Scope Of Physical Link Capacities Using Markov Decision Process Routing And Grade Of Service Constraints

K. Sreelatha

Abstract: The problem of Dimensioning in networks is one of the most consistent problems that prevails in our world. In spite of sophisticated networks coming into the market like 4G and 5G networks still it is a long standing problem of our concern. In this paper we highlight the problems that arise in the physical link of a network and also discuss about the various methods by which we can solve this problem. The problem is divided into two sub problems. The first sub problem deals with the design of the topological structure and the second sub problem is to find the network physical links optimizing the revenue given the end to end traffic and Grade of Service constraints. It also gives some light on how to dimension with respect to the absolute and Relative Grade of Service constraints. An attempt is made to calculate the dimensioning of the physical link capacities after acquiring the smooth blocking functions. The problem is reduced by knowing the optimal physical link capacities of the network. A performance model is specified to assess the accuracy of the analytical model with respect to simulation results. Two categories of calls are concerned, a narrow-band call and a wide-band call. A narrow-band call may be a voice application and a wide-band call may be a video application. The system operates in a loss mode meaning if the incoming call finds the network resources like capacity busy, it is lost. Performance measures of the network’s overall blocking probability and the blocking probabilities of the narrow and wide-band call categories are determined. The results from the measurements and the exact model are compared. The gradients of the objective function and the blocking function with respect to capacities are determined and the optimal physical link capacities are determined. Queue length distribution is also studied.

1. INTRODUCTION

1.1 BACKGROUND AND MOTIVATION
Communication services can be provided by various Multi Service Networks [1] such as ATM and IP networks to provide all kind of multimedia services. LAN and MAN multi-service networks were introduced in late 80’s and early 90’s. All of them are connected to internet to provide us with good services. Various switching and multiplexing techniques are used to implement B-ISDN using ATM Networks. Network Interfaces mostly serve as interfaces to create point-to-point point interconnections using ATM links. One needs a virtual channel with which networks are fundamentally connection-oriented, which means that a virtual channel (VC) [4] must be set up across the ATM network prior to any data transfer. Virtual channels are formed and identified by the combination of a VPI and a virtual channel identifier (VCI).

1.2 Objective and Contributions Modeling
The main objective of this paper is to provide physical link capacities for the Markov Decision Routing method such that the average revenue is maximized with respect to absolute blocking constraints. GoS constraints are also taken into consideration.

1.3 Problem formulation
The Problem can be formulated as Maximization of revenue with respect to Absolute Grade of Service (GoS) constraints. The following quantities are assumed:
- Topologies of the physical networks.
- Traffic demand between node pairs.
- Per-class revenue parameters.

Physical Network Model
The network is assumed to be of arbitrary topology. The nodes of the network are interconnected by links. The resulting graph is a bidirectional graph indicating the interconnection of the nodes that carries traffic in both directions.

1.4 Problem formulation
Traffic Assumptions
The following quantities are assumed:
- Topologies of the physical networks.
- Traffic demand between node pairs.
- Per-class revenue parameters.

The Problem can be formulated as Maximization of revenue with respect to Absolute Grade of Service (GoS) constraints.

Objective Function
The objective function is given by:

\[
\max \ W(C,r) = \sum_j W_j \lambda_j [B_j(C,r)]
\]

subject to \( B_j(C,r) \leq B_j^\tau \)

where \( j \) - call class

\( B_j^\tau \) - absolute per class blocking probability

\( C \) - physical link capacity vector

\( W_j \) - call cost

\( \lambda_j \) - traffic load

K. Sreelatha currently is working as Assistant Professor in the department of Electronics and Computer Engineering in Sreenidhi Institute Of Science And Technology (proposed as Sreenidhi University) Ghatkesar Hyderabad, India, +91 7799517667, Email: lathakiran.samudrala@gmail.com.
2. LITERATURE REVIEW

2.1 TRAFFIC MODELS
The heart of any performance evaluation of telecommunications networks is the Traffic models [7]. These models need to be accurate and able to capture the statistical characteristics of the actual traffic. Traffic models can be short-range or long-range dependent. Short-range dependent models include Markov processes and Regression models. These traffic models have a correlation structure that is significant for relatively small lags. Long range dependent traffic models such as fractional Autoregressive Integrated Moving Average (F-ARIMA) and Fractional Brownian Motion (F BM) have significant correlations even for large lags.

2.2 Call Traffic Models
The arrival distribution describes the rate at which source events occur. The arrival rate can have a different interpretation depending on the model of study that the user takes into consideration. There are two types of models that every user can use.

a) Renewal Process
Call arrivals are usually modeled by renewal processes. In a renewal traffic process, the inter arrival times {A_n} are independent and identically distributed (IID), but their distribution is allowed to be general. The superposition of independent renewal processes does not yield a renewal process with a few exceptions. The autocorrelation function of \{A_n\} vanishes for all nonzero lags.

b) Poisson Model
This is one of the oldest traffic model. A Poisson process can be characterized as a renewal process whose inter arrival times \{A_n\} are exponentially distributed with rate parameter \( \lambda \): P (A_n \leq t) = 1 - e^{-\lambda t} . Equivalently, it is a counting process, satisfying:

\[
P(N(t)) = n - \frac{e^{-\lambda t} \lambda t^n}{n!}
\]

and the number of arrivals in disjoint intervals is statistically independent. This property is known as independent increments. This property renders Poisson a memory less property. This simplifies the queuing problems involving Poisson arrivals. The superposition of independent Poisson processes results in a new Poisson process whose rate is the sum of the component rates. Poisson processes are fairly common in traffic applications that physically comprise a large number of independent traffic streams, each of which can be quite general. Many services in a multi-service networks such as voice, video-telephony, file transfer, remote login are believed to yield Poisson call arrival processes.

c) Self-similar Models
Self-similar [18] phenomena displays structural similarities over a wide range of time scales. In the case of Ethernet traffic, self-similarity is manifested in the absence of a natural length of a burst; at every time scale ranging from a few milliseconds to minutes and hours, bursts consist of bursty sub periods separated by less bursty sub periods.

