MPLS based State-Dependent Optimal Routing in IP Networks

(non-Homogenous Case)

Peng He

Department of Electrical and Computer Engineering
McGill University
Montreal, Canada

September 2003

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Engineering.

Your file Votre référence
ISBN: 0-612-98532-6
Our file Notre référence
ISBN: 0-612-98532-6

Acknowledgements

First and foremost, I wish to thank my supervisor, Professor Jean Régnier. This thesis would not be possible without his guidance and kindly help. I would like to thank him greatly for his support throughout my graduate study at McGill University.

It is also a great honor and pleasure to acknowledge Professor Tho Le-Ngoc, for his help and support.

Grateful acknowledgment is expressed to Nortel Networks for sponsoring our research and particularly to Dr. James Yan and Dr. f Beshai from Advanced Network Performance, Nortel Networks, for their supervision and helpful suggestions.

I also want to thank my friends, Zhibin Wang, Xiaojian Lu, Benoît Pelletier and all the people in the TSP Lab for their friendship and timely help.

Abstract

This study investigates routing in a MPLS-based IP network with heterogeneous holding time traffic (for example, an IP call versus an IP conference). Our basic idea is to exploit the large differences existing in the holding time of different types of traffic to make more efficient resource allocation decision in the admission and routing processes. In particular, this study investigates the concept of vacating, in which requests with short holding times vacate the bandwidth on direct links in favor of requests with long holding times under some traffic conditions.

Based on an analytical framework we developed, we propose several state-dependent routing schemes, namely preventive-vacating routing (PVV), preemptive-vacating routing (PEV) and restricted-access routing (RAR). Additionally, we deduce an approximated expression to compute the cost of accepting a long or short request. This leads to an approximated least cost routing (A-LCR) scheme directly.

Furthermore, along with the simulation study, some of the significant results we obtained are:

- The effective range in traffic mix is $[0.60, 0.99]$, within which our vacating schemes outperform the traditional LLR+TR (least loaded routing + Trunk Reservation) in terms of network throughput.
- Besides its particular flow control mechanism, the A-LCR scheme shows a constant outperformance compared to LLR+TR.
- The routing schemes we proposed perform better than the differentiated shortest distance routing (Diff-SDR) scheme, which is the only currently published dynamic routing scheme addressing the question of heterogeneous holding times.

We also study the inter-network routing issue, which focuses on how to select the best among several gateway nodes to a foreign network. Simulation results show that the intra-network links play a much more important role than inter-network links when making gateway selection decision.

Sommaire

Cette étude porte sur l'acheminement dans les réseaux IP (basés sur MPLS) dans lesquels differents flots possèdent des durées de service hétérogènes (par exemple, un appel IP versus une conference IP). Notre idée maitresse consiste à exploiter les differences significatives existant dans les durées de service de different types de flots pour permettre une allocation plus efficace des resources par les processus d'admission et d'acheminement. Plus spécifiquement, cette étude analyse le concept de dégagement, dans lequel les flots possédant de court temps de service libèrent les liens directs au profit des flots de plus longues durées sous certaine conditions de charge de traffic.

Basé sur un modèle analytique inspiré du concept du cout d'utilisation des resources, on propose plusieurs algorithmes d'acheminement dépendant de l'état, à savoir PVV (preventive-vacating routing), PEV (preemptive-vacating routing) et RAR (restricted-access routing). De plus, on déduit une expression approximative pour calculer le cout d'accepter une demande longue ou courte. Cela mène au A-LCR (approximated least cost routing) pour les flots avec des durées de service hétérogènes.

De surcroit, avec l'étude de simulation, certains résultats significatifs qu'on a obtenu sont les suivants:

- La région efficace de mélange du trafic est $[0.60, 0.99]$. Dans cette région, nos algorithmes de dégagement surpassent le LLR+TR traditionnel (least loaded routing+ Trunk Reservation) en terme de débit maximal atteignable.
- Outre son mécanisme inhérent du contrôle du débit, l'algorithme A-LCR démontre dans tout les cas une performance supérieure à LLR+TR
- Les algorithmes d'acheminement qu'on propose fonctionnent mieux que l'algorithme Diff-SDR (differentiated shortest distance routing) qui est le seul algorithme d'acheminement dynamique actuellement publié traitant les temps de services hétérogènes.

Cette étude analyse aussi la question de l'acheminement inter-réseau, qui porte sur le choix de la meilleure passerelle pour transiter d'un réseau source à un autre réseau externe connexe. Les resultats de simulation démontrent que le choix de la route à l'intérieur du réseau source joue un rôle beaucoup plus important que le choix du lien vers le réseau externe dans la sélection de la passerelle.

Table of Contents

List of Tables

List of Figures

Acronyms

AS:	Autonomous System
CCS7:	Common Channel Signaling 7
DAR:	Dynamic Alternate Routing
DCR:	Dynamically Controlled Routing
DMS:	Digital Multiplexed System
DNHR:	Dynamic Non-Hierarchical Routing
GOS:	Grade of Service
LDP:	Label Distribution Protocol
LLR:	Least-Loaded Routing
LPR:	Load Profiling Routing
LSP:	Label Switching Path
LSR:	Label Switched Routers
MDP:	Markov Decision Process
MLLR:	Multi-Rate Least-Loaded Routing
MLLRP:	Multi-Rate Least-Loaded Routing with Packing
MLR:	Most-Loaded Routing
MPLS:	Multi-Protocol Label Switching
NS2:	Network Simulator Version 2
QoS:	Quality of Service
FRVT:	Fractional Routed Volume Traffic
RSVP:	Resource Reservation Protocol
RTNR:	Real-Time Network Routing
RUC:	Resource Utilization Costs
SDR	State-Dependent Routing
SLA	Service Level Agreement
TR:	Trunk Reservation

Chapter I: Introduction

The traditional Internet, which in the past has supported only a best effort service, has transformed very quickly into commercial broadband multi-service IP networks demanding support for quality of service (QoS). Thus, compared with conventional public switched telephone networks (PSTN), a variety of challenges have been experienced by such multi-service IP networks in dealing with much more complicated routing. One of the main issues concerning routing is how to dispose of *traffic heterogeneity*.

Traffic heterogeneity includes both the differences in bandwidth requirements among flows, which have been studied broadly and heuristically [30,31,33,36,37,39,42,43,44], as well as the differences in holding time among flows, which have not yet attracted enough necessary attention. In this thesis, we focus on the impact of holding time heterogeneity among flows on routing performance.

Deploying Multi-Protocol Label Switching (MPLS) technologies [50,52], IP networks have become similar to conventional circuit-switched telephone networks in some aspects. For instance, a flow in MPLS capable IP networks could correspond to a call in circuit-switched networks. Therefore, it is possible that the ideas, research results, and experiments [1,3,11,12,14,16,20,22,23,45] on routing in circuit-switched networks can be used as a guide, or at least as references, in solving some routing problems in IP networks.

1.1 The Goal of this Thesis

The major goal of this thesis is to study the following routing issue: How can we take the large heterogeneity of holding time among traffic flows (for example, an IP call versus an IP conference) into consideration when making routing decisions in a MPLS-based IP network. Generally speaking, our basic thought is to extend the theoretical studies and practical experiments on state-dependant routing in circuit-switched networks to IP networks, and then to find a solution to this issue.

Besides the above routing issue, which is within the scope of one network, we will also study an issue associated with routing across networks, that is to say, inter-network traffic. Inter-network traffic comprises a significant fraction of the traffic in many networks and will increase in the future as networks become more inter-connected. Thus, how can we select the best one among several gateway nodes to a foreign network? This applies of course to inter-network traffic that can exit through multiple gateways, and is another issue that will be studied in this thesis.

1.2 Related Work

Traffic with non-homogenous holding times belongs to the multi-service traffic. There are, in general, two categories of methods used to study the routing issues in multi-service networks.

The first is the *Markov Decision Process-based (MDP-based)* approach. The MDP approach formulates the routing problem as a Markov decision process and obtains the "cost" for carrying a connection by the network. According to the Markov decision theory [13], an optimal routing policy, which minimizes the expected "cost", can be found with a finite number of policy iterations. Literature about the MDP approach is rich in traditional telephone networks [22,23,37] and in multi-service networks [26,30,31,32,36,40,41,42,63].

The second can be called the *"packing"* approach. The argument for packing is based on the observation that in order to maximize the utilization of available resources, a routing policy in a multi-rate environment should implement *packing* of narrow band traffic (having relatively small bandwidth requirement) on some routes so as to leave room on other routes for wideband traffic (having relatively large bandwidth requirement). Examples of schemes [43,44] using the packing techniques are Most-Loaded Routing (MLR), Multi-Rate Least-Loaded Routing with Packing (MLLRP), Multi-Rate Least-Loaded Routing (MLLR) and Load Profiling Routing.

Besides MDP and packing, Kelly [20] focuses on an approach that optimizes routing in networks with call revenue. He introduces the notion of *implied cost/shadow price*, which has been subsequently widely used [27,33,34,35]. Implied costs measure the expected knock-on effects of accepting a call upon the other routes in the network. Similar work has been done by A. Girard [38].

Actually, most of the work on multi-service networks focuses on the multi-rate traffic. Only a few [10,17] pay attention to the non-homogenous holding times of traffic. In [10], the authors introduce a new *hybrid* approach that performs dynamic routing (widest-shortest scheme) only to long-lived flows (long holding time), while forwarding short-lived flows (short holding time) on static pre-provisioned routes.

While in [17], a dynamic shortest-distance routing scheme has been adopted. The idea is to use *differentiated* link metrics to compute the shortest distance. For short requests, the link metric is the most up-to-date (called *dynamic*). For long requests, the link metric is averaged over a given time scale (called *adaptive*). It has been demonstrated in [17] that this differentiated shortest-distance routing works better than a non-differentiated one under heavy network load.

1.3 Chapter Contents

This thesis is organized as follows: in the next chapter, we review the concept of the resource utilization cost and its application to state-dependent routing in conventional circuit-switched telephone networks. Then, we define the system model for our routing issues. In Chapter III, we develop an analytical framework to analyze our vacating idea thoroughly. Based on this, several routing schemes are proposed to address the intra-network routing issue. Similarly, an analysis about inter-network routing and the resulting gateway selection schemes are presented in Chapter IV. This is followed by a detailed description about our simulation framework in Chapter V. We then present the results and interpretations of the simulations in Chapter VI and VII, respectively for intra-network and inter-network routing. Finally, the conclusions and future work are recapitulated in the closing discussion.

Chapter II: On the Basis of this Study

This chapter offers an introductory discussion on the resource utilization cost and its application for state-dependent routing. We then define the system model for our routing issues, including the basic definitions and assumptions used throughout this study.

2.1 Resource Utilization Cost and State-dependent Routing

In a network environment, when a request grabs one or a set of resources for a given time, it may deprive future upcoming requests from accessing these resources. Thus, assigning resources to a request entails a ***cost***, which can be viewed as the risk that future requests may also need the resources, and may not have access to them because they have already been assigned.

This is the classical definition of the ***resource utilization cost***; this sort of thinking, or mechanism, can guide the design of routing algorithms in networks.

2.1.1 Theoretical Average Resource Utilization Cost

Consider a system in which a pool of N resources serves requests. Each request requires one resource for the duration of its holding time. The arrival process of the requests is Poisson, with intensity λ. The holding times of the requests are independent and exponentially distributed, with an average holding time $1/\mu$. Thus the traffic intensity, A, is λ/μ.

Suppose $p < N$ resources are busy. A new request, labeled *g*, arrives at time 0, requiring one resource for its holding time h. Upon accepting request *g*, a resource utilization cost will be induced. Let $\bar{C}_p$ be the average resource utilization cost of assigning the resource to request *g*, given that p resources are currently busy. Then $\bar{C}_p$ can be defined as the probability that a subsequent request arrives while *g* holds its resource, and finds all resources busy. In other words, $\bar{C}_p$ is the probability that the process reaches state N during a period of length h, starting from state p. Then $\bar{C}_p$ is equal to:

$$\bar{C}_p = \frac{E_b(A,N)}{E_b(A,p)} \qquad (2.1),$$

where $E_b(A,n)$ is the Erlang-B blocking formula, defined by $E_b(A,n) = \frac{A^n / n!}{\sum_{i=0}^{n} A^i / i!}$. We can see that $\bar{C}_p$ depends on p, A and N, and has an obvious state-dependent character. Hence, in general we will refer to this as $\bar{C}_{A,N}(p)$.

Equation (2.1) was derived by Krishnan and Ott [23] in 1986 through formulating the problem of state-dependent routing as a Markov Decision Process (MDP).

2.1.2 State-dependent Routing based on Resource Utilization Cost

We now consider a typical trunk group in a well-engineered circuit-switched telecommunication network. Suppose the trunk group is offered a traffic of A=80 Erlang, and is engineered for a blocking of 1%. The trunk group thus contains 96 trunks. Figure 2-1 presents $\bar{C}_{A,N}(p)$ as a function of p for this trunk group.

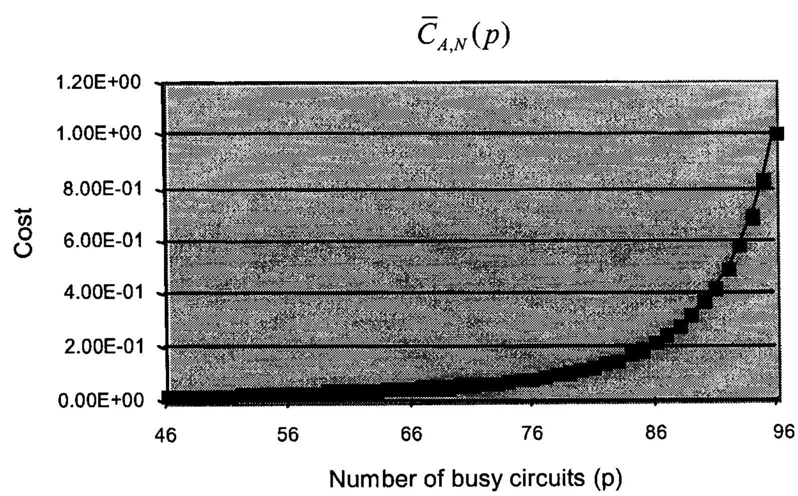

Figure 2-1: $\bar{C}_{A,N}(p)$ as a function of p for a trunk group of N=96 trunks serving a traffic demand of A=80Erl.

It can be noted from Figure 2-1 that $\bar{C}_{A,N}(p)$ is monotonically increasing in p, and is always less than one. Thus, if a call requires only one trunk, it is always profitable to use the trunk for it. Blocking the call would immediately entail a cost of one unit, while allowing it to use the trunk may, at most, entail a cost of one unit. In addition, allowing the call to use the trunk becomes increasingly costly as the number of circuits that it finds busy upon its arrival is high. It can also be noted from Figure 2-1 that $\bar{C}_{A,N}(p)$ increases rapidly as p approaches N.

In general, a call may require more than one trunk to get from its origin to its destination in a telecommunication network. For instance, this arises if there is no direct trunk group between the origin and the destination, or if the direct trunk group is fully busy. Then the cost C^R of assigning a call on a route R is the *sum* of the $\bar{C}_{A,N}(p)$ on each of its links. The optimal route for the call is that for which C^R is smallest, in other words, the *least costly one*.

If the cost C^R of assigning a call on route R is greater than or equal to one, the call can be expected to cause the blocking of at least one other call. Therefore, the call should not be assigned to route R. If the cost of the call is greater than or equal to one on all the routes that it can potentially attempt, the call can be expected to cause the blocking of at least one other call, no matter where it is routed. Thus, the call should be immediately rejected from the network. Therefore, the criterion of routing is

$$\begin{aligned} &\textit{If } C^R < 1, \textit{ then the call could be accepted on route } R; \\ &\textit{If } C^R \geq 1, \textit{ then the call is rejected from route } R; \end{aligned} \qquad (2.2)$$

The state-dependent call routing scheme in circuit-switched networks, then, operates on the basis of the observations made above; i.e.,

1) First, attempt the direct route.
2) If the direct route does not exist or is fully busy, find the route R whose overall cost is lowest (least cost route).
3) Assign the call to route R if C^R is strictly less than (<) one, and block the call otherwise.

Here, the state refers to the "state" of the network, which is the current occupancy of the links in the network. This state-dependent routing is an example of dynamic routing, and its overall operation can be summarized in three words: "*Think before routing*". The "Think" indicates the use of current state information in making routing decisions that do not blindly follow a predetermined pattern.

2.1.3 Trunk Reservation

Computing $\bar{C}_{A,N}(p)$ requires knowledge of the traffic demand, and is impractical to do in real time. For this reason, practical state-dependent routing schemes rely on an approximation to $\bar{C}_{A,N}(p)$, called "Trunk Reservation" (TR).

Trunk reservation consists of declaring that the cost of using a trunk group is one, when the number of idle circuits on the trunk group reaches or becomes less than a threshold. This amounts to approximating $\bar{C}_{A,N}(p)$ to zero when the number of idle trunks is above the threshold, and to one when at or below the threshold. Figure 2-2 illustrates the trunk reservation approximation, with a threshold of five idle trunks.

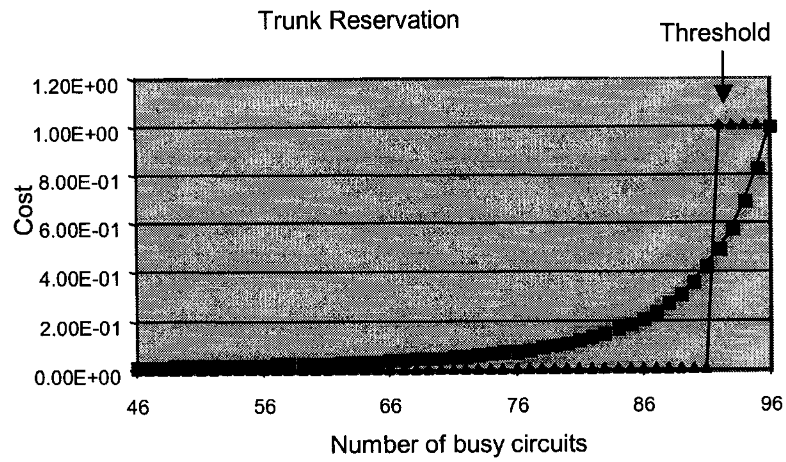

Figure 2-2: The trunk reservation approximation to $\bar{C}_{A,N}(p)$.

Note that the trunk reservation threshold does not need to correspond to an integer number of idle trunks and that it could also be expressed as the percentage of link capacity.

Note also that, with the trunk reservation approximation, the total cost of a route is always a non-negative integer. Each link whose number of idle trunks equals or is less than its threshold adds one to the total cost. As a call should be rejected from the route if the total cost exceeds (≥) one, it thus suffices to find out if at least one link is at or below its threshold to reject the route. In the practical implementation of optimal state-dependent routing schemes, this allows the cost summation operation to be replaced by the simpler operation of finding the maximum cost among the links of the route.

In addition, the cost of a link increases monotonically as a function of the resources currently busy on the link (Figure 2-1). Then the least cost route is the one with the maximum free capacity where the free capacity of a route is defined as the minimum free capacity of the links in the route. Thus, finding the least cost route could become a matter of finding the least-loaded one. Therefore, we will name this state-dependent routing with trunk reservation as **LLR+TR** (Least-Loaded Routing with TR) in this report.

It has also been observed that the TR threshold should be small for a light network load, while it should increase when the network load becomes heavy. In other words, the optimal TR threshold should be adaptive to the network load. However, finding the optimal TR threshold to approximate equation (2.1) is not an easy thing. In most cases, the optimal value of the TR threshold has to be determined experimentally (around 3~5% [57]).

Some researchers have explored methods to find an adaptive and continuous TR threshold. In [7] [8], Mitra et al. derived a load-dependent trunk reservation level by asymptotic analysis of the Fixed Point Model. In [19], Ren P. Liu argued that a linear approximation to the theoretic state-dependent cost would give results not too far away from the optimum. Based on this argument, Liu presents a relatively simple formula for computing a dynamic TR threshold. Another interesting method described in [4] uses the square root of the link capacity multiplied by a value named the "Trunk Reservation Parameter (TRP)". The TRP is a constant parameter whose optimal value needs to be found empirically.

2.1.4 Practical Dynamic Routing

The rapid deployment of stored program control networks, consisting of electronic switching systems (e.g., DMS systems) interconnected by common channel signaling (CCS7) links, makes the replacement of conventional fixed hierarchical static routing by dynamic routing possible. Here the term *dynamic* describes routing methods that are time-sensitive, or, possibly, real-time state-dependent, as opposed to time and state independent.

An implementation example of a dynamic routing scheme based on the resource utilization costs, as discussed previously, is the *Dynamically Controlled Routing* (DCR)

[16,45] developed by Nortel Networks. DCR uses a central processor to track the state of network trunks and determines the best alternate route choices based on status data collected every 10 seconds. The DCR scheme has been implemented and deployed successfully in the conventional telecommunication (circuit-switched) networks both in Canada and in the U.S.

Other well-known dynamic routing schemes in telephone networks include AT&T's Real-Time Network Routing (RTNR) [12] and British Telecom's Dynamic Alternate Routing (DAR) [14]. Generally speaking, all of these schemes use a common simple routing rule: an arriving call is always offered to the direct link first, if there is a free circuit. Otherwise, the call attempts the two-hop alternative route, via a tandem node with trunk reservation applied to both links.

After reviewing the success of resource utilization costs and associated state-dependent routing in conventional telecommunication networks, it is natural to consider how to apply or extend this line of thinking to *IP networks* in order to solve our routing issues. Let us start this with the introduction of MPLS, which, to a certain extent, makes IP networks similar to telephone networks, enabling the application of similar solutions.

2.2 Multi-Protocol Label Switching in IP Networks

The traditional Internet is in fact a connectionless network, where a packet-forwarding decision is taken independently at each hop, as the packet is sent from one hop to the next. However, with the process of commercialization, the traditional Internet that supports only the best effort service has transformed very quickly into a commercial environment, demanding support for quality of service (QoS) for many applications.

The IETF has already proposed several frameworks and mechanisms for QoS support: the Integrated Services and RSVP [48] framework, the Differentiated Services framework [49], MPLS (Multi-Protocol Label Switching) [50] and QoS based Routing frameworks [51]. Integrated Services and Differentiated Services provide a framework for service classification, differentiation and prioritization, the objective being to guarantee network performance, avoid network congestion, and satisfy customer requirements. MPLS and QoS-Policy based routing provide the mechanisms to control and influence the way traffic is forwarded across the network with the same objectives as the former two. In particular, MPLS, combined with QoS-Policy based routing, makes possible the use of the concept of resource utilization costs in IP networks.

2.2.1 MPLS

MPLS [50] is a packet label-based switching technique. Packets are assigned a label as they enter a MPLS capable IP network. Subsequent packet treatment in the network is based on the label only.

The labeling of a packet allows the use of advanced forwarding techniques. A packet entering the network at a particular router can be labeled differently than the same packet entering the network at a different router. As a result, some kind of policy routing can easily be made. Since MPLS decouples forwarding from routing, it is able to support a large variety of routing policies that are either difficult or impossible to support with just conventional network layer forwarding.

MPLS uses the Label Distribution Protocol (LDP) [52] to exchange the label and traffic trunk binding between Label Switching Routers (LSRs). A traffic trunk in MPLS terminology is an aggregation of traffic flows of the same service class, which are placed inside the same Label Switched Path (LSP). A LSP is defined as a sequence of labels from an ingress LSR to an egress LSR. LSPs are very similar to unidirectional ATM virtual circuits. The route taken by a LSP between two LSRs can be the same as the conventional network layer route, or the sender LSR can specify an *explicit route* for this LSP (an explicit route is specified as a sequence of hops rather than being determined by conventional layer-three routing algorithms on a hop-by-hop basis). Thus, apart from conventional IP routing facilities, MPLS can use the routing technique called *explicit routing* [53,55], which can support policy routing and traffic engineering [54]. An explicit route needs to be specified at the time that labels are assigned and does not have to be specified with each IP packet.

The ability to set up explicit routes is one of the most useful features of MPLS, because it allows network administrators to control how traffic flows through their network. In MPLS, an explicitly routed LSP is also considered a tunnel. When a packet enters the network, its path, QoS, and forwarding class are already fully determined. In addition, explicit routes can be selected either by configuration (manual setup by the network administrator) or dynamically.

2.2.2 Possible Usage of Resource Utilization Costs in MPLS

Although IP networks seem very different from conventional circuit-switched networks, one finds that these two networks become similar if MPLS is adopted. Actually, there are several ways that one can map the circuit-switched model we discussed previously into the MPLS framework. The key point is that a call in a circuit-switched model can correspond to a *flow* in MPLS and setting up explicit routes dynamically in MPLS is very similar to dynamic routing in telephone networks.

When a new flow is to be routed through the network, the router can determine the path it will take according to the cost induced by the flow, and then assign an effective bandwidth to the flow to meet its QoS requirements. Thus, a state-dependent scheme could be possible in MPLS capable IP networks. It would route each flow to minimize the risk of blocking future flows, and respond to the current state of the network on the basis of certain assumptions about future traffic demands. Therefore, the concept of resource utilization costs (plus induced state-dependent routing) in circuit-switched networks could still be adopted, or serve as a reference, in studying how to adapt to the large difference of requests' holding times when making routing decisions in IP networks.

2.3 System Model: Topology

The system model that will be used to study our routing issues includes three parts: the topology, the traffic, and the routing control. We start with the basic definitions and assumptions regarding the topology.

2.3.1 General Definitions

The definitions of essential concepts about topology are:

- **Network:** By network, we mean here a collection of nodes and links placed under a common administrative domain, often referred to as an autonomous system (AS). A network topology is comprised of one *local network* connected with several *foreign networks*. We mainly focus on the local network in this thesis.

- **Node:** A node is usually a switch/router, which forwards packets and computes routes.
- **Gateway Node:** A kind of node, which acts as a gateway connecting the local network to foreign networks.
- **Interior Node:** A kind of node, which exists in the local network and does not have any physical connection to foreign networks.
- **Link:** The physical trunk connecting two nodes. In addition, the links in the networks are *unidirectional*, i.e., carrying traffic in one direction only. $link_{i,j}$ is the notation for the link from node i to node j. Physically, intra-network links connect nodes within the local network, while inter-network links connect the local network to a foreign network.
- **Route-set**: The route-set is the set of all the routes of an origin-destination pair. $route-set_{i,j}$ is the notation for the route-set from node i to node j.
- **Route**: An element in a route-set, a particular route, which is comprised of a sequence of connected links. There are two types of routes in the network: direct route (one-hop) and alternate (two-hop) route.
- **Direct Route**: The direct route is the direct link serving as a route (one-hop) in a route-set. $route_{i,j}$ is the notation for the direct route from node i to node j.
- **Two-hop Route**: The two-hop route comprises two links that are physically connected. $route_{i,k,j}$ is the notation for the two-hop route from node i to node j through node k.
- Resource utilization of a link: $\frac{occupied\ bandwidth\ in\ the\ link}{capacity\ of\ the\ link}$.

2.3.2 Basic Topology Assumptions

1) We model a well-connected and well-engineered packet-switched network.
   - By well connected, we mean that many origin-destination pairs are directly connected and many two-hop routes exist for each origin-destination pair. Within such a network, there is no need to consider a route with more than

two hops. We only consider either a direct route or a two-hop route in the local network, as it is generally an exceptional case when no direct or two-hop route exists.

- By well engineered, we mean that the links are placed and sized so that the bulk of the traffic can be carried over a shortest route, most often the direct link.
- The above assumptions are readily satisfied in many backbone IP networks.

2) We assume that the traffic generated by each traffic source is small compared to the capacity of the links that it may use. This is easy to justify, based on the fact that a network needs by definition to share its resources among a large number of concurrent users.
3) We set the connectivity of the local network at around 50%. The connectivity of a network is defined as the ratio of its actual number of links relative to the number of links if it was fully meshed.
4) We assume that we can monitor the resource utilization not only on the links within the local network but also on the inter-network links, which connect the local network with foreign networks.

2.3.3 Network Examples in this Study

According to the above definitions and assumptions, we use three network examples to do simulation studies:

- A 4-node network, which is fully connected and symmetrical.
- A 12-node network, which is not fully connected and non symmetrical, but well-engineered. Kruithov's method (see Appendix B for details) is adopted to generate the traffic distribution, and to obtain reasonable individual link capacity.
- A 30-node network, not fully connected and non symmetrical, but well-engineered. Similarly, the traffic distribution and network capacity are generated through Kruithov's method and follow our assumptions made regarding the topology.

Note: The 4-node and 12-node networks focus on studying intra-network routing; while the 30-node network centers on inter-network routing. In addition, the selection of the number of network nodes accounts for the constraints imposed by the simulation time, the simulator running time, and the desired accuracy of the simulation results.

In the 30-node network, in order to have an inter-networking environment, we set up inter-network links that connect the local network with three foreign networks. For each of these three foreign networks, the local network has two gateway nodes directly connected to them. (Figure 3-1)

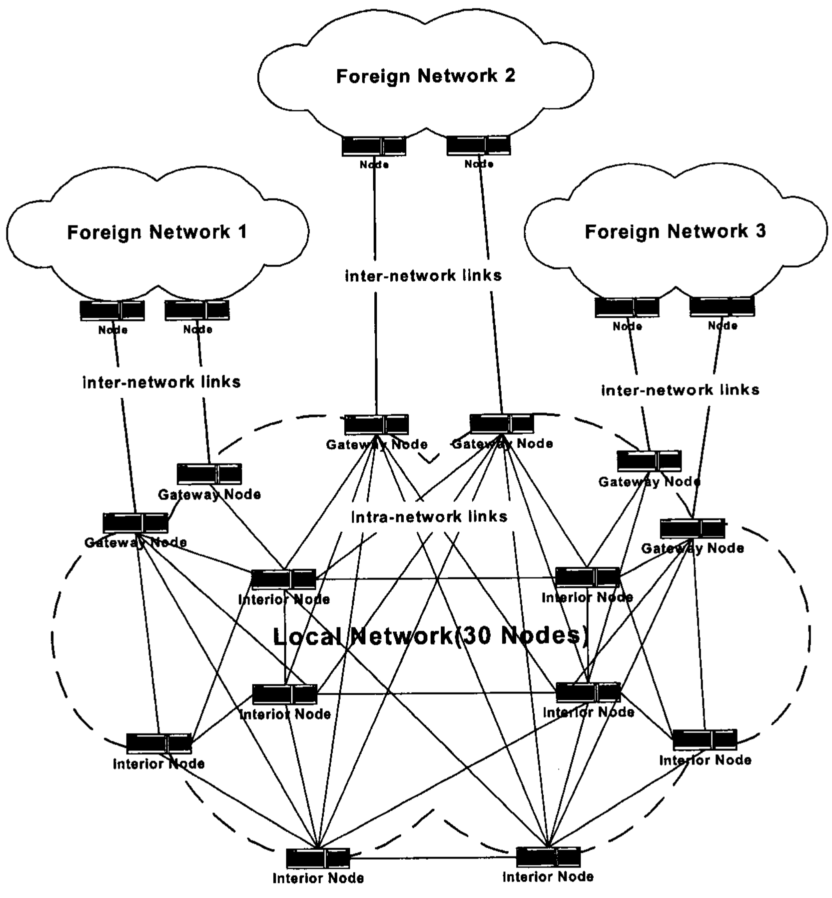

Figure 2-3: Conceptual Network Topology of the 30-node network

2.4 System Model: Traffic

In this study, we consider only the traffic sources that generate a long series of packets over some time interval. We call the traffic generated by these sources "connection" traffic or flow traffic. Furthermore, we assume that the arrivals of connection traffic requests are Poisson with mean arrival rate λ, and the holding times of connection traffic requests are independent and exponentially distributed. Connection traffic is characterized by several parameters:

- Bandwidth required
- Origin in network
- Destination in network
- Holding time

Compared with the voice traffic in telephone networks, there can be a lot of variation in the bandwidth and holding times of different connection traffic requests. For example, holding times may vary from days (e.g., VPN), to hours (videoconference), to minutes (VoIP call), to seconds (HTTP).

2.4.1 Bandwidth Required

This can be either a static parameter negotiated in advance as part of an SLA (Service Level Agreement), or it can be negotiated upon admission. We assume that the bandwidth a connection requires can be defined in terms of an "equivalent bandwidth" in the sense of Kelly [29]. That is, if a connection is provided with its equivalent bandwidth during its holding time, then its QoS (quality of service) objective is met. Furthermore, the equivalent bandwidth possesses some other beneficial properties such as additivity.

Note that from this point, the connection request is very similar to a call connection in the traditional telephone network. We assume that all connections in the network can be characterized by a common distribution for the bandwidth and we focus mainly on the holding time heterogeneity among them in this study.

2.4.2 Holding time

Different types of connection traffic have widely varying holding times, and this is assumed to be known by the system. The holding time may be explicitly negotiated as part of the admission process or the service agreement. Alternately, the holding time may be known implicitly through attributes of the connection such as the protocol and port used.

We assume here that for the same type of traffic, its holding time has the same exponentional distribution with a mean value $1/\mu$. The system knows *a priori* the mean holding time of each type of traffic.

2.4.3 Origination and destination of Connection Traffic

Within the scope of the local network, the connection traffic can be divided into three categories, according to the origination and destination of each connection call:

	Type of Call	Origination Node	Destination Node
1	Outgoing Inter-network Call	Any node in local network	Foreign Network
2	Incoming Call	Gateway nodes	Any node in local network
3	Local Call	Any node in local network	Any node in local network

Table 2-1: Three types of calls in the local network

2.4.4 Homogenous and non-Homogenous Traffic

In this study, the traffic in the network could be either homogenous or non-homogenous. They are distinguished according to the holding time of traffic flows:

- **Homogenous traffic**: There is only one type of traffic in the network. The holding times of all the connections follow the same exponentional distribution.
- **Non-Homogenous traffic**: There are at least two types of traffic in the network; the holding time of different traffic types follows *different* exponentional distributions.

For simplicity, we only consider two types of traffic in the network. The one with a longer holding time is called *long request traffic*, with mean holding time $h_L = 1/\mu_L$. The one with a shorter holding time is called *short request traffic*, with mean holding time $h_S = 1/\mu_s$. Thus, an offered load ρ_L Erlang by long requests is given by $\rho_L = \lambda_L \times h_L = \lambda_L / \mu_L$ where λ_L is the average Poisson arrival rate of long requests. An offered load ρ_S Erlang by short requests is given by $\rho_S = \lambda_S \times h_S = \lambda_S / \mu_S$, where λ_S is the average Poisson arrival rate of short requests.

2.4.5 Traffic Mix

We define the traffic mix as $\frac{\rho_L}{\rho_L + \rho_S}$, which is generally expressed in percentage form.

2.4.6 Holding Time Ratio

We define the holding time ratio as $\frac{h_L}{h_S}$.

2.4.7 Non-coincidence of peak hours

The different nodes do not necessarily sustain their peak traffic all at the same time. Here, we consider the concept of time zones. Not all the nodes in the network are in the same time zone, thus the time of their peak traffic varies (see Chapter V and Appendix B for the details).

2.5 System Model: Routing Control

We suppose that we can obtain the information of detailed link occupancy in the network when making the admission and routing decisions.

2.5.1 Basic Routing Control Assumptions

1) Our model supports the MPLS mechanism. For connection traffic, routing decisions apply on the basis of the connection. All packets of a connection follow the same route in the network, so that those packets are delivered in the same order as they are sent.
2) If a connection request is accepted and a route is assigned for it, then the bandwidth on each link of the route is simultaneously held for the duration of the connection.
3) If the connection request is finished, then the bandwidth it held on each link should be released, so that it can be re-used by future connection requests.
4) If a connection request is blocked, then the request is lost. Therefore, our model is a loss network.
5) We do not consider the overhead for obtaining the information (state of link, holding time of the request, etc.) in the calculation of cost.

6) We assume that the route selection process is state-dependent. That is, the outcome of the process can depend on the current offered traffic and loading conditions on the various links of the network. Therefore, our routing mechanism is a state-dependent.

2.5.2 Information Requirement of Routing Control

To compute the cost of an arrival connection request, the routing control mechanism of our model should know the:

- Route-set for each origination-destination pair
- Capacity of each link
- Current resource utilization of each link (and, if needed, we can also measure the traffic intensity on each link) in the route-set
- Bandwidth requirement of the request
- The type of traffic that the connection request belongs to (long request or short request)

2.5.3 Quality of Service

Quality of service (QoS) is provided by allocating sufficient bandwidth to each connection. That is, if a connection is provided with its equivalent bandwidth during its duration, then we consider that its QoS objective is met.

2.5.4 System Performance Metric

The performance metrics are:

- Fractional Routed Volume Traffic (FRVT) or *Network Throughput*:

$$\frac{No.\ of\ carried\ long\ requests \times x + No.\ of\ carried\ short\ requests}{No.\ of\ offered\ long\ requests \times x + No.\ of\ offered\ short\ requests},$$

where x is the holding time ratio. (2.3)

Note: FRVT is the fraction of the total offered load that is accepted/routed by the network, i.e., the percent routed volume of traffic. In fact, it can be understood as a kind of network throughput. Therefore, in the remainder of this report, we use the Network Throughput to represent the FRVT as a major performance metric.

- Blocking Rate of long and short requests, respectively.

$$Blocking\ Rate_{Short} = \frac{number\ of\ short\ requests\ blocked}{number\ of\ short\ requests\ offered} \quad (2.5)$$

$$Blocking\ Rate_{Long} = \frac{number\ of\ long\ requests\ blocked}{number\ of\ long\ requests\ offered} \quad (2.6)$$

Note: When the traffic is homogenous, the Network Throughput is equal to *(1- Blocking Rate)*.

2.5.5 Routing Scopes

Both of the following two routing scopes are considered in this thesis:

- **Intra-network routing**: This routing mechanism only considers the routing question within the local network. Intra-network routing seeks to control the selection of routes between origins and destinations within the scope of the local network.
- **Inter-network routing**: **Gateway Selection**. In contrast, inter-network routing seeks to control the selection of routes between the origins in the local network and the access points to foreign networks. As we can monitor the resource utilization of inter-network links, then inter-network routing actually seeks to select the best among several gateway nodes to a foreign network. This applies of course to inter-network traffic that can exit through multiple gateways. Our second routing issue focuses on this.

In summary, our first routing issue is routing the traffic with non-homogenous holding times. Thus, the traffic is non-homogenous, the routing scope is intra-network routing, and the simulation study is based on the 4-node and 12-node networks. Our second routing issue is the gateway selection. Thus, the traffic is homogenous, the routing scope is inter-network routing, and the simulation study is based on the 30-node network.

Besides, several concepts are interchangeable in the later chapters (apply to both routing issues), they are:

- Call, flow, connection, and request
- Route and Path

Chapter III: Intra-Network Routing with non-Homogenous Holding Times

In this chapter, we focus on routing in a MPLS-capable IP network with non-homogenous holding time traffic, which is the first and major routing issue in this study. Our basic idea is called the *short request vacating*. It means that under some traffic conditions, short requests will be routed on the two-hop routes instead of the direct one so as to vacate the bandwidth on the direct routes for future coming long requests.

We proceed as follows. We first illustrate the idea of vacating through a simple routing scenario. Then, based on the previously defined system model, we develop an analytical framework for our vacating idea. From the analytical results and its further generalization, two routing schemes are proposed:

- Preventive-vacating Routing (PVR)
- Preemptive-vacating Routing (PER)

Next, besides the basic vacating idea, other ideas are also analyzed to address our routing issue and the associated resulting routing schemes are proposed. They are:

- Approximated Least Cost Routing (A-LCR), in which the cost of accepting a long or short request is computed according to an approximated expression. We deduce this expression based on the work done in [63].
- Re-routing (RER)
- Restricted Access Routing (RAR), in which long and short requests have unequal right to access the network resources. It can be viewed as a modified version of Preventive-vacating Routing (PVR)

At last, we will introduce the Differentiated Shortest Distance Routing (Diff-SDR) scheme proposed in [17], which is the only currently published scheme to address the question of non-homogenous holding time traffic. Furthermore, the issues about the multi-TR thresholds and the implementation will also be considered at the end of this chapter.

3.1 Our Basic Idea: Vacating

We now consider the system model defined in Chapter II, which is:

- A well-connected and well-engineered packet-switched network.
- This network applies some form of state-dependent routing.
- There are two types of traffic flows carried by the network: long request flows with much longer holding time, and short request flows with much shorter holding time.

Then, we further consider a particular node pair *O-D* (Figure 3-1). Suppose that the route-set from node *O* to node *D*, $route-set_{O,D}$, has only two routes: a direct-route $route_{O,D}$ and a two-hop route $route_{O,T,D}$. In addition, each type of requests offers a volume of traffic for this *O-D* pair. The traffic that the direct route $route_{O,D}$ cannot carry must then be carried on the two-hop route $route_{O,T,D}$.

