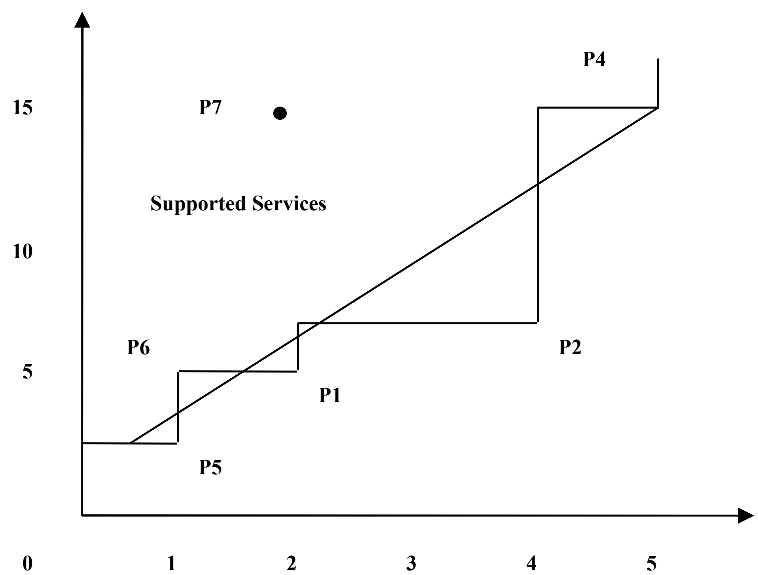

Figure 3.3 Service Staircase and it's approximates for (BW, delay)

The router will first reject all connection requests that lie below the approximation curve. The actual path will be probed and the resources will be reserved after that. The flow will be rejected if the resource requirement can not be satisfied in the actual path [4].

This outlines the areas of supported services. Any request whose bandwidth and delay requirement fall with in the supported service curve can be admitted. The service curve is approximated using the polynomial curve, cubic spline and polyline approach. These achieve more accuracy both locally and globally, depending on the space used to store the curves.

3.4 Probability Based Routing:

QoS routing makes routing decisions based on the QoS requirements and the network resource availability information. These are called as availability based routing [5]. The major barrier for QoS routing is the imprecise information about the availability of resources in a large network. Certain causes for the unreliability of the state information are as below [8]:

- Topology Aggregation: Due to topology aggregation the exact availability of resources in the individual nodes and link is lost. Now the routing algorithms must now consider both the availability of the resources and the certainty that the specific resource is available.
- Periodic Updates: Every time a call is accepted in the network the QoS availabilities in the links change. To maintain the updated information frequent exchange of messages occurs both in the inter domain and intra domain. These update message consumes a lot of bandwidth on all the links the update messages are sent. These time intervals are set depending on certain threshold policies. This periodical exchange of update messages can causes the state information in other nodes to be out dated.
- Propagation Delay: The updated messages between nodes are never instantaneous due to the propagation delay or the delay in transmission in certain links. Though a node sends updated messages the other nodes in the network may still have outdated messages.

The above drawback has given some light to probability based routing. By using the resource availability and the QoS parameters we can either infer or calculated the probability of the number of requests that will be served depending on the link state over a period of time. With the probability based routing we can predetermine the amount of traffic that would be accepted in each domain and accordingly route the path that would have the highest amount of resources. Probability based routing offers a solution to the imprecise networks information since the resource availability probability distribution for a link changes much slower than the resource availability of the link [5]. Probability based routing is better than the traditional based routing only when the global state information is imprecise.

CHAPTER IV

METHODOLOGY

4.1 Stochastic Knapsack:

By using the concepts of Stochastic Knapsack framework, each domain in the network can determine the dropping probability of requests in other domains before it can route a path. Each domain in the network can be referred to as a stochastic knapsack and the dropping probability can be found by using Stochastic Knapsack framework. The dropping probability in a domain can be determined based on the distribution of arrival and loss of requests over a period of time. Consider a Stochastic Knapsack consisting of *C* resources with objects from *K* classes arriving which can be distinguished by their size, arrival rate and mean holding time.

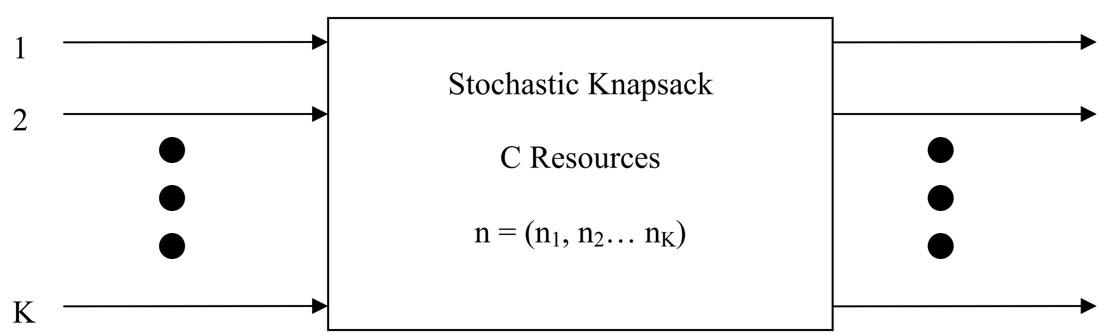

Figure 4.1 Stochastic Knapsack

The arrival processes (*K* class) are independent from each other and arrive with arrival rate λ_k. If an object with class k is accepted into the Knapsack then b_k resource are held for the holding time that is exponentially distributed with mean $1/\mu_k$. The resources allocated for class k must be less than the total amount of resources available in the knapsack. The total amount of resources utilized by k objects in the knapsack is $b \cdot n$ and

$b = (b_1 \ldots b_k)$ – Resources allocated to each class.

$n = (n_1 \ldots n_k)$ – Number of objects of the particular class in the knapsack. Then the resources utilized by k class objects are:

$$b.n = \sum_{k=1}^{K} b_k n_k . \qquad \text{--------- (1)}$$

The knapsack admits an arriving object only if there is available resources requested by the arriving object. That is the requested resources for an arriving class *k* object must satisfy the below condition [9].

$$b_k \le C - b.n \qquad \text{---------- (2)}$$

The state space in the knapsack can be defined as: $S = \{n \in I^k : b \cdot n \le C\}$ where

b: Number of resources being utilized,

n: Number of k class objects and

I: Set of non-negative integers.

Let $X_k(t)$ denote the random number of class *k* objects in the knapsack at time *t*. Let $\mathbf{X}(t) = (X_1(t) \ldots X_k(t))$ be the state of the knapsack at time *t* and $\{\mathbf{X}(t)\}$ is the associated stochastic process. This process is a periodic and irreducible Markov process over the finite space *S* [9].

The equilibrium behavior of the stochastic knapsack can be addressed as follows [9]: Consider $\pi(\mathbf{n})$ denote the probability that the knapsack is in state **n** in equilibrium such that each of $n \in S$. The offered load (ρ_k) for class *k* objects with arrival rate λ_k and mean holding time $1/\mu_k$ is λ_k/μ_k. The equilibrium distribution for the stochastic knapsack is:

$$\pi(n) = \frac{1}{G} \prod_{k=1}^{k} \frac{\rho_k^{n_k}}{n_k!}, n \in s, \qquad \text{---------- (3)}$$

Where

$$G = \sum_{n \in S} \prod_{K=1}^{K} \frac{\rho_k^{n_k}}{n_k!} . \qquad \text{---------- (4)}$$

The expression for $\pi(\mathbf{n})$ is the product form solution for the stochastic knapsack. The constant *G* is the normalization constant for the stochastic knapsack. Equation 3 holds for both exponential holding time distributions and for arbitrary distributions.

4.2 Blocking Probability:

Suppose that $\boldsymbol{B_k}$ is the blocking probability for class *k* objects. $\boldsymbol{B_k}$ is proportional to the b_k for the knapsack with large capacities C and typical traffic conditions. Since larger objects require more room than smaller objects, they have dropping probabilities that are higher than the object that request for less space. More precisely, $\boldsymbol{B_k} > \boldsymbol{B_l}$ if $b_k > b_l$ [9].

4.3 Performance Evaluation:

To derive the expression to find the dropping probabilities consider $\boldsymbol{S_k}$ as the subset of states in which the knapsack admits an arriving class-*k* objects that is:

$$S = \{n \in S: b \cdot n \leq C - b_k\}$$

For example: Consider a knapsack with capacity C=8 and two classes of objects with b_1=1 and b_2=2. The sets S_1 and S_2 can be illustrated from the below diagram.

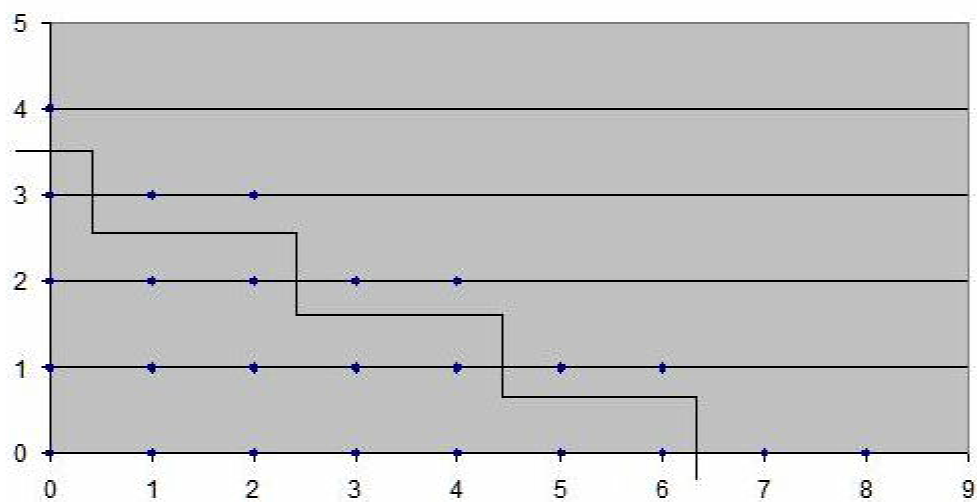

Figure 4.2 State diagram that accept two class of traffic

All the points in the diagram show the feasible states of the knapsack which is the set S_1. The points below the line (set S_2) indicate the states of the knapsack that can admit class 2 objects [9]. The above diagram shows that the dark points below the line are those states that can accept class 2 objects.