Assume \( X_k \) be a wide-sense stationary process with

\[
E[X_k] = \bar{X} \quad \text{and autocorrelation function } r(k) .
\]

Consider the process \( X^m_k \) (m = 1, 2, ..., k) that are constructed out of \( X_k \) as

\[
X^m_k = X_k = m^{-1-H} x^m_k \quad \text{ie i.e. by averaging over non overlapping blocks of size } \text{.}
\]

The process \( X^m_k \) are k. Also a wide sense stationary, with mean \( X \) and autocorrelation function \( r_m(k) \). The process \( X_k \) is called exactly self-similar if it satisfies

\[
V_k = m^{1-H} V_k
\]

where \( H \) is the degree of self-similarity. That is, the processes \( X_k \) and \( m^{1-H} V_k \) should have the same finite-dimensional distributions for all aggregate levels of m. The process \( X^m_k \) is called asymptotically self-similar if it satisfies \( r_m(k) = r(k) \) for m \( k \rightarrow \infty \). The self similar arrivals of Web connections can be accurately modeled by a renewal process with inter arrival times following a Weibull distribution.

2.3 Call Holding Time Models
The service time distribution describes the amount of time a packet remains in the server. Holding times of many interactive services such as telephony are accurately described by the exponential distribution with an of exception Internet sessions established at an ISP which have holding times more accurately described by a Weibull distribution. The traditional model of call holding times \( B_n \) is the negative exponential distribution with rate parameter \( \mu : P(B_n \leq t) = 1 - e^{-\mu t} . A more recent model is the Pareto distribution which is a

\[
Pr[B_n \leq t] = 1 - (k/t^a), \quad k > 0 . A \text{ distribution is said to be heavy tailed if } Pr[X > x] \sim CX^{-a} . a = 1 .
\]

Intuitively, a heavy tailed holding time distribution
means that if the call has not been completed for some time it becomes more and more unlikely that it will be completed soon.

3. RESOURCE MANAGEMENT

3.1 INTRODUCTION

Resource Management [15] may be defined as the use of control mechanisms to regulate access to network transmission, switching, and control equipment in such a way as to support the offered traffic at an acceptable grade and quality of service. A preventive QoS and GoS control approach is typically taken, in which network resources such as link capacity and switch/router buffer capacity, are reserved before transferring any information. From the network operators point of view resources should be allocated in such a way that the revenue is maximized. In addition to the freedom in design of the different resource allocation functions, the call charging function provide a tool for controlling the revenue.

3.2 Call Admission Control

Call Admission Control (CAC)[1] is one of the primary mechanisms for the traffic management in ATM networks. CAC is used for networks which provide QoS guarantees such as circuit switched networks, ATM networks and QoS enhanced IP networks. CAC is a part of preventive congestion control scheme. Call admission control deals with the problem of accepting a new connection or not. CAC is implemented along with the routing. Typically, the decision to accept or reject a new connection is based on the following:

1. The QoS requirements for network calls.
2. The Grade of Service (GoS)[1] for network call classes.
3. The call reward parameters.

CAC is realized by two sub functions: \( \text{CAC}_{\text{QoS}} \) and \( \text{CAC}_{\text{GoS}} \). The purpose of \( \text{CAC}_{\text{QoS}} \) function is to decide whether a particular path is expected to offer sufficient QoS to existing calls as well as the new call. If sufficient resources are expected, the call may be accepted on the particular path, otherwise it must be rejected. The \( \text{CAC}_{\text{GoS}} \) function decides if carrying a call on a certain path is economical or not and also decides whether the path is consistent with the GoS parameters. For example if wide band calls produce larger reward than the narrow band calls the \( \text{CAC}_{\text{GoS}} \) function may decide to reject the narrow band calls when the path just has free capacity to one more wide band call. Call admission schemes [8] may be classified into:

1. Non-static allocation or peak bandwidth allocation.
2. Static allocation
3. Non-static allocation

Suppose a source has an average bandwidth of 2 mbps and a peak bandwidth of 5 mbps. Peak bandwidth allocation requires that 5mbps be reserved at the output port for the specific reason, independent of whether the source transmits continuously at 5mbps. Peak rate allocation is used in CBR services, which are suitable for applications such as: PCM-encoded voice and other fixed rate applications, un-encoded video and very low bandwidth applications such as telemetry.

2. Static Allocation

In static allocation, bandwidth for a new connection is not allowed on per peak rate basis rather the allocated bandwidth is less than the peak rate of the source. As a result, the sum of all peak rates may be greater than the capacity of the output link. This allocation makes economic sense when dealing with bursty sources, but it is difficult to carry out effectively due to its inability to characterize the arrival process of ATMs.

3.3 The notion of Effective Bandwidth

Effective bandwidth [9] approach is one of the proposed methods to perform CAC in B-ISDN networks. The essential idea behind this approach is to estimate the required bandwidth (i.e. service rate) for a set of traffic sources from a queue model, given the source characteristics, the buffer size B and QoS constraints (e.g. the allowable packet loss probability e, which typically is chosen in the order of 10^{-5} - 10^{-9}). This estimate for the effective bandwidth will vary between the mean and peak rate of the cumulative traffic; the burstier the traffic is, the closer the effective bandwidth will be to the peak rate. To simplify the CAC in B-ISDN networks, it is more convenient to use the notion of effective bandwidth. To decide if a new connection can be accepted, we then simply compare the sum of effective bandwidths for the traffic streams with the available link capacity.

