Performance Optimization of Integrated Network Control Schemes in Packet-Switched Networks

Philip H. Enslow Jr. Ian F. Akyildiz Kongxun Wang

College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332

Abstract

Integrated schemes of routing, flow control, and congestion control in packet-switched networks are investigated for performance optimization. By utilizing the features of entry-to-exit routing and sliding window flow-control mechanism, the packet-switched networks are modeled as multi-chain, closed queueing networks. After the control parameters and objective function are introduced, an efficient method is derived to compute the gradient vector and Hessian matrix of such an objective function. Our method, used with a general non-linear programming strategy, solves this non-linear optimization problem. Several examples have been investigated which showed an efficient and stable behavior. The results obtained by our method are compared with exact values. It is shown that they are accurate.

1 Introduction

Packet-switched networks have enjoyed great popularity for bursty traffic communication in the last two decades because of the high efficiency gained through extensive resource sharing. However, without a good control method, it is very hard to achieve a high degree of sharing, hence the high efficiency. The major components of the network control include *routing, flow control,* and *congestion control.*

In the past, most studies have been concentrated on separate consideration of each of these control mechanisms, because of the complexity involved in an integrated study. However, this separation does not reflect the behavior of real systems since the problems are closely related [21, 10, 11, 23, 14, 24]. In this paper, we view the control mechanisms from an integrated point of view that will include certain important strategies such as window flow control schemes and entry-to-exit routing schemes.

One classification of routing strategies is based on how packets are forwarded. If the complete route used to forward a packet is determined before the packet is sent out from the entry node, it is called *entry-to-exit* routing. Otherwise, it is called *hop-by-hop* routing. Although the latter is simple, the former is more efficient and makes it easier to consider user performance requests and flow control [6, 7, 8, 24].

The most widely used mechanism for flow control is *window* mechanism whose primary parameter is *window sizes*. By assigning and/or modifying window sizes appropriately, the input traffic for each user request can be allocated and regulated. Thus, congestion control can also be achieved.

A window controlled network is usually modeled as a multi-chain, closed queueing network by which the performance of the network can be analyzed [20, 4]. Consequently the network design and control can be formulated as various optimization problems. For example, the optimization problem in [1, 2] is formulated as choosing best service rates to optimize network performance and those in [10, 16] are formulated as obtaining best routing configuration with fixed window sizes. To our knowledge there is no work so far that optimizes both routing and window sizes simultaneously.

In this paper we describe the packet-switched networks in Section 2. Introduce the multi-chain, closed queueing network model in Section 3. Based on the model, we formulate network control as a performance optimization problem in Section 4. We solve the optimization problem in Section 5. Section 6 contains examples to show the modeling and optimizing processes as well as the accuracy of the results. In Section 7 we analyze the time complexity and storage requirement of the method and the accuracy of the results. Section 8 concludes the paper and points out some of the possible research directions for the future.

CH2996-7/91/0000/0036$01.00 © 1991 IEEE

2 Network Description

We describe the general environment and some assumptions.

1. The packet-switched subnetwork consists of a set of nodes meshed together by a set of full duplex trunks.

2. The trunk communication protocols will provide four channels on each trunk such that two data channels in opposite directions are responsible for transmitting data packets on the trunk and two control channels in opposite directions for transmitting control packets on the trunk.

3. The control channels have higher priority over the data channels, and acknowledgement packets will be transmitted over the control channels.

4. A user request to the subnetwork is a traffic request from one node to another, which is independent of other user requests.

5. Packet processing time at a node is negligible compared with packet transmission time and queueing time.

6. There are enough buffers at each node. That is, the total number of buffers at a node is larger or equal to the sum of the window sizes of the chains that pass through the node.