In a general model when objects of class k arrive, the blocking probability of class-k objects is:

$$B_k = 1 - \sum_{n \in S_k} \pi(n) \qquad \text{---------- (5)}$$

Using the equation (3) and (5), the explicit expression for blocking probability:

$$B_k = 1 - \frac{\sum n \in S_k \prod_{j=1}^{k} \frac{\rho_j^{n_j}}{n_j}}{\sum n \in S \prod_{j=1}^{k} \frac{\rho_j^{n_j}}{n_j}} \qquad \text{---------- (6)}$$

B_k is the dropping probability that an arriving class k object is blocked. The blocking probability is equivalent to the long run fraction of arriving class k objects that are blocked. Larger objects would require more room than smaller objects, so they have higher dropping probabilities.

In our model the stochastic knapsack is referred to a domain (say D) in a network, the capacity C is the total amount of resources in the domain. Domain D advertises the distribution of traffic it can handle to other domains in the network. Depending on the advertised information the neighboring domains can calculate the dropping probability in domain D. For example: In a multimedia traffic there can be two different class of traffic (Video and Audio) entering the domain in a network graph. Each class of traffic has an independent arrival rate λ, mean holding time 1/μ and requesting for bandwidth. Based on these parameters the dropping probability can be found from the source.

With the determined dropping probability the acceptance probability can be evaluated. Here the path with the highest acceptance probability can be found by using Dijkstra's algorithm. The multiplicative metric of the acceptance probability must be converted into its additive property as Dijkstra's algorithm works only with additive metric. The multiplicative metric of the acceptance probability can be converted into additive metric by using logarithms. By using Dijkstra's algorithm the path with the highest acceptance can be found. Each router computes its routing table using Dijkstra's single source shortest path algorithm. When change in any connectivity in the network is detected, the information is flooded in the network. As each router computes its own most acceptable path, it takes much less time to react to change in the network and recalculate accurate table. By using Dijkstra's algorithm the path with the highest acceptance probability can be calculated from the source node to every other node in the network by iteratively growing the set of nodes (S) to which it already knows the highest acceptance probability path. At each step in the algorithm, the node in the set V – S with the highest acceptance probability is added to S (V is the set of all nodes in the network). This algorithm loops back, processing the next node in V – S with the highest acceptance probability. The algorithm completes when S contains all nodes reachable from the source node.

CHAPTER V

IMPLEMENTATION

5.1 Resources:

The code was developed and implemented in Sun Solaris (KDE Environment) operating system that is offered by the department. C language in UNIX is being used for the development due to its flexibility such as portability, file handling, library functions, system calls, shell programming and UNIX utilities. The GCC compiler (formerly called the GNU C Compiler) which is open source software is being used to compile the code.

5.2 Network Design:

A complex high level network graph is modeled such that results from different scenarios are obtained. To compare our approach to that of traditional routing we must consider two different scenarios that occur during routing.

- Route a path with source, destination pair such that few paths tends to overlap.
- Route a path with source, destination pair such that more overlap tends to overlap.

Each node in the graph indicates a domain and each domain are linked by directed edges. The network graph shown below is represented as a text file which indicates the advertised information such as the domain capacities and the adjacent domains.

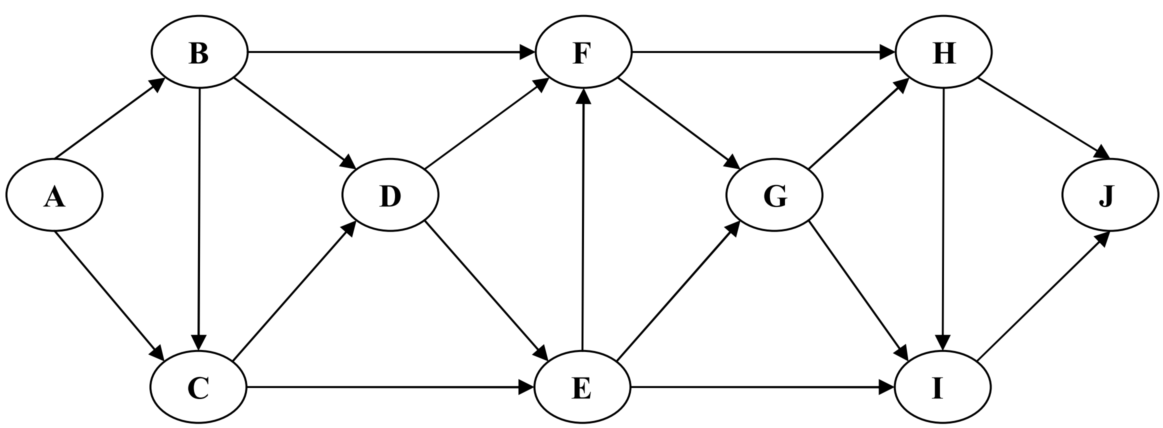

Figure 5.1 Network Graph

All domains advertise their domain capacities over the network. From the above graph the advertised domain capacities as A:50, B:70, C:40, D:80, E:40, F:60, G:90, H:40, I:90 and J:100. Now that the network graph is designed there are several call bundles that are distinguished by their QoS constraints. The QoS constraints that are involved are:

- **λ** Arrival rate of request along the path.
- **1/μ** Mean holding time of the resources in the domain.
- **BW** Bandwidth that is allocated if the request is accepted in the domain.

Thus if a call bundle is accepted in any domain (D) the requested resources are allocated in the domain and the network load will be increased by **(λ / μ).** The call bundles that are accepted in the domains will exists throughout. So when another call bundle is accepted the network load is increased by adding the current network load to the new network load caused by the next accepted class of traffic.

5.2.1 Routing using Best Effort:

Using the best effort protocols the path from node A to node J can be routed as shown in the diagram below:

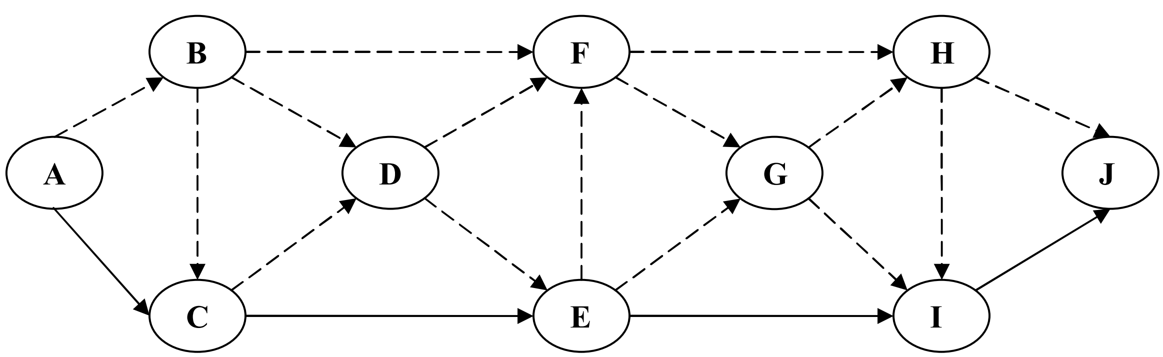

Figure 5.2 Network graph: Best Effort routing

Best effort routing routes the path that has the least hop count. In the network graph the path A → C → E → I → J has the least hop count from node A to node J. There are 5 calls each demanding for 10 units of bandwidth so the total amount of bandwidth that is requested is 50 units.

Stochastic Knapsack framework is used to calculate the dropping probability of the call bundle in each domain by using the demanded QoS constraints and the traffic load in each domain. Using the evaluated dropping probability the number of call bundles that are accepted in the domain is determined. In best effort routing protocols for the source, destination pair (*a, j*) the path taken is (A → C → E → I → J). This path cannot support a call bundle that request for 50 units of bandwidth because the domain capacity in domain I is only 40. Though the availability of bandwidth in domain A, C, E and J is greater than 50, most of the requests are dropped in domain I as it can support only 40 units of bandwidth.

5.2.2 Routing using Dropping Probability and Dijkstra's Algorithm:

Using the same network graph and QoS constraints the path routed using our approach is shown as below:

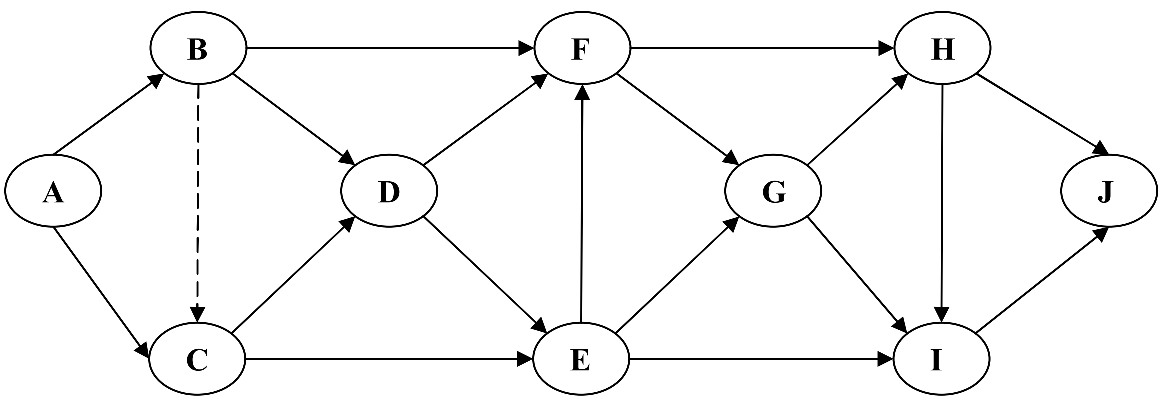

Figure 5.3 Network graph: QoS routing using Stochastic Knapsack Framework

Based on the network load, resource availability and the demanded QoS constraints, source node A can estimate the dropping probability of calls in each domain using the stochastic knapsack framework. Using the dropping probability the acceptance probability is computed. Based on Dijkstra's Single Source Shortest path algorithm the shortest path can be computed. To find the path that has the highest acceptance probability the multiplicative metric of the acceptance probability must be converted to its equivalent additive metric which is done by using the concepts of logarithms. The basic property of logarithms is: $log_b(mn) = log_b(m) + log_b(n)$. So to convert the multiplicative metric of acceptance probability to its equivalent additive metric we use logarithms. For path (*P*) from say A → B → C with for dropping probabilities m, n and o respectively, their multiplicative metric is converted to its additive metric as:

$$P = \log_b (1 / m) + \log_b (1 / n) + \log_b (1 / o).$$

In the given example the acceptance probability is calculated in all domains and its inverse logarithmic value is computed. Now using dijkstra's shortest path algorithm we can route a path that has the highest acceptance probability. Using our approach if all the domains in the network are heavily loaded then the selected path is A → B → D → F → G → I → J. As the bandwidth metric is concave, the bandwidth that is accepted along the path *P* = (A → B → D → F → G → I → J) is given as [1] $d(p) = \min\{d(a, b), d(b, d), d(d, f), d(f, g), d(g, i), d(i, j)\}$ which is 60 as domain F is along the path and has the minimum bandwidth as 60. The selected path can serve the demanded bandwidth that is 50 units. When a connection has been made then the traffic load in the intermediate nodes are increased. The dropping probability is now calculated using stochastic knapsack based on the new traffic load. Using our approach the route is never the same. The routing path keeps changing dynamically with availability of resources and traffic load.