3.4 Call Set-up modes

ATM is a connection-oriented service. Before a user starts transmitting over an ATM network, a connection has to be established. This is done at Call set-up time. The main objective of this procedure is to establish a path between the sender and the receiver. This path may involve one or more ATM switches. On each of these ATM switches, resources are released after the connection is terminated. In delay networks the call is placed in the queue before the call is set-up. This mode is known as Delayed call set-up mode. In loss networks if a wide band call arrives the call has to be set-up on demand as a wide band call will fetch more reward. This is known as on demand call set-up mode. The network operates in loss mode for narrow band calls where in if the network is found busy the call is rejected. We have also mixed loss delay networks which operate in both delay and loss mode.

3.5 Routing Methods

Routing [11] is the act of moving information across an internetwork from a source to a destination. Routing involves two basic activities: determining optimal routing paths and transporting information groups (typically called packets) through an internetwork.
Routing in circuit switched networks
The routing algorithms[1] for non-hierarchical circuit switched networks are as follows:

1. Non alternative routing
   a) In Fixed non-alternative routing calls are offered only to a single path. If the single path is busy the call is rejected. Optimal fixed routes can be determined by solving a constrained optimization problem which tries to assign call flows to paths and links so that the unused capacity is minimized under flow conservation and capacity constraints. In Fixed load sharing routing the call is routed according to a set of fixed load sharing probabilities which assigns probabilities for selecting different routes for a call from each class. Calls are only offered to one path, and if the path is busy, the call is blocked.
   b) In Sticky random routing the call is offered to the direct path. If it is available, the call is established. Otherwise, the call is offered to a previously chosen alternative path. If the alternative path is available, the call is established. Otherwise the call is blocked and the alternative path is reselected (the path is chosen at random according to uniform probabilities). The new alternative route is used by the next call that finds the direct path busy.
   c) Fixed alternative routing chooses paths for new calls according to fixed alternative sequences. The first path in the sequence to be tried is the direct path. If it is busy, the remaining paths are tried, in order, until a path that is able to accept the call is found, or there are no more paths in the sequence in which case the call is rejected. The optimal sequence can be determined by evaluating the average reward rate of different sequence choices.
   d) Least loaded routing is the first example of state-dependent alternative routing. The call is offered to the direct path. If it is busy, an alternative path is searched according to a state-dependent routing rule. If no such path is found the call is rejected.
   e) The state-dependent routing rule selects a path with sufficient capacity that also has the largest free capacity of its bottleneck link (the link with least free capacity along the path). The call blocking probability for the classes can be determined by solving Erlang fixed point equations (reduced load approximation) and Trunk reservation[1] may be used to protect wide-band calls from narrow-band calls for all the above methods.

4. Routing in IP networks
The Routing in IP networks is divided into two categories Intra-domain and Inter-domain routing.

4.1 Intra-domain routing
The intra-domain routing protocols are either based on shortest path routing (implemented by Bellman ford algorithm) or on link state routing (implemented by Dijkstra algorithm) used to compute the shortest path routes between source and destination. The context for the shortest path routing is a network with two weights associated each link-one weight for each direction. The weights measure the distance or cost of crossing the link. The metric (weight) could be packet delay or bandwidth or some link cost. Distance vector routing[1] operates by having each router maintains a routing indexed by, and containing one entry for each router in the domain. This entry contains two parts; the preferred outgoing line to use for that destination, and an estimate of cost to that destination. The tables are updated by exchanging information with the neighbors. It is assumed that each node knows the weight of the link to each of its directly connected neighbor.

4.2 Inter-domain routing
The inter-domain routing is more focused on finding loop-free paths which can reach the destinations than on finding optimal paths. The Autonomous Systems AS (which is a portion of the network under control of a single administrative unit such as a campus) use a set of policies to guide the selection of paths. The Border Gateway Protocol (BGP) is the current inter-domain routing protocol in the Internet. When configuring BGP, the administrator of each AS pick at least one node to be a BGP speaker, which is essentially a spokes person for the entire AS and establishes BGP sessions (used to exchange reach ability information among ASs) to other BGP speakers in other ASs.

5. Routing in ATM networks
ATM network routing is implemented by Private Network-Network Interface (PNNI)[13] which consists of a routing protocol and a signaling protocol.

5.1 PNNI protocol
PNNI is a hierarchical routing protocol. The network is divided into peer groups with each peer group electing a peer group leader representing the peer group as a single logical node in a hierarchical level. Each node determines the status of links and neighbors and synchronizes its topology database with that node and provides topology state parameters information describing characteristics to other nodes in the same group to which it has already synchronized.

5.2 Network Dimensioning Methods
In order to simplify the Resource management the concept of virtual networks is introduced. This concept allows the separation of the resource management functions[1] in order to allow customization to particular needs of some services and user groups and the virtual separation of resources to provide Grade of Service (GoS) guarantees for some services and user groups. A Virtual Network [1] is defined by a set of network nodes and a set of virtual network links connecting the nodes. Allocation of resources to virtual and physical networks is referred to as virtual/physical link capacity dimensioning. It must be able to determine the optimal size (bandwidth requirement) of the network links (physical or virtual) given the end to end traffic and GoS constraints optimizing an objective function. Grade of
service = number of blocked calls/total offered calls. The GoS constraints can be absolute in which case the call blocking probability for a given OD pair and call category should be less than a given value or relative in which case the ratio between the call blocking probability and the maximum call blocking probability over the OD pairs for same category is specified.