Assumption 1 is a general description of any switched computer network. The only exception is the full-duplex requirement on trunks which is not a problem for most packet-switched networks. From the concepts of layered communication architectures, assumptions 2 and 3 are exactly what they require, that is, assuming and utilizing the well-defined services provided by underlying layers. Here, we are also using the concept of *out-of-band control signalling* [15]. Assumptions 4, 5, and 6 are usual assumptions made in almost all studies in the literature, and they are commonly accepted by the researchers in the field.

In addition to the above assumptions, it is also appropriate and necessary to consider the characteristics of the specific control schemes used in the network model. The control schemes used are given as follows.

- The routing scheme utilized is entry-to-exit routing [6, 7, 8]. There are a set of physical routes available for each user input traffic. All physical routes in the network are *predefined* at *network installation* time, or during network design phase. They change only when there is a topology update or a major statistical change of network loads. The task of routing is to determine the best splitting of the traffic over these physical routes.

- The window mechanism is exercised on each user request. The task of these controls is to find the best window size for each user request such that all the user input traffic can be regulated to achieve overall optimal performance.

3 Queueing Network Modeling

The subnetwork described above is modeled as a multi-chain, closed queueing network as follows.

- A server station in the queueing network models either a data channel in the *subnetwork* or an *access link* from a communicating station to a node in the subnetwork[1]. Such server stations are denoted as $1, 2, \ldots, N$ where N is the total number of server stations in the queueing network. Note that N also denotes the index set of all server stations.

- There are R chains denoted as $1, 2, \ldots, R$, with each corresponding to one user request. R also denotes the index set of all chains.

- The source of a chain is represented by exponential message length distribution and a Poisson message interarrival time distribution. The arrival process will halt if the window is full. That is, the packets will be lost.

- The destinations are modeled by an exponential absorption time distributions.

- To model entry-to-exit routing, it is assumed that there are P_r physical routes available for chain r, denoted as $1, 2, \ldots, P_r$. P_r also denotes the index set of all physical routes available for chain r.

- $1/\alpha_{ir}\mu_i$ is the service time that server station i needs to serve packets of chain r, where μ_i is the

[1] To make the analysis simple the acknowledgement time and the propagation delay in the network will be ignored. Hence, the infinite-server stations can be dropped out from consideration. This is because

- the assumptions on the underlying layers state that the control channels have higher priority,
- the acknowledgement packets are very short compared with data packets, and
- the queueing delay is large compared with the propagation delay.

service rate of server station i and α_{ir} characterizes the relative rates for different packet chains in the same server station. Moreover, the service times are independent of messages interarrival time.

- The control parameters used in window mechanism are the window sizes, with W_r being the window size for chain r.
- The routing parameters are denoted by ϕ_{pr}, where $r \in R$ and $p \in P_r$, which represents the input traffic on chain r split over the P_r physical routes.

To describe the multiple physical routes for a chain in the multi-chain, closed queueing network, mathematically we define for route p in chain r

$$d_{ipr} = \begin{cases} 1 & \text{if it passes server station } i \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

Then, predefinition of physical routes means that all the d_{ipr} are fixed when our control algorithm is used. How to create and maintain these routes during network operation was discussed in [8, 24].

Using the multi-chain, closed queueing network model introduced above, the task of the network control can be accomplished by controlling or assigning appropriate values to the control parameters of the schemes incorporated in the model. In other words, what we are trying to do is to find the optimal values for the control parameters. The objective function in this case can be any performance measure that depends on the control parameters. The control variables in our model are the window sizes for every chain and the routing configuration parameters. Written in vector form, they are row vectors $\mathbf{W} = (W_1, W_2, \ldots, W_R)$ and $\Phi = \left(\Phi_1{}^T, \Phi_2{}^T, \ldots, \Phi_R{}^T\right)$, [2] where $\Phi_r = (\phi_{1r}, \phi_{2r}, \ldots, \phi_{P_r r})^T$. The total number of control variables is equal to $R + \sum_{r=1}^{R} P_r = R + P$, where P is the total number of physical routes in the network.

The window sizes have the following simple bounds constraints.

$$1 \le W_r \le u_r \qquad \forall\, r \in R \qquad (2)$$

where u_r depends on the maximum capacity of chain r.