5.3 Routing with less overlapping connections:

In a network graph for a (source, destination) pair there can be multiple different paths. This scenario forms the one of the best case scenario for the best effort routing protocol as less paths tend to over lap. In this section we shall see the routes selected using best effort and our approach.

5.3.1 Best Effort routing:

Consider the network graph as shown below. The dotted lines indicate that there is no traffic load in the network.

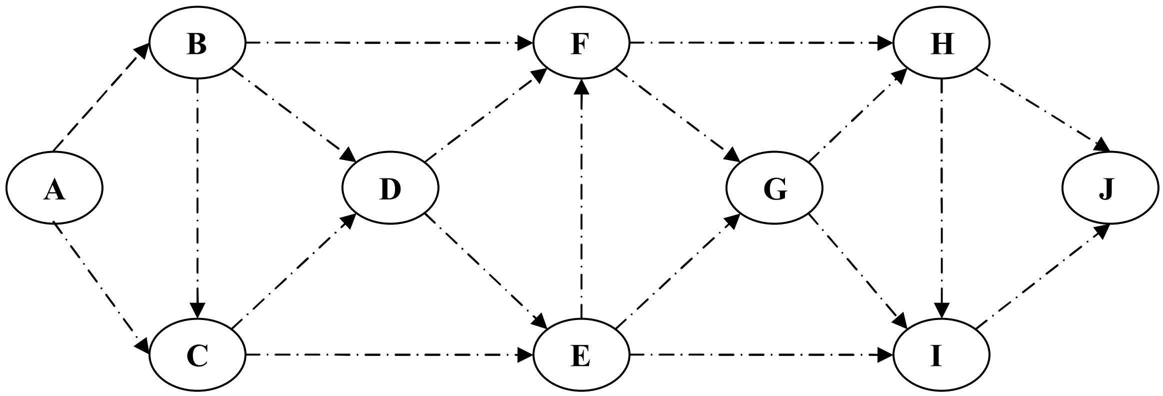

Figure 5.4 Network graph with no traffic load

With a traffic from source, destination pair as (A, J). The network graph will now be:

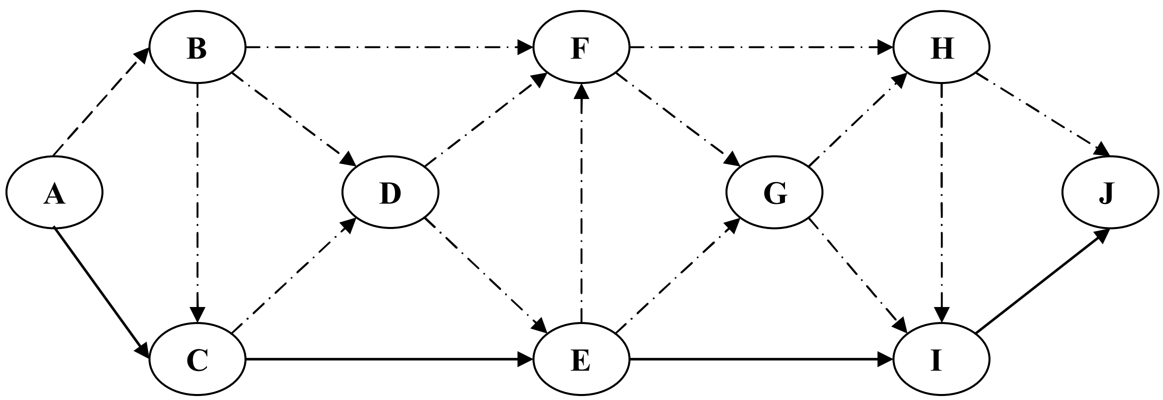

Figure 5.5 Best Effort: Network graph (less overlapping) for connection 1 (A to J)

The above figure shows the path selected for a route with source and destination pair (A, J). With static route the dropping probability will increase with increasing traffic. Consider another call request with source, destination pair as (C, G).

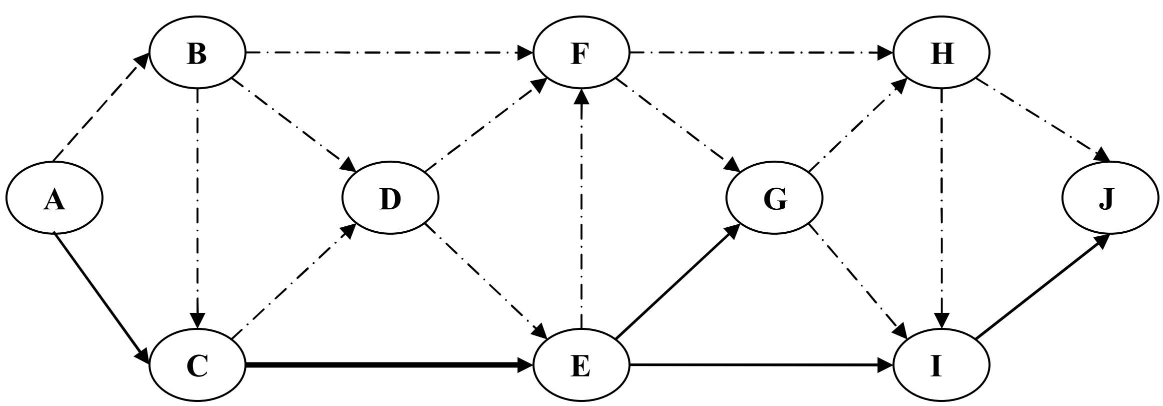

Figure 5.6 Best Effort: Network graph (less overlapping) for connection 2 (C to G)

In best effort approach the path taken for call (C, G) is C → E → G, though there is an alternative better path C → D → F → G. Using this method will effect the traffic load along the path for (A, J) and (C, G). Consider another call request for (F, J).

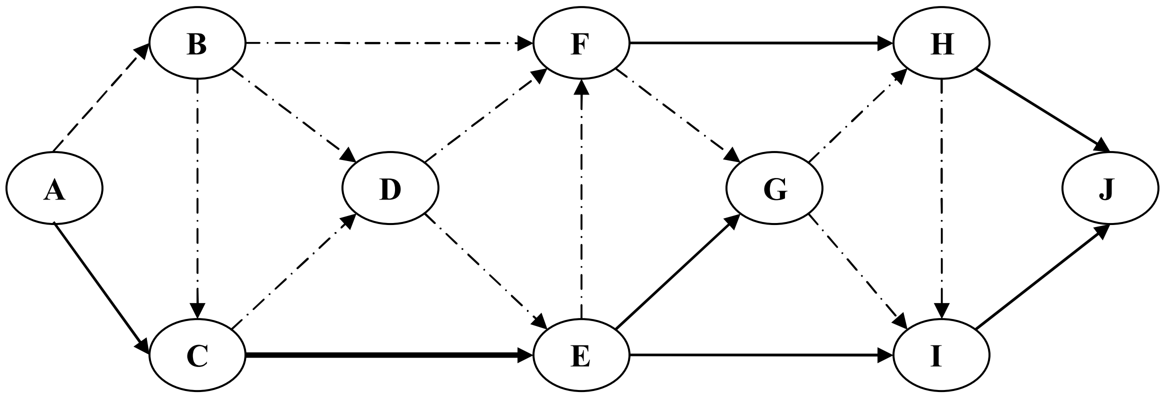

Figure 5.7 Best Effort: Network graph (less overlapping) for connection 3 (F to J)

The path from F to J does not affect much in the routing initially. But as the traffic load keeps increasing the dropping probability increases as the path is static. Consider another path with (source, destination) pair (D, I).

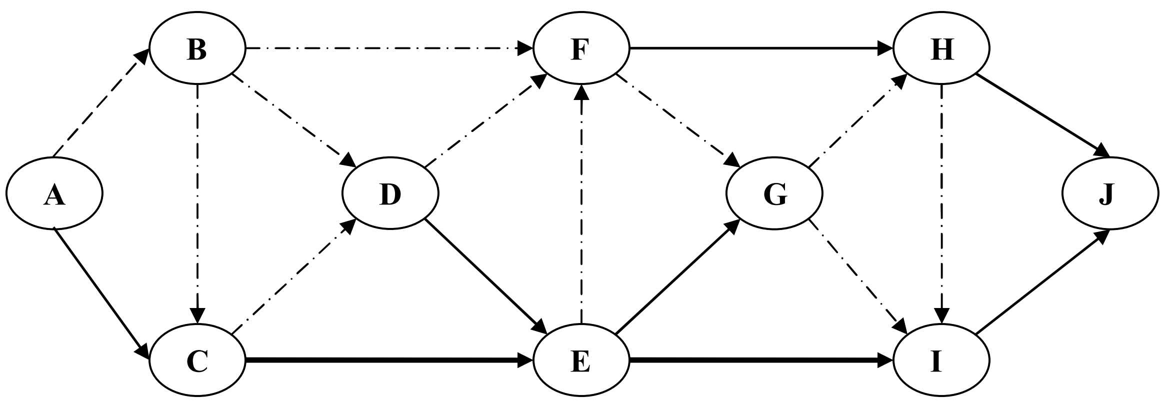

Figure 5.8 Best Effort: Network graph (less overlapping) for connection 4 (D to I)

Using best effort approach the path taken is D → E → I. By using this approach almost all the resources in domain E is exhausted which causes a drastic increase in the dropping probability for calls A → J, C → G and D → I. The traffic load is not uniform as there is no traffic load in the links A → B, B → C, B → D, B → F, C → D, D → F, E → F, F → G, G → H, G → I and G → I. By using these links the total dropping probability can be reduced .To use these resources efficiently we use QoS routing.