6. MDP Based Call Admission Control and Routing

An optimal solution of the Call admission control and routing problem [19] in multi-service loss network, in terms of average reward per unit time, is possible by modeling the network behavior as a Markov Decision Process (MDP). The objective is to maximize the revenue from carried calls, while meeting the constraints on the QoS and the GoS on the packet and call level, respectively. Two categories of calls are considered: a narrow-band (NB) requesting a bandwidth of $b_1$ Mbps and a wide-band (WB) $b_W$. The required bandwidth is represented by the call's peak bandwidth in case of deterministic multiplexing, and by the call's effective bandwidth in case of statistical multiplexing. MDP based CAC routing mechanism is state-dependent rather than static, which means the decision to reject the request for a new call, or to accept it on a particular path depends on the current occupancy of the network. A state-dependent mechanism offers advantages both in terms of achievable revenue and ability to control the QoS and GoS.

6.1 State Dependent Routing

The network is assumed to consist of a set of switching nodes, interconnected by bi-directional links according to some network topology. Each bi-directional link consists of two unidirectional links, carrying traffic in opposite directions. The network is offered traffic from K classes which are, for sake of simplicity, assumed to be subject to deterministic multiplexing. The j-th class is characterized by the following: origin destination (OD) pair, bandwidth requirement, bandwidth requirement $b_j$[Mbps], Poison call arrival process rate $\lambda_j$ [calls/s], Exponentially distributed call holding time with mean $1/\mu_j$ [s], set of alternative routes $W_j$, Reward parameter $R \in (0, \infty)$. The classes are classified into G bandwidth categories. The i-th category is characterized by: bandwidth requirement $b_i$ [Mbps], average call holding time $1/\mu_i$ [s], average reward parameter $r_i$. The task is to find an optimal routing policy $\pi$ which maximizes the mean reward from the network, defined as:

$$R(\pi) = \sum_{j \in J} \tau_j \lambda_j \pi,$$

where $\lambda_j \pi$ denotes the average class-j call acceptance rate.

6.2 MDP Modeling

6.2.1 Network Decomposition

In the exact MDP framework, the network state and policy spaces can be very large even for moderate-sized networks. We therefore decompose the network into a set of links assumed to be independent traffic and reward processes, respectively. The network Markov process is decomposed into a set of independent link Markov Processes, driven by state-dependent Poisson call arrival processes with mean rate $\lambda_j(x,\pi) s^{-1}$, where $s$ denotes the link index; $x$ denotes the link state and $\pi$ denotes the CAC and routing policy. In particular, a call connected on a path consisting of I links is decomposed into I independent link calls characterized by the same mean call holding time as the original call. The network reward process is decomposed into a set of separable link reward processes. The link call reward parameters $\tau_j = \sum_{s \in S_k} r_j s(\pi)$ where $S_k$ denotes the set of links constituting path k, specified by the routing policy.

Exact link MDP model:

The state in the exact link model is given by $x-\{xi\}$, where $xi$ denotes the number of category calls on the link. The state space for $X$ for the exact link model is given by $X = \{x-\{xi\}\}$.

$$\sum_{j \in I} b_j x_j \leq C_s$$

where $C_s$ denotes the capacity of the link s. The size of the state pace grows like $S \sim \frac{1}{g} \prod_{i \in I}(N_i + 1)$ where

$$N_i = \left[ \frac{C_s}{b_i} \right]$$

denotes the maximal number of category calls on the link. The Markov decision action $a$ is represented by a vector $\{ai\}$ corresponding to admission decisions for presumptive call requests. The action space is given by $A = \{a = \{ai\}: a_i \in \{0,1\}, i \in I\}$. The corresponding admission decisions for presumptive call requests. The permissible action space is a state-dependent subset of $A$:

$$A(x) = \{a \in A: a_i = 0 \text{ if } x + \delta_i \notin X\}, i \in I$$

where $\delta_i$ denotes a vector of zeros except a one in position $i \in I$.

The Markov chain is characterized by state transition probabilities $p_{x,y}(a)$ which express the probability that the next state is $y$, given that action $a$ is taken in state $x$. In our case the state transition probabilities become

$$P_{x,y}(a) = \begin{cases} \mu_i(x,a), y = x - \delta_i \in X, i \in I \\ 0, \text{ otherwise} \end{cases}$$

where $\lambda_j^s(x,\pi)$ denotes the i-th category arrival rate to the link in the state $x$ under routing policy $\pi$, $\mu_i$ denotes the average departure rate of category-i calls and $\xi(x,a)$
denotes the average sojourn time in state \( x \). The link call arrival rates \( \lambda_{ij}^{s}(x, \pi) \) are given by:
\[
\lambda_{ij}^{s}(x, \pi) = \sum_{\text{net}} x_{ij}^{t}(x, \pi) \varphi_{ij}^{s}(x, \pi) \prod_{c \in S_k(x, \pi)} \left(1 - B_{jc}(\pi)\right)
\]
where \( s \in S_r \) and this denotes denotes the probability that link \( c \) has not enough capacity to accept a class \( j \) call, and \( \varphi_{ij}^{s}(x, \pi) \) denotes a filtering probability defined as
\[
\varphi_{ij}^{s}(x, \pi) = P \left\{ \sum_{c \in S_k(x, \pi)} p_{ij}^{c}(x, \pi) < p_{ij}^{s}(x, \pi) \right\}
\]
here \( B_{jc}(\pi) \) denotes the condition that no link on path \( k \) is in the blocking state. \( \lambda_{ij}^{k}(\pi) \) denotes the average rate of accepted class \( j \) calls on path \( k \), and \( \lambda_{ij}^{k}(\pi) \) denotes the arrival rate of class \( j \).