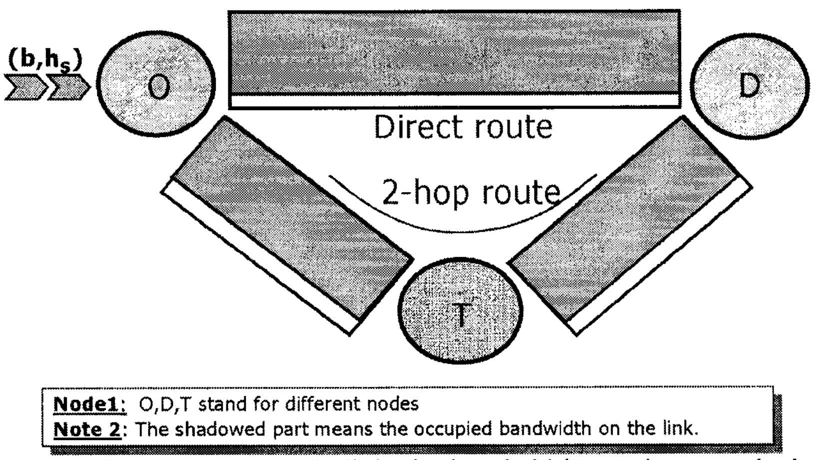

Figure 3-1: a simple routing scenario (b is the bandwidth requirement, h_S is the mean holding time of a short request)

3.1.1 The Routing Problem

Still in Figure 3-1, we further suppose that there is only *one bandwidth unit, b,* available in both the routes $route_{O,D}$ and $route_{O,T,D}$. Then, at time *t*=0, a short request (from *O* to *D*), with equivalent bandwidth requirement b and holding time h_S, arrives. The question is: how to select a route for this short request?

3.1.2 Two Routing Options

We can envision the following two routing options to address this situation:

i. We can allocate the short request to the direct route, which is the shortest path (this is what LLR+TR will do).

ii. We can use the two-hop route to serve the short request, and *vacate* bandwidth in the direct route for upcoming long requests.

We claim that option *ii* should provide a superior performance to option *i* (Figure 3-2), under certain conditions. Let us see what happens next.

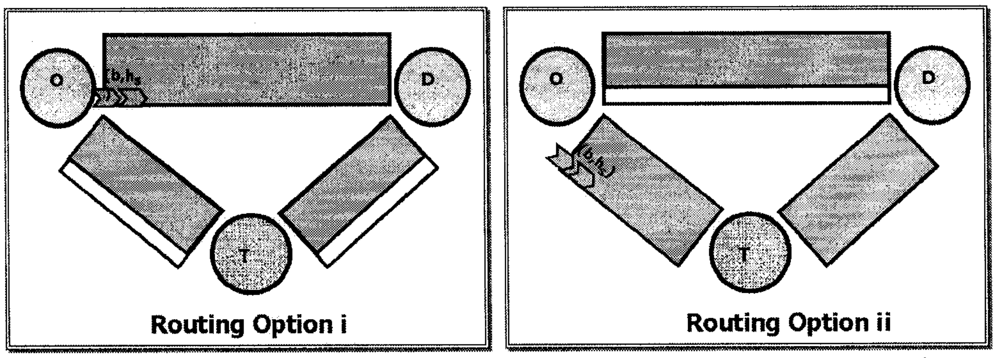

Figure 3-2: two routing options at time $t=0$. (b is bandwidth requirement, h_S is mean holding time of short request)

After a duration t_1 ($t_1 < h_S$), a new long request flow arrives, with required equivalent bandwidth b and holding time h_L, and $h_L >> h_S$. Then, according to Figure 3-2, in the scenario of option *i*, the coming long request flow must take the two-hop route $route_{O,T,D}$. As a result, capacity would be removed on the two links of the route for a very long time, measured on the time scale of the short requests. Meanwhile, in the scenario of option *ii*, the long request flow will take the direct route $route_{O,D}$, and only the capacity on one link would be removed for a long time (Figure 3-3).

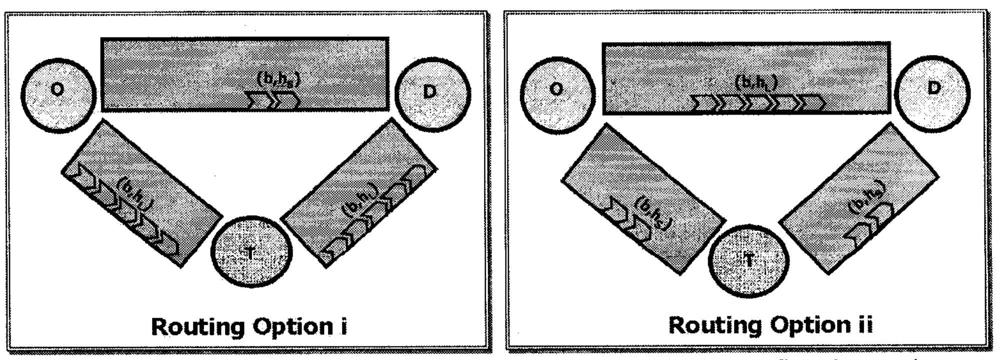

Figure 3-3: what happens at time $t=t_1$ when a long request flow is coming in the two routing options.

Then, after another duration $h_S - t_1$, the short request flow is over, and the resource utilizations in the two scenarios with different routing options are shown in the following Figure 3-4:

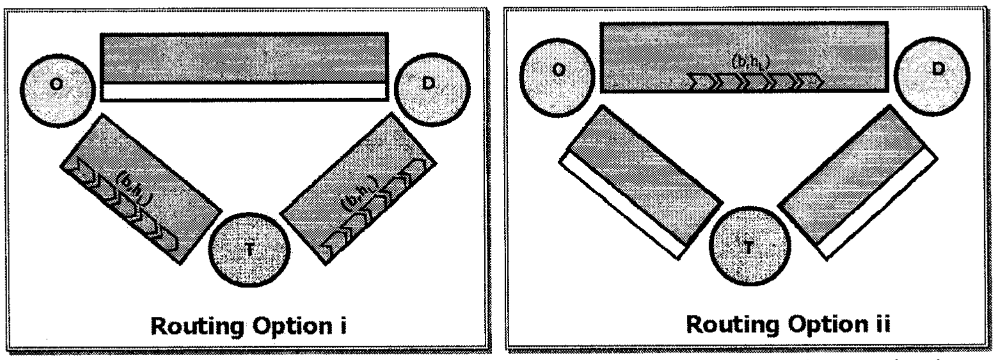

Figure 3-4: what happens at time t=h_S when the short request is over in the two routing options

It is evident that in the two scenarios, although two requests (short- and long-) were served, their induced costs, or the impact on future upcoming flows, are different:

- In routing option *i*, two links are full and only one link is idle.
- On the contrary, in routing option *ii*, only one link is full and two links are idle.

Therefore, in the above scenario, option *ii* performs better than option *i*.

3.1.3 Analysis

Actually, the level of activity on the routes is driven by two types of dynamics. The first dynamic is generated by the arrival and departure of long requests, while the second dynamics is generated by the arrival and departure of short requests. These dynamics occur on different time scales. Namely, the dynamic associated with the short requests evolves much faster than that associated with the long requests.

In routing option *i*, when we allocate the long request on the two-hop route, we remove the capacity on the two links of the route for a very long time, measured on the time scale of the short requests. The load induced by the short requests may vary considerably during this long period. As a result, the two links of the two-hop route that were idle when the direct route was selected may become busy for some time. In addition, these links are likely to overflow the traffic, which in turn will induce more cost.

In routing option *ii*, it is the short request that is allocated to the 2-link route. As the short request is driven by the fast dynamic, the state of the links that it uses will change much less during its existence. Thus, if the request was allocated to idle links, the probability that they change significantly enough to become busy will be much less. In other words, if the short request is allocated to an idle multiple-link route, those links are likely to stay idle while the short request still lives. If the state of the network evolves and another multiple-link route becomes a better choice, the next short request can then be assigned to another alternate route. Thus, option *ii* is also more flexible than option *i*.

In summary, the idea presented above is: in a non-homogenous traffic environment, the direct route may *not* always be the best and the first option for all kinds of traffic. Sometimes, an alternate route may become more "attractive". Obviously, the appropriate values of traffic mix should play an important role here. For instance, if the traffic mix is too low or extremely high, then option *ii* will become almost not necessary.

Furthermore, in Figure 3-1, regarding the issue of what to do when a long request flow arrives first, the optimal routing option is to put the long request flow on the direct route. If this is done, then whether a short request or long request flow comes next, no additional cost will be induced.

3.2 Formulation of the Basic Idea

To draw an analytical framework enabling the analysis of the idea expressed above, we use a concept called *cost rate* (c_r) to deal with the resource utilization cost of accepting a long request or short request flow. Suppose the cost of accepting a long request flow is $c_r \times h_L$, and the cost of accepting a short request flow is then $c_r \times h_S$. This means the cost of accepting a request is proportional to the holding time of this request [41]. Obviously, since $h_L > h_S$, the cost induced by a long request is much larger than the cost induced by a short request. Regarding the property of cost rate, please refer to the resource utilization cost and its curve (Figure 2-1) for details.

3.2.1 Assumptions

Having the same scenario as was used in the previous section, we suppose:

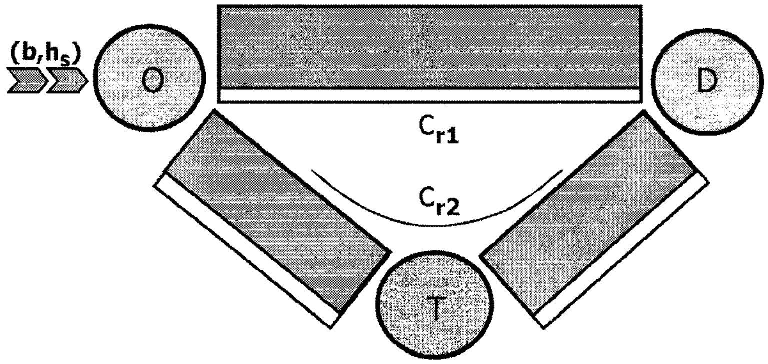

Figure 3-5: the routing scenario used in previous section

- Only one bandwidth unit is left on $route_{O,D}$ and $route_{O,T,D}$ (heavy traffic);
- The cost rate on the direct route $route_{O,D}$ is c_{r1}. The cost rate on the two-hop route $route_{O,T,D}$ is c_{r2}, and c_{r2} is the sum of cost rates on $link_{O,T}$ and $link_{T,D}$;
- There are two Poisson-type traffic: long request flows with arrival rate λ_L and exponentially distributed holding time (mean h_L), and short request flows with arrival rate λ_S and exponentially distributed holding time (mean h_S);
- Suppose holding time ratio, x, is 10. Then $h_L = 10 \times h_S$. Hence, h_L is one order of magnitude larger than h_S.

Now, if a new request (either a long or a short request) arrives from node O to node D, with required equivalent bandwidth b and holding time h, the question is still how do we select the route for it? Would we consider a direct route or a two-hop route, and why?

3.2.2 Induced Costs by different Routing Options

For Poisson traffic flows, we know:

- Pr(a request arriving during h) $= (1-e^{-\lambda h})$, where $\lambda = \lambda_L + \lambda_S$;
- Pr(a short request arriving first during h) $= (1-e^{-\lambda h}) \times p$;
- Pr(a long request arriving first during h) $= (1-e^{-\lambda h}) \times (1-p)$;
- $p = \dfrac{\lambda_S}{\lambda_S + \lambda_L} = \dfrac{\lambda_S}{\lambda}$ and $(1-p) = \dfrac{\lambda_L}{\lambda_S + \lambda_L} = \dfrac{\lambda_L}{\lambda}$.

Moreover, similarly to section 3.1.2, we have

- Routing option *i*: the request takes the direct route (LLR);
- Routing option *ii*: the request takes the two-hop route.

Then the cost induced by option *i* is

$$c_{r1}\times h+(1-e^{-\lambda h})\times[p\times c_{r2}h_S+(1-p)\times c_{r2}h_L] \quad (3.1)$$

where $c_{r1}\times h$ is the cost induced by accepting the request, and $(1-e^{-\lambda h})\times[p\times c_{r2}h_S+(1-p)\times c_{r2}h_L]$ is the cost induced by the new arriving request during h. Similarly, the cost induced by option *ii* is

$$c_{r2}\times h+(1-e^{-\lambda h})\times[p\times c_{r1}h_S+(1-p)\times c_{r1}h_L] \quad (3.2)$$

Then, the difference of the two costs is:

$$\begin{aligned}
&Cost_induced_by_option_i \; - \; Cost_induced_by_option_ii \\
&=(c_{r1}-c_{r2})\times h+(1-e^{-\lambda h})\times[p\times(c_{r2}-c_{r1})\times h_S+(1-p)\times(c_{r2}-c_{r1})\times h_L] \\
&=(c_{r1}-c_{r2})\times h+(1-e^{-\lambda h})\times(c_{r2}-c_{r1})\times[p\times h_S+(1-p)\times h_L] \\
&=(c_{r2}-c_{r1})\times\{[p\times h_S+(1-p)\times h_L]\times(1-e^{-\lambda h})-h\} \\
&=(c_{r2}-c_{r1})\times\{[p\times h_S+(1-p)\times x\times h_S]\times(1-e^{-\lambda h})-h_S\} \\
&=(c_{r2}-c_{r1})\times h_S\,\{[x-(x-1)\times p]\times(1-e^{-\lambda h})-1\} \qquad (3.3)
\end{aligned}$$

3.2.3 Short Arriving Request

if $h=h_S$, the arriving request is a short request, then equation (3.3) becomes:

$$\begin{aligned}
&(c_{r2}-c_{r1})\times\{[p\times h_S+(1-p)\times x\times h_S]\times(1-e^{-\lambda h_S})-h_S\} \\
&=(c_{r2}-c_{r1})\times h_S\times\{[x-(x-1)\times p]\times(1-e^{-\lambda h_S})-h_S\} \\
&=(c_{r2}-c_{r1})\times h_S\times\{[10-9p]\times(1-e^{-\lambda h_S})-1\} \qquad (let\ x=10) \qquad (3.4)
\end{aligned}$$

Hence, we have:

$$if\ \ either \begin{Bmatrix} c_{r2}>c_{r1}, \quad \text{and} \\ (1-e^{-\lambda h_S})>\dfrac{1}{x-(x-1)\times p}\overset{let\ x=10}{=}\dfrac{1}{10-9p} \end{Bmatrix} \quad (3.5.1),$$

$$or \begin{Bmatrix} c_{r2}<c_{r1}, \quad \text{and} \\ (1-e^{-\lambda h_S})<\dfrac{1}{x-(x-1)\times p}\overset{let\ x=10}{=}\dfrac{1}{10-9p} \end{Bmatrix} \quad (3.5.2),$$

then equation (3.4) > 0. As equation (3.4) is the cost difference of two routing options, this means the short-request should take the two-hop route.

Actually, in $\left\{\begin{matrix} c_{r2} > c_{r1}, \text{ and} \\ (1-e^{-\lambda h_S}) > \dfrac{1}{10-9p} \end{matrix}\right\}$, $(1-e^{-\lambda h_S}) > \dfrac{1}{10-9p}$ ensures that the arrival rate of requests should be large enough. In other words, during h_S, the probability of long request's arrival should be high enough. We notice that if $p \downarrow$ (decreasing the value), then $\dfrac{1}{10-9p} \downarrow$, which makes $(1-e^{-\lambda h_S}) > \dfrac{1}{10-9p}$ easier to be satisfied, where p is the probability that the incoming request is a short request ($p = \dfrac{\lambda_S}{\lambda_S + \lambda_L}$). Thus, $\lambda_L \uparrow$ (increasing the value) will lead to $p \downarrow$. This also means that the volume of long-request traffic should be large enough. Moreover, $c_{r2} > c_{r1}$ indicates that if the incoming long-request has to take a two-hop route, it will induce a higher cost. Combining these two conditions, then, clearly, the first arriving short request should "***vacate***" bandwidth on the direct route for an incoming long-request and take the two-hop route.

$\left\{\begin{matrix} c_{r2} < c_{r1}, \text{ and} \\ (1-e^{-\lambda h_S}) < \dfrac{1}{10-9p} \end{matrix}\right\}$ describes the contrary scene: $(1-e^{-\lambda h_S}) < \dfrac{1}{10-9p}$ ensures that during h_S, the probability of a long-request's arrival is low enough. We also notice that if $p \uparrow$, then $\dfrac{1}{10-9p} \uparrow$, which makes $(1-e^{-\lambda h_S}) < \dfrac{1}{10-9p}$ easier to be satisfied. Thus, during h_S, with a high probability, no long-request will come in. Therefore, the volume of long request traffic should be small enough. Then the short request should, of course, take the least cost route. According to $c_{r2} < c_{r1}$, it is the two-hop route. On the other hand, as the volume of long request traffic is low enough, then this "non-homogenous" traffic becomes much like a homogenous one, in which case LLR+TR is good enough.

Moreover, comparing [3.5.1] and [3.5.2], we should say that [3.5.1] is much more common. This is also due to the fact that in a well-engineeringed network, $c_{r2} > c_{r1}$ is a common case, while $c_{r2} < c_{r1}$ is not. (c_{r2} is the sum of two links' cost rates.).

Still from equation (3.4), we also have:

$$\textit{if either} \left\{\begin{matrix} c_{r2} > c_{r1}, \text{ and} \\ (1-e^{-\lambda h_S}) < \dfrac{1}{x-(x-1)\times p} \overset{let\ x=10}{=} \dfrac{1}{10-9p} \end{matrix}\right\} \quad (3.6.1),$$

$$\textit{or} \left\{\begin{matrix} c_{r2} < c_{r1}, \text{ and} \\ (1-e^{-\lambda h_S}) > \dfrac{1}{x-(x-1)\times p} \overset{let\ x=10}{=} \dfrac{1}{10-9p} \end{matrix}\right\} \quad (3.6.2),$$

then the cost difference < 0, which means the short-request should take the direct route.

About $\left\{\begin{matrix} c_{r2} > c_{r1}, \text{ and} \\ (1-e^{-\lambda h_S}) < \dfrac{1}{10-9p} \end{matrix}\right\}$, as the probability of a long-request arriving during h_S is low according to $(1-e^{-\lambda h_S}) < \dfrac{1}{10-9p}$, the short-request should take the least costly route, which is the direct route according $c_{r2} > c_{r1} \Rightarrow cost_{r2} > cost_{r1}$.

Regarding $\left\{\begin{matrix} c_{r2} < c_{r1}, \text{ and} \\ (1-e^{-\lambda h_S}) > \dfrac{1}{10-9p} \end{matrix}\right\}$, as the probability of a long request arriving during h_S is high according to $(1-e^{-\lambda h_S}) > \dfrac{1}{10-9p}$, then the short-request should take the most costly route, which is the direct route according to $c_{r2} < c_{r1}$, and ***vacate*** the least cost route, which is the two-hop route, for an incoming long-request.

3.2.4 Long Arriving Request

If $h=h_L$, the arriving request is a long request, from equation (3.3) we have:

$$\text{if either } \left\{\begin{matrix} c_{r2} > c_{r1}, \text{ and} \\ (1-e^{-\lambda h_L}) > \dfrac{x}{x-(x-1)\times p} \overset{let\ x=10}{=} \dfrac{10}{10-9p} \end{matrix}\right\};$$

$$\text{or } \left\{\begin{matrix} c_{r2} < c_{r1}, \text{ and} \\ (1-e^{-\lambda h_L}) < \dfrac{x}{x-(x-1)\times p} \overset{let\ x=10}{=} \dfrac{10}{10-9p} \end{matrix}\right\}, \tag{3.7}$$

then the cost difference is >0, and the long-request should take the two-hop route.

And,

$$\text{if either } \left\{\begin{matrix} c_{r2} > c_{r1}, \text{ and} \\ (1-e^{-\lambda h_L}) < \dfrac{x}{x-(x-1)\times p} \overset{let\ x=10}{=} \dfrac{10}{10-9p} \end{matrix}\right\};$$

$$\text{or } \left\{\begin{matrix} c_{r2} < c_{r1}, \text{ and} \\ (1-e^{-\lambda h_L}) > \dfrac{x}{x-(x-1)\times p} \overset{let\ x=10}{=} \dfrac{10}{10-9p} \end{matrix}\right\}, \tag{3.8}$$

then the cost difference is <0, and the long-request should take the direct route.

However, as $(1-e^{-\lambda h_L}) < 1$, $\dfrac{x}{x-(x-1)\times p} > 1$ and $p \neq 0$,

$$\Rightarrow \left\{(1-e^{-\lambda h_L}) > \frac{x}{x-(x-1)\times p} \overset{let\ x=10}{=} \frac{10}{10-9p}\right\} \text{ is always FALSE;}$$

$$\Rightarrow \left\{(1-e^{-\lambda h_L}) < \frac{x}{x-(x-1)\times p} \overset{let\ x=10}{=} \frac{10}{10-9p}\right\} \text{ is always TRUE.}$$

Therefore,

if $c_{r2} < c_{r1}$, then the long-request should take the two-hop route;

if $c_{r2} > c_{r1}$, then the long-request should take the direct route.

This means that the long request should always be routed to the route with the least cost. In most cases, especially under a heavy traffic load in the network, the cost rate of a two-hop route is larger than the cost rate of a direct route, say, $c_{r2} > c_{r1}$ (as c_{r2} is the sum of the cost rates on two links). Thus, if the arrival rate of long request flows is high, long request flows should take the least cost route as usual, while the short request flows occasionally have to *vacate* bandwidth for long request flows.

3.2.5 Least Cost Route

Notice that if $c_{r2} < c_{r1}$, the least cost route is the two-hop route; if $c_{r2} > c_{r1}$, then the least cost route is the direct route. Combining the notion of least cost route with the results of the above two sections together, we have:

	IF	Then Routing	Note
a.	$\left\{ \begin{array}{c} c_{r2} < c_{r1}, \text{ and} \\ (1-e^{-\lambda h_s}) < \dfrac{1}{10-9p} \end{array} \right\}$ (3.5.2),	**Long Request**: take the least cost route	This can be called Least Cost Routing, which is almost the same as LLR+TR.
	$or \left\{ \begin{array}{c} c_{r2} > c_{r1}, \text{ and} \\ (1-e^{-\lambda h_s}) < \dfrac{1}{10-9p} \end{array} \right\}$ (3.6.1),	**Short Request**: take the least cost route	
b.	$\left\{ \begin{array}{c} c_{r2} > c_{r1}, \text{ and} \\ (1-e^{-\lambda h_s}) > \dfrac{1}{10-9p} \end{array} \right\}$ (3.5.1),	**Long Request**: take the least cost route	This is different from LLR+TR.
	$or \left\{ \begin{array}{c} c_{r2} < c_{r1}, \text{ and} \\ (1-e^{-\lambda h_s}) > \dfrac{1}{10-9p} \end{array} \right\}$ (3.6.2)	**Short Request**: take the 2nd least cost route	

Table 3-1: Least Cost Route with Least Cost Routing.

From the above table, we find that LLR+TR is still effective under a traffic condition $(1-e^{-\lambda h_s}) < \dfrac{1}{10-9p}$; while in other traffic condition $(1-e^{-\lambda h_s}) > \dfrac{1}{10-9p}$, our vacating idea is effective. In other words, with non-homogenous holding times, both

LLR+TR and vacating have their own *non-overlapping effective range.* Furthermore, the effective-range should be a *tradeoff* among Pr[one long request arrives during the holding time of a short request], the holding time ratio and the arrival rates of short and long requests.

3.3 Generalization of the Analytical Framework

To find out the detailed effective range of the vacating idea, we need to generalize the analytical framework set up in previous section. Assuming:

- holding time ratio $h_L / h_S = x$
- traffic mix $\dfrac{\lambda_L \times h_L}{\lambda_L \times h_L + \lambda_S \times h_S} = y$
- the overall traffic volume is T

Then, we have:

$$\lambda_L \times h_L + \lambda_S \times h_S = T;$$

$$\Rightarrow \lambda_L = \frac{yT}{h_L} = \frac{yT}{xh_S};\ \lambda_S = \frac{(1-y)T}{h_S};\ \lambda = \lambda_L + \lambda_S = (1-y+y/x) \times \frac{T}{h_S};$$

$$P = \frac{\lambda_S}{\lambda} = \frac{1-y}{1-y+y/x};\ 1-P = \frac{\lambda_L}{\lambda} = \frac{y/x}{1-y+y/x}. \tag{3.9}$$

Thus, when the arriving request is a short request, the cost difference between LLR+TR and our vacating idea (equation 3.4) becomes:

$$(C_{r1} - C_{r2}) \times h_S + (1 - e^{-\lambda h_S}) \times [p \times (C_{r2} - C_{r1}) \times h_S + (1-p) \times (C_{r2} - C_{r1}) \times h_L]$$

$$= (C_{r1} - C_{r2}) \times h_S + (1 - e^{-\lambda h_S}) \times (C_{r2} - C_{r1}) \times [p \times h_S + (1-p) \times h_L]$$

$$= (C_{r2} - C_{r1}) \times h_S \times \{[x - (x-1) \times p] \times (1 - e^{-\lambda h_S}) - 1\} \tag{3.10}$$

$$= (C_{r2} - C_{r1}) \times h_S \times \left[(\frac{1}{1-y+y/x}) \times (1 - e^{-(1-y+y/x) \times T}) - 1 \right] \tag{3.11}$$

Let $x=10$, equation (3.10) is exactly the same as equation (3.4). Now let us focus on equation (3.11) in $c_{r2} > c_{r1}$ case, which is the general case. What will happen if we vary the traffic mix and holding time ratio? We draw the curve of equation (3.11) (assuming $c_{r2} > c_{r1}$) with the varying traffic mix and holding time ratio in the following Figure.

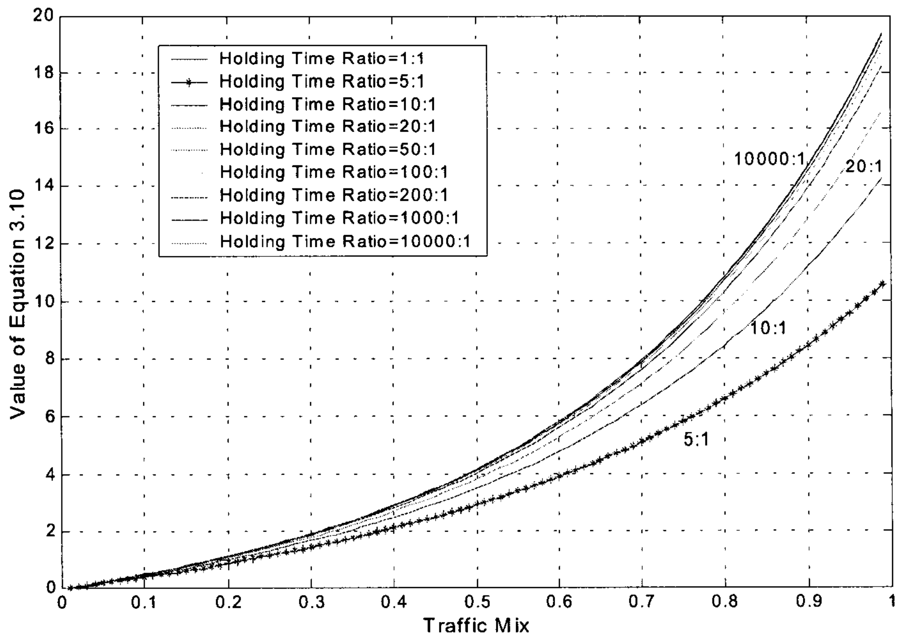

Figure 3-6: Curves of Equation 3.11 with varied traffic mix and holding time ratio.

It is obvious from the above Figure that when the traffic mix and holding time ratio increase, the cost difference also increases, which means that vacating is more valuable. However, until now, we have only analyzed one single vacating action, rather than *the performance of overall vacating actions*. That is to say, all the arriving short requests that find there is only one bandwidth unit left on the direct routes will start to vacate instantly. To find out the cost difference of LLR and overall vacating actions, we need to make two *fixedness assumptions* as follows.

Keeping the overall traffic volume, T, fixed, while the traffic mix and holding time ratio can be varied, for each single vacating action, we assume:

- Fixedness Assumption One: the value of $(C_{r2} - C_{r1})$ is fixed;
- Fixedness Assumption Two: the percentage of short requests taking vacating actions, denoted by v, is also fixed. v is computed as $\frac{\text{number of short requests taking vacating actions}}{\text{number of short requests offered}}$.

(Note: these two assumptions will be checked through experiments in Chapter VI, here, we suppose they are true.)

Hence, we have:

As all the arriving short requests will start to vacate when there is only one bandwidth unit left on the direct route, the cost difference for ONE single vacating action becomes:

$$(Cost\ by\ LLR - \ Cost\ by\ vacating)$$
$$= \{c_{r1} \times h_S + (1-e^{-\lambda h_S}) \times [p \times c_{r2} h_S + (1-p) \times c_{r2} h_L]\} -$$
$$\{c_{r2} \times h_S + (1-e^{-\lambda h_S}) \times [p \times c_{r2} h_S + (1-p) \times c_{r1} h_L]\}$$
$$= (c_{r1} - c_{r2}) \times h_S + (1-e^{-\lambda h_S}) \times (1-p) \times (c_{r2} - c_{r1}) \times h_L$$
$$= (c_{r2} - c_{r1}) \times h_S \times [(1-e^{-\lambda h_S}) \times x \times (1-p) - 1] \quad (3.12)$$

Now, based on *Fixedness Assumption One* and *Fixedness Assumption Two*, substitute equation (3.9) into equation (3.12), the overall cost difference in unit time is:

$$(arrival\ rate\ of\ short\ requests) \times v \times equation\ (3.12)$$
$$= \frac{(1-y)}{h_S} \times T \times v \times (C_{r2} - C_{r1}) \times h_S \times \left[(\frac{y}{1-y+y/x}) \times (1-e^{-(1-y+y/x) \times T}) - 1 \right]$$
$$= T \times (C_{r2} - C_{r1}) \times v \times (1-y) \times \left[(\frac{y}{1-y+y/x}) \times (1-e^{-(1-y+y/x) \times T}) - 1 \right] \quad (3.14)$$

3.3.1 Discussion

If the vacating wants to be better than LLR, we need equation (3.14)

$$T \times v \times \underbrace{(C_{r2} - C_{r1})}_{\text{named as } condition\ 1} \times (1-y) \times \underbrace{\left[(\frac{y}{1-y+y/x}) \times (1-e^{-(1-y+y/x) \times T}) - 1 \right]}_{\text{named as } condition\ 2} > 0\ .$$

(*Obviously, the bigger the cost difference, the more effective vacating will be.*)

Equation (3.14) is decided by:

- T: *T* is the overall traffic volume, and is always a positive value. It only has an impact on the amplitude of the cost difference.
- *v*: the percentage of short requests taking vacating actions, which is also non-negative $(0 \le v \le 1)$.
- Condition 1: $(C_{r2} - C_{r1})$ also has an impact on the amplitude of the cost difference. Since we consider a well-engineered network, it could be generally valid that $(C_{r2} - C_{r1}) > 0$. This is because C_{r2} is the summation of the cost

rates on two links (two-hop route), while C_{r1} is just the cost rate of one link (direct route). Furthermore, we believe that the traffic distribution on the direct and the two-hop routes as well as the choice of the TR threshold should have a significant impact on the value of $(C_{r2} - C_{r1})$. This will be studied experimentally.

- Condition 2: $(1-y)\times\left[(\frac{y}{1-y+y/x})\times(1-e^{-(1-y+y/x)\times T})-1\right]$ includes the traffic mix, holding time ratio and Pr [one long request arrives during the holding time of a short request]. It will give the tradeoff of the above parameters, which is exactly what we are looking for. Suppose $(C_{r2} - C_{r1}) > 0$, we draw the curves of equation (3.14) with varied traffic mix and holding time ratio in the following Figure.

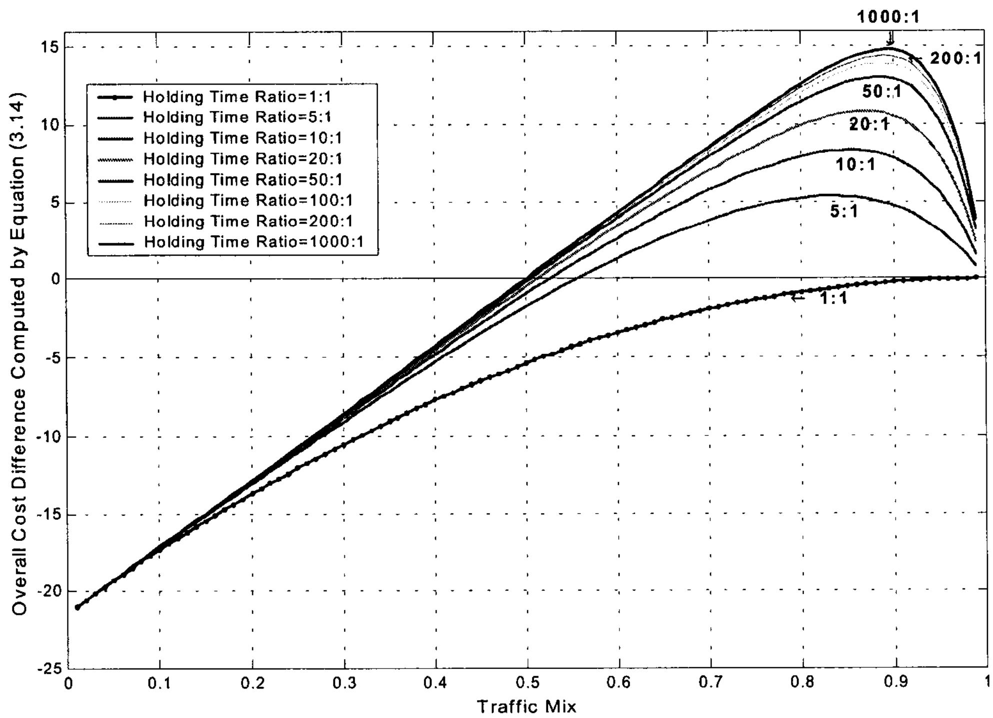

Figure 3-7: Curves of equation (3.14) with varied traffic mix and holding time ratio

Figure 3-7 shows that:

- There is an effectiveness range for vacating, around 60% ~99% in traffic mix;

- Beyond this range, the cost difference is negative, which means that the vacating idea is less efficient than LLR+TR. In another words, this is the effective range for LLR+TR;
- As the holding time ratio increases, so does the cost difference;
- But the increase in the cost difference seems to become *saturated* when the holding time ratio becomes very large;
- When the traffic mix is close to one, cost difference drops sharply. This is reasonable as there will be no vacating actions if the traffic mix is one;
- The maximum positive cost difference occurs around 85% in the traffic mix, while the minimal negative cost difference is located at 0% in the traffic mix.

Now, we can answer two key questions about vacating:

Question One: When should short requests start to vacate?

The answer is:

1) When the resource utilization of the direct link reaches a certain level (for instance, only one last bandwidth unit left); and
2) $C_{r2} > C_{r1}$ (this is generally satisfied in a well-engineered and well-dimensioned network); and
3) within the effective range of vacating.

Question Two: How big the improvement of vacating to LLR will be? (suppose the improvement is proportional to the value of cost difference.)

The answer is: the amplitude of the improvement is decided by

1) T (the overall fixed traffic in Erlang); and
2) v, the percentage of short requests taking vacating actions, which should be large enough; and
3) $(C_{r2} - C_{r1})$ (the traffic distributions on links, as well as the choice of TR thresholds, have influence on $(C_{r2} - C_{r1})$); and
4) the value of traffic mix in the effectiveness range of vacating. Within the range, the maximum improvement occurs at traffic mix around 85%.

3.3.2 Preventive-vacating and Preemptive-vacating

Vacating, as we discussed above, can be precisely defined as follows: in a certain range of resource utilization of a direct route, short requests should vacate bandwidth on the direct route for future arriving long requests, even if the direct route is not fully busy at this time. The short request should then be allocated to the least loaded two-hop route. Therefore, generally, vacating is to alternate short request traffic from direct routes to two-hop routes.

There can be two types of vacating. The first type of vacating is called *preventive-vacating*, which follows directly on the above line of thinking and discussions. For instance, when the resource (bandwidth) utilization on a direct route reaches a certain level, the arriving short requests should be allocated to the two-hop routes instead of the direct one, so that more future arriving long requests can take the direct route instead of the two-hop routes.

However, even if a short request flow vacates the direct route and takes a two-hop route, during its lifetime there may be or may *not* be a long request flow coming in. If a long request flow is coming in, then the vacating by the short request flow is valuable. If no long request flow is coming in, then the vacating will be much more costly than the short request taking the direct route. Therefore, not every instance of preventive-vacating is a beneficial action. What we are looking for is *statically-beneficial* vacating.

There could be another type of vacating, called *preemptive-vacating*. In preemptive-vacating, the vacating is started by long request flows in contrast to preventive-vacating, which is started by short request flows. In preemptive-vacating, the short request flows are always allocated to the least loaded route, that is to say, the direct route in general. The preemptive-vacating happens when a long request flow arrives and finds the direct route fully busy. Then, a suitable in-progress short request flow on the direct route is chosen, preempted (not interrupted but re-routed), and moved to a two-hop route. The vacated bandwidth on the direct route is then occupied by the arriving long request flow.

Obviously, the preemptive-vacating could be considered as an ideal scenario of preventive-vacating, illustrated by $(1-e^{-\lambda h})=1$ and $p=0$ in equation (3.10). We can thus take the preemptive-vacating as a kind of performance benchmark for the preventive-vacating. Then equation (3.10), which is the cost difference between LLR+TR and preventive-vacating, becomes:

$$\begin{aligned}&\textit{cost difference between LLR and one preemptive-vacating}\\&=(C_{r2}-C_{r1})\times h_S\times(x-1)\end{aligned}\qquad(3.15)$$

Hence, unlike preventive-vacating, preemptive-vacating needs just one condition, $(C_{r2}-C_{r1})>0$, to outperform LLR+TR. If this single condition is satisfied, then the effective range of preemptive-vacating will be the *full* range of the traffic mix.

Remember what we have done in generalizing the formulation of preventive-vacating, it is natural to question whether or not we can do the same thing here and obtain an equation for overall preemption-vacating, just like equation (3.14) with preventive-vacating. Still keeping the overall traffic volume, *T*, fixed, while the traffic mix, y, and holding time ratio, x, can be varied, for each preemptive-vacating, we suppose:

- $(C_{r2}-C_{r1})$ is fixed;
- the percentage of long requests taking preemptive-vacating actions, denoted by w, is also fixed. w is computed as $\frac{\text{number of long requests taking preemptive vacating actions}}{\text{number of long requests offered}}$.

From equation (3.9) we know the arrival rate of long requests is

$$\lambda_L=\frac{yT}{h_L}=\frac{yT}{xh_S};$$

Then, the overall cost difference between LLR+TR and preemptive-vacating in unit time

$$\begin{aligned}&=(\textit{arrival rate of long requests})\times w\times(\textit{equation } 3.15)\\&=\frac{yT}{xh_S}\times w\times(C_{r2}-C_{r1})\times h_S\times(x-1)\\&=T\times\frac{x-1}{x}\times w\times y\times(C_{r2}-C_{r1})\end{aligned}\qquad(3.16)$$

If $x\uparrow$, $(\frac{x-1}{x})\approx 1$. Thus, the holding time ratio, x, should have little influence on the overall outperformance of preemptive-vacating compared to LLR+TR, especially when x is large. We cannot predict now how the value of $w\times y$ will change, which should be investigated through simulation. Nevertheless, we could say that the shape of the curve of equation (3.16) should be decided by the value of $w \times y$. This should also be verified by simulation.

3.4 Proposed Vacating Routing Schemes

Based on the analytical results of the previous two sections, we can propose routing schemes for a network with non-homogenous holding time traffic. Furthermore, the two types of vacating result in two vacating routing schemes.

Remember that in Section 2.1, we summarized the state-dependent routing scheme in circuit-switched networks as "*Think before routing*". Here, our proposed vacating routing schemes can be summarized as "*Think again before routing*" in IP networks. The "again" indicates the *differentiated* routing: according to the arriving flow being a short- or a long request, varied detailed state-dependent routing procedures are followed.

Before giving a precise description for the routing schemes, we must define several necessary concepts:

- Idle capacity: The *idle capacity* of a link is defined as the amount of link bandwidth that is currently *not* in use. We define the idle capacity of a route as the minimum idle capacity of all of its links.
- QoS-permissibility: A route, direct or two-hop, is said to be *QoS-permissible* if it has sufficient idle capacity to carry the request.
- TR-permissibility: For a request, a two-hop route is said to be *TR-permissible* if its idle capacity minus the trunk reservation threshold is greater than or equal to the requested equivalent bandwidth of the incoming request. Note that if a two-hop route is TR-permissible then it is also QoS-permissible.
- Preemption-permissibility: A *preemption-permissible* short request is defined as an in-progress and not-alternatively-routed short request on the direct route. (i.e. the source and destination nodes of the short request are connected by this direct route.)