5.3.2 QoS Routing using Dijkstra's Algorithm:

With the same network graph and same source, destination pair .For the same (source, destination) pair different routes can be taken for different classes of traffic. The example shows the route for a single class of traffic. The network graph for a call A → J is:

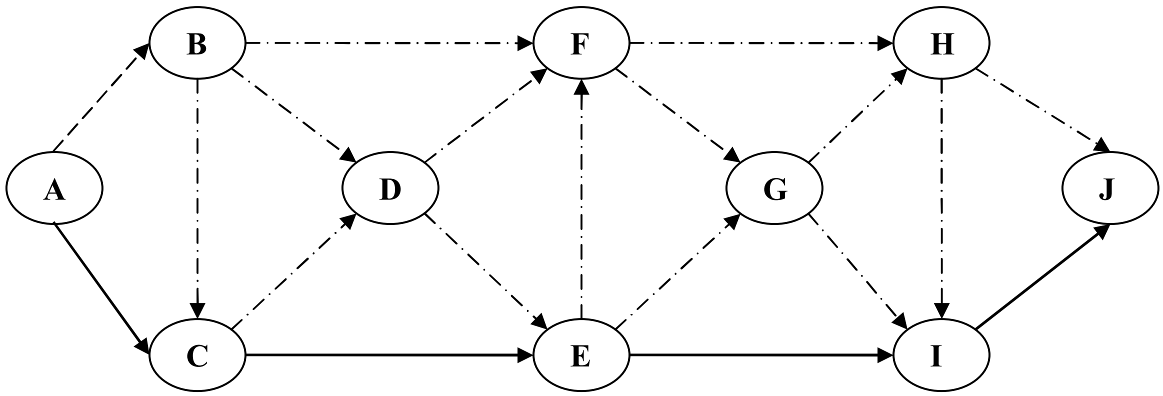

Figure 5.9 Routing by Stochastic Knapsack: Network graph (less overlapping) for connection 1 (A to J)

In the above network graph the path is selected based on the dropping probability. As there is no traffic load this would first work similar to that of Best Effort approach. As the traffic load in the selected route increases and the dropping probability increases the route is switched to an alternative path. Here the path can be as A → B → F → H → J or A → B → F → G → I → J. The path mainly depends on the availability of network resource and traffic. Now when another connection request from C → G arrives the network graph changes as below:

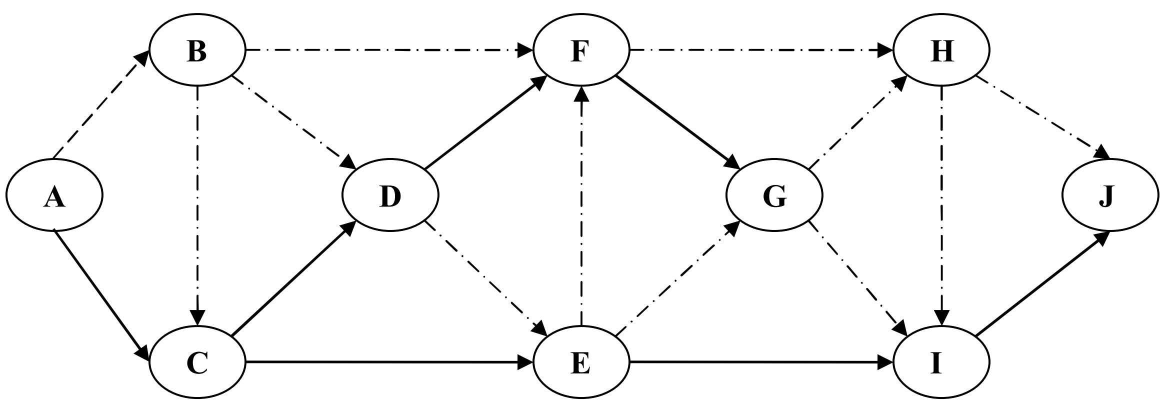

Figure 5.10 Routing by Stochastic Knapsack: Network graph (less overlapping) for connection 2 (C to G)

In this scenario the traffic load in node E is high as resources are allocated for the call A → J. So the selected path is C → D → F → G, if the dropping probability in this path is higher than the alternative paths then the route can be switched over to the alternative path. Consider another call from F → J whose routing would be similar to that of Best effort initially.

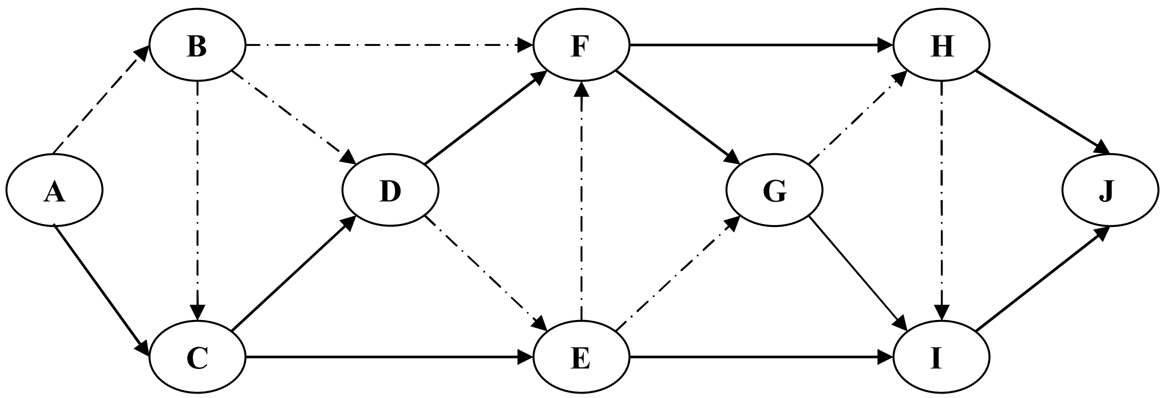

Figure 5.11 Routing by Stochastic knapsack: Network graph (less overlapping) for connection 3 (F to J)

As the domain capacity for node H is 40 while G and I is 90, when the resources in domain H is exhausted the route can be changed to F → G → I → J. Consider another call from (source, destination) pair as (D, I).

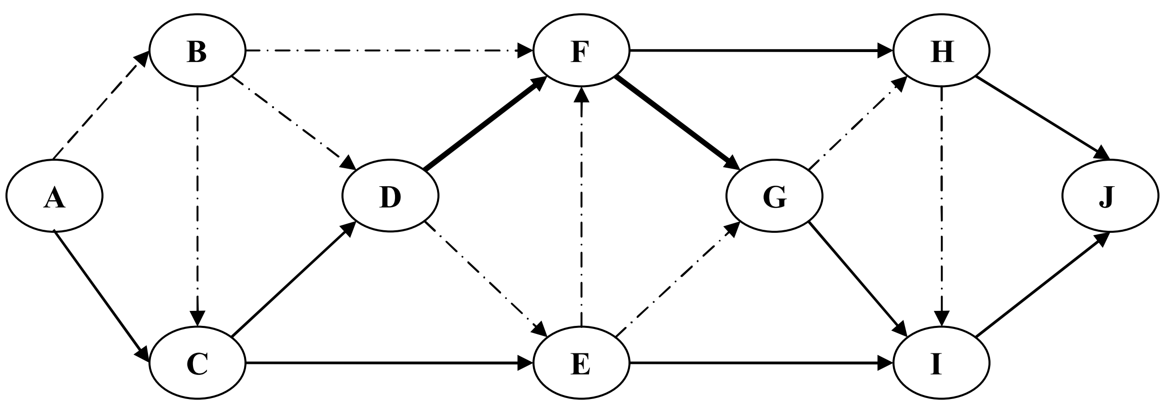

Figure 5.12 Routing by Stochastic Knapsack: Network graph (less overlapping) for connection 4 (D to I)

Initially in the above network the path selected is D → F → G → I. If the path gets congested then another alternative path as D → E → I is selected if the acceptance probability is higher.

All the routes shown above in our method are not static and changes based on the availability of the resources and the QoS constraints. While the routes shown for the best effort approach are static as this selects the path with minimum hop count.

5.4 Routing with more overlapping connections:

When there is a common destination most of the requests are bottlenecked at certain domains as most of the routes are likely to follow the same path. In such situations there are more possibilities of the requests to be dropped. We can analyze this and show that our routing algorithms are more efficient and better than best effort approach in this situation too.

5.4.1 Best Effort routing:

As best effort approach selects a path that has the minimum hop count all the selected paths are static. Here the paths gets congested very easily as no better alternative path is chosen. Let consider the same network graph with different source destination pair. For (source, destination) = (A, J), the network graph is:

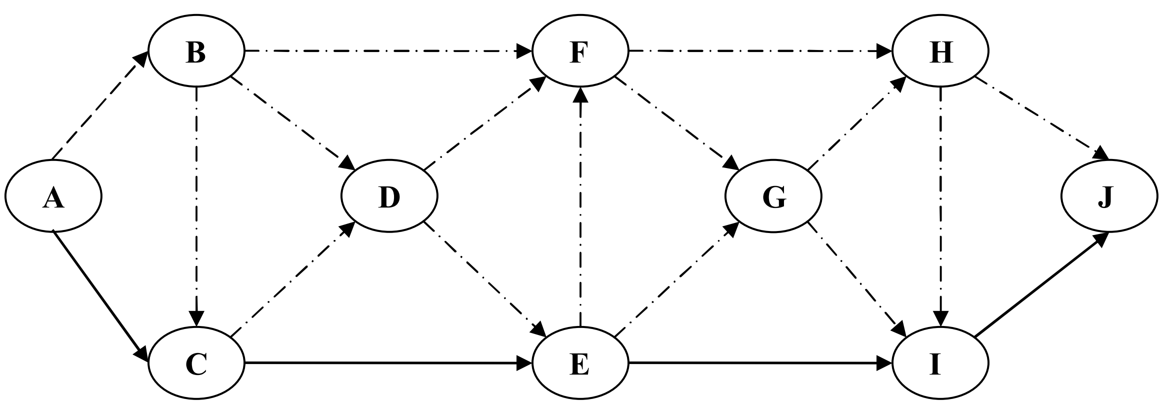

Figure 5.13 Best Effort: Network graph (more overlapping) for connection 1 (A to J)