The expected reward in state \( x \) is given by
\[
R_{ij}^{s}(x) = \mu_{ij}^{s}(x, \pi) \tau(x, a)
\]
where \( p_{ij} \) denotes the probability that an arbitrary single call found on the link is from class \( j \). The average sojourn time in state \( x \) is given by
\[
\tau(x, a) = \left\{ \sum_{i \in I} x_{ij}^{t}(x, \pi) + a_{ij} x_{ij}^{t}(x, \pi) \right\}^{-1}
\]

### 6.2.2 MDP COMPUTATIONAL PROCEDURE

The central idea is to compute path net-gain functions, \( g^{k}(y, \pi) \) which estimate the increase in the long-term reward due to admission of a class \( j \) call on path \( k \) in network state \( y \). The CAC and routing policy is simply to choose, given the state of the network and class of the call request, a path which offer maximal positive path net-gain among the paths with sufficient QoS. The call is rejected if the path net-gain is negative, or if no path would offer sufficient QoS. This is shown in Fig.1.

The call is offered to a path which has sufficient QoS and maximal positive net-gain among the \( H = |W_j| \) alternative paths. The state-dependent path net-gain is defined as:
\[
g_{ij}^{k}(y, \pi) = p_{ij}^{s}(y, \pi) - q_{ij}^{s}(y, \pi)
\]

where \( y \cdot x \) denotes the network state in the decomposed network model. The link shadow price \( p_{ij}^{s}(x, \pi) \) can be interpreted as the expected cost for accepting an \( i \) -th category call in state \( x = (x_i) \). In the reward maximization MDP framework, the link shadow price is defined as follows:
\[
R_{ij}^{s}(x_0, \pi, T) = E \left[ \int_{t_0}^{t_0+T} q_{ij}^{s}(x(t))\,dt \right]
\]

where \( q_{ij}^{s}(x(t)) \) denotes the ejected reward accumulation rate in state \( x(t) \), the process \( x(t) \) driven by a probabilistic law of motion specified by certain state transition probabilities. The relative values can now be written as:

\[
g_{ij}^{k}(y, \pi) = V_{ij}^{s}(x + \delta_{ij}, \pi) - V_{ij}^{s}(x, \pi)
\]

where \( V_{ij}^{s}(x_0, \pi) \) denotes the relative value for category \( i \) in state \( x \) and \( \delta \) denotes a vector of zeros except for one in position \( i \). The expected link reward, obtained in the interval \((t_0, t_0+T)\) of length \( T \) assuming state \( x_0 \) at time \( t_0 \), is given by
\[
R_{ij}^{s}(x_0, \pi, T) = E \left[ \int_{t_0}^{t_0+T} q_{ij}^{s}(x(t))\,dt \right]
\]

where \( q_{ij}^{s}(x(t)) \) denotes the ejected reward accumulation rate in state \( x(t) \), the process \( x(t) \) driven by a probabilistic law of motion specified by certain state transition probabilities.
\[ V^s(x_0, \pi) = \lim_{T \to \infty} [R^s(x_0, \pi, T) - R^s(x_r, \pi, T)] \]

That is the relative value in the state \( x_0 \) is defined as the difference in future reward earnings when starting in the given state, compared to reference state \( x_r \). In practice the relative value is obtained by solving linear equations. The algorithm of CAC is used for this purpose. The algorithm for determining the CAC and routing policy can be summarized as follows:

1. **Startup**: Initialize the relative values \( V^j(x) \) in a way that make all link net-gains with permissible admission positive.
2. **Online operation phase**: Measure per-path call acceptance rates \( \lambda_i^s(\pi) \) and per-link blocking probabilities \( B_j^s(\pi) \) while employing the maximum path net-gain routing rule. Perform the measurements for a sufficient large period for the system to attain statistical equilibrium.

   that maximizes the relative value in each state.

### 7. NETWORK DIMENSIONING

The problem of Network Dimensioning [14] is divided into two sub problems. First the design of the topological structure (location of the nodes and interconnection among them) and determine the optimal size of the network physical links given the end to end traffic and GoS constraints of each service class.

There are two approaches used in the dimensioning problem[1]. The first approach to use link capacities as optimization variables and the second approach uses link performance characteristics as optimization variables. In the first approach the application of the general optimization methods is practical only when the continuous gradients of the constraint functions with respect to the CAC and routing parameters can be obtained. This is feasible only when the CAC and routing strategy is defined by a set of continuous parameters, as in the case of load sharing routing and MDP routing. In fact, in these cases the inequality constraints can be replaced with equality constraints which can simplify the solution. The first approach is preferred in this case.

### 8. PERFORMANCE EVALUATION

The primary objective of performance evaluation [20] is to assess the accuracy of the analytical model with respect to simulation results.

#### 8.1 The Performance Model

The independence assumption requires constructing two functions: link function and link loading function. The link function relates the link performance information vector to each link input (arrival rates, holding times etc). The link loading function defines vector of link inputs as a function of the link performance vector and a given routing policy. The performance information can be full state probability distribution and can include only macro state probabilities. The two sets of equations defined by the

3. **Policy iteration cycle**: At the end of the measurement period, perform the following steps for all the links \( s \) in the network:

   - Identify the link MDP model:
     - Determine per-category reward parameters and link arrivals
   - Value determination: Find the relative values \( V^j(x, \pi) \) for the current routing policy \( \pi \).
   - Policy improvement: Improve the link CAC policies \( \pi \), based on the new relative values.

4. **Convergence test**: repeat from step 2 until average reward per time unit converges.

According to MDP theory an optimal policy is found after a finite number of policy iterations in case of a finite-state and policy space. The value determination step for link \( s \) determines the relative values \( V^j(x, \pi) \) for all sates and the policy improvement step for link \( s \) consists of finding the action.