3.4.1 Preventive-vacating Routing Scheme

When a new long request arrives,

i. Route this long request to the direct route if the direct route is QoS-permissible. Otherwise, go to step ii.

ii. If no TR-permissible alternate (two-hop) routes are available, then the arriving long request is rejected. Otherwise, the long request is routed to a TR-permissible alternate route with the largest idle capacity.

When a new short request arrives,

i. Route this short request to the direct route if idle capacity of the direct route is greater than vacating threshold. Otherwise, go to step ii.

ii. If there is at least one TR permissible alternate route, route this short request to a TR-permissible alternate route with the largest idle capacity. Otherwise, go to step iii.

iii. Reject the short request if the direct route is not QoS-permissible. Otherwise, Route this short request to the direct route.

Note: The vacation threshold is generally less than the TR. It could be, for instance, the last bandwidth unit. (see Figure 3-8). In other words, the TR first reserves the link for direct traffic (both short and long requests). In a second step, the vacating threshold reserves the link only for long requests. See the flow chart for short request routing in Figure 3-9.

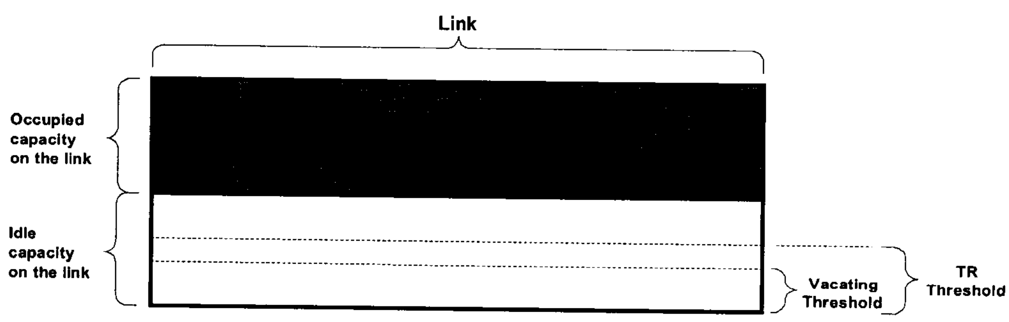

Figure 3-8: The link, Vacating Threshold and TR Threshold

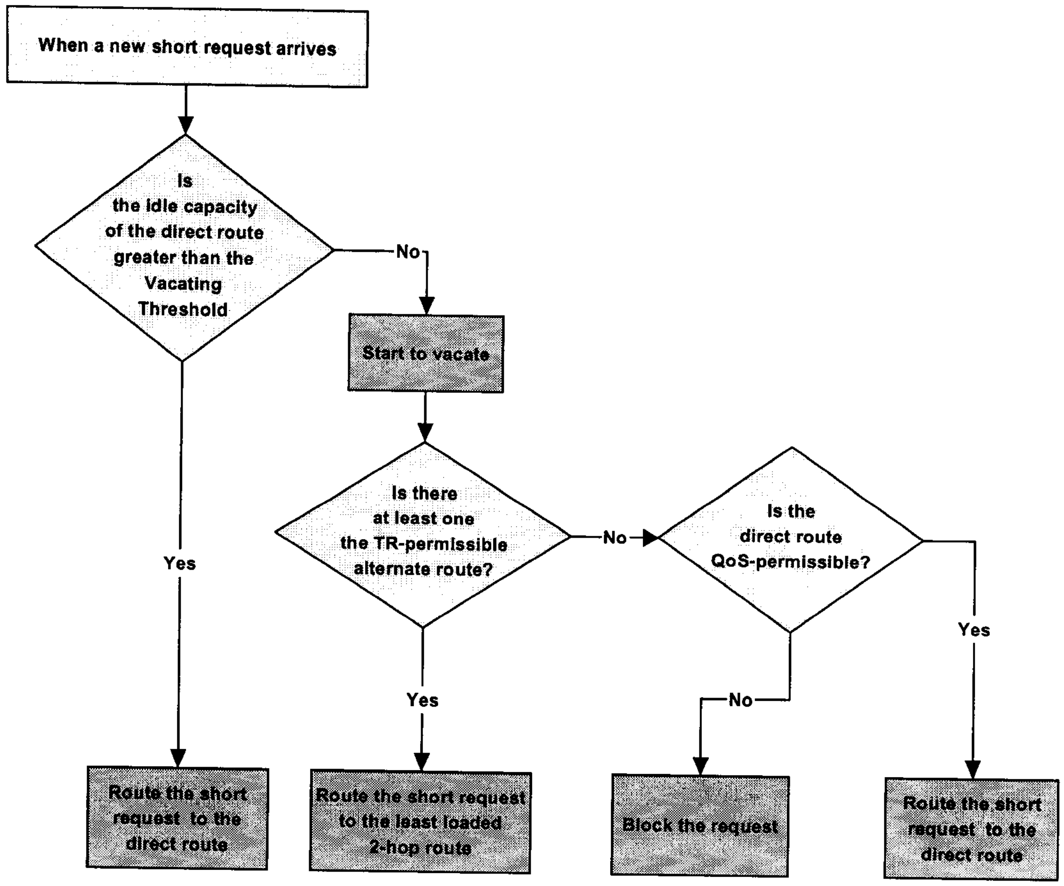

Figure 3-9: Routing flowchart for short requests in Preventive-vacating routing.

3.4.2 Preemptive-vacating Routing Scheme

When a new short request arrives,

i. Route this short request to the direct route if the direct route is QoS-permissible. Otherwise, go to step ii.

ii. If no TR-permissible alternate routes are available, then the arriving short request is rejected. Otherwise, the short request is routed to a TR-permissible alternate route with the largest idle capacity.

When a new long request arrives,

i. Route this long request to the direct route if the direct route is QoS-permissible. Otherwise, go to step ii.

ii. If no TR-permissible alternate routes are available, then the arriving long request is rejected. Otherwise,

a. if there is at least one preemption-permissible short request on the direct route, then start the vacating action: preempt a randomly-selected preemption-permissible short request from the direct route to a TR-permissible alternate route with the largest idle capacity. After this, route the long request to the direct route.

b. if there is *no* preemption-permissible short request on the direct route, then route the long request to a TR-permissible alternate route with the largest idle capacity.

For performance comparison purposes, we also define the **LLR+TR** routing scheme in detail as follow:

i. An arriving request, whether long or short, is routed to a direct route if the direct route is QoS-permissible. Otherwise, go to step ii.

ii. If no TR-permissible alternate routes are available, then the arriving request is rejected. Otherwise, the request is routed to a TR-permissible alternate route with the largest idle capacity, i.e. the least loaded.

Note that in the preventive-vacating routing scheme, the routing procedure for long request is the same with LLR+TR. While in preemptive-vacating, it is the routing procedure for short request that is the same as LLR+TR.

3.4.3 Discussion

Actually, the considerations similar to our vacating idea can also be found among several *packing* routing schemes in multi-rate loss networks [43,44]:

- Most-Loaded Routing (MLR): The MLR scheme attempts to pack traffic with lower bandwidth requirements on alternative routes by favoring the most utilized. Thus, it attempts to leave other alternative routes very lightly loaded, and increases the chances of admitting arriving traffic with large bandwidth requirements.

- Multi-Rate Least-Loaded Routing with Packing (MLLRP): the MLLRP scheme attempts to pack some kinds of traffic request by considering the next higher class of traffic. It focuses on the differences of bandwidth requirements among traffic classes. While accepting current-arriving traffic with lower bandwidth requirement, it also increases the chance of admitting future-arriving traffic with large bandwidth requirements.
- Multi-Rate Least-Loaded Routing (MLLR): The MLLR scheme forces classes of traffic requests with lower bandwidth requirements on alternative routes when the load of the given direct route reaches a certain threshold. Then, even if the direct route can still support the traffic request, the MLLR algorithm may route the request on the least loaded alternative route, provided that it is sufficiently "attractive". The alternative route is said to be attractive if its idle capacity exceeds z times the idle capacity of the direct route. The value of z is chosen to be higher for classes with higher bandwidth requirements. It is shown in [44] that MLLR performs well only in a very ill-dimensioned network.

Notice that MLLR is similar to our preventive-vacating routing, but our vacating scheme focuses on the non-homogenous holding time, while all these packing schemes, as we described above, center only on the difference in bandwidth requirements among requests. They do not care about holding time.

The biggest difference between routing in a multi-rate environment and in a non-homogenous holding time environment is whether the right of different requests to access the network resource is equal or not.

In a multi-rate network environment, the right to access the network resource is *unequal* for different types of requests. Obviously, a request with larger bandwidth requirement has a lower priority/chance/possibility than a request with smaller bandwidth requirement to access the network resource. The result is that the blocking rate of the requests with small bandwidth requirement is much lower than that of the requests with larger bandwidth requirement. Hence, all packing routing schemes try to equalize the access right among the different types of requests, so that they can have similar blocking rates so as to improve the overall network throughput.

On the contrary, in a non-homogenous holding time environment, the requests with various types of holding times have an equal right to access the network resources. Moreover, the blocking rates of various types of request are equal and our vacating schemes focus more on the different costs and the impact on network congestion induced by long and short requests.

3.5 Approximated Cost Rate and Least Cost Routing

In previous sections, we only use the concept of cost rate. Now, we try to find out an approximated expression for the cost rate, from the previous work on MDP approaches in multi-service networks. We first have a review on MDP approaches in multi-service networks. Next, we focus on the work done in [63]. Based on it, we deduce the expression for the cost rate, and then the cost of accepting a long or a short request. Finally, a resulting least cost routing scheme will be proposed.

3.5.1 MDP Approaches in Multi-Service Networks

In multi-service networks, due to the huge state space required for exact modeling the routing problem into a Markov decision process, all previous researchers made two assumptions: a *link independence assumption* and a *route cost separability assumption*. First, the link independence assumption assumes that a call carried on an n-link (n-hop) route behaves as n independent calls. Second, the route cost separability assumption assumes that the cost of carrying a call on a route is the sum of the cost of each individual link of the route. In addition to these two assumptions, various further approximations are used to reduce the complexity of cost computation. But so far, no general closed form expression, similar to equation (2.1), has been published for multi-rate networks.

Kolarov [42] considers a one-link model with two kinds of traffic and shows that the system of linear equations (in computing the relative values of costs) associated with the two-dimensional Markov chain can be decomposed into a system of linear equation associated with the one-dimensional Markov, which can be easily solved. However, his analysis is strictly based on a less general complete-partitioning policy with capacity borrowing. Krishnan and Hübner [31] proposed annother efficient numerical scheme for calculating an approximation (through state-space aggregation) of the optimal one-step improved policy and its expected costs, using CS (complete sharing) as the initial policy. However, in [32], through simple examples, the authors demonstrated the possibility that the approximations made in [31] might lead to degraded performance, instead of the expected improvement. Using the concept of "quantization", Lea and Ke [60] propose an approximation scheme to implement state aggregation.

Dziong and Mason [26, 40,41] use an approach similar to [22,23], but instead of the one-step policy iteration used in most MDP approaches, they employed the policy improvement lemma [13] repeatedly to yield successively better policies in a continuous manner. In addition, they originally adopted the decomposition of network reward process into a set of separable link reward processes to achieve an implementable solution.

Hwang's work [36,63] is very similar to Dziong and Mason's, but his approximation scheme provides more accurate routing information while requiring no additional computation. Hwang got a MDP formulation much like the one in the telephone network case (equation 2.1) [22,23] with the notable difference that call loss has been extended to revenue loss in order to account for multiple traffic classes. We use Hwang's result [63] as the basis to find out the expression for the cost rate.

3.5.2 Hwang's Work [63]

The decrease of MDP computational complexity in Hwang's work [63] is based on two ideas. First, the link independence assumption and the route cost separability assumption are made so that the links can be modeled as independent Markov decision process. Next, in order to make the link models tractable, the multi-rate traffic is replaced by a single state-dependent Poisson arrival stream consisting of single rate traffic so that it can be modeled as a birth-death process. This is inspired from the approximation proposed by Chung and Ross [59]. Then, Hwang presents the following:

Consider that a multi-rate network handles K traffic classes labeled $k=1,\ldots,K$. Assume that class k calls require b_k units of bandwidth, and its holding time is exponentially distributed with mean $1/\mu_k$.

When a call arrives in the network, it will be either carried on a route or lost (blocked), depending on the routing policy being used. Further assume that each class k call carried on the network produce r_k units of revenue/reward. From another perspective, the network will lose r_k units of revenue for each class k request that is rejected. Note that although $r_k \in [0,\infty)$, it is usually set to be b_k/μ_k [63,41,42].

Describe the link state of link l by the number of busy bandwidth units. The cost (or state-dependent link shadow price) of adding a class k request to link l at link state i, $p_k^l(i)$, is computed as :

$$p_k^l(i)=\left[v^l(i+b_k)-v^l(i)\right]/\mu_k, \tag{3.17}$$

where the difference of relative values, $\delta^l(i)=v^l(i+1)-v^l(i),\ 1\le i\le N^l$, is computed by the following set of equations:

$$\delta^l(N^l-1)=v^l(N^l)-v^l(N^l-1)=\frac{\sum_{j=1}^K r_j^l\lambda_j^l}{\bar{\lambda}_{C-1}{}^l}\times\frac{E_b(\bar{\lambda}_0{}^l,N^l)}{E_b(\bar{\lambda}_0{}^l,N^l-1)}, \tag{3.18}$$

$$\delta^l(i-1)=v^l(i)-v^l(i-1)=\frac{g^l}{\bar{\lambda}_{C-1}{}^l E_b(\bar{\lambda}^l,i-1)},\ 1\le i<N^l, \tag{3.19}$$

where $E_b(\ ,\ )$ is Erlang-B formula, and

$$g^l=\sum_{j=1}^K r_j^l\lambda_j^l-N^l\times\left[v^l(N^l)-v^l(N^l-1)\right], \tag{3.20}$$

$$\bar{\lambda}_i{}^l=\frac{\left(\sum_{j=1}^K b_j\lambda_j^l/\mu_j\right)^2}{\sum_{j=1}^K b_j^2\lambda_j^l/\mu_j}+i\left(1-\frac{\sum_{j=1}^K b_j\lambda_j^l/\mu_j}{\sum_{j=1}^K b_j^2\lambda_j^l/\mu_j}\right). \tag{3.21}$$

Notation:

- r_k : call reward (the reward of accepting a class k call);
- r_k^l : the link average reward of accepting a class k call on link l;
- λ_k^l : the offered arrival rate of class k call on link l.

Note that the relationship between r_k and r_k^l is called the reward distribution rule. Several reward distribution rules in multi-rate networks were given in [63,40,41].

It should also be noted that the concept of state-dependent link shadow price is very similar to the resource utilization cost. They are both used to evaluate the impact of accepting a call on the blocking of future coming calls. In the single-rate case (traditional telephone network), the state-dependent link shadow price has exactly the same expression (equation 2.1) as the resource utilization cost.

Accepting a class k call will generate r_k units of revenue to the network, but will also induce p_k units of cost (state-dependent shadow price) at the same time. Thus, we could have the notion of the *route net_gain* [40,41] to evaluate whether accepting this call is benefical or not.

According to the route cost separability assumption, the cost of adding a class k call on a route with links $l_1,...,l_M$, in respective states $i_1,...,i_M$, is given by [63]:

$$route\_cost = \sum_{n=1}^{M} p_k^{l_n}(i_n). \tag{3.22}$$

Thus, the *route net_gain* is equivalent to $(r_k - route\_cost)$.

3.5.3 Cost Rate for non-Homogenous Traffic

Based on Hwang's results, we then try to find out the cost rate on link l in our case, say, $K = M = 2,\ b_L = b_S = one\ unit\ bandwidth$;

let $\rho_L{}^l = \lambda_L{}^l / \mu_L$ and $\rho_S{}^l = \lambda_S{}^l / \mu_S$,

from (3.17), we have

$$p_k^l(i) = \left[v^l(i+1) - v^l(i)\right] / \mu_k, \quad k \text{ is } L, \text{ or } S, \tag{3.23}$$

Thus, the cost rate of adding a long request or a short request to link l at link state i, $C_r{}^l(i)$, is $v^l(i+1) - v^l(i)$.

From (3.21),

$$\bar{\lambda}_i{}^l = \frac{\left(\lambda_L{}^l / \mu_L + \lambda_S{}^l / \mu_S\right)^2}{\lambda_L{}^l / \mu_L + \lambda_S{}^l / \mu_S} = \rho_L{}^l + \rho_S{}^l; \tag{3.24}$$

From (3.18),

$$\delta^l(N^l - 1) = v^l(N^l) - v^l(N^l - 1) = \frac{r_L{}^l \lambda_L{}^l + r_S{}^l \lambda_S{}^l}{\rho_L{}^l + \rho_S{}^l} \times \frac{E_b(\rho_L{}^l + \rho_S{}^l, N^l)}{E_b(\rho_L{}^l + \rho_S{}^l, N^l - 1)}; \tag{3.25}$$

Bring equation (3.25) into equation (3.20), we have

$$g^l = \left(r_L{}^l\lambda_L{}^l + r_S{}^l\lambda_S{}^l\right) - N^l \times \left[\frac{r_L{}^l\lambda_L{}^l + r_S{}^l\lambda_S{}^l}{\rho_L{}^l + \rho_S{}^l} \times \frac{E_b(\rho_L{}^l + \rho_S{}^l, N^l)}{E_b(\rho_L{}^l + \rho_S{}^l, N^l - 1)}\right]$$

$$= \left(r_L{}^l\lambda_L{}^l + r_S{}^l\lambda_S{}^l\right) \times \left[1 - \frac{N^l}{\rho_L{}^l + \rho_S{}^l} \times \frac{E_b(\rho_L{}^l + \rho_S{}^l, N^l)}{E_b(\rho_L{}^l + \rho_S{}^l, N^l - 1)}\right]$$

$$= \left(r_L{}^l\lambda_L{}^l + r_S{}^l\lambda_S{}^l\right) \times \left[1 - \frac{N^l}{\rho_L{}^l + \rho_S{}^l} \times \frac{\dfrac{(\rho_L{}^l + \rho_S{}^l)^{N^l} / N^l}{\sum_{i=0}^{N}(\rho_L{}^l + \rho_S{}^l)/i!}}{\dfrac{(\rho_L{}^l + \rho_S{}^l)^{N^l - 1} / (N^l - 1)}{\sum_{i=0}^{N^l - 1}(\rho_L{}^l + \rho_S{}^l)/i!}}\right]$$

$$= \left(r_L{}^l\lambda_L{}^l + r_S{}^l\lambda_S{}^l\right) \times \left[1 - \frac{N^l}{\rho_L{}^l + \rho_S{}^l} \times \frac{\left(\sum_{i=0}^{N^l - 1}(\rho_L{}^l + \rho_S{}^l)/i!\right) \times \left((\rho_L{}^l + \rho_S{}^l)^{N^l} / N^l\right)}{\left(\sum_{i=0}^{N^l}(\rho_L{}^l + \rho_S{}^l)/i!\right) \times \left((\rho_L{}^l + \rho_S{}^l)^{N^l - 1} / (N^l - 1)\right)}\right]$$

$$= \left(r_L{}^l\lambda_L{}^l + r_S{}^l\lambda_S{}^l\right) \times \left[1 - \frac{\sum_{i=0}^{N^l - 1}(\rho_L{}^l + \rho_S{}^l)/i!}{\sum_{i=0}^{N^l}(\rho_L{}^l + \rho_S{}^l)/i!}\right]$$

$$= \left(r_L{}^l\lambda_L{}^l + r_S{}^l\lambda_S{}^l\right) \times \frac{(\rho_L{}^l + \rho_S{}^l)^{N^l} / N^l!}{\sum_{i=0}^{N^l}(\rho_L{}^l + \rho_S{}^l)/i!}$$

$$= \left(r_L{}^l\lambda_L{}^l + r_S{}^l\lambda_S{}^l\right) \times E_b(\rho_L{}^l + \rho_S{}^l, N^l) \tag{3.26}$$

From (3.19) and (3.26), we then have

$$v^l(i+1) - v^l(i) = \frac{g^l}{\bar{\lambda}_{N^l-1}{}^l E(\bar{\lambda}^l, i)} = \frac{r_L{}^l\lambda_L{}^l + r_S{}^l\lambda_S{}^l}{\rho_L + \rho_S} \times \frac{E(\rho_L{}^l + \rho_S{}^l, N^l)}{E(\rho_L + \rho_S, i)},\ 1 \le i < N^l, \tag{3.27}$$

Combine (3.27) with (3.25), we get the expression for state-dependent cost rate in link l, with state i, :

$$C_r{}^l(i) = \frac{r_L{}^l\lambda_L{}^l + r_S{}^l\lambda_S{}^l}{\rho_L{}^l + \rho_S{}^l} \times \frac{E(\rho_L{}^l + \rho_S{}^l, N^l)}{E(\rho_L + \rho_S, i)},\ 1 \le i \le N^l, \tag{3.28}$$

Therefore, the cost of adding a long request to link l at link state i is computed as:

$$C_r{}^l(i)/\mu_L = \frac{r_L{}^l\lambda_L{}^l + r_S{}^l\lambda_S{}^l}{\rho_L{}^l + \rho_S{}^l} \times \frac{E(\rho_L{}^l + \rho_S{}^l, N^l)}{E(\rho_L{}^l + \rho_S{}^l, i)} \times \frac{1}{\mu_L}, \tag{3.29}$$

the cost of adding a short request to a link at link state i is computed as:

$$C_r{}^l(i)/\mu_S = \frac{r_L{}^l\lambda_L{}^l + r_S{}^l\lambda_S{}^l}{\rho_L{}^l + \rho_S{}^l} \times \frac{E(\rho_L{}^l + \rho_S{}^l, N^l)}{E(\rho_L{}^l + \rho_S{}^l, i)} \times \frac{1}{\mu_S}. \tag{3.30}$$

Regarding the homogenous holding time case (let $r_L^{\,l} = r_S^{\,l} = 1$, $\lambda_L^{\,l} = \lambda_L^{\,l} = \lambda$, $\mu_L = \mu_S = \mu$), the cost rate is:

$$C_r(i) = \frac{r_L\lambda_L + r_S\lambda_S}{\rho_L + \rho_S} \times \frac{E(\rho_L + \rho_S, N)}{E(\rho_L + \rho_S, i)} = \frac{\lambda}{\rho} \times \frac{E(\rho, N)}{E(\rho, i)}, \quad 1 \le i \le N,$$

Thus the state-dependent cost (link shadow price) of adding a request to a link at state i is

$$\frac{C_r(i)}{\mu} = \frac{\lambda}{\rho} \times \frac{E(\rho, N)}{E(\rho, i)} \times \frac{1}{\mu} = \frac{E(\rho, N)}{E(\rho, i)},$$

which is exactly same as the resource utilization cost in equation (2.1).

3.5.4 Least Cost Routing with non-homogenous Holding Time

Then, based on equations (3.28, 3.29 and 3.30), we can have an *approximated least cost routing* (A-LCR) for non-homogenous holding time traffic, and the criterion of routing is:

When a new request, whether long or short, arrives,

1. *Compute the largest net_gain among all the routes*; (k *is* L *or* S)
2. *If the largest net_gain is positive, allocate the request to the route with the largest net_gain; otherwise, block the request.*

Note: In this study, only the holding time is non-homogenous. Moreover, we are studying well-engineered networks, according to [63], we use a simple reward distribution rule, just like the one in telephone networks. That is, $r_k = r_k^l$, k is L or S.

3.5.5 The Reward Parameters

Besides computing cost, another important advantage of reward parameters is that a flow control mechanism is automatically provided by the A-LCR scheme. This flow control mechanism is also self-adaptive. Thus, through simply adjusting the value of r_k (k is L or S), we can control the Grade of Service (GoS, i.e. blocking rate) of either long requests or short requests. It is worth noting that among all the routing schemes we proposed, *only* A-LCR has this special capability.

In addition, with the reward parameters, we can have another kind of system performance metric, beyond the Network Throughput defined in Section 2.5.4. It is the *Fractional Revenue*:

$$\frac{No.\ of\ carried\ long\text{-}requests \times r_L + No.\ of\ carried\ short\text{-}requests \times r_S}{No.\ of\ offered\ long\text{-}requests \times r_L + No.\ of\ offered\ short\text{-}requests \times r_S};$$

or, the *Fractional Revenue Loss* [69]:

$$1 - \frac{No.\ of\ carried\ long\text{-}requests \times r_L + No.\ of\ carried\ short\text{-}requests \times r_S}{No.\ of\ offered\ long\text{-}requests \times r_L + No.\ of\ offered\ short\text{-}requests \times r_S}.$$

Obviously, if r_L and r_S are proportional to the holding time associated with long and short requests, the Fractional Revenue is equal to the Network Throughput. Unlike other routing schemes, A-LCR is inherently suitable to routing under this new system performance metric.

3.6 Re-routing

We know that in dynamic routing, a routing decision must be made at call-arrival time based on the network information available at that time. However, the decision once made is final. Re-routing (also called call repacking) has a different mechanism. It is the practice whereby calls on alternate routes can be re-routed back to direct routes or other less congested alternate routes as the situation warrants.

Literature about re-routing is plentiful; it has been studied widely in regard to traditional telephone networks [5,6,46,47], ATM networks [16,35], circuit-switched wavelength-division-multiplexed all-optical networks [34], MPLS networks [21], and even in advanced cellular networks [28]. For instance, during a handoff in wireless networks, a call is re-routed from one base-station/mobile-switch-center to another base-station/mobile-switch-center. In [21], a novel approach for the optimal routing of new label switched paths (LSPs) in a MPLS-based network is proposed. This approach is based on the idea of allowing the re-routing of an already established LSP when there is no other way to route the new one. It is shown in [21] by numerical evaluation that this approach improves largely the success probability of setting up new LSPs.

Actually, the preemptive-vacating routing scheme we proposed in Section 3.4 has adopted re-routing partially. Now, we will introduce how the "pure" re-routing works in networks with homogenous traffic and then show how we are going to study it in non-homogenous holding time traffic.

3.6.1 Re-routing in Homogenous Traffic

Almost all the previous related studies have focused on the behavior of re-routing in homogenous traffic conditions (homogenous bandwidth requirement, homogenous holding time). In homogenous traffic (e.g., telephone networks), the primary property of re-routing scheme is its outstanding performance. It is shown in [5,6,46,47] that the re-routing, without trunk reservation, can provide a significant throughput increase over LLR+TR under all conditions.

The second most significant property of re-routing is *self-stabilizing* ability. It is well known that when using alternate routing freely in telecommunication networks, high blocking rate can happen under a heavy network load. This was already investigated well in traditional telephone networks [1,2], and is called the *instability* of the network. The high blocking rate is due to the excessive use of alternate routes. Two solutions have been found to solve the problem of instability: Trunk Reservation and re-routing.

Re-routing is shown [46,47] to be an effective means for maintaining the stability of the network under dynamic routing. Hence, trunk reservation becomes no longer necessary with re-routing. Actually, trunk reservation will increase the blocking probability when re-routing is used [5,6,47].

3.6.2 Re-routing in non-homogenous Holding Time Traffic

To study re-routing in non-homogenous holding time traffic, we will focus on the

- performance of re-routing;
- comparison with the preventive-vacating routing scheme;
- two new triggering policies.

The triggering policy determines under what conditions re-routing is executed. Several policies with homogenous traffic have been studied in [40] and it shows that just a simple arriving triggering policy is sufficient. As we now have non-homogenous traffic, long requests and short requests, two more new arriving triggering policies are arising: *long request arriving triggering policy* and *short request arriving triggering policy.*

The long request arriving triggering policy means that only when a long request arrives and finds its direct route not available, a re-routing process is initiated; while for short requests, the LLR (no TR) routing scheme is followed. The short request arriving triggering policy is exactly the contrary: only when a short request arrives and finds its direct route not available, a re-routing process is initiated; while for long requests, the LLR (no TR) routing scheme is used.

3.6.3 The Re-routing Scheme

Although many versions of re-routing schemes are possible [46], in this study we only focus a simple scheme described in [47] as follows:

1) A new arriving call will first be routed to its direct route, link l, if there is available bandwidth on that link.
2) If link l is not available, one alternate call on link l, if any, is picked randomly and rerouted back to its direct route (if possible) to make room for the new call.
3) If the rerouting of this alternate call is unsuccessful, another alternate call on link l is picked at random, and so on.
4) If none of the alternate calls on link l can be rerouted, the new call is routed to an alternate route with the maximum number of free bandwidth, i.e., the least loaded alternate route, as in LLR+TR routing.
5) If none of the alternate calls on link l can be re-routed, and no two-hop routes are available, the arriving call is blocked.

3.7 Restricted Access Routing (RAR)

Imagine such a scenario: with preventive-vacating routing, if the idle capacity of the direct route is less than the vacating threshold, a coming short request should start to vacate. But if there is *no* TR-permissible alternate route at this time, where should the short request be routed?

According to the preventive-vacating routing, the short request should be routed to the direct route if the direct route is still QoS-permissible. Otherwise, block the short request. However, if the short request was routed on the direct route and then during its lifetime, a long request arrived and found no resource available, we would lose the long request due to accepting the short request previously. When this short request is over, the direct link may be idle for some time as the arrival rate of long requests is, in general, much less than that of short requests. Furthermore, according to our system performance metric, the network throughput, the contribution of a long request is much larger than that of a short request.

Following the above line of thinking, the priority of resource access should be given to long requests under specific traffic conditions (for instance, if the arrival rate of long requests is relatively frequent). This raises the following issue: can a scheme be developed which will block the short request irrespective of whether the direct route is QoS-permissible or not, as in the scenario described above?

We call such a routing scheme as Restricted Access Routing (RAR). RAR can be viewed as a modified version of preventive-vacating routing. In RAR, a certain amount of capacity on the direct route is reserved *only* for the long requests and the short requests have no right to access this portion of bandwidth. This certain amount of bandwidth is named as the Restricted Access (RA) threshold. Generally, we believe that the value of the RA threshold should be equal to or less than the vacating threshold.

3.7.1 Restricted Access Routing scheme

When a new long request arrives,

i. Route this long request to the direct route if the direct route is QoS-permissible. Otherwise, go to step ii.

ii. If no TR-permissible alternate (two-hop) routes are available, then the arriving long request is rejected. Otherwise, the long request is routed to a TR-permissible alternate route with the largest idle capacity.

When a new short request arrives,

i. Route this short request to the direct route if idle capacity of the direct route is greater than RA threshold. Otherwise, go to step ii.

ii. If there is at least one TR permissible alternate route, route this short request to a TR-permissible alternate route with the largest idle capacity. Otherwise, block the short request.

3.7.2 Preemptively Restricted Access Routing

Obviously, in RAR, as its name indicates, short requests have restricted access to the direct route. Or, in other words, long requests are given a higher priority to access network resources. This results in the blocking rate of short requests being larger than that of long requests. Hence, the *fairness* of routing to these two types of traffic is lost. Note that this is different from all the previous routing schemes we proposed: preventive-vacating, preemptive-vacating, approximated least cost routing, re-routing and even LLR+TR.

Furthermore, the RAR scheme may be inefficient for short requests as short requests may be rejected frequently, while certain amount of capacity is idle in the portion reserved for long requests. One way to improve this situation is to allow the short requests access to the reserved capacity but at the risk of being preempted should a long request arrive and find that no other idle capacity can be used. When this occurs, an in-progress short request is randomly selected from among the short requests that have accessed the reserved capacity. The preempted call is lost and contributes one to the overall blocking number. The vacated bandwidth is occupied by the arriving long request. This is the preemptive version of restricted access routing.

3.8 Differentiated Shortest Distance Routing

The shortest distance routing (SDR) scheme is defined [15,18] as follow:

When a new request arrives,

i. Block the request if there is no QoS - permissible route available;

ii. Otherwise, route this request to the QoS - permissible route with the shortest distance.

The distance function of a route R is defined by

$$Dist(R) = \sum_{j=1}^{M} \frac{1}{d_j},$$

where d_j is the residual (idle) bandwidth of link j, and the route R includes link $1, 2, \ldots, M$. (Note that in this study, $M \leq 2$.)

In [17], a *differentiated* version of SDR is proposed to address the routing issue with heterogeneous holding times. The key idea is that the requests with different holding times use different ways to evaluate the link metric $\frac{1}{d_j}$. A *dynamic* link metric is the one with the most up-to-date d_j; while an *adaptive* link metric d_j has been averaged/filtered over time. Thus, actually, we could have three versions of SDR according to the link metrics they used:

Different Versions of SDR		Link Metric (d_j) Used	
1.	Dynamic Shortest Distance Routing (**Dynamic-SDR**) (Note this is the *original* SDR)	Long Request:	Dynamic
		Short Request:	Dynamic
2.	Adaptive Shortest Distance Routing (**Adaptive-SDR**)	Long Request:	Adaptive
		Short Request:	Adaptive
3.	Differentiated Shortest Distance Routing (**Diff-SDR**)	Long Request:	*Adaptive*
		Short Request:	*Dynamic*

Table 3- 2: three versions of SDR

It is shown in [17] that by using differentiated SDR with respect to traffic holding times, one can enhance network performance in terms of throughput under heavy network loads. We should point out that [17] is the only paper we found that proposed dynamic routing schemes to investigate the routing issue with non-homogenous holding time traffic until now. In this thesis, we will:

- Examine why better performance can be obtained by using Diff-SDR? (What's the secret behind the magic?)
- Give a clearer comparison of the three versions of SDR. Does Diff-SDR always work better than the other two?
- Discover if Diff-SDR has an effectiveness range, like our vacating schemes. If it does, then what is the range?
- Compare the performance of Diff-SDR with our vacating schemes.

3.9 Multi-TR Thresholds

Multi-TR thresholds mean applying different trunk reservation thresholds to different kinds of traffic. Multi-TR is used in multi-rate networks to equalize the blocking rates of various kinds of traffic [26,31,32,40,65]. Here, on the contrary, we can use multi-TR to *differentiate* the blocking rates of traffic with various holding times. We want to study how this idea works and its impact on network performance.

3.10 One Implementation Issue

Although we do not focus on the practical implementation in this thesis, some issues are still worthy of being considered. Among them, the most interesting one is how to evaluate the detailed traffic mix in practice, so that we can apply the effective vacating range to implement, for instance, the preventive-vacating scheme. Our idea is as follows:

- Each link maintains a measurement of the average holding time $\bar{h}$ of requests using it. The link can do this locally, thus facilitating implementation.
- When a new request (long or short) wants to use the link, its holding time is compared to the average, and vacating is encouraged if the holding time is well below the average. Based on that criterion, and the effective range of vacating, a decision is made for the request to vacate the link. The idea here is to use a compact metric reflecting the traffic mix. For example, if there is a lot of long request traffic (traffic mix is high), the holding time of a short request will be well below the average, hence will favor vacating.

We know the mean holding time for long requests is h_L, and for short requests, it is h_S. In addition,

$$h_L = x \times h_S,$$

where x is the holding time ratio. Actually, the relationship among $\bar{h}$, h_S, and h_L is

$$\bar{h} = h_L \times y + h_S \times (1-y) \tag{3.31}$$

where y is the traffic mix. Since we already have the average holding time $\bar{h}$ on the link, then y can be computed through

$$y = \frac{\bar{h} - h_S}{(x-1) \times h_S}. \tag{3.32}$$

Therefore, traffic mix can be obtained easily through practical measurement.

Chapter IV: Inter-Network Routing: Gateway Selection

The second routing issue we want to study in this thesis is routing in an inter-network environment, in which the traffic is homogenous. We focus on *how to select the best among several gateway nodes to a foreign network*. This applies, of course, to inter-network traffic that can exit through multiple gateways.

Actually, this routing issue belongs to routing of *multidestination traffic*. Multidestination traffic refers to a traffic type that can be routed to any one of a multiple number of destinations [66]. In our case, several gateway nodes, which connect to the same foreign network, can be taken as one multidestination set. Thus for an outgoing inter-network call originating from the local network, it is equally possible that this call will be successfully routed to any individual gateway node in the multidestination set of the foreign network.

4.1 Two Legs of an Inter-Network Route

Intra-network routing seeks to control the selection of routes between origins and destinations within the scope of the local network. In contrast, inter-network routing seeks to control the selection of routes between the origins in the local network and access points to foreign networks. An inter-network route can be generally partitioned into a first intra-network leg, linking the origin to the gateway node serving as exit point in the local network, and a second leg linking the gateway node to the foreign network access point. Thus, inter-network routing depends on intra-network routing for the selection of the first leg.

On the other hand, through the resource utilization cost concept, inter-network gateway selection couples naturally with intra-network route selection. Determining the best gateway for a given origin node can then be formulated as a two-link routing problem, where the first link (leg) is the best intra-network route to reach each gateway and the second link (leg) is the external link to the foreign network. Finding the best gateway therefore amounts to finding the two-link path with minimal *overall cost*. This is illustrated in Figure 4-1: the origin node S selects the gateway node with minimal overall resource utilization cost, say, 0.4+0.2=0.6.

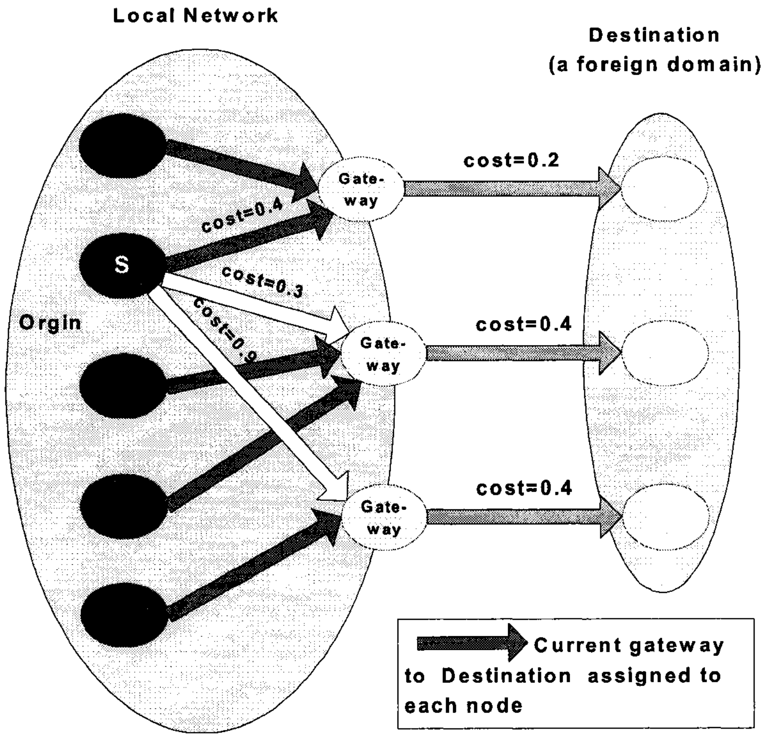

Figure 4-1: Gateway selection using resource utilization costs.

4.2 Gateway Selection Approaches

Keeping the above line of thinking, several gateway selection approaches are proposed:

i. **Theoretical Resource Utilization Cost Approach**

Select the gateway with minimal overall cost (summation of the costs on the intra-network link(s) and on the inter-network link). The cost on each link, intra-network or inter-network, is computed according to equation (2.1), say, $\bar{C}_p = \dfrac{E_b(A,N)}{E_b(A,p)}$ (2.1).

ii. **Trunk Reservation Approximation Approach**

Select the gateway with minimal overall cost. The cost on each link, intra-network or inter-network, is computed through the Trunk Reservation approximation, that is to say, approximating $\bar{C}_p$ to *0* when the number of idle trunks is above the threshold, and to one at or below the threshold. (Please see section 2.1.3 for the details of Trunk Reservation).

iii. Intra-Network Link First Approach

Select the gateway node with the minimal intra-network cost and available inter-network link.

Note: Remember the two legs of an inter-network route. This approach only emphasizes the cost on the intra-network leg. While for the inter-network leg, it just makes sure that it is not fully busy. This approach allows each outgoing inter-network call to be routed through the least loaded intra-network route. Trunk Reservation is used to compute the cost of the intra-network leg.

iv. Inter-Network Link First Approach

Select the gateway node with the minimal inter-network link occupancy cost and available intra-network link.

Note: This approach is the contrary of the Intra-network Link First approach. It emphasizes only the detailed link occupancy (resource utilization) computed as $\frac{occupied\ bandwidth\ of\ the\ link}{capacity\ of\ the\ link}$ of the inter-network leg. While for the intra-network leg, it only makes sure that it is not fully busy. This approach allows each outgoing inter-network call to be routed through the least loaded inter-network link.

v. Fixed Order Selection Approach

Select the gateway node according to a fixed and predetermined selection sequence. For instance, suppose the selection sequence is *0,1,2,3....* Then for every inter-network call, gateway node *0* is always attempted first. If the inter-network routing through gateway node *0* is not available, then according the sequence, gateway node *1* is attempted, and so on.