However, the routing configuration need some elaborations. In a generic queueing network, in order to describe its routing and/or topology either of the two following sets of parameters could be used,

1. transition probabilities $\{p_{jir}\}$, where p_{jir} is the probability that a chain r packet after completing service at server station j proceeds to server station i, and
2. relative throughputs $\{y_{ir}\}$, where y_{ir} is the throughput of the chain r at server station i, normalized to the throughput of chain r. The y_{ir} are also called the mean number of visits that a chain r packet makes to server station i, and denoted by e_{ir} in the literature.

The relationship between them is a simple equation

$$y_{ir} = \sum_{s=1}^{R} \sum_{j=1}^{N} y_{js} \cdot p_{jis} \qquad \forall\, r \in R \,\&\, i \in N \qquad (3)$$

The transition probabilities $\{p_{ijr}\}$ are suitable to specify routing configuration in hop-by-hop routing since they only describe the flow of packets from one station to another and do not reflect the predetermination of routes before packets are sent out. The relative throughput $\{y_{ir}\}$, on the other hand, are more suitable for entry-to-exit routing because of their conceptual relationship with the control variables $\{\phi_{pr}\}$ and the computational efficiency. This can be seen in the equations below. Using the definition (1) and writing them in a matrix form we have

$$\mathbf{Y}_r = \mathbf{D}_r \cdot \Phi_r \qquad \forall\, r \in R \qquad (4)$$

where $\mathbf{Y}_r = (y_{1r}, y_{2r}, \ldots, y_{Nr})^T$ are column vectors and $\mathbf{D}_r = [d_{ipr}]$ is an $N \times P_r$ matrix which specifies the given P_r physical routes in chain r. This equation states that through a series of simple logical tests and arithmetic additions, y_{ir} can be computed from routing configuration variables.

As for the constraints on these routing variables, from the description of the model in the last chapter, obviously,

$$0 \le \phi_{pr} \le 1 \;\&\; \sum_{p=1}^{P_r} \phi_{pr} = 1 \quad \forall\, r \in R \,\&\, p \in P_r \qquad (5)$$

Note that the routing variables over one chain are not independent of each other.

Figure 1 gives a general picture of the model.

4 Optimization Problem

Let $F(\mathbf{W}, \Phi)$, be the objective function, which can be any performance measure that depends on the control variables. A general optimization problem for the

[2] That Φ is written in this form, instead of $(\phi_{11}, \phi_{21}, \ldots, \phi_{P_1 1}, \ldots, \phi_{1r}, \phi_{2r}, \ldots, \phi_{P_r r}, \ldots, \phi_{1R}, \phi_{2R}, \ldots, \phi_{P_R R})$ is only for notation convenience. It does not imply that there need any structure among the variables. In fact, they are just sequentially listed in the vector.

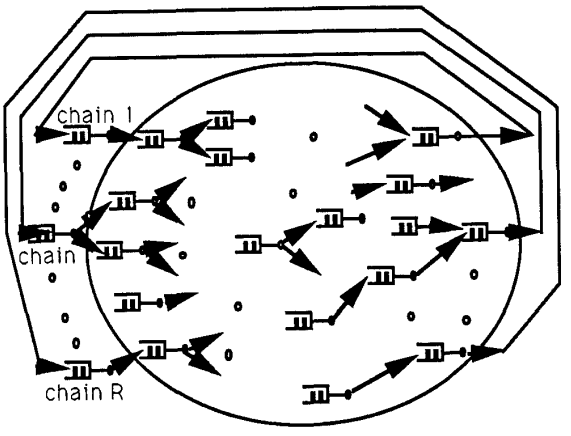

Figure 1: A multi-chain, closed queueing network

network control can be formulated as follows.