From the above resources in nodes C, E, I and J are allocated for the traffic and serve the requested bandwidth. When the request keeps increasing the dropping probability increases, even though there is an alternative path. The network graph from B to J is:

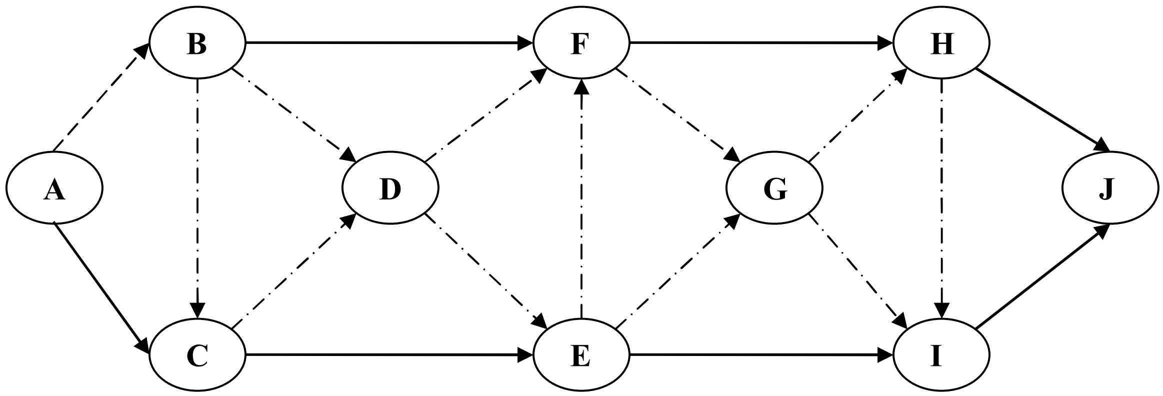

Figure 5.14 Best Effort: Network graph (more overlapping) for connection 2 (B to J)

The dropping probability in domain H is high as it has a domain capacity of 40. Dropping probability increases for requested that demands bandwidth more than 40 units. For another connection from node E to node J the network graph is:

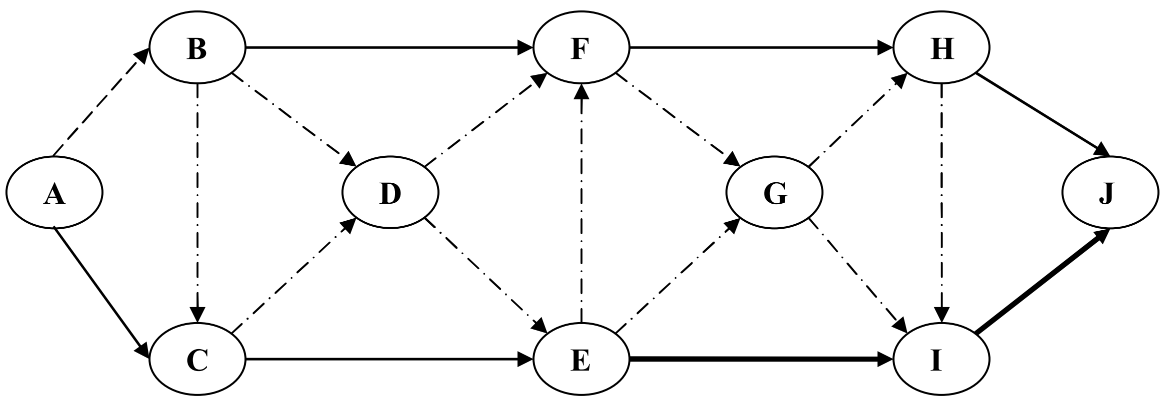

Figure 5.15 Best Effort: Network graph (more overlapping) for connection3 (E to J)

The resources in node I get exhausted as the domain already has traffic load from A to J. In such cases the calls from node A to node J is bottlenecked in node E. For another network connection from node C to I the network graph is:

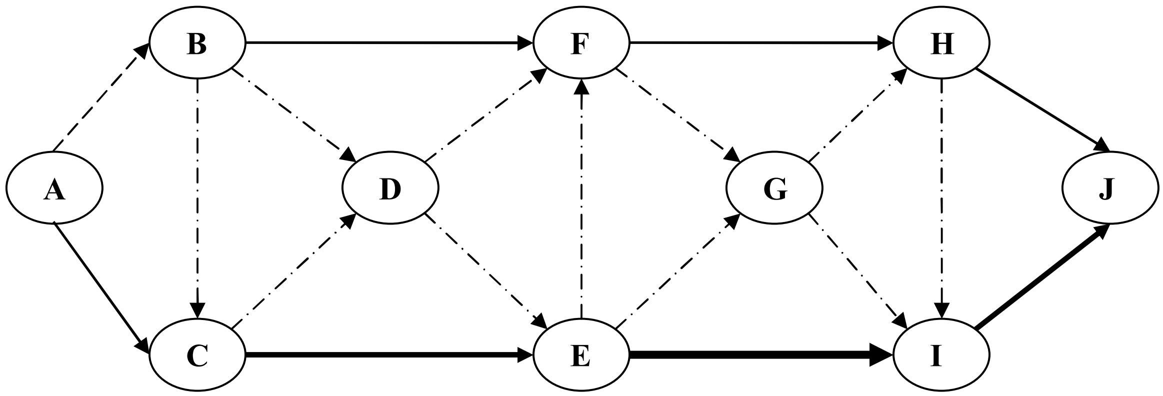

Figure 5.16 Best Effort: Network graph (more overlapping) for connection 4 (C to I)

In this situation the traffic load in domain E get higher and the routing is effected for all the other calls too. This is not a feasible solution when the domain is heavily loaded.

5.4.2 QoS Routing using Dijkstra's Algorithm:

Using the same source, destination pairs we can analyze the performance of best effort routing with Single Constraint QoS Routing. In this case we would find the performance of Best Effort and our approach to be same during the earlier stages of routing. Later when the network gets congested the dropping probability increases as the network gets overloaded. The network graph for connection from A to J is shown below:

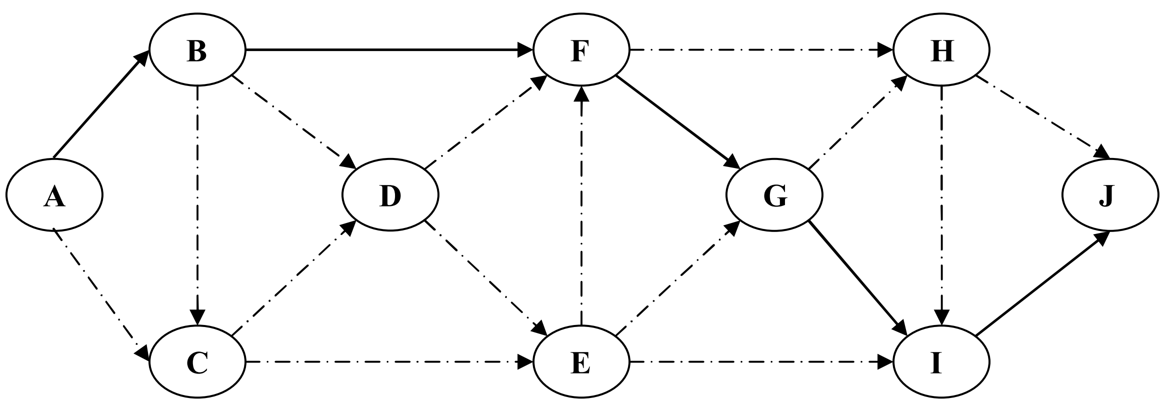

Figure 5.17 Routing by Stochastic knapsack: Network graph (more overlapping) for connection 1 (A to J)

If the traffic request more than 40 unit of bandwidth then the above path is taken else it would take the route A → B → F → H → J. If the traffic in the above path gets congested then another alternative path is taken which is lightly loaded. Consider another connection taken from node B to node J where the network is lightly loaded.

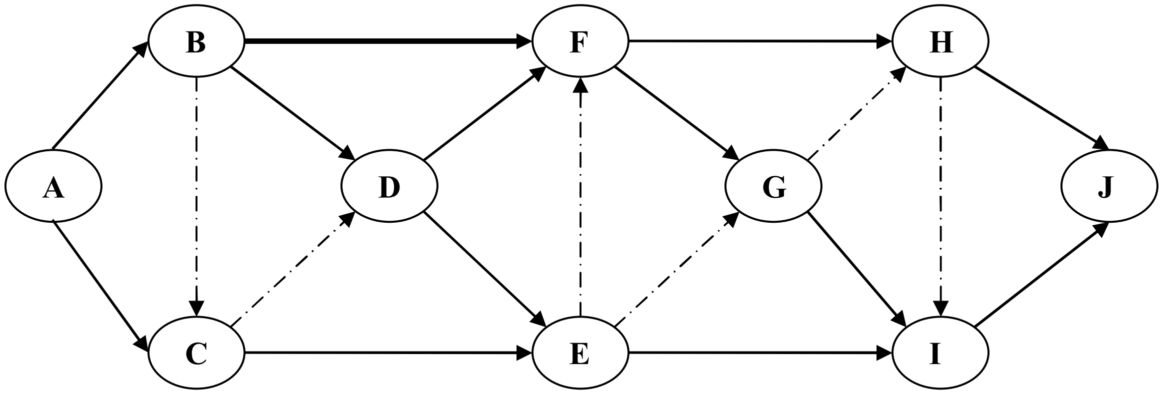

Figure 5.18 Routing by Stochastic Knapsack: Network graph (more overlapping) for connection 2 (B to J)

For traffic from B to J the resources in F and H are used. If any of the paths gets congested the alternative path is taken. If the traffic in the domains increases then the path from domain A to J will also change. When a new call is accepted from source node E to destination J is made then traffic load is increased and must be distributed evenly.