Important Notes: Among all the above gateway selection approaches except the first one, the routing scheme for intra-network routing within the local network scope is still **LLR+TR**. This applies to both of the following cases:

- Pure intra-network routing, where both the origination and destination nodes are within the scope of the local network.
- Intra-network route selection for an inter-network routing. The origination node is within the scope of the local network, while the destination is a foreign network. Then, given a gateway node, select the LLR+TR route from the origination node to the gateway node.

4.3 Blocking Rule of Inter-Network Routing

In inter-network routing, the route could be either a two-hop (one intra-network link, plus one inter-network link) or a three-hop route (two connected intra-network links plus one inter-network link). There is no direct route (one-hop) for inter-network routing. Hence, the request should only be blocked when all routes in the route-set are inaccessible. An inaccessible route means that there is at least one link fully busy among the links of the route.

Note that the TR threshold in inter-network routing is only used to *evaluate* or *quantify* the cost.

4.4 Performance Metric

As the traffic is homogenous, the performance is the overall blocking rate:

$$blocking\ rate = \frac{number\ of\ flows\ blocked}{number\ of\ flows\ offered},$$

where the flows include both intra-network and inter-network flows.

Chapter V: Simulation

The network simulator used in this study is NS2 (network Simulator Version 2), which is an event driven network simulator developed at UC Berkeley that simulates a variety of IP networks [56]. It provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. The platform (operating system) for running NS2 is FreeBSD 4.3, and the version of NS2 is 2.1b7a. Based on NS2, we implement our simulation.

5.1 The Simulator: NS2

NS2 is also an object-oriented simulator, written in C++, with an OTcl interpreter as a front-end. (Figure 5-1)

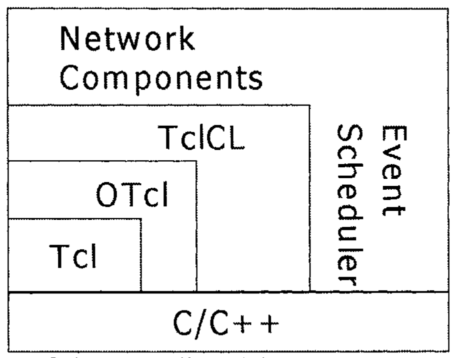

Figure 5-1: Extending Tcl Interpreter by NS2

5.1.1 C++ & OTcl in NS2

NS2 supports a class hierarchy in C++ (also called the compiled hierarchy), and a similar class hierarchy within the OTcl interpreter (also called the interpreted hierarchy). The two hierarchies are closely related to each other. From the user's perspective, there is a one-to-one correspondence between a class in the interpreted hierarchy and one in the compiled hierarchy (Figure 5-2). The root of this hierarchy is the class *TclObject.* Users create new simulator objects through the interpreter. These objects are instantiated within the interpreter, and are closely mirrored by a corresponding object in the compiled hierarchy. The interpreted class hierarchy is automatically established through methods defined in the class *TclClass*. User instantiated objects are mirrored through methods defined in the class *TclObject.*

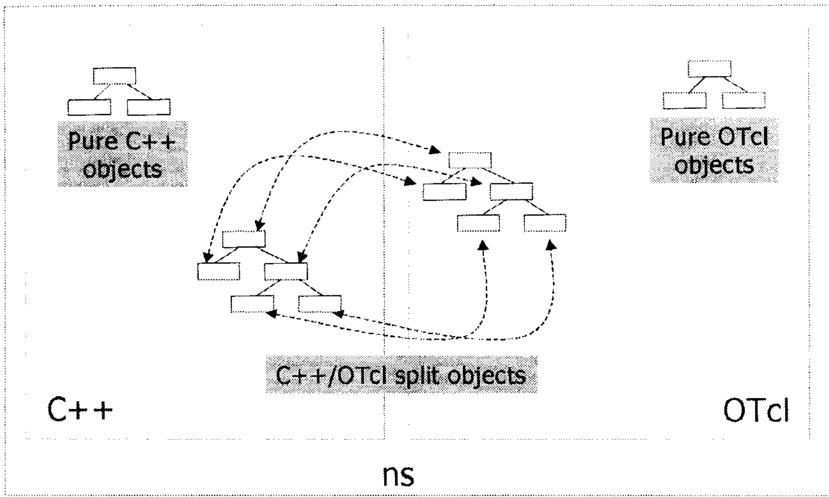

Figure 5-2: OTcl and C++: The Duality

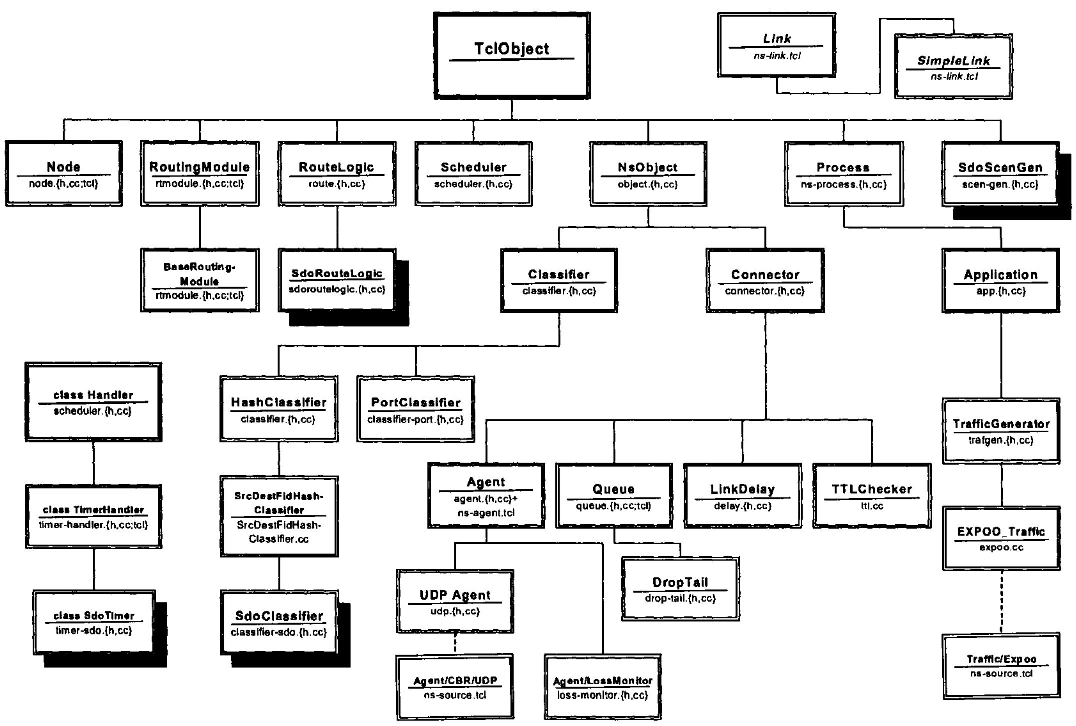

Figure 5-3: Main part of the C++ hierarchy in NS2 (the shadowed blocks are our code)

5.1.2 Running a Simulation

Generally speaking, a network simulation can be described by the following diagram (Figure 5-4). We set up a network topology, which is comprised of nodes and links. Then we attach traffic into the topology. The traffic is composed of agents and traffic generators. At last, we use user-specified routing control to control the routing.

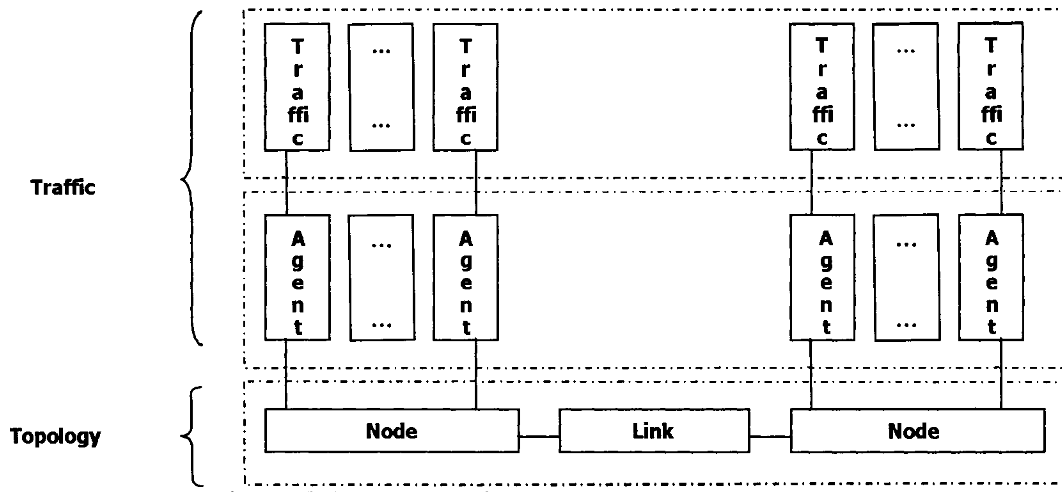

Figure 5-4: Concept of a network simulation in NS2

For a single simulation trial, an OTcl script file, where the action-sequence and related functions of the simulation are defined, is needed for NS2 to run the simulation directly, i.e.:

```
#ns  <OTcl_script filename>
```

The action-sequence of a simulation is:

1. **Initialization**
2. **Get traffic demand matrix**
3. **Define the topology by creating nodes and links**
4. **Attach traffic to the topology**
5. **Trigger user-implemented routing control mechanism**
6. **Start to run the simulation**

5.1.3 Our Coding Work in NS2

To implement the proposed routing schemes, our programming work includes coding both in OTcl and C++:

- OTcl script files: to implement different simulation scenarios, collect statistical information, and draw associated Figures.
- C++ files: The original NS2 is actually a basic IP routing platform that does not provide the state-dependent optimal (SDO) routing we need. Thus, we implement the routing functions by ourselves. For the purpose of performance, packet processing and routing computations are coded in C++. Our programming work in C++ includes:

File Name	Contents
Scen-gen.{h,cc}	Implementation of Kruithov's method (Appendix B) to generate traffic demands and link capacities
Classifier-sdo.{h,cc}	Implementation of SdoClassifier that use source/destination/flow_id as hash key to distinguish
Sdoroutelogic.{h,cc}	Tracking link usage information, implementing the detailed routing algorithms, computing optimal routes, and setting label paths
Timer-sdo.{h,cc}	Implementation of flow timer.

Table 5-1: Our programming work in C++. (the C++ Class we implemented are shadowed in Figure 5-3: C++ hierarchy in NS2)

5.2 Traffic Distribution

We need to decide the exact details of the amount of traffic requests between nodes. Kruithov's method (see Appendix B) is adopted to generate the traffic distribution, and to obtain reasonable individual link capacity. The steps are:

1. Set the total originating traffic from each node
2. Set the non-coincidence of peak hours
3. Allocate the traffic demand to individual destinations
    a. *Allocate the originating intra-network traffic demand*
    b. *Allocate the originating inter-network traffic demand*
    c. *Allocate the terminating traffic demand to individual destinations*
4. Size the network to find out the exact amount of traffic for each node-pair, and the capacity of each link.

For details of the steps 1 to 4, please refer to Appendix B.

5.3 Topology

To implement the topology (according our system model) in NS2, we need to create nodes (interior nodes and gateway nodes) and links (see Figure 3-1). All the links are unidirectional.

5.3.1 Node

A node in NS2 acts as a container of classifiers and agents. It provides interfaces between its classifiers and links.

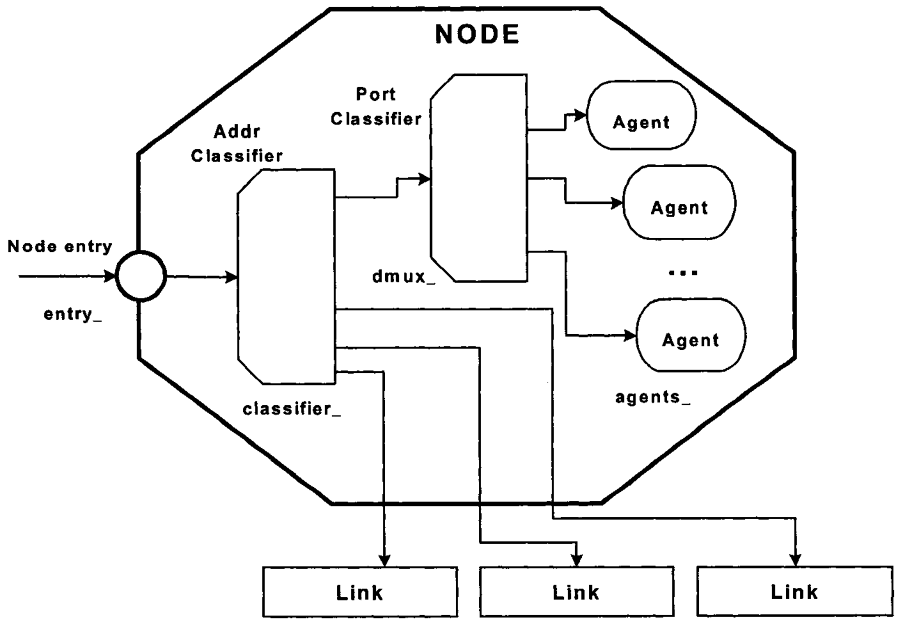

Figure 5-5: Structure of Unicast Node in NS2

At the beginning of a simulation, nodes are responsible for selecting and installing the right classifiers, and hooking the classifiers in correct sequence. Note that we replace the "*Addr Classifier*" (Figure 5-5) with our *Sdo Classifier* in each node. Thus, we name this kind of nodes as Sdo Nodes (Sdo means state-dependent optimal).

5.3.2 Link

The link is the second part to define the topology. It is used to connect two nodes. Similar to a node, a link is also like a container, which contains several objects, illustrated in the following Figure 5-6.

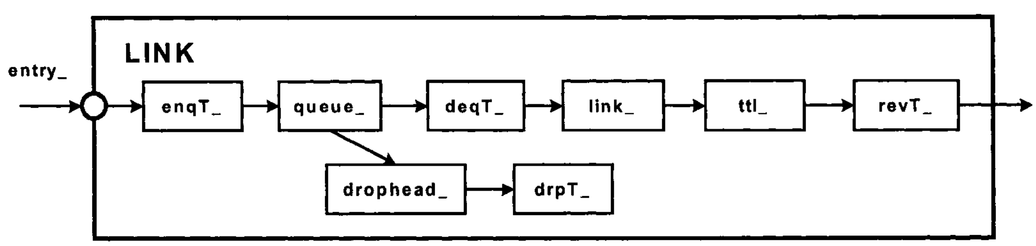

Figure 5-6: Composite Construction of an Unidirectional Link (see [56], for details)

5.4 Attaching Traffic to the Topology

How to add traffic into the topology? There are two steps:

1) set up enough logical connections (source-destination pairs) among nodes according to detailed traffic distributions.

2) attach appropriate traffic generator to each logical connection.

5.4.1 Connections Set up by Agent-Sink Pairs

Agents in NS2 represent endpoints where network-layer packets are constructed or consumed. They are the objects that actively drive the simulation, and can be thought of as the processes and/or transport entities on nodes. To run the traffic, it is necessary to set up an end-to-end logical connection (agent-to-sink). In general, each of such end-to-end connections comprises one agent in the origin node, and one sink agent in the destination node. It can be called an agent-sink pair. The packets are transmitted from the agent in the origin node to the sink in the destination node, through the link(s) connecting the origin and destination nodes.

5.4.2 Attaching Traffic Generators to Connections

According to our system model, we define two kinds of Poisson-type traffic generators. They are the long-request-generator and the short-request-generator, corresponding to the traffic generators for long and short holding times respectively.

Then, we attach different traffic generators to logical connections in a one to one manner, i.e., one traffic generator is attached to one agent-sink pair. This can induce various types of traffic in the network, for example, homogenous traffic, or non-homogenous traffic.

The mean holding time of short requests is set as 5 seconds. The traffic intensity of each traffic generator, long or short, is set as 0.7 Erlang. Thus, according to the detailed value of the traffic demands, we can determine the number traffic generators needed for each origination-destination pair.

5.5 Routing Control

According to a detailed routing algorithm, the routing control determines the routing options for each incoming request and then sets up the label path. If there are not enough resources, the routing control blocks the request. To understand this in NS2, let us review some pertinent concepts first.

5.5.1 Some Pertinent Concepts in NS2

Classifiers

A classifier is the object that is responsible for the packet classification required for forwarding packets to the next node.

The function of a node, when it receives a packet, is to examine the packet's fields, usually its destination address, and on occasion, its source address. It should then map the values to an outgoing interface object that is the next downstream recipient of this packet.

In NS2, the classifier object performs this task. The classifier provides a way to match a packet against some logical criteria, and retrieves a reference to another simulation object based on the match results. Each classifier contains a table (*slot table*) of simulation objects indexed by a slot number. The job of a classifier is to determine the slot number associated with a received packet, and forward that packet to the object referenced by that particular slot number. The object could be a port classifier, a head of a link, or an agent.

Hash classifier

A hash classifier is used to classify a packet as a member of a particular flow. As the name indicates, hash classifiers use a *hash table* internally to assign packets to flows. Several "flow granularities" are available. In particular, packets may be assigned to flows based on flow ID, destination address, source/destination addresses, or the combination of source/destination addresses plus flow ID.

In our implementation, a new kind of classifier, a *SdoClassifier*, is derived from the hash classifier. The *SdoClassifier* uses the combination of source/destination_addresses/flow_ID (*src/dst/fid*) to distinguish flows.

Port classifier

A port classifier is used to classify the received packets from a *SdoClassifier* to different local agents, according to the port number of the received packet. A *port table* is maintained in the port classifier to match the port number and associated agent. See Figure 5-7 for details about the above classifiers in a node.

Centralized Virtual Routing Decision Node

The Centralized Virtual Routing Decision Node is an object independent of the network topology, created from the class *SdoRouteLogic*. It performs the following tasks:

- Keep track of the route-set information for all source and destination pairs (in the *RouteSet* Table);

- Keep the capacity information and real-time usage information for all links (in the *Resource* Table);
- Keep the status of each on-going call, including effective bandwidth;
- Find the optimum route for each call according to the detailed routing algorithms
- Set up the label paths (see Figure 5-7,8).

Scheduler

The NS2 is an event-driven simulator. The scheduler runs by selecting the next earliest event, executing it to completion, and returning to execute the next event. The unit of time used by the scheduler is seconds. If more than one event is scheduled to be executed at the same time, the execution is performed in a first scheduled - first dispatched manner.

Flow Timer

We attach to each on-going call/flow a flow timer (set to 0.5 second) to determine the end of the call/flow. If the idle time between two sequentially-incoming packets (for example, packet *p1* and *p2*) is less than 0.5 second, then the two packets (*p1* and *p2*) are assumed to belong to the same flow (for example, *f1*). If not, then the two packets are assumed to belong to two different flows (*p1* is in *f1*, and *p2* is in *f2*). In addition, in this latter case, the flow *f1* is terminated and a new flow, *f2*, is created.

5.5.2 Actions Taken for a New Arriving Request

When a new call/request comes in, the following procedures are carried out:

1. The *SdoClassifier* of the source node informs the Centralized Virtual Routing Decision Node.
2. The Centralized Virtual Routing Decision Node then checks the route-set table and the resource table for link capacity and real-time usage.
3. Based on these results, according to a detailed routing algorithm, the Centralized Virtual Routing Decision Node computes the route for the incoming request.
4. A label path is set up. The Centralized Virtual Routing Decision Node installs a hash entry for this request in all the nodes along the selected route.

Then, it also adds the trunk usage entries in the resource table for the link(s) along the selected route (see Figure 5-8).

5. A timer (*SdoTimer*) is created and started for this request to keep track of its status.
6. After the above operations, all the packets belonging to this request are forwarded along the pre-established label path.

5.5.3 Actions Taken When the Request is Over

When the request is over and has been triggered by the flow timer, the following procedures are carried out:

1. The Centralized Virtual Routing Decision Node deletes and releases the label path in the nodes along the route. Also, it modifies the resource table accordingly.
2. The flow timer attached to this request is released.

5.6 Consideration about our Simulation Framework

In practice, the routing decision for a particular instant-connection (and other similar decisions such as for admission) are extremely unlikely to have access to current state information, at least on the remote links. Instead, they will have access to *near real-time* information, gathered some time in the past (e.g., 10 seconds in DCR). This poses no problem if the update cycle time is fast compared to the dynamics of the traffic that we are trying to control. It means that we may have to consider certain traffic only on the basis of their flow. That is, the routing decisions would attempt to only balance the flow of these instant-connections on different paths, and not on the basis of individual detailed instant-connections. This supposes that stable state-dependent routing schemes can be devised, with acceptable *overhead* of messaging and processing. In this thesis, with MPLS and a centralized bandwidth broker (Centralized Virtual Routing Decision Node), we assume the above to be possible.

Furthermore, our routing schemes proposed in Chapter III could be applied to long durations flows, e.g., lasting from several minutes to several hours (IP calls to IP conferences), for which an update time (for system state information) on the order of several seconds would be amply sufficient. This makes our schemes more practical and also focusing on routing control when it is most important, i.e., when the decision will last for several minutes to hours.

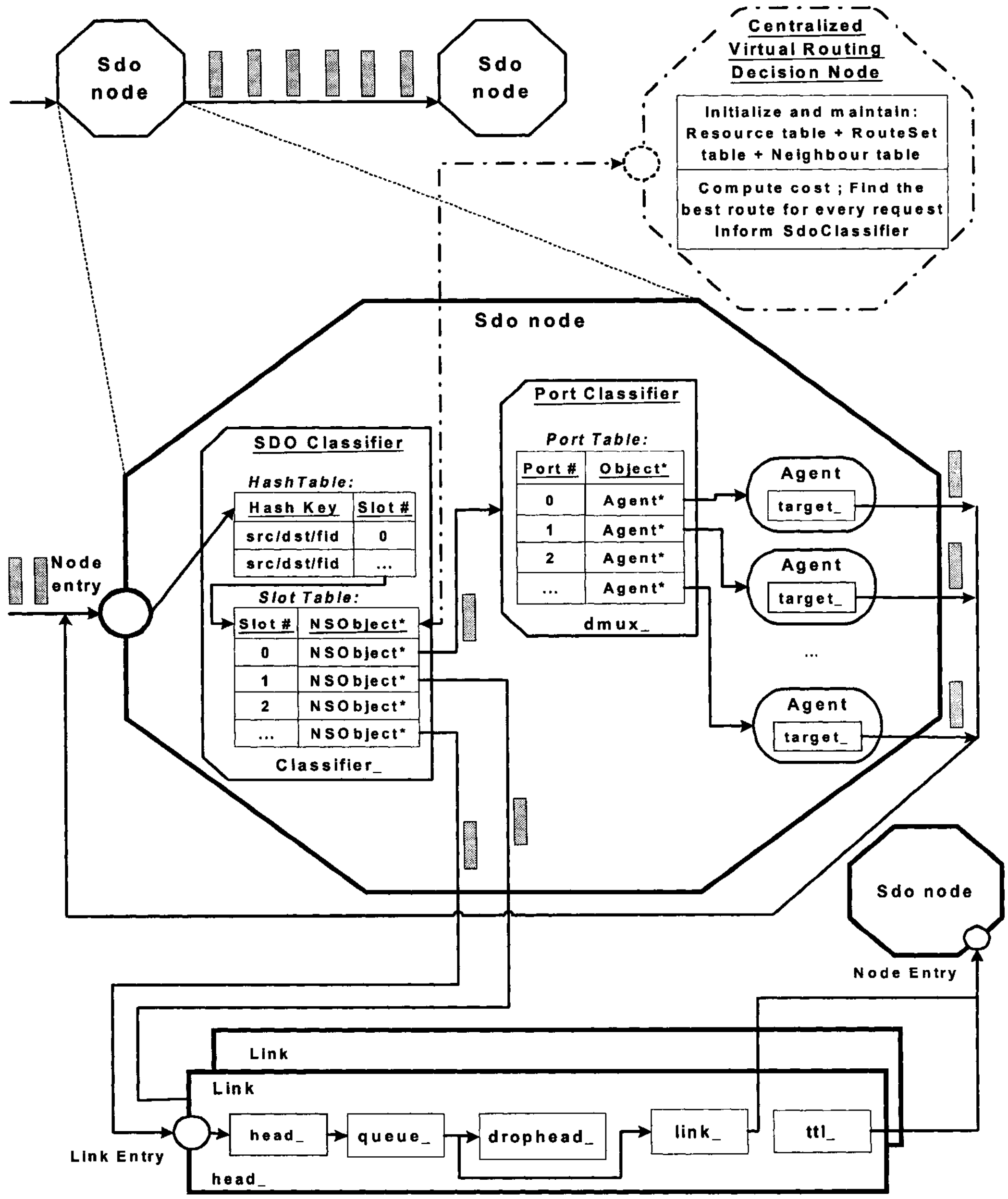

Figure 5-7: Structure of a Node. (A Node installs one *SdoClassifier* and one *PortClassifier*. In the SdoClassifier, an incoming request, according to its src/dst/fid, can be matched to a slot number in the *hash Table*. Indexed by this slot number in the *Slot Table*, the request can be sent either to the *PortClassifier* and then to an agent, or to a link for transmission.)

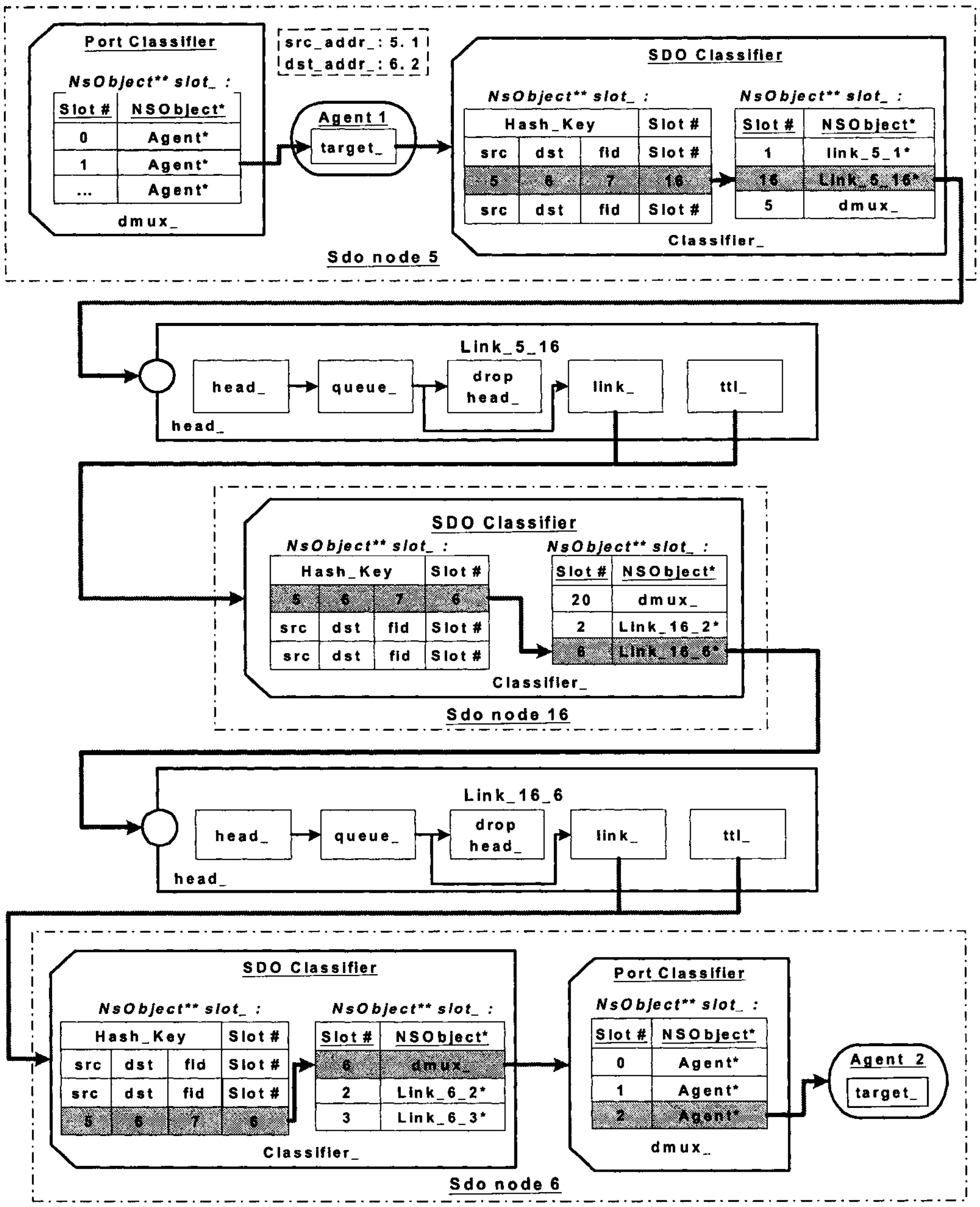

Figure 5-7: Label Path: Node 5 -> Link_5_16 -> Node 16 -> Link_16_6 -> Node 6. (The flow id of the flow taking this 2-hop path is 7. The hash key (label) of this flow is 5/6/7, which has been installed in the hash tables of Node 5, 16, and 6 respectively. It matches the different slot numbers in different nodes through hash tables. These slot numbers also match to various objectives, either the head of a link or the PortClassifier in nodes.)

Chapter VI: Simulation Results of Intra-network Routing with non-Homogenous Holding Times

In this chapter, we present and discuss the simulation results of all the routing schemes proposed in Chapter III to address our major routing issue in this study, namely intra-network routing with non-homogenous holding time traffic.

We start with the network set-up for the simulation. Then we proceed to study our proposed routing schemes sequentially. For each of these proposed schemes, its performance is evaluated through the comparison with LLR+TR. Moreover, the performance-impacting considerations, such as traffic mix, holding ratio and TR thresholds, are also investigated. Finally, we compare all the proposed routing schemes with each other in a thorough way.

6.1 Network Set-up

Call-level simulations in NS2 were performed to test the performance of our proposed routing schemes. The performance study is based on two network examples: a fully connected 4-node network and a practical 12-node network. Through random number generators, the traffic mix is set to be similar across the origination-destination pairs in the network.

6.1.1 4-Node Network

The 4-node network is described as:

Item	Value
Number of nodes	4
Number of single-way links	12
Basic Bandwidth Unit (BBU)	20kB
Capacity of each link (N)	1020kB=51 BBU
Connection ratio	100% (fully connected)
Fully Symmetrical	Yes
Nominal Total Traffic (Erlang)	529
Overload Conditions	+10%

Table 6-1: Description of the 4-node Network

According to the above table, the 4-node network is fully connected and symmetrical. In addition, the Basic Bandwidth Unit (BBU) is the mean equivalent bandwidth of each request. Under *nominal* traffic load, the blocking rate for long requests or short requests is around 0.5~1%.

6.1.2 12-Node Network

The 12-node network is described as:

Item	Value
Number of nodes	12
Number of single-way links	76 (55~101 BBU)
Basic Bandwidth Unit (BBU)	20kB
Overall Network Capacity	100MB=5k BBU
Connection ratio	60%
Fully Symmetrical	No
Nominal Total Traffic (Erlang)	4k
Overload Conditions	+15%

Table 6-2: Description of the 12-node Network Example

The 12-node network is not fully connected and is non-symmetrical, but well-engineered. In general, this network carries approximately 90% of its traffic directly. The 10% left attempts to route over two-hop routes either because it overflows from the direct route (around 5%) or because it has no direct route.

All the simulation trials in the 12-node network were run with the traffic in three time patterns: 10h00, 12h00 and 13h00. The associated simulation curves are generated by averaging the results obtained from the three time patterns of traffic.

6.1.3 Outperformance Metric

In Section 2.5.4, we defined the performance metric for the intra-network routing schemes as the Network Throughput. Here, in order to compare various proposed routing schemes with LLR+TR in a clearer way, we define the *outperformance metric* as:

Outperformance of a proposed routing scheme compared to LLR+TR (%)

$$= (Network\ Throughput\ of\ this\ scheme) - (Network\ Throughput\ of\ LLR + TR)$$

6.1.4 The Simulation Time

The simulation time is set long enough to achieve a sufficiently small 95% confidence interval (**±0.1%**) in *all* the simulation trials. Suppose that the mean holding time of long request is one unit time. In the 4-node network, for instance, the related simulation trial is run for a total of 160 units of time. Each related simulation trial was warmed up in the initial 80 time units starting from an idle network. We only consider the data collected within the remaining 80 time units. The relation between initial and stable periods is illustrated in Figure 6-1.

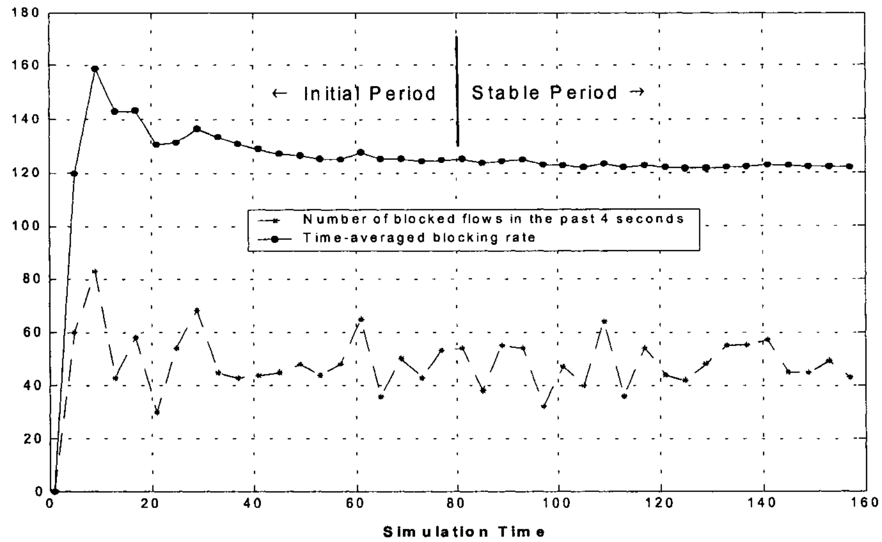

Figure 6-1: A piece of simulation cut from a trial to illustrate the relation of the initial period and the stable period. For example, in the 4-node network, suppose that the holding time for long request is 50 seconds, and the overall simulation time is $50\times160=8000$ seconds. Then the first $50\times80=4000$ seconds is the initial period. Our data is obtained only from the remaining $50\times80=4000$ seconds (stable period).

It is worth mentioning the difference between the simulation time and the simulator (NS2) running time. The simulation time is the time simulated by the simulator. The simulator running time is the actual running time of the simulator. For instance, to simulate 10000 seconds simulation time in the 4-node network under nominal traffic load, NS2 needs to run about 50 minutes in actual running time. While for a 5000 seconds simulation time in the 12-node network under normal traffic load, NS2 needs to run for approximately six hours.

Most of the simulation trials have been done both in the 4-node and in the 12-node networks, except the extremely time-consuming ones, such as when the holding time ratio is 100, which is simulated only in the 4-node network.

6.2 Preventive-vacating Routing (PVV)

Preventive-vacating routing (PVV) is the first routing scheme we proposed, and is derived directly from our basic vacating idea. In this section, we study the performance of PVV routing compared with LLR+TR. Moreover, several important performance-impacting considerations are also investigated.

6.2.1 Performance Comparison with LLR+TR

All the routing schemes we propose for non-homogenous holding time traffic are evaluated mainly through comparing with the LLR+TR scheme. Hence, we first test LLR+TR with various combinations of traffic mix, holding time ratio, and traffic load. We find that the performance of LLR+TR, in terms of network throughput, changes only very slightly under diverse traffic mix and holding time ratio combinations if the overall traffic volume in the network is fixed (see Appendix A). Furthermore, the blocking rates for long requests and short requests are also very similar, which is to be expected since LLR+TR does not distinguish whether an arriving request is long or short when making routing decision.

Then we compare the performance of preventive-vacating routing (PVV) with LLR+TR in Figure 6-2 and 6-3a,b for nominal and overload traffic conditions, respectively.

As can be seen in Figure 6-2 and 6-3a,b, the outperformance curves of PVV (in two networks) are *matched* very well with the curves of the cost difference obtained from our theoretical analysis (Figure 3-10), both in amplitude and in the trend of curves (especially under overload traffic condition). We can thus conclude that the outperformance of PVV in network throughput is proportional to the associated cost difference, which is rational. Indeed, this has already confirmed the correctness and exactness of our analytical framework in Chapter III in general.

In nominal traffic condition (Figure 6-2), the preventive-vacating only provides a marginal positive outperformance to LLR+TR. This is due to the fact that the Network

Throughput of LLR+TR in such traffic condition is already good enough (around 99%), and there is very little room for improvement (1%) left.

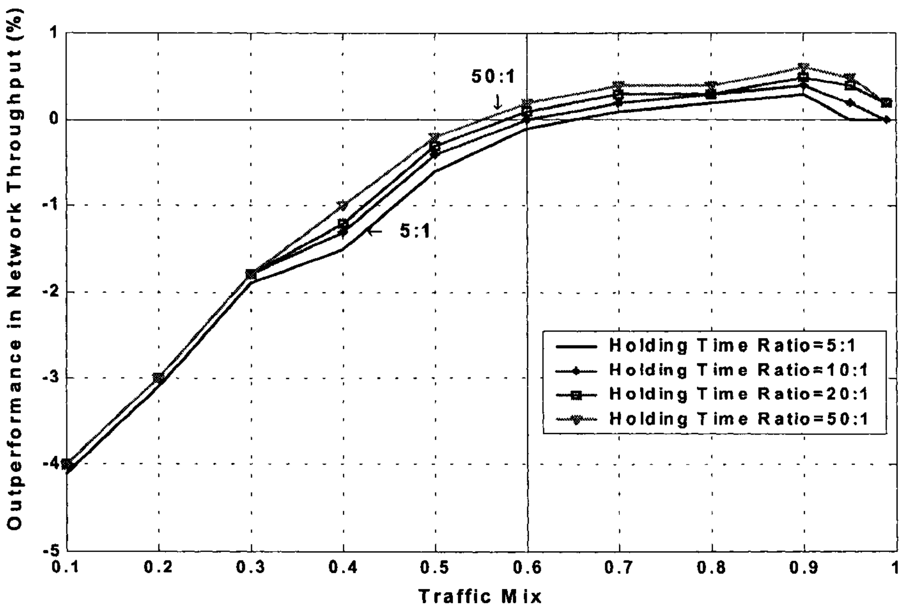

Figure 6-2: Outperformance of Preventive-Vacating scheme compared to LLR+TR in nominal traffic condition. TR threshold=vacating threshold=2%. 4-node network.

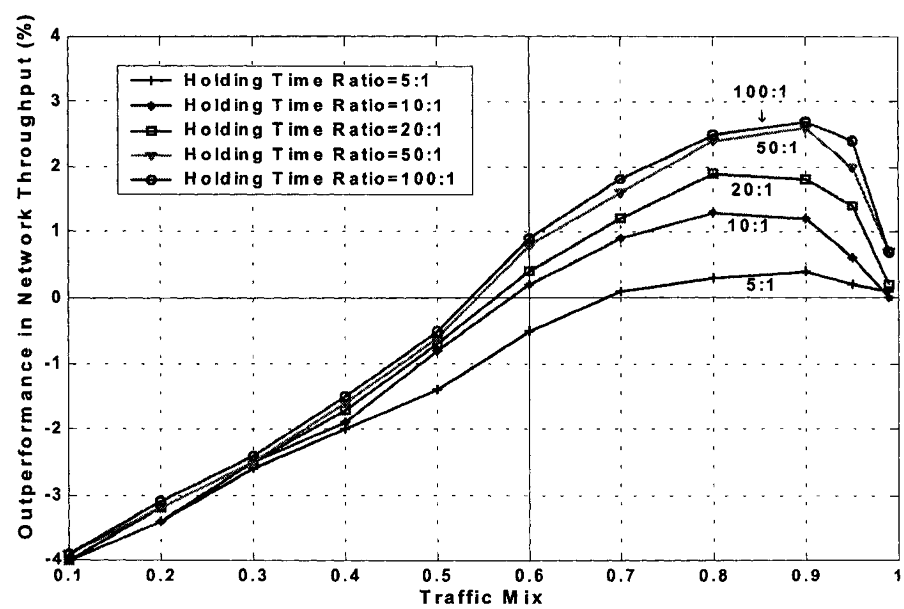

Figure 6-3a: Outperformance of Preventive-vacating scheme compared to LLR+TR in overload traffic condition. TR threshold=vacating threshold=2%. 4-node network.