The link function and link loading function are solved by means of repeated substitutions. This scheme is known as fixed point equations or reduced load approximation. The performance measures are the blocking probability for each \( a-d \) pair and the average blocking in the network. The link function and the link loading function define a set of non linear equations.

\[
\begin{align*}
\lambda &= f(\Pi, \pi) \\
\Pi - g(a) &= \text{(eq2)}
\end{align*}
\]

where

- \( \Pi \) is the link performance vector
- \( \pi \) is the given routing policy
- \( a \) is the set of link inputs

These equations are solved by repeated substitutions and are known as fixed point equations. In this model the link input is defined by the state dependent Poissonian arrival rates, \( \lambda_j^s(x, \pi) \) and the link performance is defined by state dependent link shadow prices \( p_j^s(x, \pi) \). Thus the link loading function and link function are described as

\[
\begin{align*}
\lambda(x) &= f(Q(x), p(x, \pi)) \\
Q(x) &= g(\lambda(x))
\end{align*}
\]

The main difference compared with the traditional set of fixed point equations is that the values of the shadow prices (determining the policy) are not given and the equations cannot be solved in the present form. The solution is to find shadow prices from the policy iteration algorithm. To evaluate shadow prices of the vectors \( \lambda = \{ \lambda_j^s \} \) and \( r = \{ r_j^s \} \) should be calculated. In the policy iteration algorithm these values are calculated based on statistics and a simplified analytical model.
Since these statistics are functions of current shadow prices and the network state distributions, in the decomposed model we have
\[
(\lambda, r) = \nu(Q(x, p), p(x, \pi)) \quad \text{----------(eq 5)}
\]
Then the improved policy, \(\pi'\) is defined by new values of shadow prices
\[
P(x, \pi') = u(\lambda, r) \quad \text{----------(eq 6)}
\]
Equation 5 can be seen as the link loading function for the shadow price link model and equation 6 as the link model for the shadow price evaluation. The final policy \(\pi^*\) cannot be evaluated from these equations since there is no model for \(Q(x)\), evaluation.
Note the complementary structure of the fixed point equations (3,4) and policy iteration equations (5,6) with respect to unknown variables. This feature suggests the solution of both equation sets in one iteration procedure applied to the following extended set of equations:
\[
\lambda(x) = f(Q(x), p(x)) \quad \text{----------(eq 7)}
\]
\[
Q(x) = g(\lambda(x)) \quad \text{----------(eq 8)}
\]
\[
(\lambda, r) - \nu(Q(x), p(x)) \quad \text{----------(eq 9)}
\]
\[
p(x) - u(\lambda, r) \quad \text{----------(eq 10)}
\]
Assuming that by repeated substitutions solution is achieved both optimal policy and performance will be evaluated.

8.2 The Computational Procedure
The equations 7, 8 and 10 are computed using the MDP model and the equation 9 is computed using the load sharing model.

8.2.1 MDP Blocking Evaluation
The network is offered calls from \(K\) classes which are divided into \(G\) categories. Let \(j \in \{1, ..., K\}\) denote a class, and \(i \in \{1, ..., G\}\) denote a category. Consider a path with \(n\) links .the state of the link \(s\) in the path is
\[
X^s_j = \{x^s = (x^s): C^s - \sum b_i x_i^s \geq b_j \}
\]
The set of the rest of the path is
\[
X^S = \{x^1, ..., x^{s-1}, x^{s+1}, ..., x^n\}
\]
Let \(n^k\) denote the number of links in path \(k\). Let \(W_j\) denote the set of alternative paths for class \(j\). Let \(p^k\) denote the random variable for the shadow price on the path \(k\). The OD pair blocking is
\[
Y^r = C^r - \sum b_i X_i^r \quad \text{for path } j \in \{1, ..., W_j\} \quad \text{Pr}\{Y^r \geq b_j\}
\]
where \(Y^r\) denotes free capacity on the link rel. Let \(P_j^k(x^s)\) denote the probability of routing a class \(j\) call on path \(k\) when the link \(s\) is in the state \(x^s\). We have
\[
P_j^k(x^s) = \sum_{x^s} \frac{\pi_j^k}{\Pi_{i \in W_j(k)}} \prod_{i \in W_j(k)} \Pr[p^k < p^i, y_1 \in W_j(k)]
\]
The arrival rate in the state \((x^s)\) is given by
\[
\lambda_j^s(x^s) = P_j^k(x^s)\lambda_j
\]
Once the arrival rate are found from the state dependent probabilities \(Q^s(x^s)\) can be found from balanced equations. The probability \(P^k(x^s)\) can be written as,
\[
P_j^k(x^s) = \sum_{x^s} \frac{\pi_j^k}{\Pi_{i \in W_j(k)}} G_p p^1 \left\{ (p^k) \overline{Q} \right\}
\]
since paths are independent by assumption we have
\[
P_j^k(x^s) = \sum_{x^s} \frac{\pi_j^k}{\Pi_{i \in W_j(k)}} G_p p^1 \left\{ (p^k) \overline{Q} \right\}
\]
where
\[
\left(\overline{Q} \right)(\overline{x}) = \prod_{\tau \neq s} Q^\tau \left( x^\tau \right) \quad \text{since the links are independent}
\]
Let
\[
F_{p^k}(v) = \Pr[p^k \leq v]
\]
define
\[
G_p p^k(v) = 1 - F_{p^k}(v)
\]
we have
\[
\Pr[p^k < p^k] = G_p 1 (v)
\]
which gives the result
\[
P_j^k(x^s) = \sum_{x^s} \frac{\pi_j^k}{\Pi_{i \in W_j(k)}} G_p p^1 \left\{ (p^k) \overline{Q} \right\}
\]
The probability can be obtained as
\[
G_p 1 (p^k) = \sum x^s e_{x^s} \delta(p^k x^{s+1}, ..., x^n) - p^k \prod_{x^s} Q^s(x^s)
\]
where \(\delta(x) = 1\) if \(x > 0\) and \(\delta(x) = 0\) if \(x \leq 0\).
We face two nested summation loops. First an outer summation loop over the states of the links in path \(k\) excluding link \(s\). Second, an inner summation loop over the states of the links in each competing path \(l\). This yields a quite a severe worst computational complexity in the order \(O(H S^{2N-1})\) where we have assumed that both path \(k\) and paths \(l\) contain at most \(N\)
links, the set of alternative paths \( W_j \) contain \( H \) paths, and all links have a state space with \( S \) elements.