$$\begin{array}{lll} \textbf{Optimize} & F(\mathbf{W},\Phi) & \\ \textbf{Subject to} & 1 \le W_r \le u_r & \forall\, r{\in}R \\ & W_r \text{ is an integer} & \forall\, r{\in}R \\ & 0 \le \phi_{pr} \le 1 & \forall\, r{\in}R \,\&\, p{\in}P_r \\ & \sum_{p=1}^{P_r} \phi_{pr} = 1 & \forall\, r{\in}R \end{array}$$

To solve this optimization problem, $F(\mathbf{W},\Phi)$ must be specified. Since the objective function is nonlinear and complex, we have a nonlinear optimization problem with mixed linear equality and simple bounds constraints [12, 13]. In this paper we will utilize the structure of our queueing network model and the AMVA (Approximate Mean Value Analysis) method [3, 22] to build a realistic objective function which incorporates performance measures such as the throughput and the delay.

For a given set of routing variables $\{\phi_{ir}\}$, $\{y_{ir}\}$ can be simply computed from (4). Based on the given $\{W_r\}$ and computed $\{y_{ir}\}$, the AMVA method [3, 22] can be utilized to obtain some performance measures, for example, the throughput, λ_r, of chain r; the mean delay, $\bar{t}_{ir}$, of a chain r packet at station i; and the mean queue size, $\bar{q}_{ir}$, of chain r at the station i. Note that $\bar{t}_{ir}$ and $\bar{q}_{ir}$ are not defined if $y_{ir} = 0$ because, from the definition of y_{ir}, this can happen only in either of the two cases described below. The first case is when no physical route of chain r passes through station i. The second case is which the routing variable(s) on the physical route(s) of chain r that pass through station i are all zeros. Both of these two cases imply that a chain r packet will not go over station i. Hence no delay or queue incurs that is related to chain r in station i. For notation purpose, we will set them equal to zero when they are not defined.

We define $F(\mathbf{W},\Phi) = \sum_{r=1}^{R} \lambda_r/\bar{t}_r$ as our objective function, where $\bar{t}_r$ is the mean total delay a chain-r packet experienced when traversing through the chain. The rationale behind this is that an objective function containing only throughput or only delay will be monotonic for some networks. Using this objective function, the optimization becomes maximization of the objective function is expressed by the following equations.

$$\begin{aligned} F(\mathbf{W},\Phi) &= \sum_{r=1}^{R} \frac{\lambda_r}{\bar{t}_r} = \sum_{r=1}^{R} \frac{\lambda_r}{W_r/\lambda_r} \\ &= \sum_{r=1}^{R} \frac{\lambda_r^2}{W_r} \qquad (6) \\ \lambda_r &= \frac{W_r}{\sum_{i=1}^{N} y_{ir}\bar{t}_{ir}} \qquad \forall\, r{\in}R \qquad (7) \\ y_{ir} &= \sum_{p=1}^{P_r} d_{ipr}\phi_{pr} \qquad \forall\, r{\in}R \,\&\, i{\in}N \qquad (8) \\ \bar{t}_{ir} &= \begin{cases} 0 & \text{if } y_{ir} = 0 \\ \frac{1}{\alpha_{ir}\mu_i}\left(1 + \frac{W_r - 1}{W_r}\bar{q}_{ir} + \sum_{s\neq r}\bar{q}_{is}\right) & \text{if } y_{ir} > 0 \end{cases} \\ & \qquad \forall\, r{\in}R \,\&\, i{\in}N \qquad (9) \\ \bar{q}_{ir} &= \lambda_r y_{ir}\bar{t}_{ir} \qquad \forall\, r{\in}R \,\&\, i{\in}N \qquad (10) \end{aligned}$$

5 Solution of The Optimization

In order to solve the nonlinear optimization problem we first relax the integer constraint. Then the problem becomes a standard nonlinear optimization problem with mixed linear equality and simple bound constraints [12, 13].