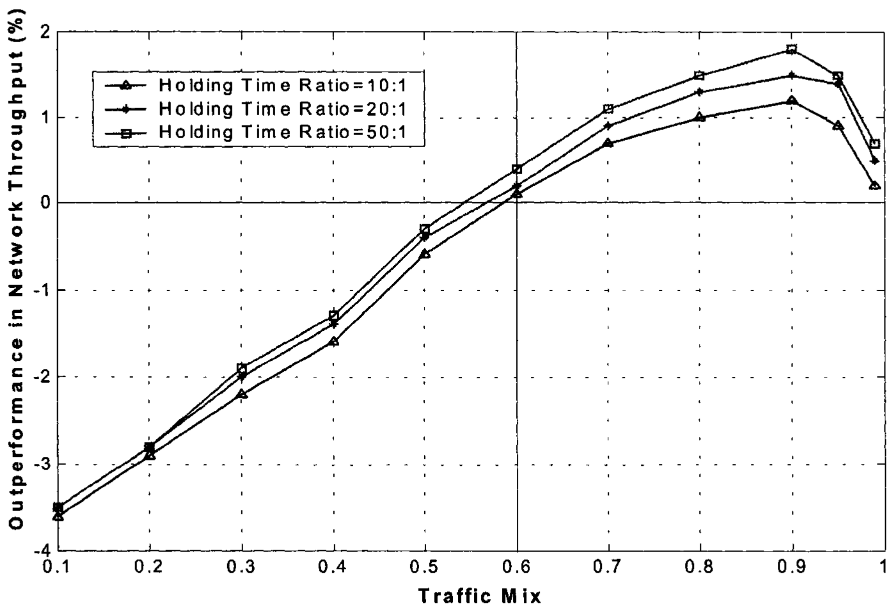

Figure 6-3b: Outperformance of Preventive-vacating scheme compared to LLR+TR in overload traffic condition. TR threshold=3%. Vacating threshold=2%. 12-node network.

In overload traffic condition (Figure 6-3a,b), the positive outperformance of preventive-vacating is significant, with the maximum value even close to 3% (holding time ratio is 100 in the 4-node network) and close to 2% (holding time ratio is 50 in the 12 node network). This can be explained through the equation (3.14) from our theoretical results in Chapter III:

$$\textit{the overall cost difference between preventive-vacating to LLR+TR}$$
$$= T\times v\times(C_{r2}-C_{r1})\times(1-y)\times\left[\left(\frac{y}{1-y+y/x}\right)\times\left(1-e^{-(1-y+y/x)\times T}\right)-1\right] \qquad (3.14)'$$

where T is the overall traffic volume in Erlang and v is the percentage of short requests taking vacating actions. When the overall traffic varies from nominal conditions to overload conditions ($T\uparrow$), we also observe that v is rising also (Figure 6-4 in next page).

When T is increasing, the cost rate on each link usually increases too. This results in a rise in C_{r2} (which is the sum of cost rates on two links), as well as a rise in the value of $C_{r2}-C_{r1}$ (C_{r1} is the cost rate of only one link). Therefore, according to equation (3.14), keeping x and y unchanged, the overall cost difference, which is the major contribution to the outperformance of vacating, is also increasing as $T\uparrow$. So does the positive outperformance of preventive-vacating compared to LLR+TR.

However, the above deduction cannot lead to such a conclusion: as T could be infinite, the outperformance can also be unlimited. The reason is that although T can be any value that is large enough, v will not always increase as T does. In fact, when T increases above a certain level, v will drop dramatically. Therefore, the product of T and v will be limited. Selecting a simulation trial, as in Figure 6-4, can easily validate this.

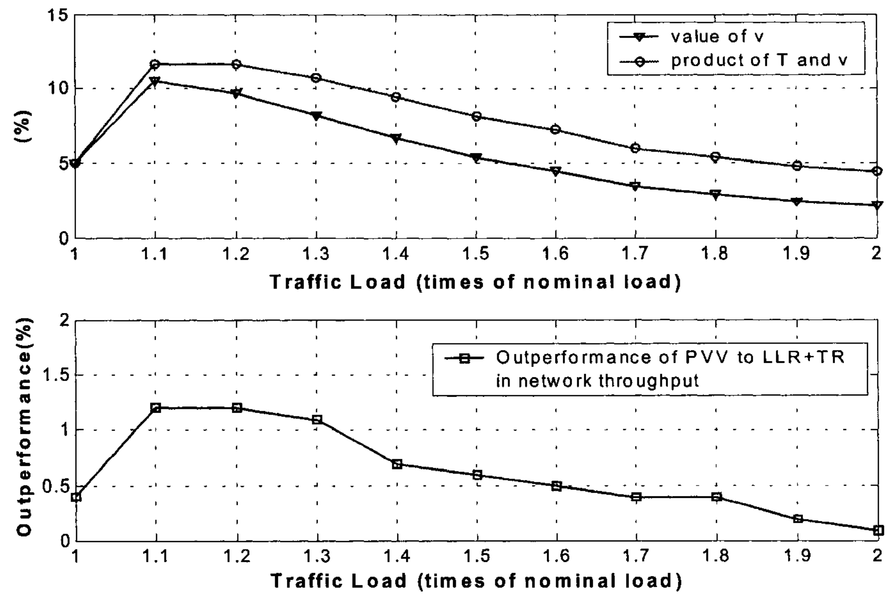

Figure 6-4: The relationship of v, $T\times v$ and outperformance of Preventive-Vacating compared to LLR+TR. (Traffic mix is 80%, holding time ratio is 10, TR=vacating threshold=0.02; 4-node network)

As can be seen in Figure 6-4, the outperformance of PVV compared to LLR+TR is roughly proportional to the value of $T\times v$. We notice that the trends of v , $T\times v$ and the outperformance are just like what we expected. When traffic is nominal, $T\times v$ is small, so does the outperformance. Then, the outperformance increases as traffic loads does. After 1.2 times of nominal traffic loads, the outperformance drops as $T\times v$ decreases. When traffic load is extremely heavy, say double the nominal load, both the outperformance and $T\times v$ are near zero. Note that at this time, direct routing becomes almost the only choice for any accepted requests ($v \approx 0$).

Furthermore, Figure 6-4 also indicates that the *effective range* of PVV in traffic load, which is wide (from 1.0 to 1.8 times the nominal traffic).

6.2.2 On the Effect of the Traffic Mix

The traffic mix plays a key role in deciding whether the outperformance of preventive-vacating to LLR+TR is positive or not. As indicated both in the theoretical analysis (Figure 3-7) and the simulation results (Figure 6-2 and 6-3a,b), there is an *effective range* of traffic mix for vacating. Within this range, the preventive-vacating is effective, that is to say, its outperformance compared to LLR+TR is positive; while outside this range, it is not. The range is around [0.60, 0.99] in traffic mix, as shown in Figure 3-7, Figure 6-2, and Figure 6-3a,b.

We also notice that within the effective range of preventive-vacating, although the outperformance is always positive, it is not a constant value. The maximal value of outperformance occurs between 80% ~ 90% in traffic mix. Leaving this maximal value range, the outperformance decreases in two directions: to 1.0 and to 0.0 in traffic mix.

Beyond the effective range, the outperformance of preventive-vacating is negative, and is dropping almost linearly as the traffic mix approaches zero. It means that in this range, LLR+TR has better performance, in terms of network throughput, than preventive-vacating.

When the traffic mix is 1.0, there will be no short requests, and thus no vacating actions will happen at all. Hence, the preventive-vacating is in fact equal to LLR+TR with zero outperformance. When the traffic mix is 0.0, the traffic is homogenous short request, thus vacating actions become fully non-beneficial and preventive-vacating reaches its worst performance.

6.2.3 On the Effect of the Holding Time Ratio

From the analytical and simulation results, we also can find out the effect of the holding time ratio on the outperformance of preventive-vacating compared to LLR+TR. Within the effective range of preventive-vacating, as the holding time ratio rises, so does the outperformance (Figure 3-7, 6-2, 6-3a,b). For instance, when the holding time ratio increases from 5 to 100 in the 4-node network, the outperformance also extends from 0.5% to almost 3%. The same thing happens in the 12-node network.

However, the rise in the outperformance is not unlimited. When the holding time ratio is above 50, the rise seems to be *saturated*: rising further the holding time ratio will only produce an almost fixed improvement on the outperformance compared to LLR+TR.

Noticing that the outperformance of vacating with a holding time ratio of five is very limited, we suggest to deploy preventive-vacating routing scheme only when the holding time ratio is not less than ten.

6.2.4 On the Effect of the TR

Generally, we study the effect of TR thresholds on a proposed routing scheme from two aspects:

- the effect on the performance of this routing scheme itself, and
- the effect on the outperformance of this routing scheme compared to LLR+TR.

Regarding PVV, these two aspects are shown in Figure 6-5 and 6-6 (next page), respectively. We observed that the varying TR threshold has a strong impact on both the performance and outperformance of preventive-vacating scheme in overload traffic condition.

As can be seen in Figure 6-5, the performance of PVV in terms of network throughput is improved with the increase of the TR threshold, especially beyond the effective range of vacating. While in Figure 6-6, it is observed that the absolute value of the outperformance compared to LLR+TR, whether positive or negative, is inversely proportional to the increase of the TR threshold. When the TR threshold becomes large enough, there will be little difference between the performance of preventive-vacating and LLR+TR. In other words, the curves of performance and outperformance of preventive-vacating are *flattened* by increasing the TR threshold.

This observation can be explained still through the equation (3.14). When the TR threshold changes, the unchanged parameters in equation (3.14) are: T, x and y, the changed parameters are: v and $(C_{r2} - C_{r1})$. Note that C_{r2} is the cost rate of a two-hop route when vacating is happening, and this two-hop route must be TR-permissible at that time. Hence, the value of C_{r2} under a high TR threshold is usually less than what it is under a low TR threshold. In addition, with a higher TR threshold, the opportunity or possibility for short requests to vacate, say v, decreases. (see Table 6-3 on next page). Therefore, with the increase of the TR threshold, both the values of $(C_{r2} - C_{r1})$ and v drop, and this will result in the decrease of the absolute value of equation (3.14).

Moreover, if the TR threshold is large enough, then the value of v could become near zero, and the overall outperformance is then near zero, too.

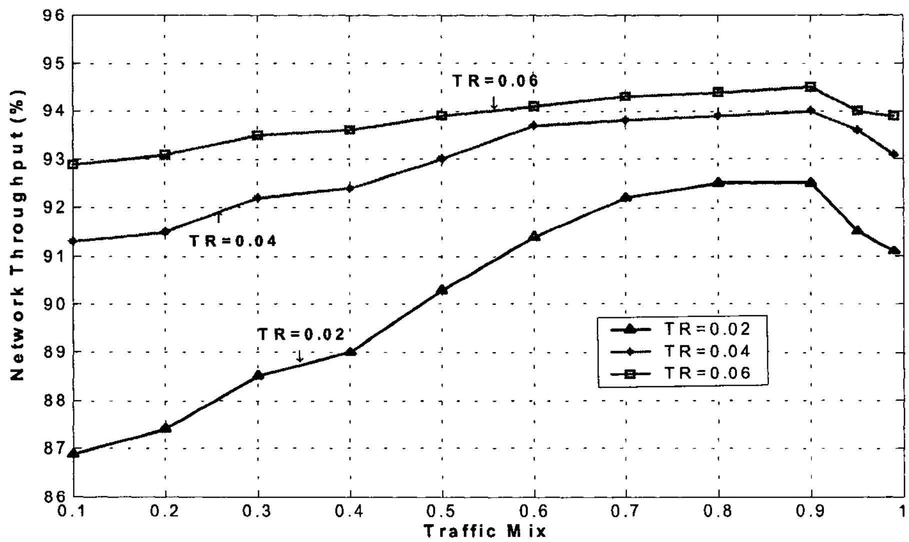

Figure 6-5: Performance of PVV with various TR thresholds. (Overload traffic condition, 4-node network, holding time ratio is 10.)

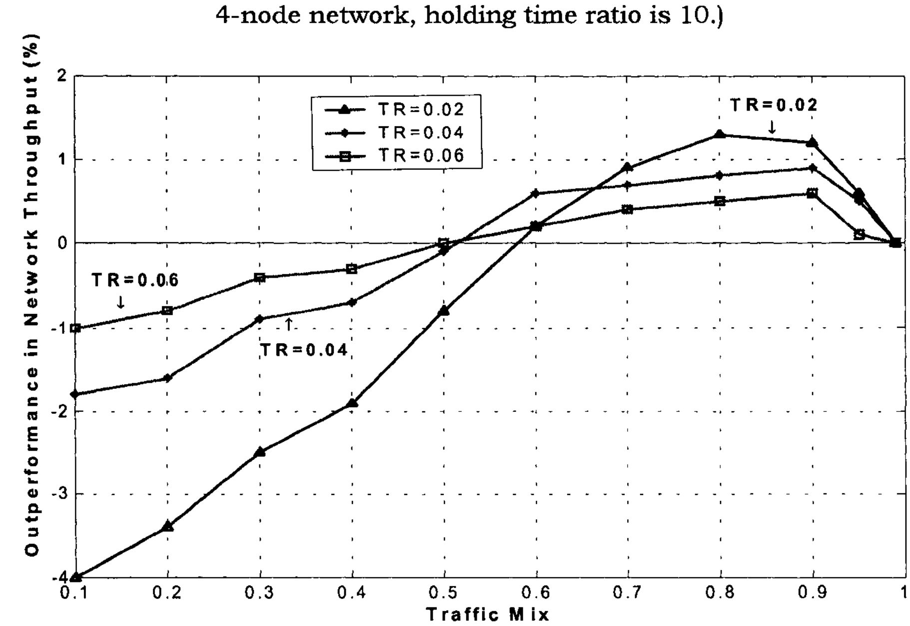

Figure 6-6: Outperformance of PVV compared to LLR+TR with various TR thresholds. (Overload traffic condition, 4-node network, holding time ratio is 10.)

We notice that the effective range of preventive-vacating does not change much as the TR threshold varies. As seen in Figure 6-6, within the effective range of preventive-vacating, the increase of the TR threshold has a "bad" effect: decreasing the positive outperformance of preventive-vacating compared to LLR+TR; while outside the effective range of vacating, the increase of the TR threshold has a "good" effect: reducing the negative-outperformance of vacating to LLR+TR. The rationale behind this is the fact that both the preventive-vacating routing and LLR+TR have improved performance (network throughput) with the appropriate increase of the TR threshold in overload traffic condition.

Moreover, as the TR threshold changes, the vacation threshold is, in general, set to be less than the TR, and could be, for instance, the last basic bandwidth unit. As we discussed in Chapter III, the TR first reserves the link for direct traffic (both short and long requests). In a second step, the vacating threshold reserves the link only for long requests.

Remember the Fixedness Assumption we made in Section 3.3? We validate it now through the statistics data collected in the simulation.

6.2.5 Fixedness Assumption Two Validating

In Section 3.3, the Fixedness Assumption Two assumes that the percentage of short requests taking the vacating actions, v, will be fixed if the overall traffic volume is unchanged. In the following table, we show the statistics data of v:

Traffic Mix→	90%	80%	70%	50%	30%	10%
	TR=0.02, Vacating Threshold=0.02					
Holding time ratio: 10	9.9%	10.5%	10.8%	10.9%	10.8%	10.6%
Holding time ratio: 50	10.5%	10.6%	11.0%	11.2%	10.9%	10.6%
Holding time ratio: 100	10.5%	10.7%	10.9%	11.2%	11.0%	10.7%
	TR=0.04, Vacating Threshold=0.02					
Holding time ratio: 10	5.4%	5.5%	5.6%	5.7%	5.8%	5.7%
Holding time ratio: 50	5.4%	5.3%	5.4%	5.6%	5.5%	5.6%

Table 6-3: Listing of the value of v. Overload traffic condition; 4-node network, and the overall traffic volume is fixed.

The data in Table 6-3 confirms the Fixedness Assumption Two, that is, within a fixed traffic+routing environment (keeping overall traffic volume, TR threshold and vacating threshold fixed), varying the traffic mix and holding time ratio will not have significant influence on the value of v. The same observation is also obtained in the 12-node network (see Appendix A).

We already know that the curves of the theoretical analysis and of the simulation results matched well. Now, the Fixedness Assumption Two is validated to be true. Hence, it is reasonable to believe that the Fixedness Assumption One (the value of $C_{r2} - C_{r1}$ is fixed) we made along with Fixedness Assumption Two is also true.

Besides, in the simulation of PVV we also observe that:

- short requests and long requests have similar blocking rates in preventive-vacating, just like LLR+TR. This is reasonable as the two kinds of requests in preventive-vacating have, in general, equal access to the overall network resources;
- comparing with LLR+TR, the short requests in preventive-vacating routing have a much higher two-hop alternating ratio and much lower direct routing ratio; while for long requests, it is the contrary. This is also reasonable and it is due to the vacating mechanism of PVV.

6.3 Preemptive-vacating Routing (PEV)

Preemptive-vacating routing (PEV) is viewed as the ideal case of preventive-vacating routing. In this section, the performance of preemptive-vacating will be studied, and compared with both the preventive-vacating and LLR+TR.

6.3.1 Performance Analysis

As can be seen in Figure 6-7 and Figure 6-8a,b, preemptive-vacating always has a *positive* outperformance compared to LLR+TR in the full range (0.0, 1.0) of traffic mix (in both networks). This means that preemptive-vacating is effective in the full range of traffic mix.

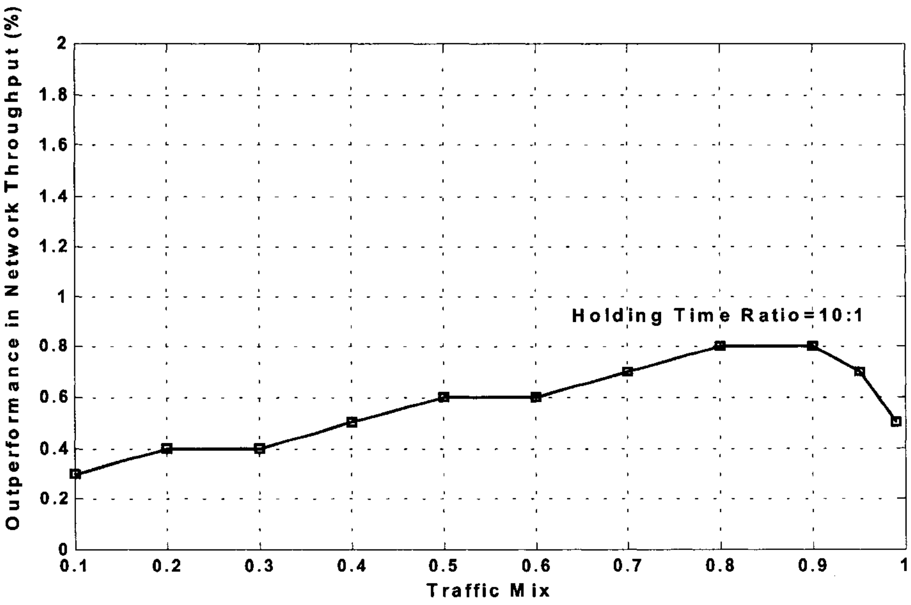

Figure 6-7: Outperformance of the Preemptive-Vacating (PEV) scheme compared to LLR+TR in nominal traffic condition. TR threshold=2%. 4-node network. **Note** that in this figure, it is really too difficulty to illustrate all the outperformance curves of PEV with various holding time ratio, as they are too close. Hence, we only draw one outperformance curve of holding time ratio as 10 to represent also the curves with other holding time ratio.

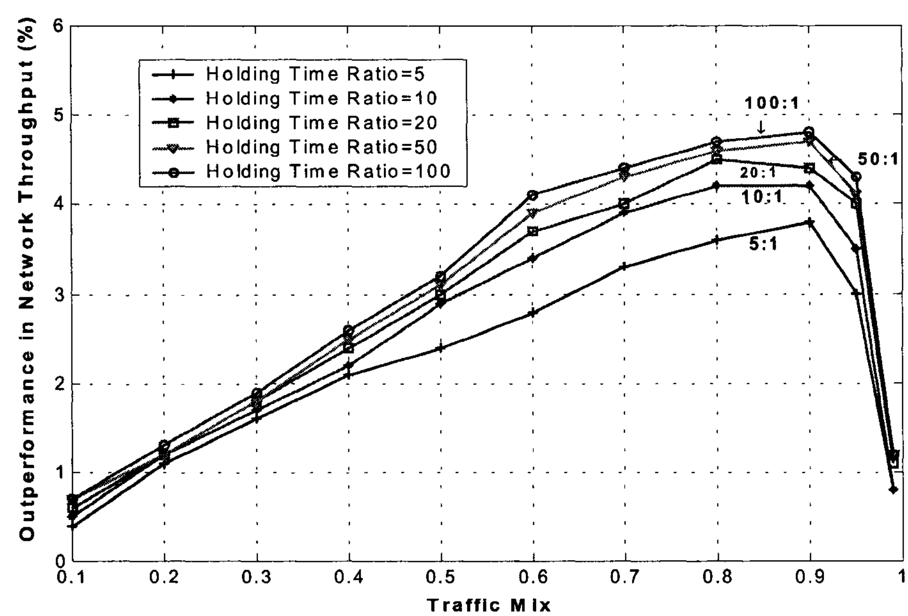

Figure 6-8a: Outperformance of the Preemptive-Vacating (PEV) scheme compared to LLR+TR in overload traffic condition. TR threshold=2%. 4-node network.

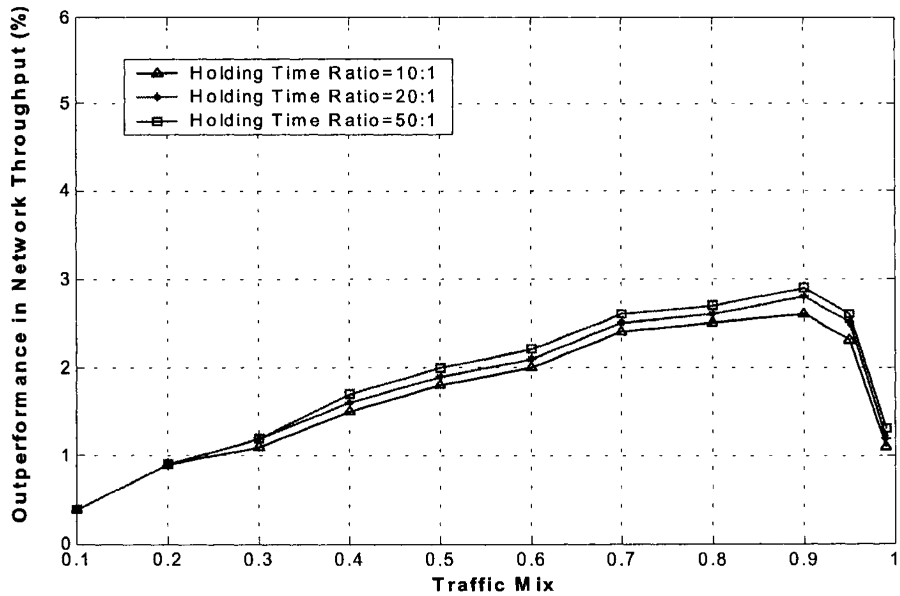

Figure 6-8b: Outperformance of the Preemptive-Vacating (PEV) scheme compared to LLR+TR in overload traffic condition. TR threshold=3%. 12-node network.

Compared with preventive-vacating routing, preemptive-vacating has a much higher outperformance to LLR+TR in terms of network throughput (around 4.5% with holding time ratio of 100 in the 4-node network and 3% with holding time ratio of 50 in the 12-node network). However, the forms of the outperformance curves of these two vacating schemes are similar (Figure 6-2, 6-3a,b, 6-7, 6-8a,b). In addition, larger values in the preemptive-vacating outperformance occur within the effective range of preventive-vacating ([0.60, 0.99]) in traffic mix, with the maximal value within the range of 80%~90%, which is the same as preventive-vacating. When leaving the effective range of [0.60, 0.99] and moving to the left extreme point 0.0, the outperformance of preemptive-vacating decreases linearly to zero. When moving to the right extreme point 1.0, the outperformance also drops very sharply to zero. This is because the traffic will become homogenous when the traffic mix is either 0.0 or 1.0. Thus, there is no preemptive-vacating and no outperformance.

Similarly as equation (3.14) for preventive-vacating, the behavior of preemptive-vacating can be explained through equation (3.16) in Section 3.3.2:

$$\textit{The overall cost difference between LLR+TR and preemptive-vacating}$$
$$= T\times\frac{x-1}{x}\times w\times y\times(C_{r2}-C_{r1}) \qquad (3.16)$$

, where T is the overall fixed traffic volume, x is the holding time ratio, y is the traffic mix, and w is the percentage of long requests taking preemptive-vacating actions (Section 3.3.2).

According to equation (3.16), whether the overall cost difference is positive or not is only decided by $(C_{r2}-C_{r1})$, since all the other parameters are positive. In our system model and simulation environment, $(C_{r2}-C_{r1})>0$ is generally true. Thus, the overall cost difference is positive in the full range of traffic mix, so is the outperformance of preemptive-vacating compared to LLR+TR.

Remember that we have claimed at the end of section 3.3.2 that the shape of the curve of equation (3.16) is decided by the value of $w\times y$. Now, let us verify this claim.

We draw the curves of $w\times y$ in the full range of traffic mix with different holding time ratio in Figure 6-9. Comparing Figure 6-9 with Figure 6-8: we find a perfect match! This confirms the correctness of claim, and furthermore, the exactness of equation (3-16).

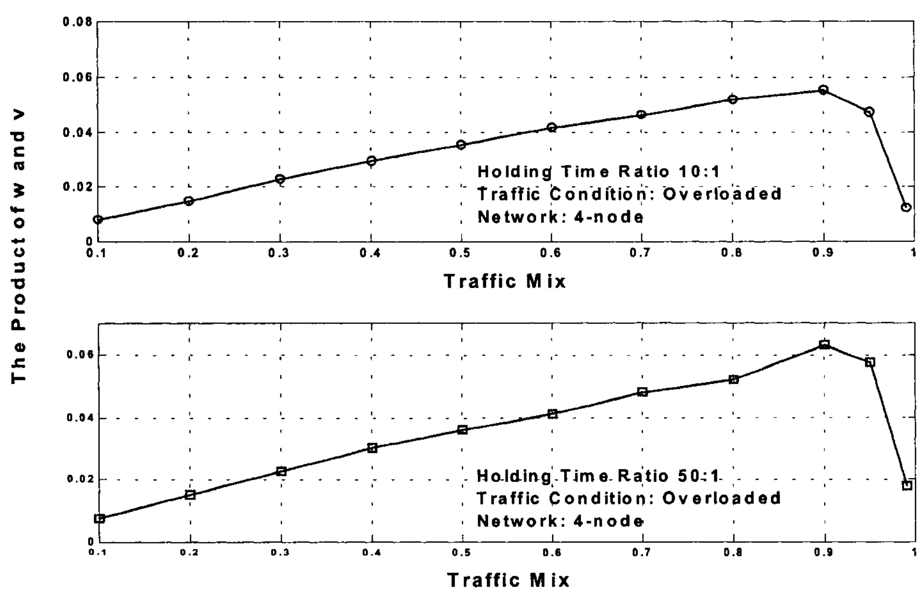

Figure 6-9: $w\times y$ vs. various traffic mix and holding time ratio

On the effect of the holding time, we notice that as the holding time ratio increases, so does the outperformance of preemptive-vacating. However, this increase is not as obvious or significant as in the case of preventive-vacating, especially when the holding time ratio is larger than ten. This can still be explained through equation (3.16): the holding time ratio can impact the amplitude of (3.16) only through the factor $\frac{x-1}{x}$, and $\frac{x-1}{x} \approx 1$ when x is large.

6.3.2 On the Effect of the TR

We present the network throughput of preemptive-vacating and its outperformance compared to LLR+TR, both with various TR thresholds, in Figure 6-10 and Figure 6-11, respectively.

As can be seen in both Figures, the outperformance of preemptive-vacating is inversely proportional to the increase of the TR threshold. Actually, increasing the TR flattens the performance and outperformance curves of preemptive-vacating, which is the same behavior as in the preventive-vacating case.

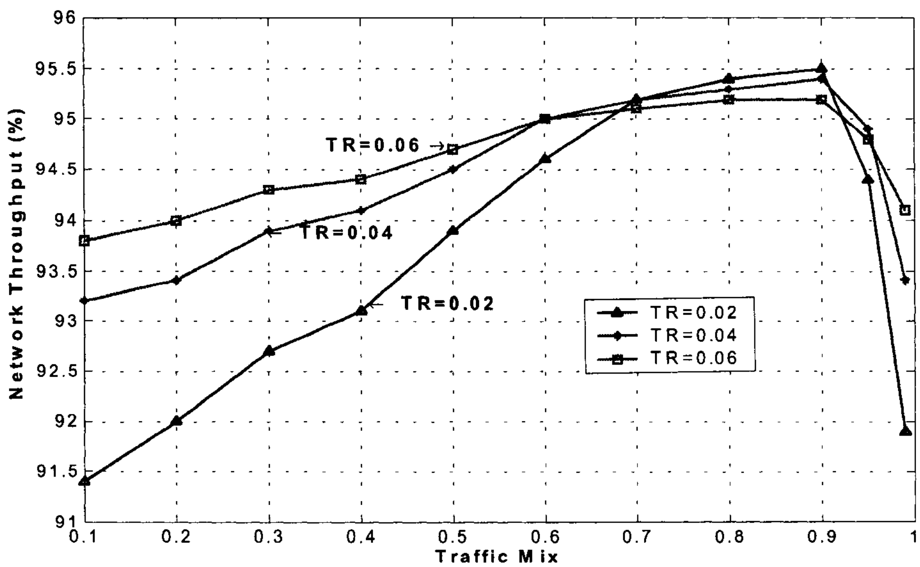

Figure 6-10: Performance of PEV compared to LLR+TR with various TR thresholds. (Overload traffic condition, 4-node network, holding time ratio is 10.)

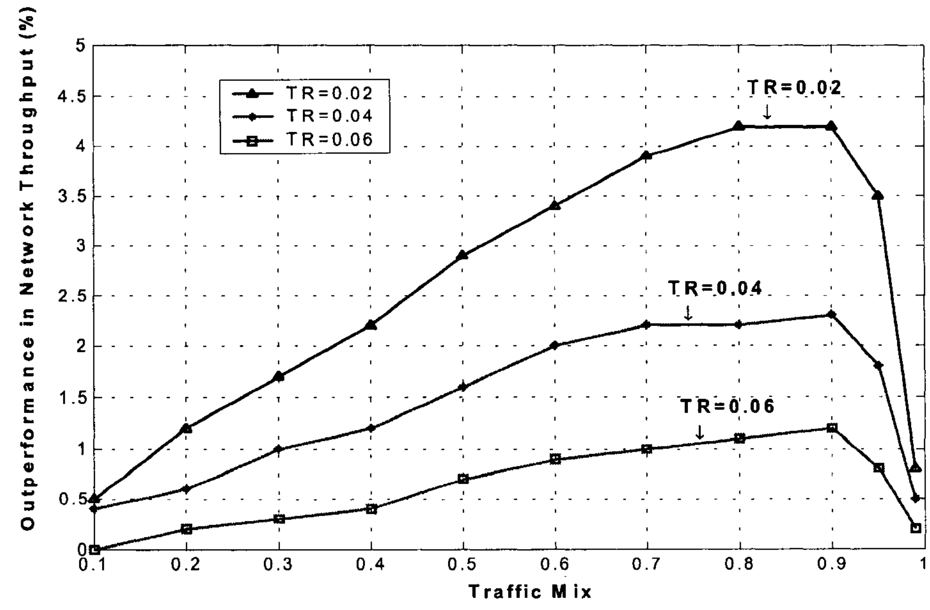

Figure 6-11: Outperformance of PEV to LLR+TR with various TR thresholds. (Overload traffic condition, 4-node network, holding time ratio is 10.)

However, there are still differences between the two vacating routing schemes:

- Within the effective range of preventive-vacating in traffic mix [0.65, 0.99], the performance of preemptive-vacating, in terms of network throughput, is not sensitive to the change of the TR threshold (Figure 6-10). Or, the network throughput even drops slightly with a higher TR. Note that this is different from both the preventive-vacating and LLR+TR, and it is mainly due to the special routing mechanism of preemptive-vacating, in which re-routing is allowed. Thus, preemptive-vacating itself can achieve part of the function of the TR, that is to say, prohibiting over-alternating. Therefore, the desire for an appropriate and sufficient TR threshold in preemptive-vacating becomes not as strong as in LLR+TR.
- When beyond the effective range, the performance of preemptive-vacating becomes sensitive to the change of TR, just as in preventive-vacating.
- In the full range of traffic mix, the outperformance of preemptive-vacating always increases as the TR threshold decreases. (Figure 6-10). There is no cross point, like in the preventive-vacating case (Figure 6-6), among the curves in

Figure 6-11. This is, partly, because preemptive-vacating is effective (positive outperformance to LLR+TR) in the full range of traffic mix.

6.3.3 Preventive-vacating vs. Preemptive-vacating Routing Schemes

Generally speaking, preventive-vacating is only a conservative approach, while preemptive-vacating is much more aggressive. As they are the routing schemes directly derived from our basic vacating idea, it is interesting to compare them in the following table:

Criteria	**Preventive-vacating**	**Preemptive-vacating**
Outperformance compared to LLR+TR	Positive within effective range; negative outside this range	Always positive, and the value is larger than PVV, especially in overload
Blocking Rates of long requests short requests	Equal	Equal
Sensitivity to changing the TR threshold	Sensitive	Not sensitive within [0.60, 0.99]; sensitive outside this range
Effective Range in traffic mix	[0.60, 0.99]	Full range
Sensitivity to changing the Holding time ratio	Sensitive	Not sensitive

Table 6-4: The comparison of the two vacating routing schemes.

6.4 Approximated Least Cost Routing (A-LCR)

Approximated Least Cost Routing (A-LCR) is based on the approximated expression for the cost rate (equation 6-28), which is derived from the results of the MDP approach for multi-rate traffic [63]. In this section, we study the performance of the A-LCR scheme, along with its particular and inherent flow control mechanism.

6.4.1 On-line Parameter Estimation

To implement the Approximated Least Cost Routing scheme in NS2, it is necessary to compute the cost rate according to equation (3.28) when making each individual routing decision. In equation (3.28), r_k^l, μ_k, and N^l are known *a priori*; the only unknown parameter is the offered arrival rate on link *l*: λ_k^l (*k* is L or S). We should measure λ_k^l in real time for long and short requests, respectively. The flowchart of the estimation process in the simulation is showed as follows.

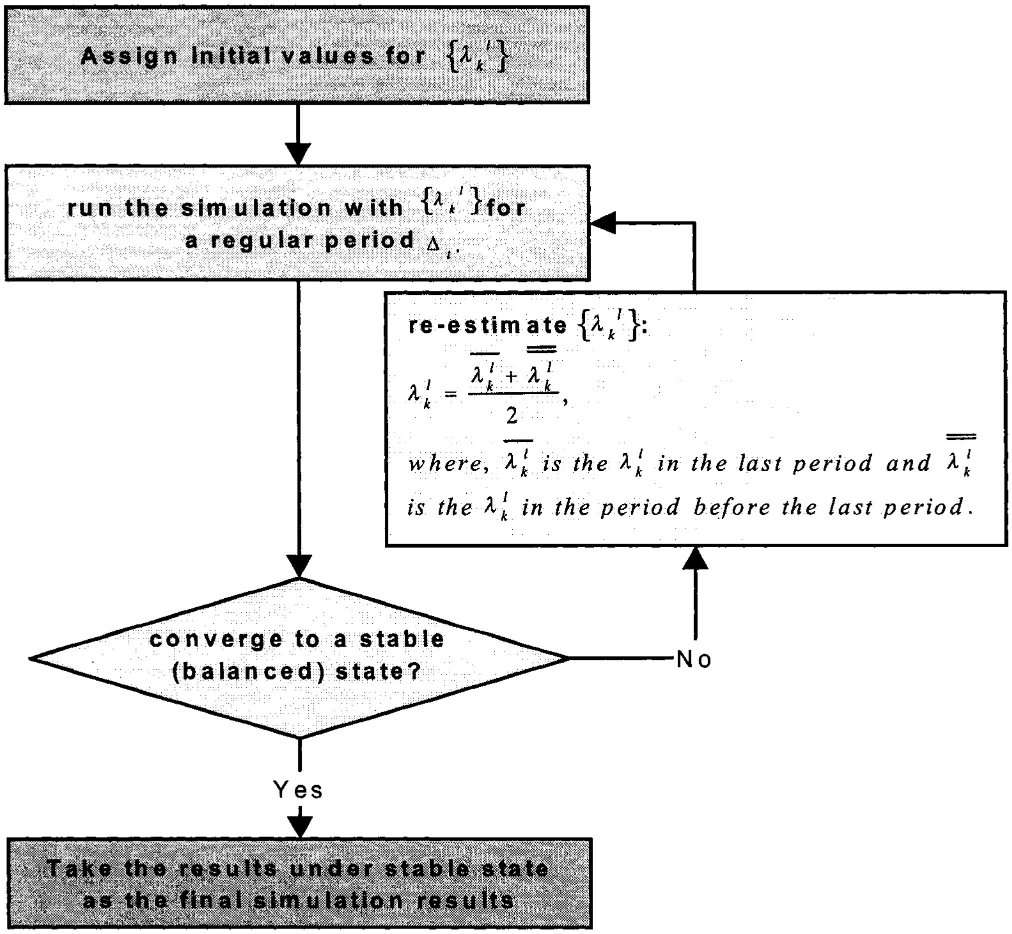

Figure 6-12: Flowchart of real-time Estimation Process of λ.

Clearly, the value of Δ_t can affect the performance of the routing scheme. In our simulation, the optimal Δ_t is set as $10 \times \frac{1}{\mu_L}$ ($\frac{1}{\mu_L}$ is the mean holding time of long request). After $10 \times \Delta_t$, in general, the estimation process of λ_k^l is stable. An example is shown in Figure 6-13.

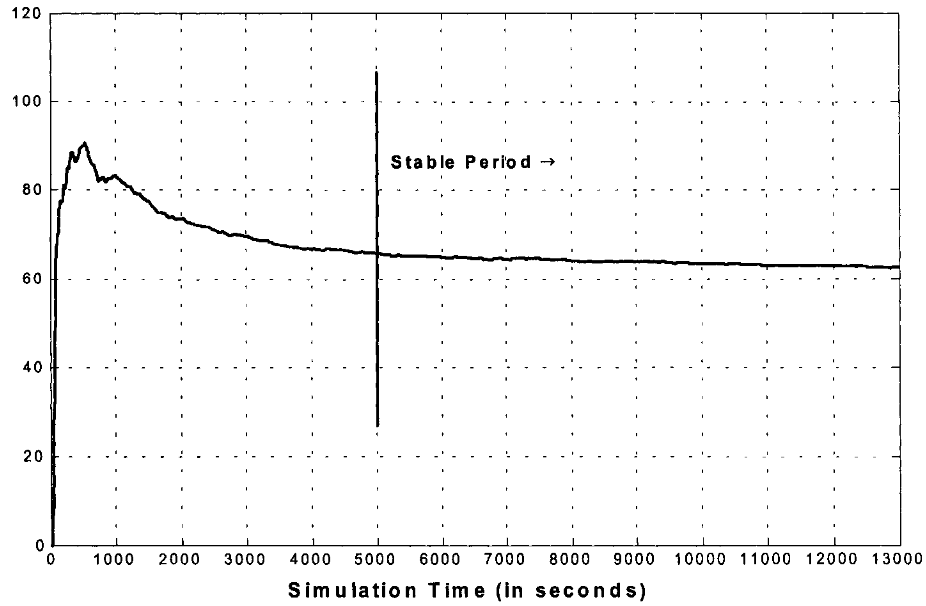

Figure 6-13: Time-averaged blocking rate vs. simulation time. (Overload traffic condition in the 4-node network, traffic mix is 80%; $1/\mu_L = 50$ Seconds, $\Delta_t = 500$ Seconds, after $10 \times \Delta_t$, the estimation process of λ_k^l is stable)

6.4.2 Performance Analysis

In the simulation of A-LCR, the call reward parameters, r_L and r_S are set to be proportional to their associated holding times: $r_L = x,\ and\ r_S = 1$. The comparison of performance of A-LCR and LLR+TR with different TR thresholds is given in Table 6-5 and 6-6. Table 6-5 displays the performance comparison under several traffic mix and for a holding time ratio of 10. While in Table 6-6, the holding time ratio is set to 50.

Traffic Mix→	**80%**	**50%**	**20%**	**80%**	**50%**	**20%**
	Nominal Traffic			Overload Traffic		
A-LCR	99.3%	99.3%	99.2%	94.1%	94.1%	93.8%
LLR+0.02	99.0%	99.1%	98.9%	91.2%	91.0%	90.8%
LLR+0.04	99.2%	99.3%	99.1%	93.1%	92.9%	92.8%
LLR+0.06	99.0%	99.1%	99.0%	93.9%	93.7%	93.6%

Table 6-5: Network Throughput of A-LCR and LLR+TR with various traffic mix and traffic conditions. Holding Time Ratio is 10, 4-node network.