8.2.2 The Load Sharing Model

The link call arrival rates \( \lambda_i^j(x, \pi) \) are given by:

\[
\lambda_i^j(x, \pi) = \sum_{i \in \mathcal{E}} \lambda_i^j(\pi) \phi_i^j(x, \pi) \prod_{c \in \mathcal{C}/c} (1 - B_i^c(\pi))
\]

The link \( c \) has not enough capacity to accept a class \( j \) call, \( \phi_i^j(x, \pi) \) denotes a filtering probability that the path net-gain is positive which can be computed using link state distributions and \( \lambda_i^j(\pi) \) denotes the arrival rate of class \( j \) to path \( k \mid W_j \), and is given by the

\[
\lambda_i^j(\pi) = \frac{\lambda_j^k(\pi)}{\sum_{i \in \mathcal{W}_j} \lambda_j^k(\pi)}
\]

where \( \lambda_j^k(\pi) \) denotes the average rate of accepted class \( j \) calls on path \( k \) and denotes the arrival rate of class \( j \).

9. The Sequential Quadratic Programming software

The Sequential Quadratic Programming algorithm is a generalization of Newton's method for unconstrained optimization in that it finds a step away from the current point by minimizing a quadratic model of the problem. The CFSQP [21] is a set of \( C \) functions for the minimization of the maximum of a set of smooth objective function subject to general smooth constraints (if there is no objective function, the goal is to simply find a point satisfying the constraints).

9.1 The Computation of Gradients

We know that the objective function is defined as

\[
W(C, \Gamma) = \sum_j W_j \lambda_j [1 - B_j(C, \Gamma)]
\]

where \( C - \{ C_i \} \) is the vector of physical link capacities, and \( \Gamma - \{ \Gamma \} \) is the vector of routing parameters, corresponding to the reward parameters \( r_j \) in MDP routing. The derivatives \( \frac{\delta W}{\delta C_i} = \frac{\delta B_j}{\delta C_i} \) are obtained from the above equation and substituted in the CFSQP software and the optimal capacity values are obtained. The software uses the gradient descent method in finding the optimal values.

9.2 The Gradient Descent method

Gradient descent is a function optimization method which uses the derivative of the function and the idea of steepest descent. The derivative of a function is simply the slope. So if we know the slope of a function, then it stands to reason that all we have to do is somehow move the function in the negative direction of the slope, and that will reduce the value of the function. Gradient descent is an iterative method, so the idea is as follows:

1. Compute the derivative of the function with respect to its independent variables. We can denote this derivative as \( F(x) \), where \( F \) is the function to be minimized, and \( x \) is the vector of independent variables.

2. Change the value of \( x \) as follows: \( x_{n+1} = x_n - \eta \nabla F(x_n) \), where the subscript \( n \) refers to the iteration number, and is a step size which must be chosen so that we don't take too big or too small of a step. Too big of a step will overshoot the function minimum, and too small of a step will result in a long convergence time.

3. Repeat the above two steps until we converge to a minimum of the function \( F(x) \).

10. NUMERICAL RESULTS

The routing algorithm considered in the numerical experiments is the MDP routing algorithm based on the Decomposition model. The performance analysis is performed for the network example w2s and w3s networks. The algorithm specific parameter settings are shown in the table 10.1 were determined heuristically based on simulation experience. The topology of the networks is also shown in the figure 7 below.

\[
\begin{align*}
\text{W2s} & \quad \text{W3s} \\
\end{align*}
\]

Figure 7: Network Topologies

<table>
<thead>
<tr>
<th>Table 10.1 Algorithm specific parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>--------------------------</td>
</tr>
<tr>
<td>Symmetrical</td>
</tr>
<tr>
<td>#nodes</td>
</tr>
<tr>
<td>#links</td>
</tr>
<tr>
<td>#OD pairs</td>
</tr>
<tr>
<td>#routes per OD pair</td>
</tr>
<tr>
<td>Route length</td>
</tr>
<tr>
<td>Link capacity</td>
</tr>
<tr>
<td>Network capacity</td>
</tr>
<tr>
<td>#max links per path</td>
</tr>
<tr>
<td>#traffic categories(i)</td>
</tr>
<tr>
<td>Mean holding time</td>
</tr>
<tr>
<td>Bandwidth ( b_j )</td>
</tr>
<tr>
<td>Traffic</td>
</tr>
<tr>
<td>Total traffic</td>
</tr>
<tr>
<td>Normalized reward parameter</td>
</tr>
</tbody>
</table>

The three performance measures network overall blocking.
probability, narrow band overall blocking probability and wide band overall blocking probability are determined. These quantities are obtained by measurements (using simulator) and also using the analytical model. The simulator calculates the measured blocking using the following formulae. The measured network overall blocking probability is given by the formula.

\[ B_{\text{measured}} = \frac{\text{Total Offered \lambda - Total Carried \lambda}}{\text{Total offered \lambda}} \]

\[ B_{\text{measured}} = \frac{\sum_{j=1}^{\infty} \lambda_j - \sum_{j=1}^{\infty} \lambda_j (1 - B_j)}{\sum_{j=1}^{\infty} \lambda_j} \]

similarly the overall blocking for narrow band and wide band are obtained. The following quantities are obtained using the analytical model. The network overall blocking probability is given by

\[ B_{\text{overall}} = \frac{\lambda_m - \lambda_m (1 - B_j)}{\lambda_m} \]