$$\begin{array}{ll} \textbf{Maximize} & F(\mathbf{x}) \\ \textbf{Subject to} & \mathbf{Bx} = \mathbf{b} \\ & \mathbf{l} \le \mathbf{x} \le \mathbf{u} \end{array}$$

where $\mathbf{x}$ is the column vector that contains $R{+}P$ control variables, $F(\mathbf{x})$ is the objective function defined by (6), (7), (8), (9), and (10), $\mathbf{B}$ is the $R{\times}(R{+}P)$ matrix whose rows contain the coefficients of the equality constraints, $\mathbf{b}$ is the column vector that contains R constants on the right hand sides of these constraints, $\mathbf{l}$ is the column vector that contains the lower bounds on control variables, and $\mathbf{u}$ is the column vector that contains the upper bounds on the control variables. Therefore,

$$\begin{aligned} \mathbf{x} &= (\mathbf{W},\Phi)^T \\ \mathbf{b} &= (1,\ldots,1)^T \\ \mathbf{l} &= (\underbrace{1,\ldots,1}_{R},\underbrace{0,\ldots,0}_{P})^T \end{aligned}$$

$$\mathbf{u} = (\underbrace{u_1, \ldots, u_R}_{R}, \underbrace{1, \ldots, 1}_{P})^T$$

$$\mathbf{B} = \begin{bmatrix} \underbrace{00\ldots0}_{R} & \underbrace{11\ldots1}_{P_1} & 00 & \ldots & 0 \\ \underbrace{00\ldots0}_{R} & 00\ldots0 & \underbrace{11\ldots1}_{P_2} & 00 & \ldots\ 0 \\ \vdots & & \ddots & & \vdots \\ \underbrace{00\ldots0}_{R} & 00 & \ldots & 0 & \underbrace{11\ldots1}_{P_R} \end{bmatrix}$$

For problems with this kind of structure, Murtagh and Saunders [19], proposed an algorithm which is the best one as far as we are aware of. This algorithm is a search algorithm. Starting from an initial point, it computes the search direction and step length from the gradient vector $\mathbf{g}(\mathbf{x}) \equiv \nabla F(\mathbf{x})$ and the Hessian matrix $\mathbf{G}(\mathbf{x}) \equiv \nabla^2 F(\mathbf{x})$, and obtain a new point. Then it repeats until an optimal point is found. The constraints are used to limited the directions and step lengths of each search.

Although an arbitrary initial point might lead to an optimal point, it makes the search inefficient and maybe not convergent. In our case, a feasible initial point can easily be obtained by letting all the window variables be equal to 1 and using an arbitrary assignment of the traffic for chain r to the P_r routes.

Another set of data we need to obtain is the upper bounds on the window sizes. They can be computed by assuming that each chain is independent of other chains and will take all the capacities of the stations on the chain.

It should be noted that during the search for the optimal point and at the end of execution, the intermediate and the resulting *window sizes* are not necessarily *integers*. By rounding them to nearest integers, we can be assured that they will be integers.

The Murtagh and Saunders' method requires some procedures to evaluate $F(\mathbf{x})$ and compute $\mathbf{g}(\mathbf{x})$ and $\mathbf{G}(\mathbf{x})$. Using equation (4) and the AMVA method [4], all quantities in equations (7), (8), (9), and (10) can be obtained. Hence, the $F(\mathbf{x})$ can be evaluated. However, without an efficient method to compute the $\mathbf{g}(\mathbf{x})$ and $\mathbf{G}(\mathbf{x})$, or if the accuracy of the computation results are poor, the Murtagh and Saunders' method performs very badly[3]. We developed an efficient method to compute $\mathbf{g}(\mathbf{x})$ and $\mathbf{G}(\mathbf{x})$. The method and the derivations were given in [9]. To save space, only the basic idea is briefly mentioned below.