Traffic Mix→	**80%**	**50%**	**20%**	**80%**	**50%**	**20%**
	Nominal Traffic			Overload Traffic		
A-LCR	99.4%	99.4%	99.2%	94.3%	94.2%	94.0%
LLR+0.02	99.3%	99.1%	99.0%	91.3%	91.2%	91.0%
LLR+0.04	99.4%	99.3%	99.2%	93.3%	93.2%	93.0%
LLR+0.06	99.1%	99.0%	99.1%	94.1%	94.0%	93.8%

Table 6-6a: Network Throughput of A-LCR and LLR+TR with various traffic mix and traffic conditions. Holding Time Ratio is 50, 4-node network.

Traffic Mix→	**80%**	**50%**	**20%**	**80%**	**50%**	**20%**
	Holding Time Ratio is 10			Holding Time Ratio is 50		
A-LCR	95.2%	95.1%	95.0%	95.4%	95.3%	95.1%
LLR+0.01	87.2%	87.0%	86.9%	87.2%	87.0%	87.0%
LLR+0.03	93.8%	93.5%	93.4%	93.9%	93.6%	93.4%
LLR+0.05	94.9%	94.7%	94.7%	95.0%	95.0%	94.9%

Table 6-6b: Network Throughput of A-LCR and LLR+TR with various traffic mix and holding time ratio. Overload traffic condition, 12-node network.

Looking closely at Tables 6-5 and 6-6a,b, the basic conclusion is that A-LCR always attains at least the optimal performance of LLR+TR under various combinations of traffic mix, holding time ratio and traffic loads. In addition, just like LLR+TR, varying the traffic mix, or varying the holding time ratio has little impact on the performance of A-LCR.

Again, in Tables 6-5 and 6-6a,b, regarding LLR+TR in nominal traffic condition, the optimal value of the TR threshold can be small; while in overload traffic condition, the optimal value of the TR threshold should become relatively larger to obtain better performance. In other words, in order to keep optimal performance, the value of the TR threshold in the LLR+TR scheme must be *adaptive* to the traffic load. While for A-LCR, its inherent mechanism gives it the property of adaptability to the varying traffic loads. Thus, A-LCR can always reach the optimal performance of LLR+TR under various traffic loads.

On the other hand, although both the A-LCR and the LLR+TR schemes are using all link states to make routing decisions, in the case of LLR+TR, only the state of the route's link with the smallest free capacity counts. Hence, the probability of choosing the route is not changed for a whole range of states on the other links, provided that their free capacity is larger or equal to that of the bottleneck. While in the case of A-LCR, the state of each link affects the route net_gain. Thus, *more* information about the network state is used when making a routing decision in the A-LCR and hence it is reasonable that A-LCR outperforms LLR+TR.

6.4.3 Reward, Cost and Net_Gain

In the previous simulation of A-LCR, both the reward parameter and cost of accepting a request, long or short, are proportional to the associated holding time. Hence, the route net_gain of accepting a request is also proportional to its holding time. Note that the route net_gain is equal to the reward of accepting a call minus the cost on the route resulting from accepting the call (equation 3.22). Now we introduce a concept called *normalized reward parameter*, $r_k^{'}$ ($r_k^{'} = r_k \times \mu_k, k$ is L or S) [40,41]. Then in our case, the normalized reward parameters for long and short requests are *equal* to each other, which is one. It means that in the unit time and in average, the *same* amount of net-gain is obtained by the network through accepting a request, whether a long or a short one, on a route. Therefore, A-LCR treats long-requests and short-requests equally, and thus they have the same priority to access network resources. This results in the behavior of long and short requests being very similar in A-LCR: similar blocking rate, similar two-hop alternating rate, and similar direct-routing rate.

In A-LCR, the net_gain by an accepted long request, whether on a direct route or a two-hop route, is always positive. This is because if the net-gain was negative, then the long-request would have been rejected. Thus, we can explain the relationship between the TR threshold and our vacating schemes from the viewpoint of reward, cost, and net_gain.

If the TR threshold is not appropriate, say a very small threshold under heavy traffic loads, then the net_gain of accepting a request, long or short, on a two-hop route becomes negative and a long request will do much more damage (negative net-gain) than a short request due to its longer holding time. Hence, preventing long requests from two-hop alternating is much more important than preventing short requests. Therefore, our vacating idea should be effective under such conditions.

However, when the TR threshold has an appropriate value, that is to say, it is adaptive to the traffic load, then the net_gain of accepting a request, long or short, on a two-hop route should always be positive in general. Thus, since a two-hop alternated long request will never induce damage, then why should vacating be necessary? In other words, using an adaptive TR threshold, the scenario (last bandwidth unit case) we described when introducing our basic vacating idea in section 3.1, will not happen. Therefore, the outperformance of vacating, preventive or preemptive compared to LLR+TR, is inversely proportional to the TR threshold.

6.4.4 GOS Distribution Control

To demonstrate the particular flow control mechanism of A-LCR, we present the traffic loss (computed as $1-(\textit{Network Throughput})$), the blocking rates for short requests and long requests versus the normalized reward parameter of long requests, $r_L^{'}$ $(r_L^{'} = r_L \times \mu_L)$, in Figure 6-14 and 6-15 (for two networks).

As we can see, the reward parameters provide a mechanism for controlling the ratio of the long request blocking rate to the short request blocking rate over a very wide range, including their equalization. Hence, through adjusting the reward parameter(s), one can achieve almost independent control of grade of service (blocking rate) for either long or short requests.

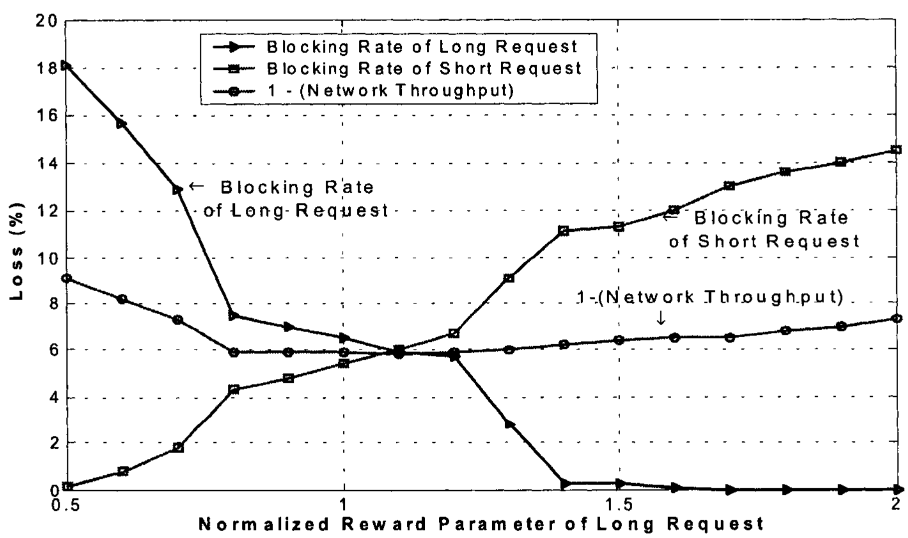

Figure 6-14: Traffic Loss vs. normalized reward parameter of long requests in the 4-node network in overload traffic. Traffic mix is 50%, holding time ratio is 10.

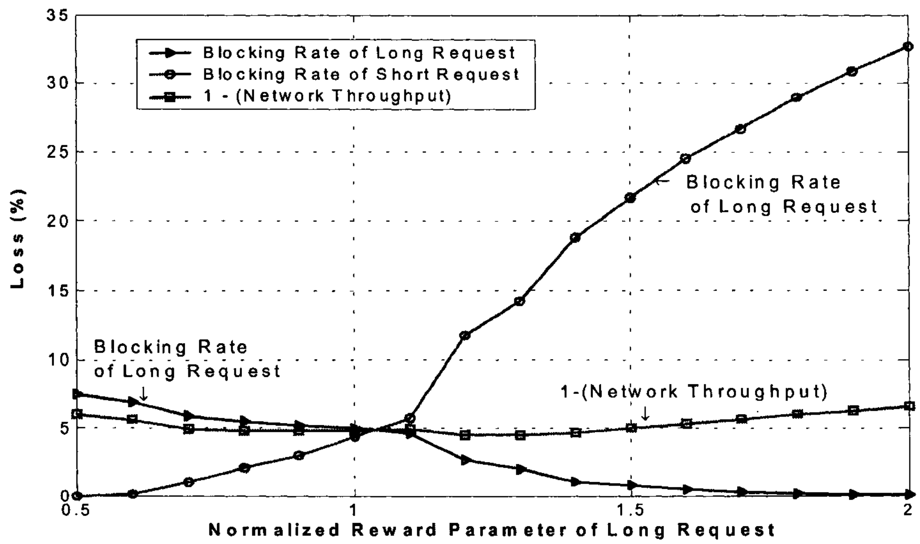

Figure 6-15: Traffic Loss vs. normalized reward parameter of long requests in12-node network in overload traffic. Traffic mix is 80%, holding time ratio is 10.

Figure 6-14 and 6-15 also show that Network Throughput maximization is achieved if the normalized reward parameters, $r_k^{'}$ ($r_k^{'} = r_k \times \mu_k$, k is L or S), equal each other ($r_L^{'} = r_L \times \mu_L = r_L^{'} = r_S \times \mu_S \approx 1.0$).

Furthermore, if the reward parameters are proportional to call tariffs, then A-LCR can also achieve network income (money) maximization.

Remember in Section 3.5.5 that with reward parameters, we could have another system performance metric, the *Fractional Revenue*, in addition to the Network Throughput. Let us use this new performance metric to compare A-LCR with LLR+TR in Table 6-7. The Fractional Revenue is computed as:

$$\frac{No.\ of\ carried\ long\ requests \times r_L + No.\ of\ carried\ short\ requests \times r_S}{No.\ of\ offered\ long\ requests \times r_L + No.\ of\ offered\ short\ requests \times r_S}.$$

	4-Node Network (TR=0.04)			12-Node Network (TR=0.03)		
$r_L^{'} \rightarrow$	0.5	1.5	2.0	0.5	1.5	2.0
A-LCR	94.0%	95.8%	96.0%	95.0%	96.2%	96.3%
LLR+TR	93.3%	93.0%	93.0%	94.0%	93.9%	93.8%

Table 6-7: Performance comparison in Fractional Revenue with varied normalized reward parameter for long requests. Overload traffic condition; Holding Time Ratio as 10, traffic mix is 80%, $r_S^{'} = 1.0$.

From the simple examples in the above Table, we can observe the inherent adaptive property of the A-LCR scheme. Due to this property, A-LCR is automatically suitable to the varying reward parameter(s) and the new system performance metric. While LLR+TR (along with PVV and PEV) is not sensitive to the changes in the reward parameters, and this results in the performance degradation shown by the new performance metric. In some conditions, this degradation is significant compared to A-LCR. From this point of view, A-LCR is better than LLR+TR, PVV, and PEV.

6.5 Re-routing (RER)

We now study how the re-routing (RER) scheme performs in the traffic environment with non-homogenous holding times.

6.5.1 Re-routing Performance

We present the performance in terms of network throughput of the re-routing scheme under different traffic mix, holding time ratio, traffic loads, and different TR thresholds in Table 6-8 and 6-9 for two network examples respectively.

	Holding Time Ratio=10			**Holding Time Ratio=50**		
Traffic Mix→	**80%**	**50%**	**20%**	**80%**	**50%**	**20%**
			Nominal Traffic			
RER+0.00	99.9%	99.9%	99.9%	99.9%	99.9%	99.9%
RER+0.02	99.8%	99.8%	99.8%	99.9%	99.8%	99.8%
RER+0.04	99.7%	99.7%	99.7%	99.8%	99.7%	99.7%
			Overload Traffic			
RER+0.00	96.5%	96.4%	96.3%	96.6%	96.5%	96.5%
RER+0.02	96.0%	96.0%	95.9%	96.3%	96.2%	96.2%
RER+0.04	95.7%	95.6%	95.5%	95.9%	95.8%	95.8%
RER+0.06	95.3%	95.3%	95.2%	95.5%	95.5%	95.5%

Table 6-8: Network Throughput of the Re-routing scheme with various holding time ratio, traffic mix, and TR thresholds in the 4-node network.

	Holding Time Ratio=10			**Holding Time Ratio=50**		
Traffic Mix→	80%	50%	20%	80%	50%	20%
RER+0.00	97.3%	97.0%	97.0%	97.5%	97.2%	97.3%
RER+0.02	96.7%	96.6%	96.7%	96.8%	96.7%	96.6%
RER+0.04	96.4%	96.2%	96.3%	96.4%	96.4%	96.3%
RER+0.06	96.0%	95.9%	96.0%	96.1%	96.0%	95.9%

Table 6-9: Network Throughput of the Re-routing scheme with various holding time ratio and traffic mix, in the 12-node network. Overload traffic condition.

From the results in Table 6-8 and 6-9, we observe that the re-routing scheme with non-homogenous (holding time) traffic has a similar behavior with homogenous traffic as described in [46,47]. In all the test cases, the re-routing scheme always provides a superior performance. The network throughput of the re-routing scheme can be almost 100% in nominal traffic condition. In overload traffic condition, the outperformance of re-routing compared to LLR+TR is even more significant (5.5% at best). Comparing with the other routing schemes (PVV, PRV, A-LCR) we studied in previous sections, the re-routing scheme is the one with the best (highest) network throughput.

Furthermore, varying the traffic mix or holding time ratio has almost no impact on the performance of the re-routing scheme. This demonstrates the useful self-adaptive property of the re-routing scheme to traffic dynamics. As long and short requests are treated blindly by the re-routing scheme, they have equal grade of service (blocking rate). Thus, the fairness of the re-routing scheme is also good.

As shown in Table 6-8 and 6-9, the biggest network throughput value with the re-routing scheme occurs when no TR (TR threshold=0) is applied. In addition, the performance of the re-routing scheme degrades when using a TR. In other words, the throughput of a network with the re-routing scheme is inversely proportional to the trunk reservation threshold. This phenomenon is the same as in the homogenous traffic case [46,47], and can be explained by the self-stabilizing property of the re-routing scheme. In the re-routing scheme, the bandwidth devoted to an alternate route is only used temporarily. A call/request is re-routed when capacity is available on its direct route. Thus, there is less incentive to be conservative and to turn down an alternately routed call/request when a route has the requested capacity.

However, trunk reservation can be used to *regulate* the rate of re-routing actions, and hereby the switching, processing, and signaling loads imposed on the network. As can be seen in the following table, the re-routed traffic rate (i.e., the rate/ratio of requests successfully re-routed) drops dramatically when the TR threshold increases.

Traffic Mix →	80%	50%	20%
Pure Re-routing +TR=0.00	4.1%	4.8%	5.2%
Pure Re-routing +TR=0.02	2.5%	3.0%	3.4%
Pure Re-routing +TR=0.04	1.5%	1.9%	2.2%
Pure Re-routing +TR=0.06	0.9%	1.1%	1.4%

Table 6-10: The re-routed traffic rate with different traffic mix and TR threshold. Holding time ratio is 10, 4-node network.

6.5.2 New Triggering Policies in the Re-routing Scheme

In Section 3.5.5, we proposed two new triggering policies, a long request arriving triggering policy and a short request triggering policy, for the re-routing scheme in non-homogenous traffic. Based on the network throughput and the re-routed traffic rate as metric, we compare these two policies with the original "pure" arriving triggering policy in Table 6-16.

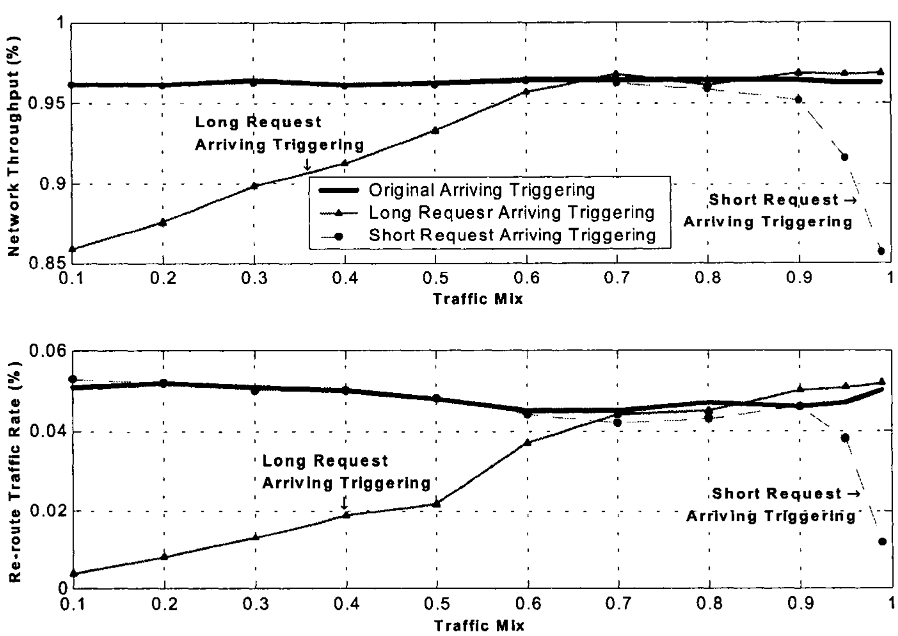

Figure 6-16: Comparison of three arriving triggering policies in the 4-node network with holding time ratio of 10. (The thick flat line is for the pure arriving triggering policy).

As illustrated in Figure 6-16, the long request triggering policy has an effective range, which is roughly equal to the previous effective range of vacating; outside this range, the performance of the long request triggering policy drops rapidly.

Compared with the long request triggering policy, the short request triggering policy has a wider effective range. This is mainly because short requests have a much higher arrival rate than long requests. Thus, they have more chances to trigger the re-routing process and obtain a higher re-routed traffic rate over a wider range. This can be validated by comparing the curves of the re-routed traffic rates in Figure 6-16. We find that the performance curves of the different triggering policies are in accordance with their re-routed traffic rate curves.

We also observe that within the effective range both the long and short request triggering policies have very similar re-routed traffic rates compared with the pure arriving triggering policy, which is around 4% ~ 5%. This indicates that with non-homogenous holding time traffic, it is generally not important which one (long or short requests) triggers the re-routing process, and the important thing is to keep a certain re-routed traffic rate.

The same conclusion is obtained in cases with other holding time ratios and in the 12-node network.

6.5.3 Re-routing vs. Preemptive-vacating Routing Schemes

As we mentioned previously, preemptive-vacating is inspired from the spirit of the re-routing scheme. Thus, the two schemes have some inherent connections, along with obvious differences.

In the re-routing scheme, the in-progress calls, whether long or short, are re-routed back from alternate routes to their direct routes. While in preemptive-vacating, the triggering policy is long request arriving. Only the in-progress *short* calls are re-routed, and re-routed from the direct routes to the alternate routes. The advantage of preemptive-vacating is that it reduces the amount of re-routing actions compared with pure re-routing, but the cost is performance degradation, especially when the traffic mix is less than 60% (see Figure 6-17 below).

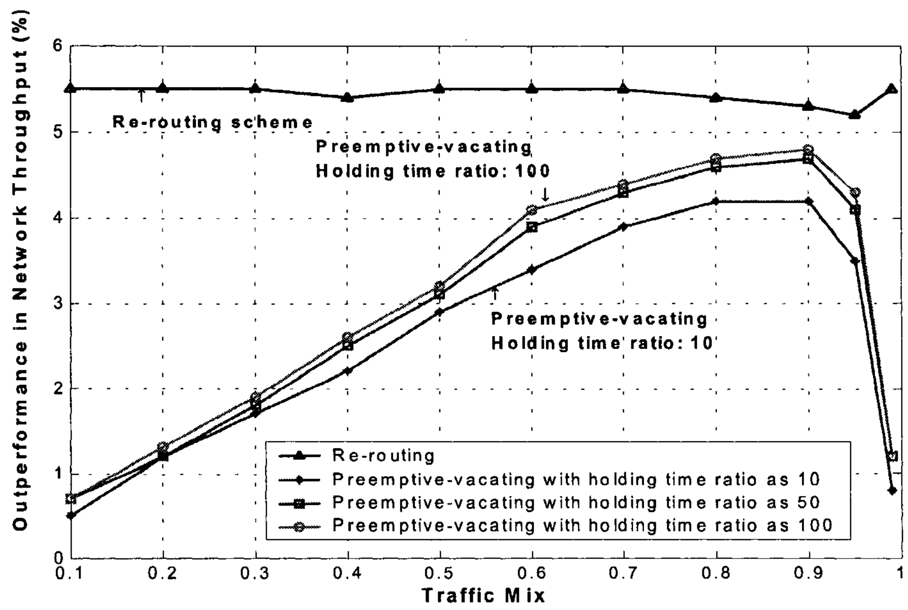

Figure 6-17: Comparing the outperformance of re-routing and PEV to LLR+TR.

On the other hand, in preemptive-vacating routing, re-routing is allowed in a special form (as described previously). Preemptive-vacating thus inherits, to a certain extent, the self-stabilizing property of the re-routing scheme. For instance, within the effective range of preventive-vacating, $[0.60, 0.99]$, where preemptive-vacating has the closest performance to the re-routing scheme, preemptive-vacating routing is not sensitive to the change of the TR threshold. In addition, increasing the TR threshold will even result in a slight degradation in the performance of preemptive-vacating (Figure 6-10). This is similar to the re-routing scheme.

6.6 Restricted Access Routing (RAR)

Restricted Access Routing (RAR) can be viewed, in general, as a modified version of Preventive-Vacating Routing (PVR). We will study its performance in this section and compare it with PVV.

6.6.1 Performance Comparison

Figure 6-18 and 6-19 present the outperformance of RAR compared to LLR+TR, in terms of the network throughput, with varying holding time ratio and in the full range of traffic mix.

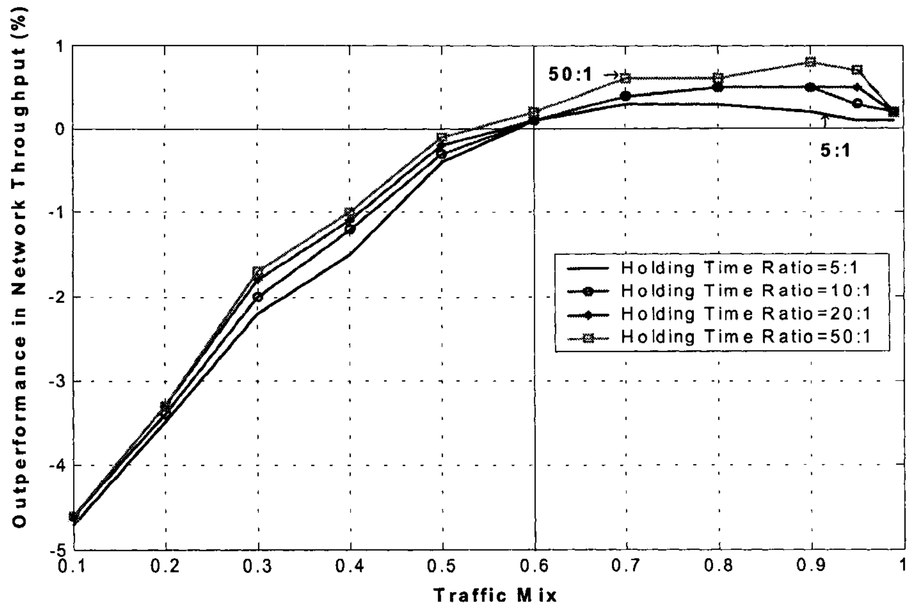

Figure 6-18: Outperformance of the Restricted Access Routing scheme to LLR+TR in nominal traffic condition. TR threshold=Restricted Access threshold=2%. 4-node network.

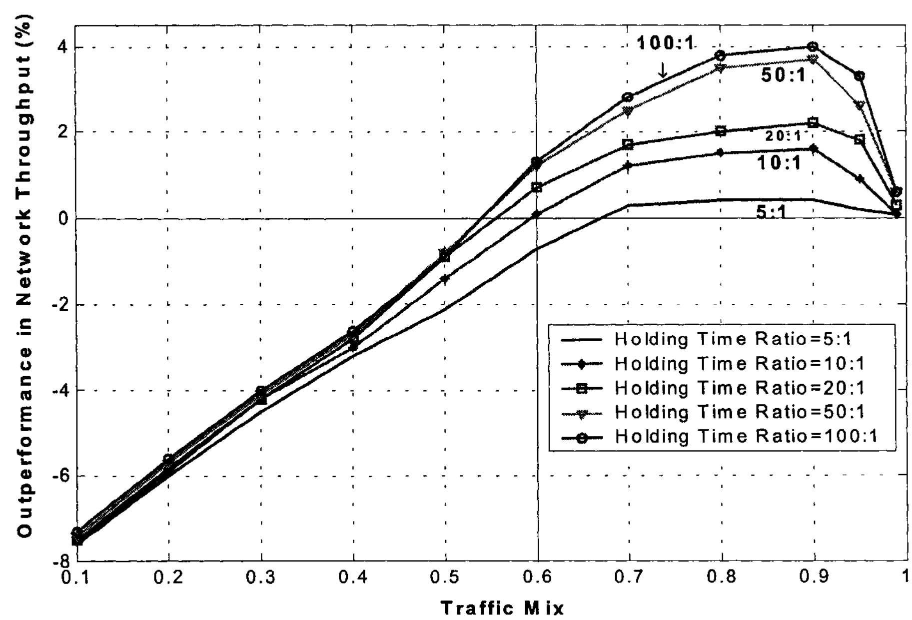

Figure 6-19a: Outperformance of the Restricted Access Routing scheme to LLR+TR in overload traffic condition. TR threshold=Restricted Access threshold=2%. 4-node network.

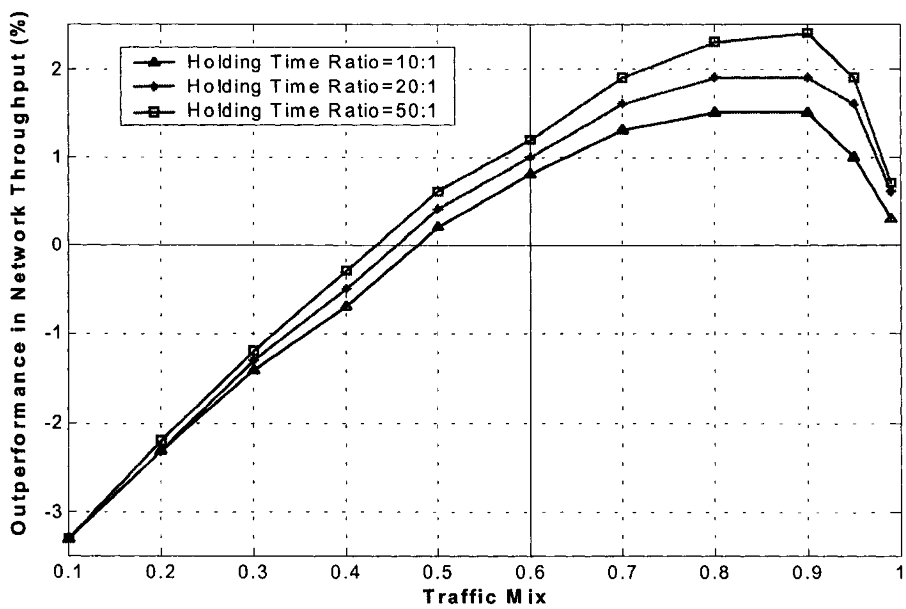

Figure 6-19b: Outperformance of the Restricted Access Routing scheme to LLR+TR in overload traffic condition. TR threshold=3%; Restricted Access threshold=2%. 12-node network.

As can been seen, the general trend of the outperformance curves for RAR are similar to those of preventive-vacating routing in Figure 6-2 and Figure 6-3a,b. At the first glance, the outperformance of VAR compared to LLR+TR also increases as the traffic load varies from nominal to overload conditions.

On the effect of the traffic mix, RAR has an effective range around [0.60, 0.99], which is the same as the effective range of preventive-vacating. Within this range, the outperformance of RAR is positive, with the maximum value lying somewhere within 80%~90% in traffic mix. Beyond this range, the outperformance is negative, and drops linearly.

On the effect of the holding time ratio, the outperformance of RAR compared to LLR+TR increases as the holding time ratio does, but when the holding time ratio becomes very large, for instance >50, we also notice the increase of outperformance slows down. Alternatively, we can say that the increasing outperformance due to a changing holding time ratio becomes saturated when the holding time ratio becomes large enough.

Compared with preventive-vacating, RAR provides a better performance within its effective range in traffic mix. For instance, the outperformance of RAR compared to LLR+TR can be as high as 4% in the 4-node network (Figure 6-19a, holding time ratio is 100), while for preventive-vacating, it is about 3% (Figure 6-3a). In the 12-node network, it is 2.4% (Figure 6-19b, holding time is 50) vs. 1.7% (Figure 6-3b).

6.6.2 On the effect of the TR

We then investigate the effect of varying the TR threshold in the RAR scheme. We present the performance of RAR and the outperformance of RAR compared to LLR+TR, with different TR thresholds, in Figure 6-19 and Figure 6-20 (next page), respectively.

Compared with Figure 6-5 and Figure 6-6 (preventive-vacating), the observation is that the impact of the TR threshold with the RAR scheme is still similar to that in preventive-vacating, i.e., increasing the value of the TR threshold improves the performance, in terms of network throughput, of RAR under overload traffic condition, but the outperformance compared to LLR+TR decreases. The only difference with preventive-vacating is that the decrease in outperformance of RAR is not so rapid as in the preventive-vacating case. This is because the RAR mechanism achieves a small portion of the function of the TR, which is prohibiting over-alternating, through restricting short request from accessing the network resources. Thus, the requirement for applying an appropriate (higher) TR threshold in overload traffic is not as strong as in LLR+TR. Even under a higher TR threshold, RAR can still provide an obvious improvement (around 0.8% with TR=0.06 in Figure 6-21). However, this only happens within the effective range. Outside the effective range, the RAR mechanism *over*-restricts short requests from accessing network, and results in a severe degradation in performance.

Regarding the Reserved Access threshold of RAR, we set it not greater than (≤) the preventive-vacating threshold in general.

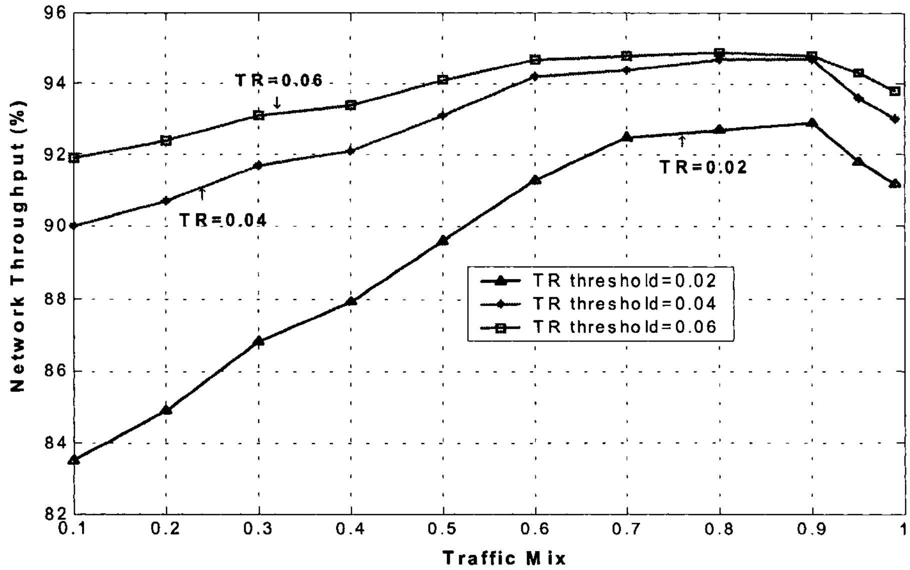

Figure 6-20: Performance of the Restricted Access Routing scheme with various TR thresholds. Nominal traffic condition. 4-node network.

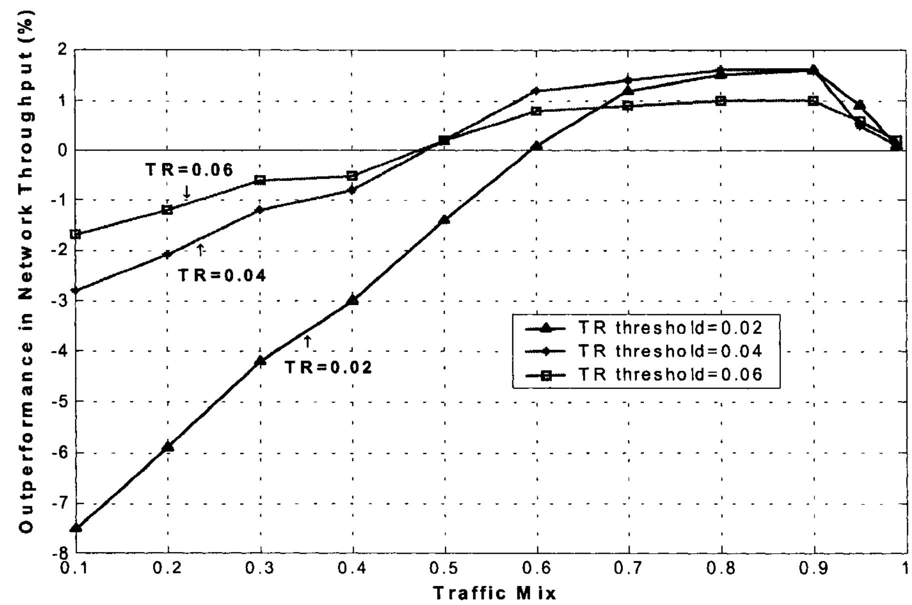

Figure 6-21: Outperformance of the Restricted Access Routing scheme to LLR+TR with various TR thresholds. Overload traffic condition. 4-node network.

6.6.3 The Fairness of RAR and Its Preemptive Version

In the RAR scheme, long requests are actually given higher priority than short requests to access the network resources. This has a significant influence on the grade of service (blocking rate) for long and short requests, respectively, as shown in the following Table.

	4-node network			12-node network		
Traffic Mix	Blocking -Long	Blocking -Short	Network Throughput	Blocking -Long	Blocking -Short	Network Throughput
90%	6.1%	17.4%	92.9%	3.4%	12.4%	95.7%
80%	4.7%	19.5%	92.7%	2.2%	12.3%	95.8%
50%	1.6%	19.2%	89.6%	0.8%	11.2%	94.0%
20%	0.4%	18.7%	84.9%	0.3%	10.8%	91.3%

Table 6-11: Performance of RAR in terms of the Blocking Rate and Network Throughput, among different traffic mix. The holding time ratio is 10 in the 4-node network, and 20 in the 12-node network; overload traffic condition. TR=0.02 in the 4-node network, TR=0.03 in the 12-node networks.

As seen in Table 6-11, although RAR performs better than preventive-vacating routing within the effective range, the improvement on performance is at the price of losing the fairness for long and short requests. In RAR, short requests have a much higher blocking rate than long requests for all of the combinations of traffic mix, holding time ratio and traffic loads.

To improve the unfairness issue in RAR, we proposed the *preemptively* restricted access routing in Chapter III: allowing a short request to access the reserved capacity but at the risk of being preempted should a long request arrive and find no idle capacity. We then present the performance of the preemptively RAR scheme in terms of blocking rates and network throughput as follows:

	4-node network			12-node network		
Traffic Mix	Blocking -Long%	Blocking -Short%	Network Throughput	Blocking -Long%	Blocking -Short%	Network Throughput
90%	6.5%	6.4%	93.5%	4.1%	3.9%	95.8%
80%	6.1%	5.9%	93.9%	4.1%	3.8%	95.9%
50%	9.0%	8.3%	91.3%	6.0%	5.3%	94.4%
20%	12.9%	11.9%	87.9%	8.2%	7.7%	92.2%

Table 6-12: Performance of Preemptively RAR in terms of the Blocking Rate and Network Throughput, among different traffic mix. The holding time ratio is 10 in the 4-node network, and 20 in the 12-node network; Overload traffic condition. TR=0.02 in the 4-node network, TR=0.03 in the 12-node networks.

As we expected, a significant improvement on fairness is obtained with the preemptively RAR scheme. Comparing the results in Table 6-11 and 6-12, we notice that the preemptively RAR scheme even gets a better network throughput than the original RAR, especially beyond the effective range of RAR. This is mainly because short requests become the dominant contribution to the traffic outside the effective range, while they are not within the range, and the preemptively RAR scheme focuses on improving the blocking rate of short requests.

6.7 Differentiated Shortest Distance Routing (Diff-SDR)

In this section, we study the only currently published dynamic routing scheme, Differentiated Shortest Distance Routing (Diff-SDR) [17], which focuses on the routing issue in networks with non-homogenous holding time traffic. In addition, we will compare it with the other two shortest-distance routing schemes, namely Dynamic Shortest Distance Routing (Dynamic-SDR) and Adaptive Shortest Distance Routing (Adaptive-SDR). It is shown in [17] that Diff-SDR performs better than the other two under non-homogenous holding time traffic.

6.7.1 Sampling Rate and Sampling Window

In shortest distance routing, the link metric is $\frac{1}{d_j}$, where d_j is the residual bandwidth of link j [15]. As proposed in [17], to obtain a better performance than the original shortest distance routing scheme (the dynamic SDR) in non-homogenous holding time traffic, long requests should use the *adaptive* link metric, say, d_j is averaged/filtered over time. While for short requests, they still use the dynamic link metric, which is the most up-to-date d_j. That is why it is called *Differentiated* Shortest Distance Routing.

Hence, to implement Diff-SDR, two parameters must be determined: the sampling rate and the sampling window. The *sampling rate* is the speed at which a link takes samples of its load/occupancy information, and the *sampling window* is the duration of the time over which the samples are kept in memory. The reciprocal of sampling rate is also called *sampling period.*

The estimation of the adaptive d_j is, in fact, based on averaging the samples obtained with a moving window (sampling window). Therefore, the size of the window may impact the routing performance. In the simulation, we vary the sampling rate and sampling window to find out an optimal combination. The results show that the routing performance of Diff-SDR is robust to the selection of the sampling rate and of the sampling window size, unless very poor choices are made.

Suppose the mean holding time of long requests is one time unit. This one time unit can represent various absolute time if we change the holding time ratio. For instance, it can be ($h_L = 10 \times h_S = 50$ seconds, supposing $h_S = 5$ seconds) for a holding time ratio of 10, or ($h_L = 100 \times h_S = 500$ seconds) for a holding time ratio of 100.

As we can see in Figure 6-22, the sampling period and the window size impact the performance of Diff-SDR. Observe that when the sampling window is small, i.e., equal to 0.01 time units, the routing performance in terms of network throughput is unsatisfactory. On the other hand, it is also unsatisfactory when the sampling period is large (sampling rate is small), i.e., with a sampling period of one time unit, the routing performance also deteriorates.

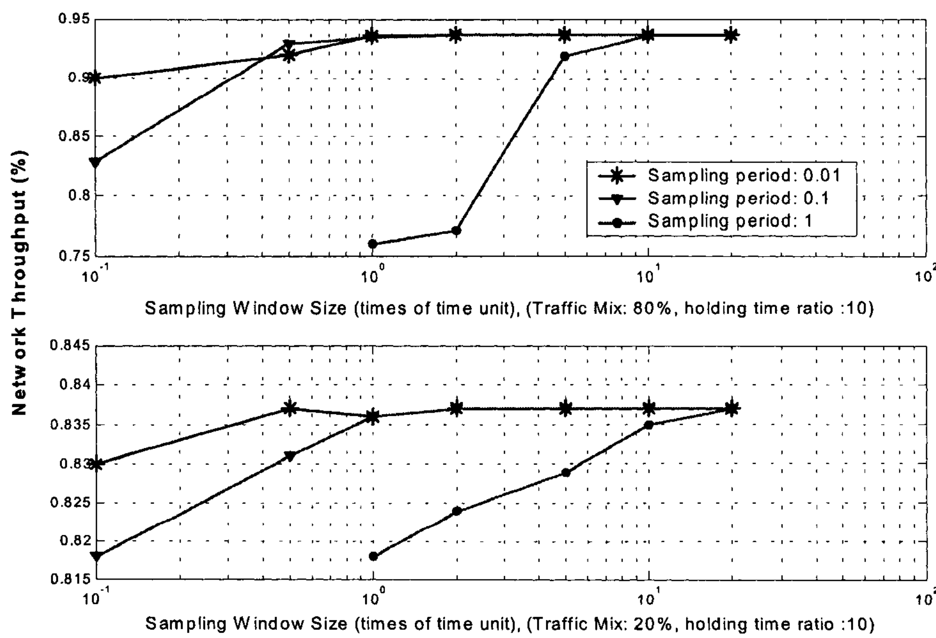

Figure 6-22: Impact of the sampling rate and window. (4-node network, holding time is 10, overload traffic condition.)