The narrow band blocking probability is given by 

\[ B_n = \frac{\lambda_n - \lambda_n (1 - B_j)}{\lambda_n} \]

The wide band blocking probability is given by

\[ B_w = \frac{\lambda_w - \lambda_w (1 - B_j)}{\lambda_w} \]

where

Fig 1. Graphs Showing the variation of Overall Blocking Probability

Fig 2. Graphs Showing the variation of Narrow Band Overall Blocking Probability for WS 3 

\[ \text{NB Traffic/WB Traffic} = \frac{\lambda_n \mu_n^{-1}}{\lambda_n \mu_n} \]

where

NB traffic and WB traffic represent the bandwidth, arrival rate and the holding time of narrow band and wide band respectively. Different mixes are obtained by varying the per-category call arrival rate to the OD pairs between the simulations, while keeping the amount of traffic per OD-pair constant. All OD-pairs were offered the same per-category call arrival rates within a simulation. The Fig. 7 and 8 are drawn by taking the Total traffic [Erlang*Mbps] on the x-axis and the overall blocking probability on the y-axis.
Fig 3. Shows the narrow band over all blocking for WS2

Fig 4. Graphs Showing the variation of Wide Band Overall Blocking Probability

Fig 5. Graphs Showing the variation of Wide Band Overall Blocking Probability

Fig 6: Variation of wide band overall blocking probability for W3s network
The queue length of the traffic in the network was taken along with the measured and modeled values [according to the queue length theory model] as shown in the Fig 9 below. As the queue length increased the modeled values decreased.

11. RESULT REVIEW
The first two graphs show the variation of network overall blocking probability with respect to the traffic ratio. We can observe that the measured overall blocking probability and the analytical overall blocking probability are almost similar for both the networks W2s and W3s. The next two graphs show the behavior of the narrow band overall blocking probability. We can observe from these two graphs that the narrow band blocking probability decreases as the traffic ratio is increased. This is due to the fact that the network load increases gradually as the traffic is increased and the calls have to be blocked when the network doesn’t find sufficient resources (bandwidth for example) for those calls. It is obvious that more NB calls can be accommodated than the WB calls in the network. So, the NB calls suffer less blocking. The next two graphs show the behavior of the wide band overall blocking probability. The blocking of WB calls decreases as the traffic ratio increases. If the WB traffic increased then the network will be overloaded with the WB calls and WB calls suffer higher blocking than the NB calls because there will not be sufficient resources to accept the WB calls. We can easily make out from the graphs that if we move from right towards the left the blocking of WB calls increases. The next two graphs show the behavior of the overall blocking probability when the total traffic is varied. We can conclude from the graph that as the traffic load increases the overall blocking probability increases.

12. CONCLUSION
A model for node-to-node blocking probabilities evaluation in loss networks with state-dependent routing maximizing the reward from carried calls is synthesized. The model is based on the link independence assumption and results in an extended set of fixed point equations solved by repeated
substitutions. In this approach both the policy maximizing the reward from the network and the network performance under this policy are evaluated at the same time. We can conclude that the analytical model is accurate for small sized networks. The analytical and theoretical blocking results were similar for small sized networks. The overall blocking probability decreases as the traffic ratio is increased. We can also conclude that the overall blocking probability increases as the traffic load is increased. Lack of smooth blocking functions restricted the thesis work to the evaluation of blocking probabilities the categories for each OD-pair and there by the last step in dimensioning which deals with the optimization of the physical link capacities was as a part of future work. It also includes the performance of system must be analyzed in delay and mixed-delay modes. The dimensioning of the physical link capacities has to be done after acquiring the smooth blocking functions. The dimensioning has to be done with respect to the relative Grade of Service constraints. The Queue length of the traffic in the network was taken along with the measured and modeled values to show the variation in the network.

13. REFERENCES
ISBN 0-07-134959-6
[4]. ATM Analyzer ASL Group <www.atmanalyzer.com>
[5]. IPNetworks Sangoma Technologies <http://www.sangoma.com/>
[6]. Guillaume Urvoy-Keller a and G’erard H´ebuterne b
MPLS Guaranteeing Deterministic QoS in a Multipoint-to-point Network
<http://www.eurecom.fr/~btroup/BBPublished/ITC17.pdf>
[7]. Abdelnaser Adas ,Traffic Models in Broadband Networks Georgia Institute of Technology<http://www.sce.carleton.ca/faculty/devetsikiotis/94581Y_00/Adas.pdf>
[9]. Thijs Harleman An overview of effective bandwidth methods
wk/ito_doc/routing.htm#xtocid2>
[12]. Yunning Song, David Cypher, and David Su, Simulation and Performance of PNNI ATM Networks
[13]. Chen-Khong Tham, Jianning Mai & Lawrence WC Wong, A QoS based routing algorithm for PNNI ATM Networks
<http://citeseer.nj.nec.com/cachedpage/460703>
[14]. Thomas Bauschert , Jochen Frings, Rainer Siebenhaa

Network Engineering for ATM with VirtualPaths<http://citeseer.nj.nec.com/cachedpage/408752/1>
[16]. Zbigniew Dziong, Yijun Xiong and Lorne G. Mason Virtual network concept and its applications for resource management in atm based networks
[17]. Xi Peng Xiao and Lionel M. Ni, Internet QoS : A BigPicture<http://citeseer.nj.nec.com/update/437>
[18]. Leland , Taqpu, Willinger Wilson On the self similar nature of Ethernet traffic<http://citeseer.nj.nec.com/cachedpage/49996/2>
[19]. Ersnt Nordstrom Near Optimal CAC and Routing for Multi-service Networks by Markov Decision Theory
[20] Craig Lawrence, Jian L.Zhou and Andre L. Tits A C Code for Solving (Large Scale) Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality Constraints