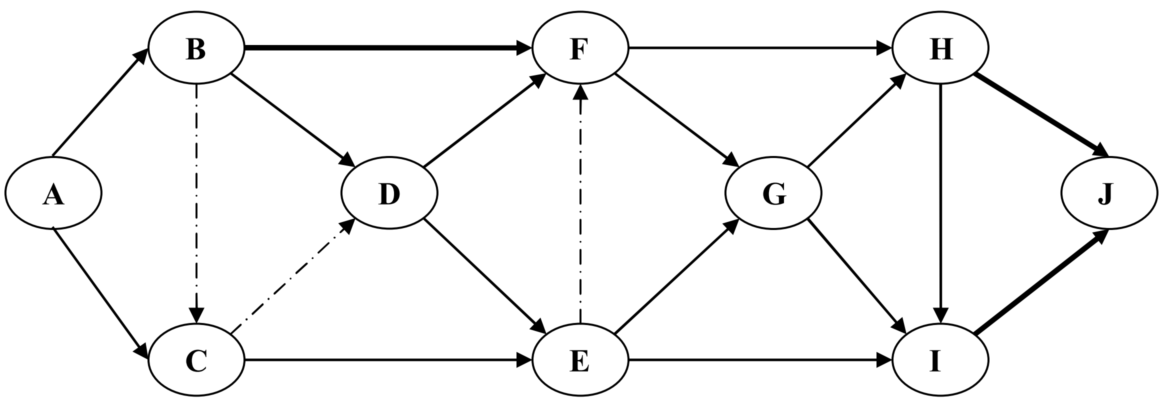

Figure 5.19 Routing by Stochastic Knapsack: Network graph (more overlapping) for connection 3 (E to J)

As resources in domain G were free the routing was done through domain G and then through domain H or domain I depending on the path that has the highest acceptance path. If resources in node G are depleted then the call for node E to node J can be routed through node F. There are two alternative paths from node C to node I, if the traffic load in domain E is low then the routing is done through E with a minimum hop count of 1. If the traffic load in E is high then the traffic is routed as below:

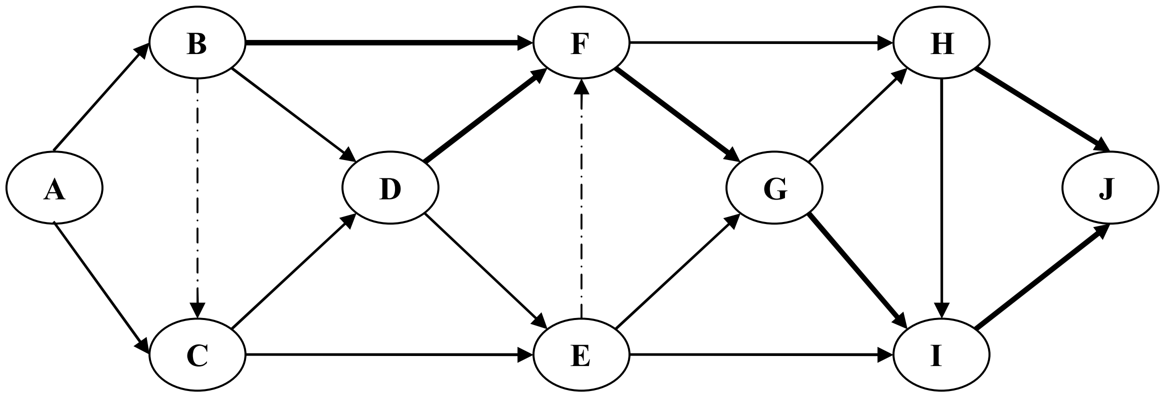

Figure 5.20 Routing by Stochastic Knapsack: Network graph (more overlapping) for connection 4(C to I)

From figure 5.20 the route taken is C → D → F → G → I, as the domain capacities are higher than that of domain E.

The above figures do not give the exact path for all calls. The path mainly depends on the domain capacity, network load the QoS constraints for each call. The above figures give a few possible cases of the paths that can be taken in two different scenarios. In both the scenarios we had noticed that our approach uses more of the network resources when compared to Best effort routing. Doing so our approach balances the network traffic and traffic engineering is done automatically.

CHAPTER VI

RESULTS

The network state information keeps changing based on the arrival rate **(λ)**, mean holding time (1 / **μ**) and the bandwidth requested **(BW)**. Based on the network state information the best path is selected in the QoS Routing. In best effort routing the network state information is not taken into consideration as it routes the path based on the least hop count.

The results are obtained from the route that is computed from two different classes of traffic. Each class of traffic has a distinguished set of QoS parameters. For class 1 traffic the QoS parameters are:

- λ_1 Arrival rate of request for class 1 traffic.
- $1/\mu_1$ Mean holding time for class 1 traffic.
- BW_1 Bandwidth request for class 1 traffic.

Similarly class 2 traffic has distinguished set of QoS parameters as:

- λ_2 Arrival rate of request for class 2 traffic.
- $1/\mu_2$ Mean holding time for class 2 traffic.
- BW_2 Bandwidth request for class 2 traffic.

Different possibilities for the QoS constraints to occur were taken into consideration during the simulation phase. The different QoS constraints were checked for both cases where there are more possibilities of more network paths to overlap and less possibility for network paths to overlap. For a path to have more over lap we route most of the connections to a common destination. This will cause a bottle neck near the destination which will be a worst case scenario in best effort method. The (source, destination) pair taken to show the worst case scenario for the best effort approach are (A, J), (B, J), (E, J) and (C, I). The (source, destination) pair taken to route a path such that different routes are taken to avoid overlapping are (A, J), (C, G), (F, J) and (D, I).

Other than considering the routes that can be taken, the different possibilities for the QoS constraints to occur are also considered. Here we increase and keep certain QoS constraints constant to determine the total amount of bandwidth accepted in each case. The different possibilities for the QoS constraint are:

Scenario	λ_1	λ_2	$1/\mu_1$	$1/\mu_2$	BW1	BW2
Scenario 1	Increasing	Constant	Constant	Constant	Constant	Constant
Scenario 2	Constant	Increasing	Constant	Constant	Constant	Constant
Scenario 3	Constant	Constant	Constant	Constant	Increasing	Constant
Scenario 4	Constant	Constant	Constant	Constant	Constant	Increasing

Based on the above conditions the total amount of bandwidth allocated is obtained. Based on these results we can calculate the ratio of bandwidth that was allocated to the amount of bandwidth requested. Each graph indicates the bandwidth ratio that was accepted when there are more possibilities for overlapping of paths and fewer possibilities for paths to overlap with different order of the QoS constraint. The different scenarios were tested on Best Effort routing and our approach of QoS routing.

6.1 Scenario 1: Acceptance Ratio with Increasing Class 1 Arrival rate:

Here the Class 1 arrival rates is in an increasing order of 1 and the class 2 arrival rates remains constantly at 2. The QoS parameters are:

- Mean Holding time$_1$ (μ_1) = ½
- Mean Holding time$_2$ (μ_2) = 1/5
- Bandwidth$_1$ (BW_1) = 8
- Bandwith$_2$ (BW_2) = 10

The acceptance ratio based for the above QoS parameters and the network graph whose requests have less overlapping paths for best effort and our approach of QoS routing are:

$Class_1$ (λ_1)	$Class_2$ (λ_2)	Proposed App.	Best Effort
3	2	0.9750	0.8364
4	2	0.7915	0.5689
5	2	0.7953	0.5815
6	2	0.6287	0.4348
7	2	0.6282	0.4207
8	2	0.5735	0.3599
9	2	0.4638	0.3257

Table 2 Acceptance ratios: Increasing order of λ_1 on less overlapping paths

In best effort routing protocols as the arrival rate of Class 1 traffic increases the resources along the selected path are exhausted and the acceptance ratio reduces drastically. While in our approach of QoS routing the paths are changed according to the traffic load and capacity in each domain. So at each stage the acceptance ratio reduces very gradually .When the network saturates the acceptance ratio remains almost the same.

Using the same QoS parameters we can find the acceptance ratio for the worst case scenario (higher over lapping paths) for Best Effort routing.

$Class_1$ (λ_1)	$Class_2$ (λ_2)	Proposed Appr.	Best Effort
3	2	0.9456	0.5574
4	2	0.7275	0.2536
5	2	0.7237	0.2412
6	2	0.4677	0.1469
7	2	0.4650	0.1422
8	2	0.2717	0.0952
9	2	0.2717	0.0904

Table 3 Acceptance ratios: Increasing order of λ_1 on more overlapping paths

In best effort routing protocol, for paths that tend to overlap there are many connections competing for resources in the same path. This causes the network to be congested, unbalanced network conditions and low acceptance ratio. On the other hand in our approach for QoS routing the path is selected with the highest acceptance probability so the acceptance ratio is higher than that of best effort.

The above values can be plotted on an X-Y plane to show the performance of best effort routing and Single Constraint QoS routing.

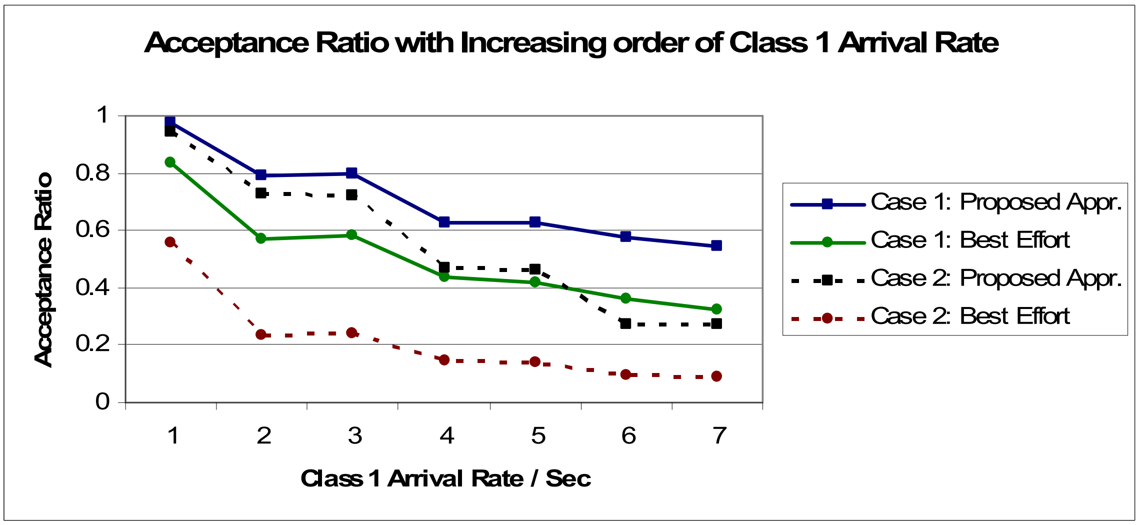

Figure 6.1 Bandwidth acceptance ratios for increasing Class 1 Arrival rate

In figure 6.1, case 1 implies the acceptance ratios in the network graph for situations where there are less overlapping paths. Case 2 implies the condition where there are more overlapping paths. The solid lines denote the acceptance ratio using our approach and the dotted lines show the acceptance ratio using best effort approach. The X-axis denotes the increasing rate of arrival rate in Class1 traffic The Y-axis implies the acceptance ratio on the amount of bandwidth requested to the amount of bandwidth provided. Initially the acceptance ratio is high for both approaches as there is no traffic load in the network. When the domains accept the connection bundle the traffic load keeps increasing thus reducing the acceptance ratio. As the arrival rate of Class 1 traffic increases the accepted bandwidth reduces. When the network saturates then the acceptance ratio reduces slowly. This is because almost all the resources are exhausted and only a few of the arriving call bundles are accepted.