However, other than the specific cases described above, the routing performance is robust to the choice of sampling rate and sampling window size. Thus in the sequel we use a sampling period equal to 0.1 time unit and a sampling window size equal to 2.0 time units.

Note that all the above observations and the values set for sampling period and window size are coherent with the study in [17].

6.7.2 Performance Analysis

We then study the performance of Diff-SDR. Again, we use LLR+TR as the baseline for the performance comparison. Figure 6-23 and Figure 6-24 present the outperformance of Diff-SDR compared to LLR+TR in terms of the network throughput for the full range of traffic mix, under nominal and overload traffic conditions respectively.

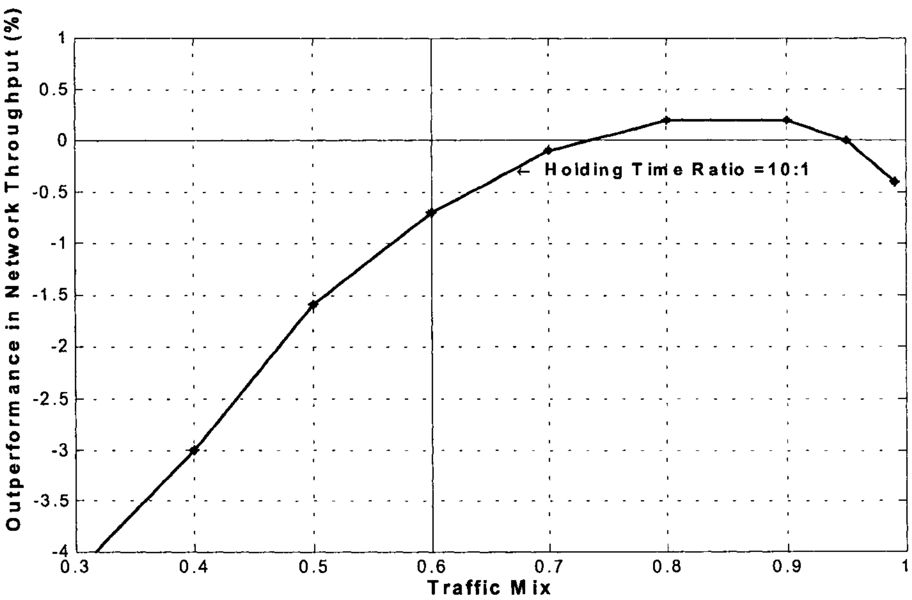

Figure 6-23: Outperformance in terms of Network Throughput of Diff-SDR compared to LLR+TR. Nominal traffic condition, 4-node network. In LLR+TR, TR=0.02. Note that in this figure, it is really too difficult to illustrate all the outperformance curves of Diff-SDR for the various holding time ratio, as they are too close. Hence, we only draw one outperformance curve for a holding time ratio of 10. It represents also the curves with other holding time ratios.

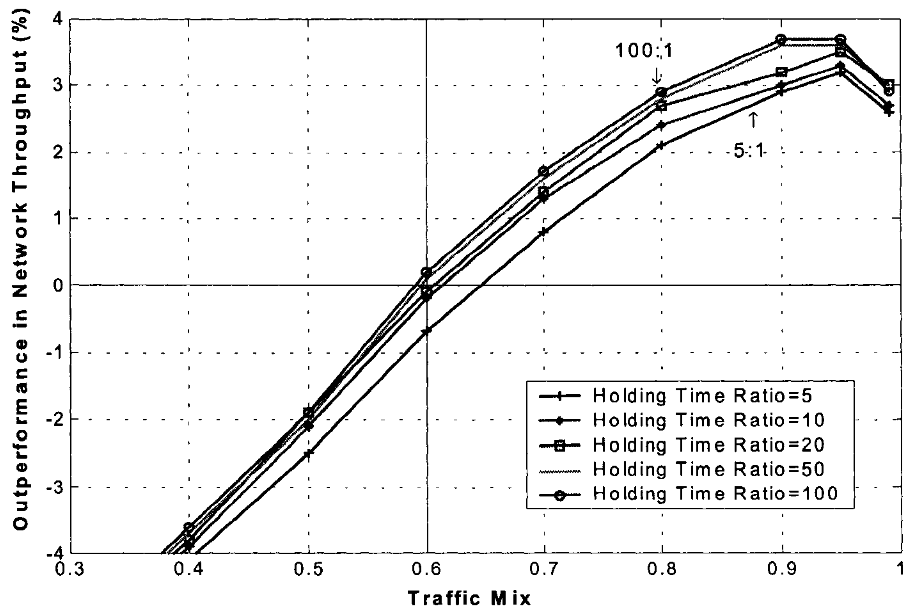

Figure 6-24a: Outperformance in terms of Network Throughput of Diff-SDR compared to LLR+TR. In LLR+TR, TR=0.02. Overload Traffic condition. 4-node network.

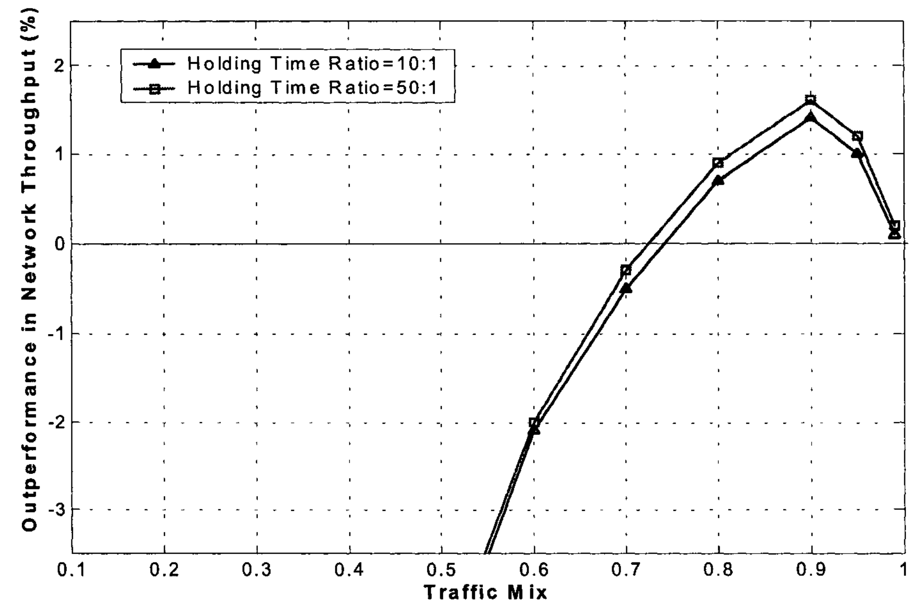

Figure 6-24b: Outperformance in terms of Network Throughput of Diff-SDR compared to LLR+TR. In LLR+TR, TR=0.03. Overload Traffic condition. 12-node network.

As we can see, first, there is also an effective range of traffic mix for Diff-SDR both in nominal and in overload traffic conditions. The effective range is around 70%-99%, which is narrower than the effective range of preventive-vacating. Within its effective range, especially in overload traffic condition, Diff-SDR performs better than LLR+TR in terms of network throughput. Beyond the effective range, the performance of Diff-SDR drops faster than any other scheme we have proposed. However, Diff-SDR still has a general trend in performance similar to that of the preventive-vacating and restricted access routing schemes.

On the effect of the holding time ratio, its influence on the performance of Diff-SDR is not so considerable as with the vacating schemes. In nominal traffic condition, we even cannot distinguish the outperformance curves for different holding time ratio in Diff-SDR (Figure 6-23). In overload traffic condition though, only limited differences can be found among the outperformance curves (Figure 6-24a,b).

On the effect of the TR threshold, there is **no** influence on the performance (network throughput) of Diff-SDR itself. However, as an appropriate higher TR threshold can improve the performance of LLR+TR in overload traffic condition, the outperformance of Diff-SDR compared to LLR+TR is thus inversely proportional to the increase of the TR threshold.

6.7.3 Reason for Diff-SDR's Outperformance

We then try to explain why Diff-SDR outperforms LLR+TR in the effective range. As presented in Table 6-13 (next page), we notice that the blocking rate for long and short requests in Diff-SDR are *not* equal, while in LLR+TR, they are equal! In Diff-SDR routing, long requests have obviously a much higher blocking rate than short requests. (Note that this is just the contrary to the restricted access routing scheme, in which short requests have a much higher blocking rate than long requests). This observation can help us to find out the secret behind the magic.

In Diff-SDR, short requests use a dynamic (real-time) link metric, while long requests adopt an adaptive one. Thus, short and long requests have different blocking scenarios.

The blocking of short requests only occurs when all of the routes (direct and two-hop routes) are completely busy. While for long requests, it is not. As long requests use the time-averaged (adaptive) link metric, the following scenario will occasionally happen. According to the adaptive link metric, assume that a long request should be routed to a route, say *R*. However, at the exact arriving time of this long request, *R* becomes fully busy suddenly due to the traffic dynamics, although route R should be available in average. Then, what should the long request do? It is blocked instantly although there may be some of the routes still available. (We have confirmed this explanation with the author of [17].) Hence, this mechanism (using an adaptive link metric) leads to long requests having a much *higher* possibility of being blocked than short requests.

In addition, we also notice that in Diff-SDR, no TR is applied. This hints us that Diff-SDR, or more precisely, the adaptive link metric, is in fact *a special form of Trunk Reservation*. Using an adaptive link metric restrains long request traffic from accessing the network to a certain extent, thus it fulfills part of the role of Trunk Reservation, which is prohibiting over-alternating, especially when the traffic load is heavy. Therefore, by applying this special form of Trunk Reservation, within its effective range where long requests are the major contribution to the overall traffic, Diff-SDR obtains a better performance.

We can even further verify this claim in two ways:

First, beyond the effective range, short requests will have a higher contribution to the overall traffic than long requests, and then only restraining long request traffic becomes not sufficient, especially when the traffic mix is small. The direct result is that the performance of Diff-SDR decreases very quickly (Table 6-13). Then what would happen if we applied an adaptive link metric to short requests while using a dynamic metric for long requests in such a case? See the following Table.

	4-node Network			12-node Network		
Traffic Mix	Blocking-Long%	Blocking -Short%	Network Throughput	Blocking -Long%	Blocking-Short%	Network Throughput
80%	7.5%	1.3%	93.6%	6.8%	0.4%	94.5%
80%	*16.2%*	*36.6%*	*80.8%*	*15.7%*	*45.5%*	*78.3%*
20%	28.4%	13.4%	83.6%	42.5%	14.0%	80.3%
20%	*1.0%*	*11.2%*	*90.8%*	*0.2%*	*10.7%*	*91.3%*

Table 6-13: Comparison of Diff-SDR with its variation (applying a dynamic link metric to long requests and an adaptive link metric to short requests). The data for the variation is in shadow. The holding time ratio is 10.

The results in Table 6-13 demonstrate the impact of the adaptive link metric on the long request blocking rate, the short request blocking rate and the network throughput. When the traffic mix is 80%, applying an adaptive link metric to long requests is better than applying it to short requests. While when traffic mix becomes 20%, applying an adaptive link metric to short requests, as we expected, has better performance than applying it to long requests.

Second, keeping the traffic mix unchanged at 80%, does Diff-SDR always perform better than the other two SDR, namely Dynamic-SDR and Adaptive-SDR? In [17], this is not very clear. It is only showed that Diff-SDR performs better than the other two in some range of traffic. Here, let us give a thorough comparison in the following Table.

	4-node Network			12-node Network		
SDR	Nominal Traffic	Overload Traffic	Severe Overload	Nominal Traffic	Overload Traffic	Severe Overload
Diff-SDR	99.1%	93.6%	81.0%	99.5%	94.5%	87.8%
Adaptive-SDR	98.2%	92.8%	83.6%	97.5%	93.8%	89.9%
Dynamic-SDR	93.0%	81.9%	70.2%	87.0%	76.6%	70.6%

Table 6-14: Network Throughput comparison of the three SDR schemes. The holding time ratio is 10, and the traffic mix is 80%. The severe overload in the 4-node network is 1.3 times of the nominal traffic load, while in the 12-node network, it is 1.2 times of the nominal traffic load.

The results in Table 6-14 show that:

- Dynamic-SDR always provides the worst performance in terms of network throughput. We ascribe this to the fact that there is no form of trunk reservation in Dynamic-SDR, thus over-alternating happens.
- In nominal and overload traffic conditions, Adaptive-SDR performs worse than Diff-SDR due to its over trunk reservation (applying the adaptive link metric to both long and short requests).
- When the traffic becomes severe overload, the need for trunk reservation becomes *stronger* than ever, only applying trunk reservation to long requests, like what Diff-SDR does, is not sufficient. Thus, as we expected, Adaptive-SDR finally performs better than Diff-SDR.
- The effective range of Diff-SDR in traffic load is much narrower, compared with PVV. (Figure 6-4)

As a brief summary, Diff-SDR can provide an improved performance in network with non-homogenous holding time traffic, compared to Dynamic-SDR, Adaptive-SDR, and LLR+TR. However, its outperformance only happens within the effective range of traffic mix, and within a limited range of traffic loads.

6.8 Multi-TR Thresholds

In a multi-rate traffic environment, one of the most useful applications of the TR is to equalize the blocking rates of the different traffic types with multi-TR thresholds [58]. If only one unique TR threshold is applied, the calls with high bandwidth requirements experience a worse grade of service (in term of blocking rate). In our case, that is to say with non-homogenous holding time traffic, multi-TR thresholds can be used to differentiate the blocking rates for long and short requests, as in the following Table:

	block_L	block_S	Throughput
trL=trS=0.04	7.2%	6.9%	92.9%
trL=0.02, trS=0.04	6.6%	10.2%	91.6%
trL=0.02, trS=0.06	5.2%	10.9%	91.9%
trL=0.02, trS=0.08	4.5%	11.0%	92.2%
trL=0.04, trS=0.02	10.1%	5.6%	92.2%
trL=0.06, trS=0.02	10.0%	4.3%	92.9%
trL=0.08, trS=0.02	9.9%	3.6%	93.1%

Table 6-15: Performance of LLR with multi-TR. The traffic mix is 50%. The holding time ratio is 10, in the 4-node network. trL is the TR threshold for long requests; while trS is the TR threshold for short requests.

Two interesting observations are obtained from Table 6-15:

- The blocking rates of long and short requests can be easily adjusted just by varying the values of the TR thresholds for long requests or short requests, or both.
- By carefully selecting individual TR threshold(s), a little bit of change in the blocking rate (the one with higher value) of one traffic type will result in relatively large change in the blocking rate of the other traffic type. In addition, this has no significant impact on the overall network throughput. The above phenomenon is even more obvious when the blocking rate of the long requests is higher than that of the short requests.

6.9 Implementation Issue

As described in Chapter III, the implementation issue is how to get the detailed traffic mix in practice. The idea is that each link keeps a measurement of the average holding time $\bar{h}$ of requests using the link; then the traffic mix of each link can be computed through equation (3.32). Hence, we can apply the effective range of vacating, so as to implement a practical version of the preventive-vacating routing scheme, denoted by new-PVV.

Very similarly as for estimating the adaptive link metric in the Diff-SDR routing scheme, we also need to determine the sampling rate and the sampling window here so as to measure $\bar{h}$. This is also based on the averaging of samples obtained with a moving window (sampling window).

The simulation results show that the routing performance of new-PVV is robust to the selection of the sampling rate and the sampling window size, unless very poor choices are made.

Suppose the mean holding time of long requests is one time unit, we use a sampling period equal to 0.1 time unit and a sampling window size equal to 2.0 time units in new-PVV, just like in the Diff-SDR case.

We compare the original PVV and new-PVV as follows:

	4-node Network			12-node Network		
Traffic Mix →	90%	80%	40%	90%	80%	40%
PVV	1.2%	1.3%	-1.9%	1.5%	1.3%	-1.4%
New-PVV	1.1%	1.2%	0.0%	1.4%	1.3%	0.0%

Table 6-16: Comparison of the original PVV and new-PVV in outperformance compared to LLR+TR in network throughput. The holding time ratio in the 4-node network is 10, and it is 20 in the 12-node network.

The results in Table 6-16 shows that the performance of new-PVV is satisfactory. Within the effective range of vacating, new-PVV performs very closely to the original PVV. While, outside the effective range, vacating is stopped in new-PVV, thus it acts the same as LLR+TR.

Note that the above observation is very meaningful to practically implement the routing schemes we proposed, e.g., PVV or RAR. From the point of view of the implementation, the major difference between LLR+TR and preventive-vacating is that preventive-vacating needs to distinguish long and short requests. However, this does not make PVV much more difficult to be implemented than LLR+TR. The reason is that the distinguishing (long and short requests) work and further statistics (to compute the traffic mix) work by PVV only operates with the traffic on the direct links. In other words, the originating nodes just need to focus on the links originating from themselves. There is no need for the information of remote nodes or links. All the extra work of PVV compared to LLR+TR can be done locally, thus facilitating the implementation. Hence, even converting LLR+TR to PVV (or RAR) is not so difficult as we imagined.

On the other hand, in this study, we talk about the case in which different types of connections have widely differing holding times, and we assume that this is known by the system. In practice, the holding time may be explicitly negotiated as part of the admission process or the service agreement (being a parameter announced by the traffic source, e.g., a 2-hours videoconference). Or, the holding time may be known implicitly through attributes of the connection such as the protocol and port used. (e.g., a TCP port). We may not know the exact value of each connection, but it is possible to know the mean value of a category, which is based on long-term statistical measurements.

Furthermore, the great diversity of emerging services in modern IP networks make it possible to categorize the traffic according to either the bandwidth requirements or the holding times. Actually, the categorizing work can be done only by the routing control part of the networks and would be based on the long-term statistics of the traffic. It is not necessary for final customers to know this. For instance, an IP call and an IP conference can be put in two different categories, because their average holding times are different widely according to long-term statistics. They are distinguished only by routing control program of the systems, and then different routing procedures should be applied on them automatically.

We use a factor of 10 to distance the average holding time. This is because we want to show that the holding times of long and short requests are different at least one order of magnitude. From our simulation results, it is known that this is the necessary condition for our vacating routing scheme being effective. And the larger the difference, the bigger outperformance of our vacating schemes compared to LLR+TR will be.

6.10 Summary

At the end of this chapter, we give an overall comparison of all the routing schemes we have proposed for network with non-homogenous holding time traffic. Similarly, LLR+TR is still used as baseline for the comparison. The outperformances compared to LLR+TR in network throughput of all the proposed schemes are presented in Figure 6-25a for the 4-node network and Figure 6-25b for the 12-node network, respectively.

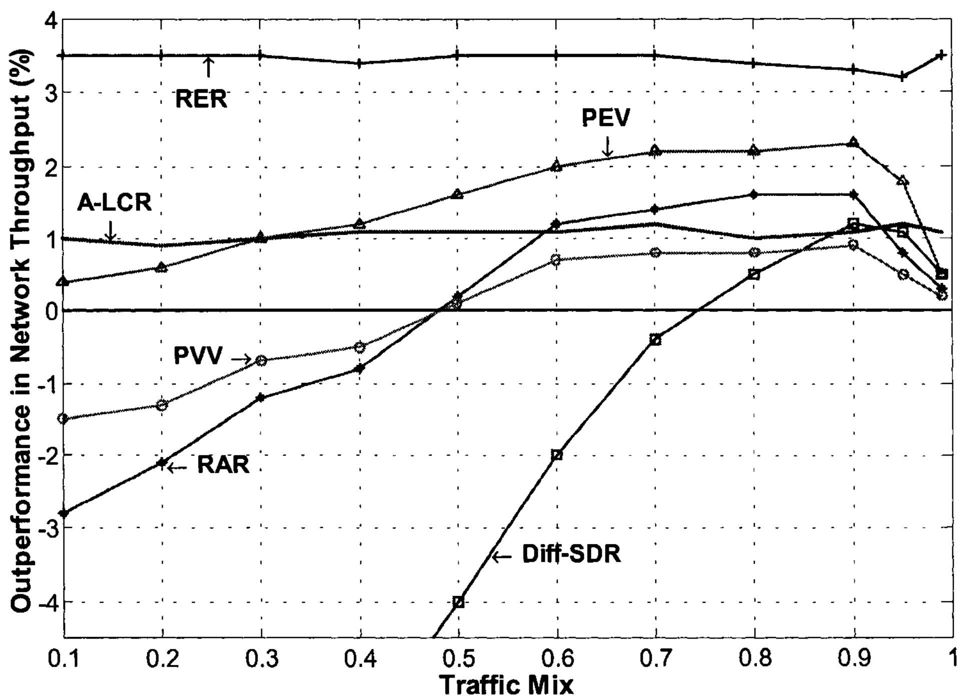

Figure 6-25a: Outperformance (in network throughput) of all the proposed routing schemes compared to LLR+TR in the **4-node** network. The holding time ratio is 10. Overload traffic condition. The TR threshold in LLR+TR, PVV, PEV, and RAR is 0.04. The vacating threshold in PVV is 0.02. The Restricted Access Threshold in RAR is 0.02.

Legend:			
RER:	Re-routing	PVV:	Preventive-vacating routing
PEV:	Preemptive-vacating	A-LCR:	Approximated Least Cost Routing
RAR:	Restricted Access Routing	Diff-SDR:	Differentiated Shortest Distance Routing

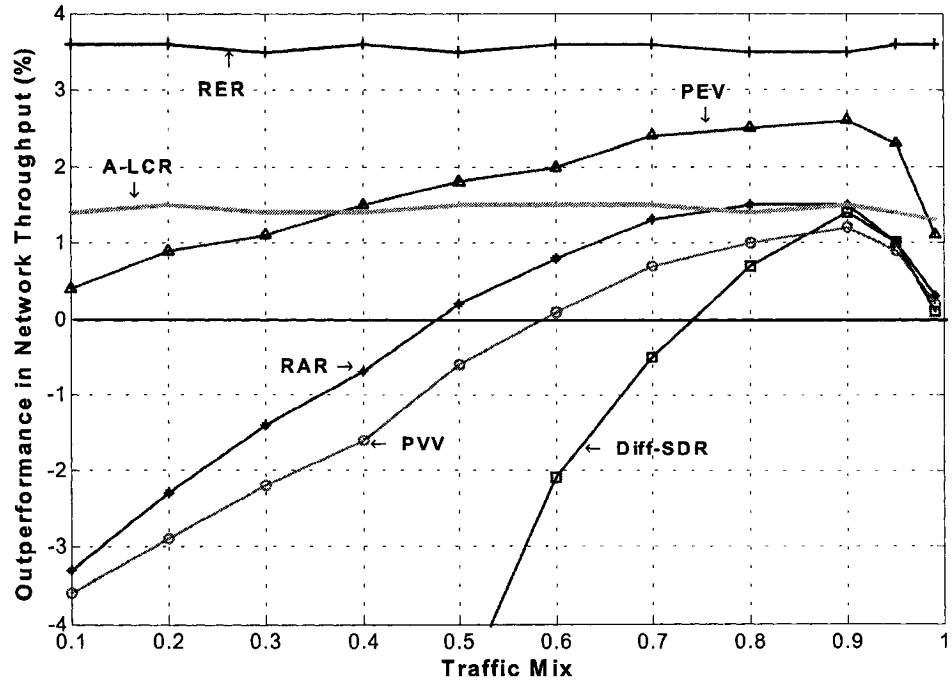

Figure 6-25b: Outperformance (in network throughput) of all the proposed routing schemes compared to LLR+TR in the **12-node** network. The holding time ratio is 10. Overload traffic condition. The TR threshold in LLR+TR, PVV, PEV, and RAR is 0.03. The vacating threshold in PVV is 0.02. The Restricted Access Threshold in RAR is 0.02.

If the network throughput is the only metric to evaluate the performance of the routing schemes, then it is obvious from Figure 6-25a,b that the re-routing scheme is the best of the best. However, there are still some other factors to be considered in comparing the routing schemes. Thus, a further comparison is given in the following Table.

Item	PVV	PEV	A-LCR	RER	RAR	Diff-SDR
Network Throughput	No.5	No.2	No.3	No.1	No.4	No.6
Fairness	Good	Good	Good	Good	Bad	Bad
Effective Range	Medium	Full range	Full range	Full range	Medium	Narrow
Holding Time Ratio ↑	↑	↑	—	—	↑	—
TR ↑ (impact on performance)	↑	↑↓*	—	↓	↑	—
TR ↑ (impact on outperformance)	↓	↓	↓	↓	↓	↓
Implementation	Easy	Not easy	Medium	Not easy	Easy	Medium

Table 6-17: Thorough comparison of all the routing schemes. Note 1: "—" means there is no impact. Note 2: regarding the impact of TR ↑ on the performance of PEV, within the effective of vacating, increasing the TR degrades its performance slightly; while outside the effective range of vacating, increasing the TR improves its performance. Thus, it is shown as "↑↓" in the table.

Besides, there are still five points needing further clarification:

The first is that all the simulation results in the 4-node and the 12-node networks are in accordance with each other.

The second is about the re-routing scheme. In all the simulations, the re-routing scheme always provides a performance (in network throughput) superior to that of any other routing scheme. It is thus believed that the network throughput achieved by the re-routing scheme can be viewed as an upper bound for the other routing schemes. The other schemes can approach this limit, but not exceed it. For instance, within the effective range of vacating, say [0.60, 0.99] in the traffic mix, through increasing the holding time ratio, the upper bound in network throughput can be approached by:

- Preventive-vacating Routing,
- Restricted Access Routing, and
- Preemptive-vacating Routing.

The third is about the effective range of preventive-vacating. It is interesting to summarize that within this range in the traffic mix, the followings happen:

- Preventive-vacating is effective;
- Preemptive-vacating achieves maximal values, and it is not sensitive to the change of the TR threshold;
- Restricted access routing is effective;
- Diff-SDR is effective in a smaller range.

The fourth is about the A-LCR scheme. A-LCR shows its good performing in both the network throughput (almost constant value) and the fairness of routing. Furthermore, its particular and inherent flow control mechanism can achieve several network management objectives. Thus, although RAR outperforms A-LCR in some specific traffic mix (Figure 6-25a,b), we still rank A-LCR as No.3 in Table 6-17, right after the re-routing and preemptive-vacating schemes.

The fifth is about the Diff-SDR scheme. As seen in Figure 6-25a,b, Diff-SDR has a narrower effective range compared with preventive-vacating (PVV) and restricted access routing (RAR). However, in a certain even narrower range in traffic mix, say [0.85, 0.95] for preventive-vacating and [0.90,0.95] for restricted access routing, Diff-SDR outperforms these two schemes slightly. However, we notice that the holding time ratio in Figure 6-25 is set as 10. As observed earlier, when the holding time ratio increases, both the performance of PVV and RAR increases as well; while for Diff-SDR, it is almost unchanged. Hence, there is a possibility that the two routing schemes we proposed become better than Diff-SDR in their entire effective range. In fact, we find that this happens when:

- the holding time ratio is greater than 50 for PVV and,
- the holding time ratio is greater than 20 for RAR.

Furthermore, beside the narrower effective range in traffic mix compared to PVV and RAR, Diff-SDR also has a much narrower effective range in traffic load. Therefore, in Table 6-17 (Network Throughput row), we put Diff-SDR at the bottom of the six routing schemes.

Chapter VII: Results and Discussions of Inter-network Routing with Gateway Selection

This chapter presents the performance comparison of inter-network routing with several gateway selection approaches proposed in Chapter IV.

7.1 Network Set-up

The network used in the simulation of this chapter is defined by following elements:

Item	Value
Overall number of nodes	30
Number of one-way intra-net links	412
Basic Bandwidth Unit (BBU)	20kB
Overall Capacity of Local Network	11k BBU
Connection ratio	50%
Symmetrical	No
Number of Foreign Networks	3
Number of Gateway Nodes	6 (2 for each foreign network)
Capacity of each inter-network link (one-way)	700 BBU
Nominal Overall Traffic (Erlang)	7.5k Erlang
Percentage of Outgoing Inter-Network Traffic	43%
Mean holding time of flows	5 seconds

Table 7-1: Description of the 30-node Network

Regarding the computation of the cost in the Theoretical Resource Utilization Cost Approach we proposed, we use the same method as the one adopted in simulating Approximated Least Cost Routing (A-LCR) in Chapter VI.

7.2 Simulation Results and Discussion

All the simulation trials were run with the traffic in three time patterns: 10h00, 12h00 and 13h00. The final simulation curves (in Figure 7-1) are generated by averaging the results obtained from the three time patterns of traffic.

For each simulation scenario, the duration of the simulation is adjusted to be long enough to ensure the 95% confidence of results is always ± 0.1%. We present the performance comparison of the gateway selection approaches we proposed in Chapter IV as follows:

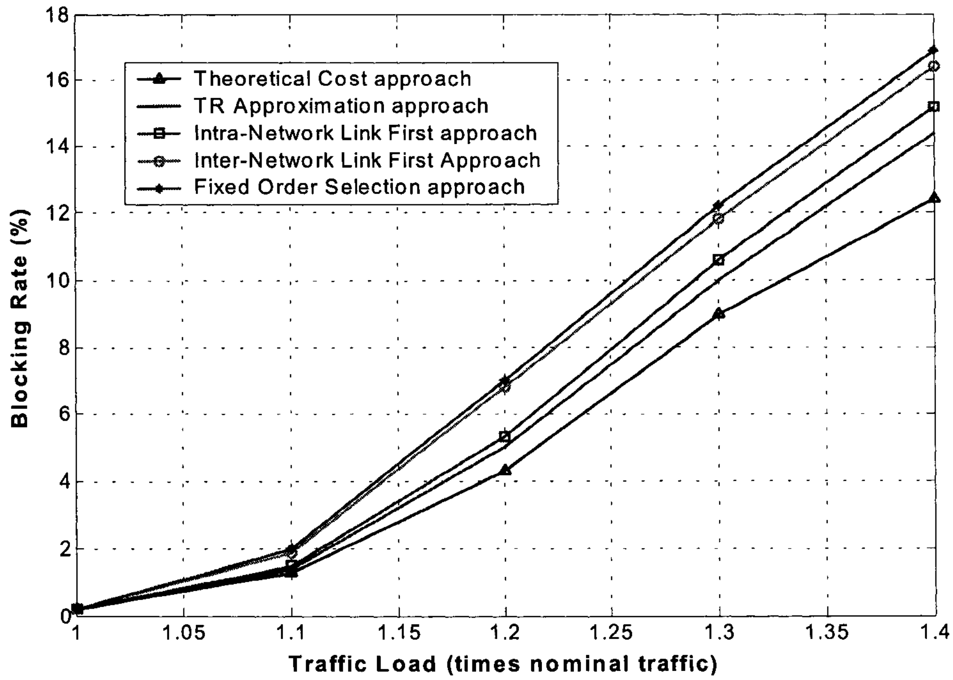

Figure 7-1: Performance Comparison (in overall blocking rate) of the different gateway selection approaches vs. the traffic load

7.2.1 Theoretical Resource Utilization Cost Approach

As shown in Figure 7-1, it is obvious that the gateway selection approach using the Theoretical Resource Utilization Cost yields, in general, a performance (in terms of blocking rate) superior to all the other approaches. The performance difference becomes more substantial when operating under heavy load condition. This can be explained by the following considerations:

- This approach uses (equation 2.1) $\frac{E_b(A,N)}{E_b(A,p)}$ to compute the resource utilization cost, and $\frac{E_b(A,N)}{E_b(A,p)}$ is *adaptive* to the traffic (through A) and to the detailed resource utilization (through p). Thus, while the traffic changes from a light to a heavy load, the routing decision for each request is still optimal (or near optimal). Or, we can say that this approach adopts a *dynamic* TR threshold, in which more bandwidth is reserved to protect the direct routed flows under heavy traffic, while a lower trunk reservation level is used under light traffic conditions.

- For gateway selection, both the information in the two legs of an inter-network route is used to make a decision. The effect of accepting an inter-network call on future incoming calls on each link of the route is reflected through the theoretical resource utilization cost. The gateway node with minimal overall impact on future incoming calls is selected. Thus, this approach selects a gateway node based on an overall view of resource utilization and an integrated impact on future calls due to accepting an inter-network call. Hence, it performs best.

7.2.2 Trunk Reservation Approximation Approach

As an approximation of the above Theoretical Resource Utilization Cost Approach, the Trunk Reservation approach provides the second best performance. It also takes into account the overall information and the integrated impact when making gateway selection decisions.

But this approach uses a fixed TR level (5%). Therefore, when the traffic load is heavy, the performance of this approach becomes worse than the first approach.

7.2.3 Fixed Order Selection Approach

The Fixed Order Selection approach provides the worst performance among all the gateway selection approaches we proposed. This is not strange, as in the fixed order selection approach no detailed overall information on the two legs of an inter-network route is considered, as it is in the above two approaches. Gateways are selected according to a predetermined sequence, and the dynamics of the traffic load both in intra-network and inter-network links are not considered. This is illustrated in Figure 7-2.

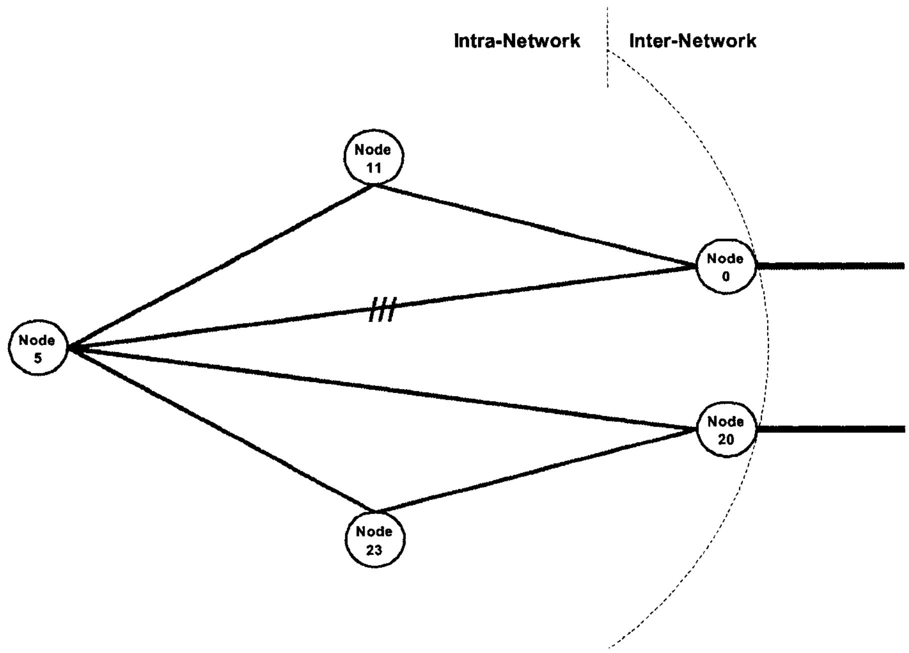

Figure 7-2: One of Network Scenarios

In Figure 7-2, suppose an inter-network request originating from Node 5 wants to reach a Foreign Network. Suppose also that only gateway node 0 and 20 are available. The direct link 5,0 is fully busy and all the other links in the figure are still available ($link_{i,j}$ is defined as the link from Node i to Node j) and have the same link occupancy.

Then, it is obvious that the inter-network route from node 5 to gateway 20 and then to the foreign network is the best choice in such a scenario, as this choice has the lowest overall resource utilization. However, according to the fixed order selection approach, this inter-network request must select the three-hop route: from node 5 to node 11 to gateway 0 and to the foreign network. This is because the gateway selection sequence is predetermined and gateway node 0 is on the top of the list.

Clearly, the fixed order selection approach cannot balance the traffic as efficiently as the other approaches, especially when the traffic becomes heavy. And thus, it has resulted in inefficient link capacity utilization and therefore performance degradation.

7.2.4 Inter-Network Link First vs. Fixed Order Selection Approach

Compared to the Fixed Order Selection Approach, the Inter-Network Link First approach uses the real-time inter-network link occupancy to select gateway node for a multidestination inter-network call. For the two legs of an inter-network route, the inter-network one is considered first. As can be seen in Figure 7-1, the performance of this approach is slightly better than the Fixed Order approach. Note that the same phenomenon has also been mentioned in [66].

This is due partially to the fact that our fixed order selection is not a pure static one. In other words, crankback is allowed. For instance, suppose the gateway selection order is 0/1/2. If the gateway node 0 is busy, then the gateway 1 is attempted; if the gateway 1 is also busy, then the gateway 2 is attempted. As the traffic is dynamic, which gateway node is busy or not is also dynamic. Thus, the fixed order is transformed into a dynamic fashion by the dynamic nature of the traffic to a certain extent. Furthermore, both of these two gateway selection approaches pay less attention on the resource utilization on the intra-network links.

7.2.5 Intra-Network Link First vs. TR Approximation Approach

We then investigate the performance of the Intra-Network Link First approach. As shown in Figure 7-1, the Intra-Network Link First approach performs worse than the TR Approximation Approach, but much better than the Inter-Network Link First Approach.

The Intra-Network Link First approach focuses on the intra-network leg of an inter-network route. The detailed link occupancy of inter-network links, unless they are fully busy, is not considered when choosing a gateway node. Basically, the resource utilization of the intra-network link(s) has the major influence on choosing a gateway. In other words, the Intra-Network Link First approach pays more attention to balancing the traffic loads within the local network.

While for the TR Approximation Approach, the link occupancy information on both the legs of an inter-network route is integrated through using the concept of the resource utilization cost. Thus, a general and overall view is used when making a gateway selection decision. Hence, the TR Approximation Approach, compared with the Intra-Network Link First Approach, obtains better performance in blocking rate. We also notice that the difference in the blocking rate of these two approaches is not large. Let us see what will happen when these two approaches make different gateway selections for a same network scenario. This is presented in the following Figure 7-3.

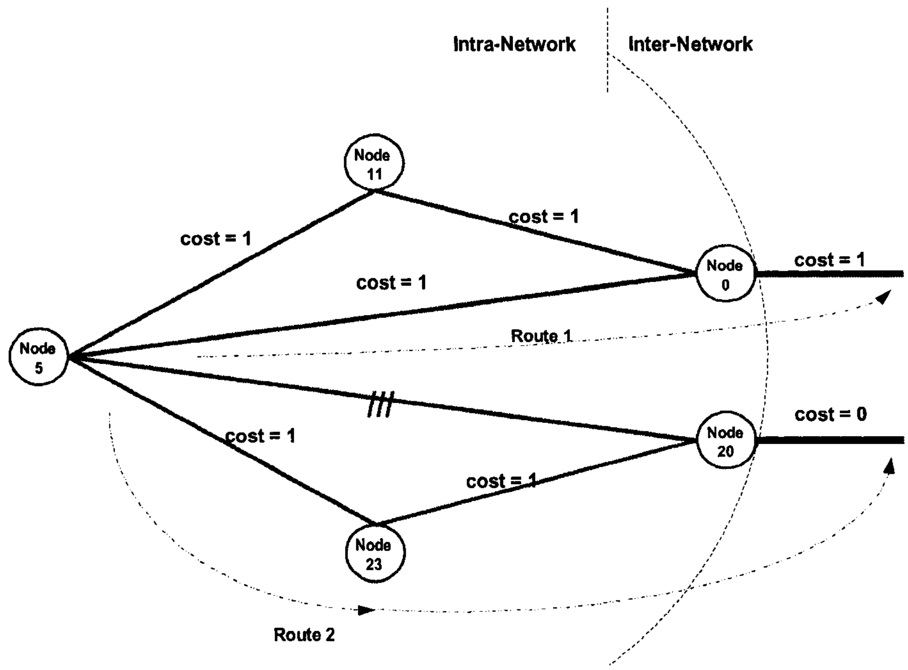

Figure 7-3: A Network Scenario

In Figure 7-3, still suppose an inter-network request originating from Node *5* needs to reach a Foreign Network. Suppose also that only gateway node 0 and 20 are available now. The direct $link_{5,20}$ is busy and all the other links in the figure are still available.

Given this situation, the two candidates for an outgoing inter-network call are: *route 1* ($link_{5,0}$ + inter-network link), and *route 2* ($link_{5,23}$ + $link_{23,20}$ + inter-network link). According to the Intra-Network First approach, *route 1* (least intra-network cost) will be selected, while according to the TR Approximation approach, there is a *50%* (only two gateway available) probability that *route 2* will be selected. This is because the overall cost for both candidates is the same value, which is 2.

Continuing in this scenario, when the cost of $link_{23,20}$ is changed to zero, then according to the TR Approximation approach, *route 2* will be selected (least overall cost). According to the Intra-Network First approach, however, *route 1* will be selected, since *route 1* has a lower hop count (2) even though both candidates have the same overall cost.

From the simulation results, the two cases described above seem rare and do not occur often enough to generate a very significant performance difference in our traffic and network environment.