From the graph in figure 6.1 shows, the acceptance ratio in our approach is higher than that of Best effort when the traffic has fewer over lapping paths and for traffic that has higher over lapping paths. Thus our approach of QoS routing is better than Best effort routing when traffic for Class 1 arrival rate is in an increasing order.

6.2 Scenario 2: Acceptance ratio with Increasing Class 2 Arrival rate:

Here the Class 2 arrival rate is in an increasing order of 1 while the Class 1 arrival rate remains constantly at 2. The QoS parameters are:

- Mean Holding $time_1$ (μ_1) = ½
- Mean Holding $time_2$ (μ_2) = 1/5
- $Bandwidth_1$ (BW_1) = 8
- $Bandwidth_2$ (BW_2) = 10

The acceptance ratio for the above QoS parameters with increasing order of class 2 arrival rate and the network graph whose requests have less over lapping paths for best effort and our approach of QoS routing are:

$Class_1$ (λ_1)	$Class_2$ (λ_2)	Proposed App.	Best Effort
2	3	0.9711	0.8042
2	4	0.9718	0.7941
2	5	0.7114	0.5427
2	6	0.7097	0.5419

Table 4 Acceptance ratios: Increasing order of λ_2 on less overlapping paths

From the values in the table 4 we can notice that the difference in acceptance ratios of Best Effort and Proposed Approach is not much initially. The difference in the acceptance ratios increase gradually because most of the paths that are routed do not overlap. But the acceptance ratio in the best effort approach drops very low when there is more number of connection bundles.

Considering another network scenario where more paths tend to overlap. Doing so will exploit more resources along the path if the route is static. The acceptance ratio for such scenarios for our approach of QoS routing and Best Effort is:

Class$_1$ (λ_1)	Class$_2$ (λ_2)	Proposed App.	Best Effort
2	3	0.9391	0.4951
2	4	0.9370	0.4755
2	5	0.6038	0.2290
2	6	0.6025	0.2280

Table 5 Acceptance ratios: Increasing order of λ_2 on more overlapping paths

Table 5 shows the performance of Best Effort routing in the worst case scenario. In Best Effort routing all the paths over lap with each other and the paths are static. So the acceptance ratio is very low though there is no traffic load in the network. When the traffic load is very high in the network then almost all the request are dropped in the best effort approach. The above tabular values are represented in a graphical order as below:

Figure 6.2 Acceptance Ratios with Increasing Order of Class 2 Traffic

Interestingly from the above graph we would notice that the network saturates much faster. This shows the significance of the QoS constraints while routing. The graph also shows that the performance of our approach of QoS routing is better than Best Effort routing when Class 2 arrival rate is in an increasing order.

So far we had seen the QoS constraints where the arrival rates were in increasing order. In section 6.3 we analyze the performance when bandwidth for Class 1 traffic is in creasing order. In the section 6.4 we discuss the performance when Class 2 traffic is in increasing order. The acceptance ratio are plotted for conditions when there are possibilities for fewer traffic is routed over the same path and higher amount of traffic routed on the same path is being analyzed.

6.3 Scenario 3: Acceptance Ratios with Increasing order of Class 1 Bandwidth:

Here the arrival rates (λ_1), (λ_2) and Class 2 bandwidth is maintained at a constant rate while the bandwidth of Class 1 is in an increasing order. The QoS parameters are as:

- Class 1 Arrival Rate (λ_1) = 3 per second
- Class 2 Arrival Rate (λ_2) = 5 per second
- Mean Holding time1 ($1/\mu_1$) = ½
- Mean Holding time2 ($1/\mu_2$) = 1/5

The acceptance ratio for the above QoS parameters and the network graph whose requests have less over lapping paths for best effort and our approach of QoS routing are:

Class 1 (BW_1)	Class 2 (BW_2)	Proposed App.	Best Effort
6	5	0.9633	0.8382
7	5	0.9504	0.7745
8	5	0.9211	0.7452
9	5	0.8947	0.6963
10	5	0.8616	0.6906
11	5	0.7956	0.6166
12	5	0.7548	0.5945

Table 6 Acceptance Ratios: Increasing order of BW_1 on less over lapping paths

From the above table Best Effort performance does not show much improvement than our approach of QoS routing. As less number of overlapping paths tend to occur the difference in the acceptance ratio is not much. But with increasing order of Class 1 Bandwidth we notice that the acceptance ratio becomes very low in the Best Effort routing. Thus we can say that the performance of Single constraint QoS routing is better than Best Effort when less number of paths overlap and with increasing order of Class 1 bandwidth.

Now to consider situations where more number of paths tends overlaps more frequently. With the same QoS constraints the acceptance ratios of bandwidth for Best Effort and our approach of QoS routing:

Class 1 (BW_1)	Class 2 (BW_2)	Proposed App.	Best Effort
6	5	0.9542	0.5547
7	5	0.9030	0.4827
8	5	0.8573	0.4380
9	5	0.7894	0.3942
10	5	0.7814	0.3869
11	5	0.6945	0.3158
12	5	0.6557	0.3032

Table 7 Acceptance ratios: Increasing order of BW_1 on more overlapping paths

Similar to all other situation the acceptance ratios in the Single Constraint QoS routing are higher than Best Effort routing. This shows that Best Effort approach does not consider the QoS constraints and routes based on the least hop count. Thus the difference between the acceptance ratios increases as bandwidth of Class 1 traffic keeps increasing.

The values on table 6 and table 7 can be plotted on an XY plane to show the performance of both approaches.

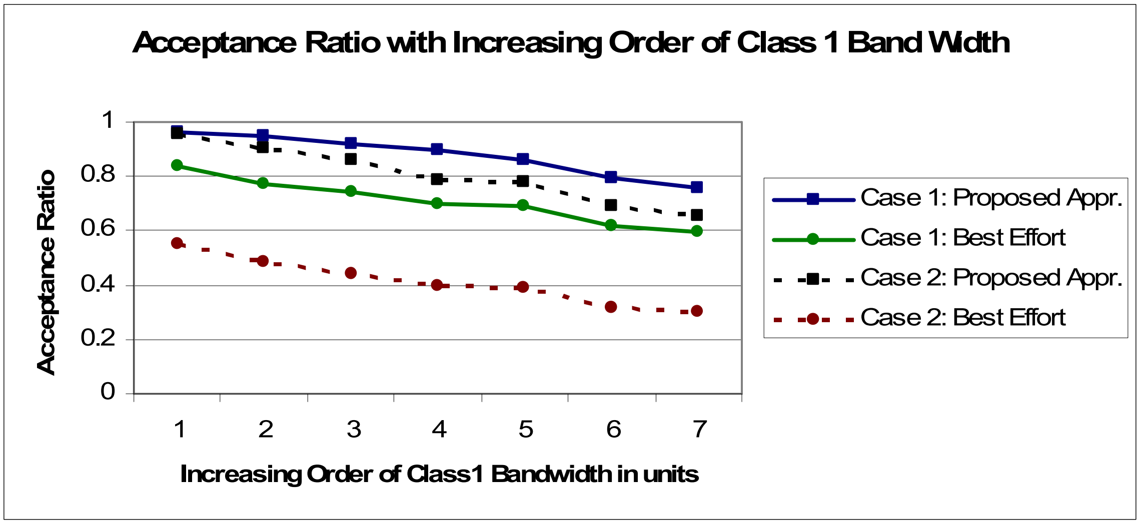

Figure 6.3 Acceptance ratios with increasing order of Class 1 BW units

The X – axis shows the increasing order of bandwidth for Class 2 traffic. The Y – axis shows the acceptance ratio. Initially the acceptance ratio for both network scenarios is almost the same in our kind of QoS routing. This is because the availability of domain resources is initially the same. As the routing process continues the traffic in all the domains increases and the acceptance ratio decreases gradually until the network is completely saturated. While in the Best Effort approach the acceptance ratios are very low in all conditions. This shows that Best Effort approach is not a feasible routing strategy while Class 1 Bandwidth is in an increasing order.

6.4 Scenario 4: Acceptance Ratio with Increasing Order of Class 2 Bandwidth Units:

Here the arrival rates (λ_1), (λ_2) and Class 1 bandwidth is maintained at a constant rate while the bandwidth of Class 2 is in an increasing order. The QoS parameters are as:

- Class 1 Arrival Rate (λ_1) = 3 per second
- Class 2 Arrival Rate (λ_2) = 5 per second
- Mean Holding time1 ($1/\mu_1$) = ½
- Mean Holding time2 ($1/\mu_2$) = 1/5

The acceptance ratio for the above QoS parameters and the network graph whose requests have less over lapping paths for best effort and Single Constraint QoS routing are:

Class 1 (BW_1)	Class 2 (BW_2)	Proposed App.	Best Effort
5	6	0.9643	0.8436
5	7	0.9455	0.7767
5	8	0.9101	0.7201
5	9	0.8809	0.6529
5	10	0.8378	0.6484
5	11	0.7622	0.5700
5	12	0.7137	0.5300

Table 8 Acceptance Ratios: Increasing BW_2 on less overlapping paths

As the bandwidth for class 2 increases the acceptance ratio decreases. Table 8 shows the acceptance ratios are higher while routing using our approach of QoS routing than Best Effort routing.