We also observe that the Intra-Network Link First approach outperforms the Inter-Network Link First approach by a significant margin. Thus, in general, it can be concluded as follows:

1) Regarding the two legs of an inter-network route, the intra-network one is the most important when making a gateway selection decision.
2) Integrating the resource utilization information from the two legs to select a gateway node achieves better performance than only considering the information of one leg.

We have already discussed the second point previously, but regarding the first point above, a question appears: from the point of view of the resource utilization cost concept, is accepting a call on an inter-network link exactly the *same* as accepting a call on an intra-network link?

They are similar of course, however, there is still a little difference. As defined in the system model (Table 2-1), there are in fact three kinds of calls within the intra-network scope:

1) Outgoing Inter-network Calls, whose origination node is a node in the local network, while the destination is a foreign network;
2) Incoming Calls, whose origination is one of the gateway nodes, while the destination is a node in local network;
3) Local Calls, whose origination and destination nodes are any node in local network.

Accepting a call on an inter-network link only affects future outgoing inter-network calls, which, according to our simulation parameters, represent 43% of the overall traffic. However, when accepting a call on any intra-network link all the three types of future incoming calls will be influenced.

Chapter VIII: Conclusions and Future Work

The major goal of this thesis is to study the following routing issue: how can we consider the large heterogeneity of holding times among traffic flows when routing in a MPLS-based IP network.

The concept of the resource utilization cost and its successful application in the state-dependent routing in telephone networks hint that the same concepts can be applied to MPLS-based IP networks, so as to find a solution to our routing issue. Through introducing the "cost rate", the resource utilization cost of accepting a long or a short request can be expressed as the product of the cost rate and the request's holding time. In other words, the cost induced by a long or short request is roughly proportional to its holding time. Based on this, an analytical framework is set up and our basic idea of "vacating" is analyzed thoroughly. As a result, several routing schemes in networks with non-homogenous holding time traffic are proposed, they are:

- Preventive-vacating Routing (PVV);
- Preemptive-vacating Routing (PEV);
- Approximated Least Cost Routing (A-LCR) and
- Restricted Access Routing (RAR).

Through the simulations in NS2, we found that the results of our analysis and experiments matched very well. They proved the correctness as well as the exactness of our analytical model.

In addition, the performance of the re-routing scheme was investigated in networks with non-homogenous holding time traffic. We also compared our proposed routing schemes with the Differentiated Shortest Distance Routing (Diff-SDR), which is the only currently published dynamic routing scheme addressing the question of heterogeneous holding times.

In general, the conclusions and observations obtained are:

- Just as in the homogenous traffic environment, the re-routing scheme still has a constant and superior performance (in terms of network throughput) in a non-homogenous holding time traffic environment than all other routing schemes we investigated. Varying the holding time ratio or traffic mix has almost no impact on the performance of the re-routing scheme. However, applying TR results in performance-degradation when using re-routing.

- Similarly, approximated least cost routing (A-LCR) also has constant superior performance (although not as good as re-routing) in a non-homogenous traffic environment. We can thus deduce that the LLR+TR scheme can still be efficient provided that the associated TR threshold is adaptive strictly to the traffic load. Moreover, the A-LCR scheme has an inherent flow control mechanism, which, through simply adjusting the values of the reward parameters, enables the control of the Grade of Service (blocking rate) of the long and short requests over a very wide range. Among all the routing schemes we proposed, only A-LCR has this special capability.
- We found that the effective-range of vacating, from both the analytical and simulation results, is around [0.60, 0.99] in the traffic mix.
- Within the effective-range of vacating, both the PVV and RAR schemes perform better (in terms of network throughput) than LLR+TR. While beyond the effective range, they do not.
- Although the preemptive-vacating scheme out-performs LLR+TR in the full range of traffic mix, the maximal value of outperformance occurs within the effective-range of vacating.
- The Diff-SDR scheme works better than LLR+TR, but only in much narrower effective ranges both in the traffic mix and in the traffic load. We found that the outperformance of Diff-SDR, in fact, results from the implementation of a TR mechanism in a special form.
- In general, the outperformance of PVV and RAR within the effective range is directly proportional to the holding time ratio. While for PEV, and especially for Diff-SDR, varying the holding time ratio does not have significant influence on their performance.
- On the effect of TR, the outperformance of PVV, PEV, RAR, and Diff-SDR compared to LLR+TR is inversely proportional to the increase of the TR threshold.

- Generally, the performance (network throughput) of PVV, PEV, RAR and Diff-SDR in networks with non-homogenous holding time traffic can be ranked as follows: PEV > RAR > PVV > Diff-SDR.
- However, in order to obtain a better performance, the RAR and Diff-SDR schemes sacrifice the fairness of routing. In RAR, the blocking rate of short requests is much higher than that of long requests; while in Diff-SDR, it is exactly the contrary.
- Multi-TR thresholds could be used to differentiate the blocking rates of long and short requests in networks with non-homogenous holding time traffic.
- A practical method is introduced to measure the real-time traffic mix. This method can be used to measure when the traffic mix is in the effective range, and for then triggering the vacating schemes. The simulation results are satisfactory.

Besides the above discussion of routing within the scope of one network and non-homogenous traffic, the second routing issue in this study is about routing across networks, that is to say, inter-network traffic (homogeneous). This routing issue focused on how to select the best among several gateway nodes to a foreign network.

Actually, an inter-network route can generally be partitioned into a first intra-network leg, linking the origin to the gateway node serving as exit point in the local network, and a second leg linking the gateway node to the foreign network access point. Several gateway selection schemes are proposed, and studied in a 30-node network. We then found that:

- Among all the proposed gateway selection schemes, the one that selects gateway nodes based on an overall view of resource utilization of the two legs and an integrated impact on the future calls performs best. This means that the optimal gateway selection scheme should base on the integrated overall view of resource utilization and induced costs along the whole inter-network route.
- Between the two legs of an inter-network route, the intra-network one is the most important when making gateway selection decision.
- Gateway selection according to a predetermined order (with crankback allowed) performs worse among all the schemes we proposed, which is reasonable.

Regarding the future work, further investigation should be oriented primarily towards the expansion of the research beyond the scope considered in this study. For instance, the routing schemes we proposed need to be generalized for a network where the holding time of traffic flows span 1000/100/10/1 time units instead of only long and short requests. This is based on the fact that holding times in a practical IP network may varies from days (e.g., VPN), to hours (videoconference), to minutes (VoIP call), to seconds (HTTP). Then, how to select the appropriate route for different flows?

In general, vacating is still the basic idea. The requests should vacate the bandwidth on the direct routes to all the requests whose holding times are at least one order of magnitude longer. There are two possible ways to implement this:

- Multi-threshold. For instance, each type of requests has its own individual vacating-threshold for PVV (or restricted-access-threshold for RAR).
- One threshold but multi-vacating-probability. For instance, each type of requests has its own probability to vacate when the resource utilization on the direct routes reaches the only vacating threshold.

Both the above two ways indicate that the requests with various holding times have different "priority" to vacate.

Next, it might prove interesting to study the different impacts of holding time (heterogeneous) and bandwidth (heterogeneous) on the cost of accepting a request. For instance, suppose two requests: one with one time unit holding time and 50 units bandwidth requirement, the other with 50 time units holding time but one unit bandwidth requirement. These two requests generate the same reward since the reward of accepting a request, e.g. k, is generally computed as b_k / μ_k (where b_k is the bandwidth and $1/\mu_k$ is the mean holding time of the request k) [41,42,63]. However, will these two requests induce the same cost? What is the relationship between the values of the costs induced by these two "equal-reward" requests? Is the expression (b_k / μ_k) to evaluate the reward of accepting a request always appropriate?

Moreover, it is necessary to consider the practical implementation of the routing schemes we proposed. For example, based on the spirit and framework of DCR (dynamic controlled routing), we may include our vacating schemes in a new version of DCR that works not only in an IP network, but also in an IP network with non-homogenous holding time traffic. Similarly, a practical version of A-LCR is also worthy of further consideration.

Furthermore, we know that GMPLS (Generalized Multiprotocol Label Switching) will be an integral part of deploying the next generation of data networks. It provides the necessary bridges between the IP and photonic layers to allow for interoperable and scalable parallel growth in the IP and photonic dimensions. As a generalization of MPLS, GMPLS applies the MPLS concepts at different levels (switching, optical). Inherently, these different levels operate on very different time scales. Therefore, the results in this study, potentially, could apply there as well. Of course, this would require further investigation, which can also be part of the future work.

References

[1] Girard, Routing and Dimensioning in Circuit-switched Networks, Addison-Wesley,

[2] F.P. Kelly, Loss Networks, Ann. Appl. Prob. Vol. 1, pp.319-378.

[3] Gerald R. Ash, Dynamic Routing in Telecommunications Networks, McGraw-Hill.

[4] A. Eriksson, Performance of a Telecommunications System Using Trunk Reservation, MSc thesis submitted to the Royal Institute of Technology, Stockholm, Sweden, 1997.

[5] A.Girard and S.Hurtubise, "Dynamic routing and call repacking in circuit-switched networks", IEEE Trans. Commun., vol. COM-41, pp. 290–1294,Dec.1983.

[6] Tam, I.M.-C.; Farber, D.J., Call repacking and trunk reservation, Proceedings of IEEE Singapore International Conference on Networks, Volume: 2 , 6-11 Sep 1993.

[7] D. Mitra and R.J. Gibbens, State-dependent Routing on Symmetric Loss Networks with Trunk Reservations II: Asymptotic, 1992.

[8] D. Mitra and R.J. Gibbens, and B.D. Huang, State-dependent Routing on Symmetric Loss Networks with Trunk Reservations I, IEEE Trans on Commun., February, 1993.

[9] R. Guerin, A. Orda, and D. Williams, QoS Routing Mechanism and OSPF Extensions. IETF Internet Draft <draft-guerin-qos-routing-ospf-00.txt>, November 1996.

[10] A. Shaikh, J. Rexford, K. Shin, Load-sensitive routing of long-lived IP flows, ACM SIGCOMM '99, Cambridge, Massachusetts, USA, pp. 215– 226, august 1999.

[11] G. R. Ash, R. H. Cardwell and R. P. Murray, Design and Optimization of Networks with Dynamic Routing, Bell Sys. Tech. Journal, Vol. 60, pp. 1787-1820, 1981.

[12] G. R. Ash, J.-S. Chen, A. E. Frey and B.-D. Huang, Real-time Network Routing in a Dynamic Class-of-Service Network, 13th Intl. Teletraffic Congress, Denmark, 1991.

[13] Howard, R., Dynamic Programming and Markov processes. Wiley, New York, 1960.

[14] R. J. Gibbens, F. P. Kelly and P. B. Key, Dynamic Alternate Routing —Modeling and Behavior, 12th International Teletraffic Congress, Torino, Italy, 1988.

[15] Q. Ma, P. Steenkiste, and H. Zhang, Routing High-Bandwidth Traffic in Max-Min Fair Share Networks. In ACM SIGCOMM96, page 206-217, Stanford, CA, 1996.

[16] A.Yamashita, R.Kawamura, and H.Hadama, "Dynamic VP rearrangement in an ATM network", in Proc. IEEE Globecom'95, 1995, pp. 1379–1383.

[17] Yang, S.; Su, X.; De Veciana, G., Heterogeneity-aware Shortest Path Routing: flow holding time, user demand and network state, 2001 IEEE Workshop on High Performance Switching and Routing, 2001; Page(s): 287 -291

[18] Q. Ma and P. Steenkiste, On path selection for traffic with bandwidth guarantees, IEEE INFOCIM98, 1998.

[19] Liu, R.P.; Moylan, P.J., Dynamic Trunk Reservation for Teletraffic Links, Global Telecommunications Conference, 1995. GLOBECOM '95.

[20] Kelly, F. P., Routing in Circuit-Switched Networks: optimization, shadow prices and decentralization, Advances in Applied Probability 20 (1988), 112-144.

[21] Juttner, A.; On-demand optimization of label switched paths in MPLS networks, Ninth International Conference on Computer Communications and Networks, 2000.

[22] Ott, T., and Krishnan, K., State-dependent Routing of Telephone Traffic and The Use of Separable Routing Schemes. In Proc. 11th International Teletraffic Congress (Kyoto, Japan, 1985).

[23] Krishnan, K. and T.J. Ott, State-dependent Routing for Telephone Traffic: theory and results, Proc. Of 25th IEEE Conference on Decision and Control, Athens, Greece, December, 1986

[24] Lazarev W.G., Starobinets S.M., The Use of Dynamic Programming or Optimization of Control in Networks of Commutations of Channels. Engineering Cybernetics (Academy of Sciences, USSR), No.3, 1977

[25] G R. Ash, R. H. Cardwell, and R. P. Murray, Design and Optimization of Networks with Dynamic Routing, Bellsys. and Tech. J., vol. 60, no. 8, Oct. 1981.

[26] Dziong, Z.; Mason, L.G., Call Admission and Routing in Multi-service Loss networks, IEEE Transactions on Communications, Volume: 42 Issue: 2, February/March/April 1994, Page(s): 2011 –2022

[27] Chatzaki, M., Sartzetakis, S., Papadakis, N., Courcoubetis, C., Resource Allocation in Multi-service MPLS, IWQoS '99. 1999 Seventh International Workshop on, 1999.

[28] V.B.Iversen, "Traffic engineering of cellular wireless systems,"_Network Inform. Processing Sysem, Oct. 1997.

[29] F. P. Kelly, Notes on effective bandwidths, In F. P. Kelly, S. Zachary, and I. Zeidins, editors, Stochastic Networks: Theory and Applications, pages 141-168, Oxford University Press, 1996.

[30] Kolarov, A. and Hui, J., On Computing Markov Decision Theory-based Cost for Routing in Circuit-Switched Broadband Networks, In Journal of Network and Systems Management, vol.3, no.4, 1995, pp.405-426.

[31] Krishnan, K. R. and Hibner-Szabo de Buts, F., Admission Control and State-Dependent Routing for Multirate Circuit-Switched Traffic, In Proceedings of the 15th

International T&traffic Congress, Washington D.C., USA, 1997.

[32] Ahlfors, U., Korner, U., and Pioro, M., Towards Efficient Call Admission Control for State-dependent Routing in Multi-rate Networks, 1999. Proceedings of the 1999 IEEE International Conference on Control Applications, Volume: 2, 1999.

[33] Montgomery, M., De Veciana, G., Hierarchical source routing through clouds, INFOCOM '98.

[34] K.C. Lee and V.O.K. Li, "A circuit rerouting algorithm for all-optical wide-area networks", in Proc. IEEE Infocom, 1994, pp.954–961.

[35] R.Cohen, "Smooth intentional rerouting and its applications in ATM networks", in Proc. IEEE Infocom, 1994, pp. 1490–1497.

[36] Ren-Hung Hwang; Kurose, J.F.; Towsley, D., MDP Routing in ATM Networks Using Virtual Path Concept, INFOCOM '94.

[37] Z. Dziong et al., On Adaptive Call Routing Strategy for Circuit Switched Networks – Maximum Reward Approach, in Twelfth International Teletraffic Congress, 1988.

[38] Girard, A., Revenue Optimization of Telecommunication Networks, IEEE Transactions on Communications, Volume: 41 Issue: 4, April 1993.

[39] Girard, A.; Zidane, R., Revenue Optimization of B-ISDN Networks, IEEE Transactions on Communications, Volume: 43 Issue: 5, May 1995.

[40] Dziong, Z.; Mason, L., An Analysis of Near Optimal Call Admission and Routing Model for Multi-service Loss Networks, In Proceedings of IEEE INFOCOM'92 (1992), pp. 141-152.

[41] Dziong, Z.; Mason, L., Control of Multi-service loss networks, Proceedings of the 28th IEEE Conference on Decision and Control, 1989, Page(s): 1099 -1104 vol.2

[42] Kolarov, A.; Hui, J., Least Cost Routing in Multiple-service Networks, INFOCOM '95. Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies. 'Bringing Information to People', 1995.

[43] Bestavros, A.; Matta, I., Load Profiling for Efficient Route Selection in Multi-class Networks, International Conference on Network Protocols, 1997, Proceedings, 1997.

[44] Matta, I.; Krunz, M, Packing and Least-loaded Based Routing in Multi-rate Loss Networks, 1997 IEEE International Conference on Communications, Montreal, Towards the Knowledge Millennium, 1997, Volume: 2, Page(s): 827 -831 vol.2

[45] Regnier, J.; Cameron, W.H., State-dependent Dynamic Traffic Management for Telephone Networks, IEEE Communications Magazine, Volume: 28, Oct. 1990.

[46] E. E. M. Wong, T.-S.P. Yum, and A. K. Chan, A Taxonomy of Re-routing in Circuit-

switched Networks, IEEE Commun. Mag., vol. 37, pp. 116-122, Nov. 1999.

[47] E. E. M. Wong, A. K. Chan, and T.-S.P. Yum, Analysis of Rerouting in Circuit-switched Networks, IEEE/ACM Transactions on Networking, , June, 2000.

[48] R. Braden et al., Integrated Services in the Internet Architecture: an Overview, RFC 1633

[49] S. Blake et al., An Architecture for Differentiated Services, draft-ietf-diffserv-arch-02.txt

[50] E. Rosen et al., Multiprotocol Label Switching Architecture, draft-ietf-mpls-arch-02.txt.

[51] E. Crawl et al., A Framework for QoS-based Routing in the Internet, RFC 2386

[52] L. Andersson et al., LDP Specification, draft-ietf-mpls-ldp-02.txt

[53] B. Davie et al., Use of Label Switching with RSVP, draft-ietf-mpls-rsvp-00.txt; and Explicit Route Support in MPLS, draft-davie-mpls-explicit-routes-00.txt

[54] D. Awduche et al., Requirements for Traffic Engineering over MPLS, draft-ietf-mpls-traffic-engineering-00.txt, Nov, 1998

[55] B. Jamoussi et al., Constrained-Based LSP Setup using LDP, draft-jamoussi-mpls-cr-ldp-00.tct, Nov, 1998

[56] UC Berkeley, LBL, USC/ISI and Xerox PARC, "The NS Manual", December 9, 2000

[57] J.M. Akinpelu, The Overload Performance of Engineered Networks with Nonhierarchical and Hierarchical Routing, in ITC-10, Montreal, 1983

[58] N.G. Bean, R.J. Gibbens & S. Zachary, The Performance of Single Resource Loss System in Multiservices Networks, Research Report 93-10, Stats. Labs, University of Cambridge, June, 1993.

[59] S.-P. Chung and K. W. Ross, Reduced Load Approximations for Multi-rate Loss Networks, IEEE Transactions on Communication, vol. 41, August, 1993.

[60] C.-T. Lea, K.-W. Ke, "Quantization and Cost Computation of MDP-based Admission and Routing", in Proc. IEEE INFOCOM'96, (San Francisco), April, 1996.

[61] Forsythe, G.E., M.A. Malcolm, and C.B. Moler, Computer Methods for Mathematical Computations, Prentice-Hall, 1977.

[62] I, Mitrani, Probabilistic Modelling, Cambridge University Press, 1998, Page 61-62.

[63] R.-H. Hwang, J. Kurose, and D. Towsley, State Dependent Routing for Multi-rate Loss Networks, Globecom'92, pp565-570, 1992.

[65] Azmodeh, M.; Macfadyen, R.N., Multi-rate call congestion: fixed point models and trunk reservation, Teletraffic Symposium, 11th. IEE Eleventh UK, 23-25 Mar 1994.

[66] Lee, Y.; Tien, J.M, Strategic parameter-driven routing models for multidestination traffic in telecommunication networks.; IEEE Transactions on Systems, Man and Cybernetics, Part B, Volume: 31 Issue: 5 , Oct 2001, Page(s): 720 -728

Appendix A: Complement Data for Chapter VI

We present the complement date for Chapter VI (Intra-network routing with non-homogenous holding time traffic), including the performance of LLR+TR, in terms of network throughput, under diverse traffic mix and holding time ratio combinations in two network examples.

	Traffic Mix								
Holding Time Ratio	**90%**	**80%**	**70%**	**60%**	**50%**	**40%**	**30%**	**20%**	**10%**
5	98.8%	99.0%	99.0%	99.1%	99.0%	99.0%	99.0%	99.0%	99.1%
10	98.8%	99.0%	99.1%	99.0%	99.1%	99.0%	99.1%	99.0%	99.1%
20	98.8%	99.0%	99.1%	99.2%	99.1%	99.1%	99.1%	99.0%	99.0%
50	98.9%	99.0%	99.2%	99.3%	99.2%	99.2%	99.1%	99.1%	99.1%
100	98.9%	99.1%	99.3%	99.3%	99.3%	99.2%	99.2%	99.1%	99.0%

Table: A-1: Network Throughput of LLR+TR in the 4-node network with various traffic mix and holding time ratio. TR=0.02, nominal traffic conditions.

	Traffic Mix								
Holding Time Ratio	**90%**	**80%**	**70%**	**60%**	**50%**	**40%**	**30%**	**20%**	**10%**
5	90.9%	90.9%	91.1%	91.2%	91.2%	91.2%	91.0%	90.9%	91.0%
10	91.1%	90.9%	91.3%	91.2%	91.3%	91.2%	91.0%	90.8%	91.1%
20	90.9%	90.9%	91.3%	91.2%	91.3%	91.2%	91.2%	90.9%	91.0%
50	90.9%	91.2%	91.2%	91.3%	91.2%	91.2%	91.2%	90.9%	91.1%
100	91.0%	91.1%	91.3%	91.3%	91.3%	91.2%	91.1%	91.0%	91.2%

Table: A-2: Network Throughput of LLR+TR in the 4-node network with various traffic mix and holding time ratio. TR=0.02, overload traffic conditions.

Holding Time Ratio	Traffic Mix 90%	80%	70%	60%	50%	40%	30%	20%	10%
10	93.6%	93.8%	93.5%	93.5%	93.5%	93.6%	93.7%	93.4%	93.4%
20	93.8%	93.9%	93.9%	93.7%	93.6%	93.4%	93.4%	93.6%	93.4%
50	93.9%	93.9%	93.9%	93.9%	93.6%	93.6%	93.7%	93.4%	93.5%

Table: A-3: Network Throughput of LLR+TR in the12-node network with various traffic mix and holding time ratio. TR=0.03, overload traffic conditions.

Traffic Mix→	90%	80%	70%	50%	30%	10%
Holding time ratio: 10	10.1%	10.2%	10.5%	10.3%	10.5%	10.0%
Holding time ratio: 50	10.0%	9.9%	10.1%	10.2%	10.2%	10.0%

Table A-4: Percentage of short requests taking vacating actions, overload traffic condition. 12-node network. Keep the overall traffic volume unchanged

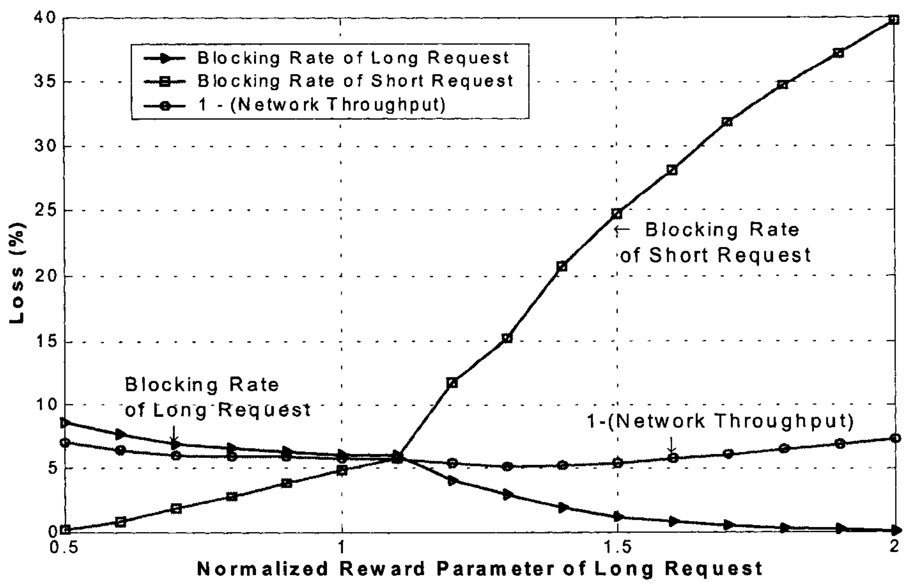

Figure A-1: Traffic Loss vs. normalized reward parameter of long requests in the 4-node network in *overload* traffic. Traffic mix is 80%, holding time ratio is 10.

Appendix B1: Kruithov's method to generate Topology and Traffic Data

B1.1 Total originating traffic from each node

The originating traffic of a node refers to the traffic generated by the customers (or networks) subtended by the node. The figure below illustrates originating and terminating traffics.

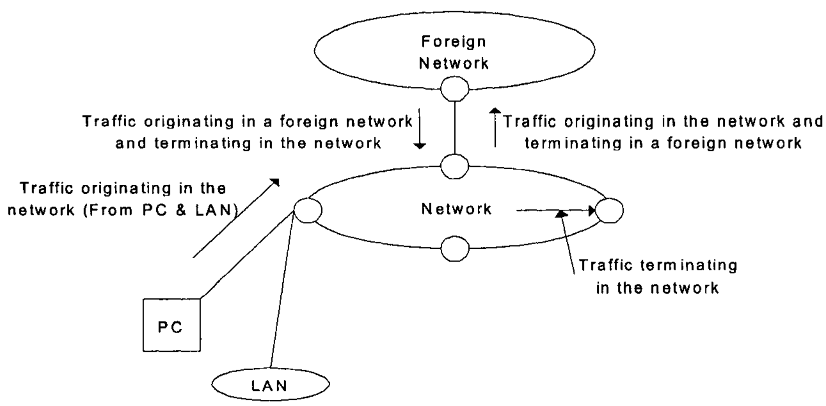

Figure B-1: Originating and terminating traffic.

Not all nodes are equal. Some nodes serve more customers than others. We can distinguish nodes as a function of the number of customers that they serve. For instance, for modeling purposes, we could use the following scale:

Nodes size	Normalized peak traffic	Prob of Occurrence
Small	0.5	.3
Medium	1	.4
Large	1.5	.2
Very large	2	.1

Let TO be the total originating traffic corresponding to normalized peak traffic of 1 (i.e., that of a medium node in the preceding table). We can generate the total originating traffic for each node by drawing its size at random following the distribution of column 3, and by then associating the proper Normalized peak traffic factor for its size. For instance, if node 1 is drawn to be a small node, its total originating traffic would be $TO_1 = 0.5\,TO$. Likewise, if node 2 is drawn to be a large node, its total originating traffic would be $TO_2 = 1.1\,TO$. Using this process, we can generate the size and the peak originating traffic for each node. (Note that many similar processes are also possible.)

B1.2 Non-coincidence of peak hours

In many networks, the different nodes do not necessarily sustain their peak traffic all at the same time. A simple example is a continental network in the United-States where nodes on the east coast typically peak earlier than nodes on the west coast, simply because the time difference. In general, we consider that the nodes of a network can be in one of the traffic levels defined in the Table below.

Traffic level	Fraction of peak traffic
Peak hour	1.0
Side peak hour	0.85
Off peak hour	0.70

Correspondingly we associate a traffic level to each node. Note that it may not make sense to associate the traffic levels randomly with the nodes here. Rather, these traffic levels should normally reflect an underlying cause (such as time zones). For example, returning to the continental United-States network example, the following traffic levels could be associated the nodes in the eastern, central and western regions in three particular hours. These three hours would represent three distinct traffic demands, all in the same network.

Hour	Region of the US where the node is located		
	East	Central	West
10h00 EST	Peak hour	Side peak hour	Off peak hour
12h00 EST	Side peak hour	Peak hour	Side peak hour
13h00 EST	Side peak hour	Side peak hour	Peak hour

From the viewpoint of our model, the traffic levels introduced here basically serve to reduce the total originating traffic of the nodes when they are outside their peak hour. For example, if node 1 is declared to be in "Side peak hour", its originating traffic would become $TO_1 \Leftarrow 0.85\,TO_1$. If node 2 is declared to be in "Off peak hour", its originating traffic would become $TO_2 \Leftarrow 0.70\,TO_2$. Repeating this process, we can generate the originating traffic demand for all nodes for a set of traffic levels representative of the network at a certain time.

B1.3 Allocating the traffic demand to individual destinations

The traffic originating at a node may be destined to a destination within the same network or to a destination in a foreign network. In the former case, the traffic remains within the network. In the latter case, the traffic is delivered to a gateway node where it progresses thereafter in the foreign network. In this latter case, the traffic thus leaves the network at the gateway node. We assume that the users originating the traffic have no knowledge of the network of residence of the destination. Thus, we assume that the destinations for the originating traffic are evenly distributed across the networks. We consider that there are X such networks; one of them being the network that we model, and the remaining $X-1$ being the foreign networks. This means that the originating traffic of a node:

- Stays within the network in proportion P. We call this traffic the originating intra-network traffic demand.
- Transfers to a foreign network in proportion $1-P$. We call this traffic the originating inter-network traffic demand.

If the X networks are identical, we should have $P=1/X$. However, the model should consider P and X as independent parameters for flexibility. The allocation of these demands to specific termination points is described in the next two subsections.

B1.3.1 Allocation of the originating intra-network traffic demand

We assume that the originating intra-network traffic demand terminates on the other nodes of the network in proportion to their total originating demand. Thus, let TO_i be the total originating demand of node i, as determined in section 2. The fraction of this demand allocated to node j in the network is: $TO_{ij} = TO_i \dfrac{TO_j}{\sum_{k \neq i} TO_k}$.

B1.3.2 Allocation of the originating inter-network traffic demand.

There are $X-1$ foreign networks, which we assume equivalent. Let these foreign networks be called $F_1, \ldots, F_{X-1}$. The originating traffic from a node i to each of these networks is:

$$TO_{iF_j} = TO_i \frac{1-P}{X-1}, \qquad j = 1, \ldots, X-1$$

We assume that foreign network F_j can be reached through a set of gateway nodes $G_{F_j} = \left\{ g_{F_j,1}, g_{F_j,N_{F_j}} \right\}$ where $g_{F_j,k}$ identifies the k^{th} node in the set and N_{F_j} denotes the number of nodes in the set. We assume that the originating traffic from a node i to the foreign network F_j is evenly equally distributed amongst its gateway nodes. This means that, for each node $g_{F_j,k}$ in G_{F_j}, the originating traffic from i to $g_{F_j,k}$ is:

$$TO_{i,g_{F_j,k}} = TO_{iF_j}\frac{1}{N_{F_j}}$$

This traffic assignment rule is very simple. Other rules, such as assigning more traffic to gateways that are closer to the origin node could also be considered. This comment could apply in particular for the nodes that are in the set G_{F_j}. These improvements are left for further study.

B1.3.3 Allocating the terminating traffic demand to individual destinations

Carrying the traffic assignment described in 3.1 and 3.2, the originating traffic demand of each node i will be assigned to individual destinations in the network. For a destination j that is not a gateway node, this demand will simply be TO_{ij} as determined in 3.1. For a destination j that is a gateway node to one or more foreign networks, this demand will be the sum of TO_{ij} as determined in 3.1 and of the foreign network traffic assigned to j as in 3.2. To complete the traffic demand, it remains to model and assign the traffic demand incoming from the foreign networks and terminating in the network. For this purpose, we make the assumption that the total traffic from a foreign network F_j terminating in the network is equal to the total traffic originating in the network and terminating in the foreign network F_j. Let TO_{F_j} be the total traffic originating in the network and terminating in the foreign network F_j. We have: $TO_{F_j} = \sum_{\text{all } i} TO_{iF_j}$.

Let TT_{F_j} be the total traffic incoming from the foreign network F_j and terminating in the network. By our assumption: $TO_{F_j} = TT_{F_j}$.

Similarly, as in 3.2, we assume that the terminating traffic incoming from foreign network F_j is evenly distributed amongst its gateway nodes. Thus, the terminating traffic of foreign network F_j incoming from gateway node $g_{F_j,k}$ is simply:

$$TT_{F_j,g_{F_j,k}} = TT_{F_j}\frac{1}{N_{F_j}}$$

Similarly as in 3.1, we assume that the terminating traffic incoming from $g_{F_j,k}$ is distributed to the individual nodes of the network proportionately to their total originating traffic. That is : $TT_{F_j,g_{F_j,k},j} = TT_{F_j,g_{F_j,k}}\dfrac{TO_j}{\sum_k TO_k}$

Following this process, the terminating traffic from all foreign networks can be assigned to individual destinations in the network.

B1.4 Sizing the network.

Sizing a network is in itself a complex optimization problem. However, this problem is not the focus of this study. For our purpose, a simple sizing algorithm should be sufficient. This section outlines such a simple algorithm. The input to the sizing algorithm is a set of node-to-node traffic demands, for instance determined through the process outlined in section 3.

1) Determine the total originating traffic from each node, for each of the traffic demands. Then determine $TO_{i,\max}$ the maximum total originating traffic demand from node i over all the traffic demands. The first principle of the sizing algorithm is that there must be enough outgoing capacity from node i to carry $TO_{i,\max}$. Assuming that the peak efficiency of the links is η, the total outgoing capacity from node i must be at least $TO_{i,\max}/\eta$ (η should be around 0.7).
2) Determine the total terminating traffic at each node, for each of the traffic demands. Then determine $TT_{i,\max}$ the maximum total terminating traffic demand at node i over all the traffic demands. By the same principle as above, there must be enough incoming capacity at node i to carry $TT_{i,\max}$. For the same reason as in 1, the total incoming capacity from node i must be at least $TT_{i,\max}/\eta$.
3) Consistency of the total originating and terminating traffic capacity. Determine the total originating capacity in the network $TO_{\max}$: $TO_{\max} = \sum_{\text{all } i} TO_{i,\max}$

   Determine the total terminating capacity in the network $TT_{\max}$: $TT_{\max} = \sum_{\text{all } i} TT_{i,\max}$

   If $TO_{\max} > TT_{\max}$, scale up the terminating capacities so that they sum up to $TO_{\max}$, i.e.:

   $$TT_{i,\max} \Leftarrow TT_{i,\max} \frac{TO_{\max}}{TT_{\max}}$$

   If $TO_{\max} < TT_{\max}$, scale up the originating capacities so that they sum up to $TT_{\max}$, i.e.:

   $$TO_{i,\max} \Leftarrow TO_{i,\max} \frac{TT_{\max}}{TO_{\max}}$$

4) Find a set of capacities matching all the $TO_{i,\max}$ and $TT_{i,\max}$. This is done by guessing an initial set of node-to-node capacities, and by successive iterations on the rows and

columns until the set of node-to-node capacities converge. Let $T=\left[t_{i,j}\right]$ be the average traffic demand matrix (where the average is carried over the set of traffic demands, and let $C=\left[c_{i,j}\right]$ be the matrix of node-to-node capacities. Row iteration:

for all i, set $c_{i,j} \Leftarrow c_{i,j}\dfrac{TO_{i,\max}}{\sum_k c_{i,k}}$, all j

Column iteration: for all j, set $c_{i,j} \Leftarrow c_{i,j}\dfrac{TT_{j,\max}}{\sum_k c_{k,j}}$, all i

5) Repeat the above row and column updates until the set of $C=\left[c_{i,j}\right]$ stabilizes. This process is known as *Kruithov's method.* It usually works fine, except in rare pathological cases.

B2.1 12-node Network Capacity

Unit: kB

	To Node											
From Node	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
0	—	2020	1400	1140	1120	1120	2020	2020	1100	1480	1100	1100
1	2020	—	1380	1120	1120	1100	2020	2020	1120	1480	1100	1100
2	1400	1380	—	—	—	—	1340	1340	—	—	1100	—
3	1120	1120	—	—	—	—	1080	1100	—	—	—	—
4	1120	1120	—	—	—	—	1080	1100	—	—	—	—
5	1120	1120	—	—	—	—	1080	1100	—	—	—	
6	2020	2020	1340	1100	1100	1100	—	2020	1100	1480	1100	1100
7	2020	2020	1320	1100	1100	1100	2020	—	1100	1480	1120	1120
8	1100	1120	—	—	—	—	1100	1100	—	—	—	—
9	1480	1500	—	—	—	—	1480	1500	—	—	—	—
10	1100	1120	—	—	—	—	1100	1120	—	—	—	—
11	1100	1100	—	—	—	—	1100	1120	—	—	—	—

Network Capacity of the 12-node network.

B2.2 30- node Network Capacity

Unit: KB

TO NODE

FROM NODE	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	-	1680	1120	560	560	280	280	560	560	1140	560	560	840	280	280
1	1540	-	1080	560	560	280	280	560	560	1140	560	560	840	280	280
2	1120	1120	-	400	200	300	300	300	400	1220	400	400	800	200	280
3	500	500	620	-	-	-	-	-	-	140	-	-	140	-	-
4	400	400	520	-	-	-	-	-	-	140	-	-	140	-	-
5	420	-	400	-	-	-	-	-	-	-	-	-	-	-	-
6	420	-	400	-	-	-	-	-	-	-	-	-	-	-	-
7	400	300	600	-	-	-	-	-	-	140	-	-	140	-	-
8	400	300	620	-	-	-	-	-	-	140	-	-	140	-	-
9	1180	1060	1260	140	140	-	-	140	140	-	140	140	140	-	-
10	400	400	420	-	-	-	-	-	-	140	-	-	140	-	-
11	400	400	420	-	-	-	-	-	-	140	-	-	140	-	-
12	860	820	920	140	140	-	-	140	140	160	-	-	-	-	-
13	-	420	320	-	-	-	-	-	-	-	-	-	-	-	-
14	-	420	320	-	-	-	-	-	-	-	-	-	-	-	-
15	400	400	400	-	-	-	-	-	-	140	-	-	140	-	-
16	400	400	400	-	-	-	-	-	-	140	-	-	140	-	-
17	-	420	400	-	-	-	-	-	-	-	-	-	-	-	-
18	400	400	400	-	-	-	-	-	-	140	-	-	140	-	-
19	-	420	400	-	-	-	-	-	-	-	-	-	-	-	-
20	1420	1260	1320	560	560	280	280	560	560	1140	560	560	860	280	280
21	1460	1200	1360	560	560	280	280	560	560	1140	560	560	860	280	280
22	1400	980	1360	560	560	280	300	560	560	1140	560	560	860	280	280
23	1100	1140	1260	140	140	-	-	140	140	180	140	-	140	-	-
24	860	860	940	140	-	-	-	-	-	160	-	-	140	-	-
25	400	400	420	-	-	-	-	-	-	140	-	-	-	-	-
26	-	420	440	-	-	-	-	-	-	-	-	-	-	-	-
27	400	400	400	-	-	-	-	-	-	140	-	-	-	-	-
28	400	400	420	-	-	-	-	-	-	140	-	-	-	-	-
29	800	880	900	140	-	-	-	-	-	160	-	-	140	-	-

FROM NODE	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	560	560	280	560	280	1500	1600	1400	1120	840	560	220	560	440	700
1	560	560	280	560	280	1060	1120	1260	1120	840	560	220	560	460	700
2	400	400	300	400	300	1360	1360	1360	1200	860	580	220	580	520	780
3	-	-	-	-	-	400	400	400	140	140	-	-	-	-	500
4	-	-	-	-	-	400	400	400	140	140	-	-	-	-	500
5	-	-	-	-	-	420	420	420	-	-	-	-	-	-	-
6	-	-	-	-	-	420	420	420	-	-	-	-	-	-	-
7	-	-	-	-	-	400	400	400	140	140	-	-	-	-	-
8	-	-	-	-	-	620	620	620	140	140	-	-	-	-	-
9	140	140	-	-	-	1160	1160	1160	160	140	140	-	380	-	500
10	-	-	-	-	-	600	600	600	140	140	-	-	-	-	-
11	-	-	-	-	-	600	600	600	140	140	-	-	-	-	-
12	-	-	-	-	-	900	900	900	160	140	-	-	420	420	500
13	-	-	-	-	-	420	420	420	-	-	-	-	-	-	-
14	-	-	-	-	-	420	420	420	-	-	-	-	-	-	-
15	-	-	-	-	-	600	600	600	140	140	-	-	-	-	500
16	-	-	-	-	-	620	600	600	140	140	-	-	-	-	500
17	-	-	-	-	-	420	420	420	-	-	-	-	-	-	-
18	-	-	-	-	-	600	600	600	140	140	-	-	-	-	500
19	-	-	-	-	-	420	420	420	-	-	-	-	-	-	-
20	560	580	280	580	280	-	1080	1080	1140	860	580	240	460	440	720
21	560	580	280	580	280	1080	-	1080	1100	860	580	240	460	440	720
22	560	580	280	580	280	1080	1080	-	1100	860	580	220	460	440	720
23	-	-	-	-	-	1200	1200	1200	-	140	140	-	-	-	-
24	-	-	-	-	-	900	920	900	160	-	-	-	140	420	240
25	-	-	-	-	-	600	600	400	140	140	-	-	-	-	160
26	-	-	-	-	-	420	420	420	-	-	-	-	-	-	-

Network Capacity of the 30-node network