Now to consider another network scenario where more number of connections tends to overlap we change most of the source, destination pair to a common destination. With the same QoS constraints the acceptance ratios of bandwidth for Best Effort and our approach of QoS routing is:

Class 1 (BW_1)	Class 2 (BW_2)	Proposed App.	Best Effort
5	6	0.9440	0.5456
5	7	0.8900	0.4553
5	8	0.8363	0.3909
5	9	0.7541	0.3269
5	10	0.6464	0.2571
5	11	0.5966	0.2366
5	12	0.5962	0.2352

Table 9 Acceptance Ratios: Increasing order of BW_2 on more overlapping paths

The above table shows that the Best Effort option is not a good approach for any kind of network scenarios, especially when more paths tend to overlap. Even when the traffic in the network is saturated the acceptance ratio for the best effort is lower than the acceptance ratio when routing using our approach of QoS routing. The graph to show the performance of both approaches of routing is:

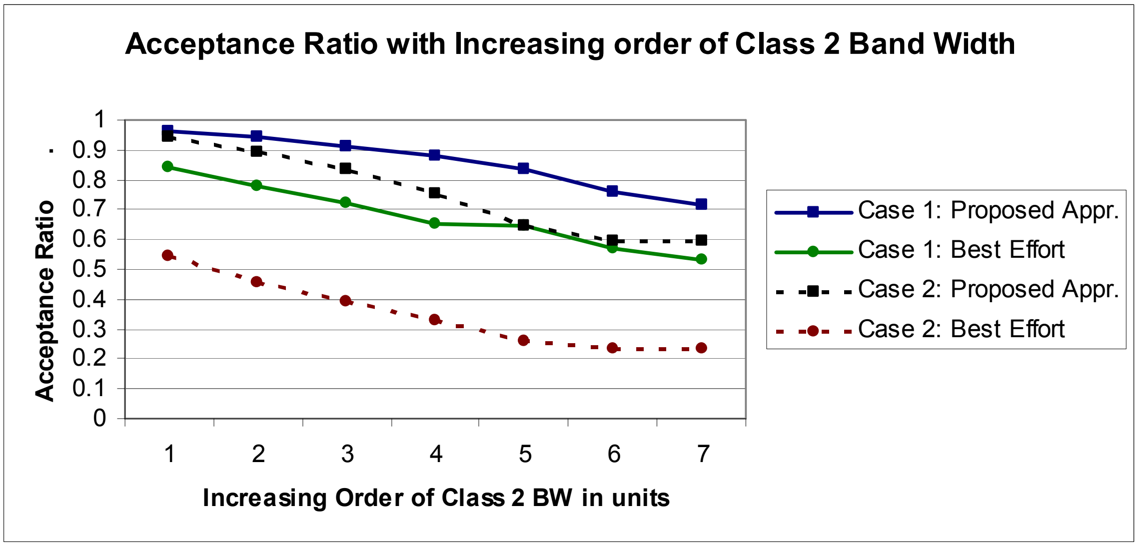

Figure 6.4 Acceptance ratios with increasing Order of Class 2 BW units

The acceptance ratio for the Best Effort is lower in the above graph than that shown in Figure 7.4. Case 1 implies the source, destination pair that route the path which does not overlap very frequently. Case 2 implies the source, destination pair that route the path that overlap frequently. The above graph shows that our approach of QoS routing is better than Best Effort routing in both scenarios when the bandwidth for Class 2 traffic is in increasing order.

From the above different scenarios that were considered so far it has proven that out methodology of QoS routing as a better strategy in routing a path. It has also shows efficient use of resources thus serving more number of requests. There is still a possibility that the Best Effort and Single constraint routing have the same performance. This happens only when there is extremely large number of available resources and the demands of the QoS constraints are small. Such a situation is highly impossible in today’s internet as we have growing applications every day.

CHAPTER VII

CONCLUSION AND FUTURE WORKS

This thesis has shown that Stochastic Knapsack framework as a good model to evaluate the dropping probability from other domains. This model finds the dropping probability based only on the domain capacity and QoS constraint. Thus stochastic knapsack framework requires lesser advertisement of network state information which saves a lot of bandwidth. By knowing the dropping probability this makes it easier to route the path and ensure the amount of resources that can be accepted which also save a large amount of bandwidth. We had also investigated problem faced by Best Effort routing under different kinds of network and QoS constraints that are most likely to occur. The simulation was performed on network conditions where the possibilities of more overlapping paths to occur and possibilities of less overlapping of paths to occur. The QoS constraints too were changed to check performance on increasing order of certain QoS constraints while keeping other QoS constraints constant. In spite of changing the QoS constraints and network condition, QoS routing using Dijkstra's algorithm shows better bandwidth utilization, throughput and efficiency than routing with Best Effort.

Future works will focus on increasing the number of domains and QoS parameters like delay and jitter. The QoS parameters can also be of mixed order where there is an increasing and decreasing order of arrival rate, mean holding time, bandwidth, delay, jitter, or with any of the QoS parameters. The Stochastic Knapsack framework can be implemented in different levels in the hierarchical model of networks. When a call bundle is accepted it is being routed through the path that has the highest acceptance in all levels in the hierarchical model of networks. The duration of the call bundle in each domain can be made finite such that after the expiration of the mean holding time the resources held by the call bundle can be released. This would give a more accurate system showing a steady acceptance ratio when the network is saturated. This will give a wider range to evaluate the dropping probability and provide a more accurate model providing traffic engineering. We can also implement many communication protocols such as UDP and TCP to check for efficiency and fault tolerance. Adding new functionalities to the existing model will ensure better performance under different network conditions. Future works will also involve in reliability and check for fault tolerance.

REFERENCES

[1] X. Xiao and L. M. Ni, "Internet QoS: A Big Picture," *IEEE Network Magazine*, vol. 13, pp. 8--18, Mar. 1999.

[2] S. Chen and K. Nahrstedt, "An Overview of Quality-of-Service Routing for the Next Generation High-Speed Networks: Problems and Solutions," *IEEE Network*, vol. 12, no. 6, pp. 64--79, November/December 1998.

[3] Fang Hao, Ellen W. Zegura: On Scalable QoS Routing: Performance Evaluation of Topology Aggregation. INFOCOM 2000: 147-156".

[4] Yong Tang, Shigang Chen, "QoS Information approximation for aggregated networks", ICC 2004 - *IEEE International Conference on Communications*, vol. 27, no. 1, June 2004 pp. 2107-2111.

[5] Xin Yuan, Guang Yang, "Empirical probability based QoS routing", ICC 2003 - *IEEE International Conference on Communications*, vol. 26, no. 1, May 2003 pp. 1713-1717.

[6] Rick Gallaher, "http://www.convergedigest.com/tutorials/mpls5/page2.asp". (Last Accessed on March 2005).

[7] Dean H. Lorenz, Ariel Orda, "QoS routing in networks with uncertain parameters", *IEEE INFOCOM - Conference on Computer Communications*, no. 1, April 1998 pp. 3-10

[8] Donna Ghosh, Raj Acharya, "A Probabilistic Scheme for QoS Hierarchical QoS Routing", *IEEE International Conference on Computer Networks*, October, 2001.

[9] G. Grimmett and D. R. A. Welsh. Flow in Networks with Random Capacities. *Stochastic*, 7:205–229, 1982.

[10] P. Ferguson and G. Huston, "Quality of Service", John Wiley & Sons, 1998.

[11] Braden, R., Clark, D. and Shenker, S., "Integrated Services in the Internet Architecture: an Overview", Internet RFC 1633, Jun. 1994.

[12] Y.Bernet et al., "A Framework for Differentiated Services", Internet draft, May 1998.

[13] E. Crawley, R. Nair, B. Jajagopalan and H. Sandick, "A Framework for QoS-based Routing in the Internet", RFC 2386, Aug. 1998.

[14] Younis, O., Fahmy, S.: Constraint-based routing in the Internet: Basic principles and recent research. *IEEE Communications Surveys* (2003).

[15] R. Guerin and A. Orda, "QoS-based Routing in Networks with Inaccurate Information: Theory and Algorithms", *IEEE/ACM Transaction on Networking*, 7(3):350-364, June 1999.

VITA

Surendran Ramasamy

Candidate for the Degree of

Master of Science

Thesis: QUALITY OF SERVICE (QoS) ROUTING FOR CONNECTION BUNDLES USING STOCHASTIC KNAPSACK FRAMEWORK

Major Field: Computer Science

Biographical:

Personal Data: Born in Kandy, Sri Lanka, on April 15th 1980, the son of Mr. Ramasamy Subramaniam and Mrs. Thangamani Ramasamy.

Education: Obtained Senior High School Diploma from Brindavan Public School, India in May 1997. Received my Bachelors in Computer Science from Bharathiar University, May 2000 and Post graduate diploma in Computer Science from Madras University, May 2002. Completed the requirements for Master of Science degree at Oklahoma State University, May 2006.

Experience: Jan 2005 – Dec 2005 Web Administrator, Zoology department, Oklahoma State University, Stillwater

Name: Surendran Ramasamy Date of Degree: May, 2006

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: QUALITY OF SERVICE (QoS) ROUTING FOR CONNECTION BUNDLES USING STOCHASTIC KNAPSACK FRAMEWORK

Pages in Study: 48 Candidate for the Degree of Master of Science

Major Field: Computer Science

Scope and Method of Study:

Today's Internet carries different class of traffic which has a distinct set of Quality of Service (QoS) constraints and requires different routes. Traditional routing protocols cannot support the demanded QoS for different traffic classes. Best Effort routing also leads to network congestion and uneven distribution of traffic. These drawbacks have led to the dawn of QoS routing. The objective of this thesis is to determine a path for a connection bundle such that the determined path has the highest probability of satisfying the connection bundle's QoS requirements. This thesis mainly concentrates on stochastic knapsack framework to compute the dropping probabilities in the network and route the best path based on the dropping probability.

Findings and Conclusions:

Many approaches have been done to compute a path from the source to the destination using multiple QoS metric such as delay, jitter, cost and bandwidth. The effectiveness of this design is shown by comparing the results obtained from our method to that of Best Effort routing. The simulation was done over different network and QoS scenarios. The results show the ratio of resources allocated is more in our approach when compared to Best Effort. The results also demonstrate when routing using the proposed approach, the traffic flow is distributed over all areas of network, such that uneven network utilization is avoided thus making traffic engineering automatic.

ADVISER'S APPROVAL: Dr. Venkatesh Sarangan