Because of the complexity involved, we adopted a numerical approach. From (8), $\{y_{ir}\}$ can be computed. Then by utilizing AMVA method, $\{\lambda_r\}$ and $\{\overline{q}_{ir}\}$ can be obtained. The objective function is obtained from $\{\lambda_r\}$ by using (6). We also derived an equation that is only involved in the control variables and the above quantities. By differentiating the both sides of the equations, the derivatives of $\{\lambda_r\}$ with respect to the control variables can be expressed by two large sets of simultaneous linear equations. By breaking these two sets of equations into a series of small simultaneous linear equations with a constant coefficient matrix $\mathbf{A}$. The problem becomes tractable. After these derivatives obtained, $\mathbf{g}(\mathbf{x})$ and $\mathbf{G}(\mathbf{x})$ can be computed.

[3]We have tested some cases by not giving the gradient vectors to the algorithm. The searches did not converge.

6 Examples

We applied our method on several examples. In this section, a small example will be presented in detail to show a step-by-step procedure from network modeling to optimization result. In [9], more examples are presented. Some of them, with reasonable sizes, will show the computation efficiency of our method. Others will be used to demonstrate the accuracy of our method.

We have coded our method into a Fortran program, running on a CDC 9000 machine. We utilized a package for linear system and related problems, called *LINPACK* [5], to compute the inverse of the matrix $\mathbf{A}$ mentioned in the previous section. We also utilized an optimization program called MINOS [18] which is based on the Murtagh and Saunders' method. We did not use the Hessian matrix $\mathbf{G}(\mathbf{x})$ because it is not utilized in MINOS. Consequently, the efficiency is compromised. But even though, the results still show that our method is efficient.

The communication network is given in Figure 2 where nodes a, b, and c are switch nodes within the subnet; and a', b', and c' are the local access parts of the corresponding switch nodes, respectively. There are 2 chains and 4 physical routes listed in Table 1.

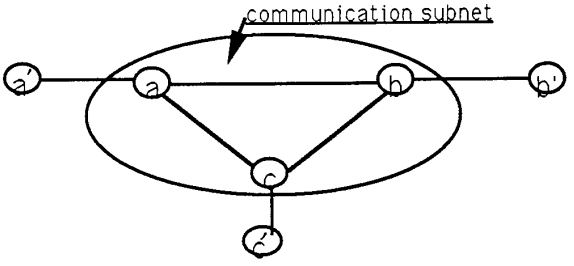

Figure 2: Communication Network

The network can be modeled as a closed multi-chain queueing network shown in Figure 3. The service times are listed in Table 2.

Table 1: Chains and physical routes

Chain		Physical routes
1	1	a'–a–b–b'
	2	a'–a–c–b–b'
2	1	c'–c–b–b'
	2	c'–c–a–b–b'

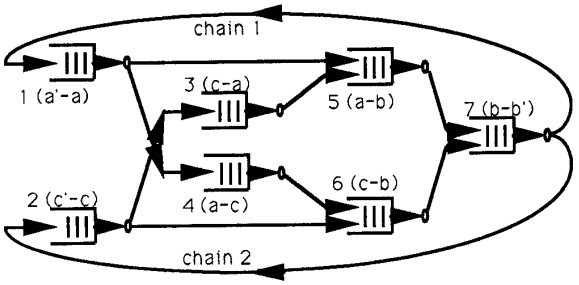

Figure 3: Closed multi-chain queueing network

From the given network and definitions of d_{ipr} in (1), the matrices, D_1 and D_2 in (4), that specify the topology of the physical routes, can be obtained.

$$\mathbf{D_1}^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

$$\mathbf{D_2}^T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

The upper bounds on the window sizes are given to be $u_1 = 5$ and $u_2 = 5$. The lower bounds are $l_1 = 1$ and $l_2 = 1$.

Now the optimization problem with the integer constraints relaxed can be expressed as follows.

$$\begin{aligned} &\textbf{Maximize} && F(\mathbf{x}) \\ &\textbf{Subject to} && \mathbf{Bx} = \mathbf{b} \\ & && \mathbf{l} \le \mathbf{x} \le \mathbf{u} \qquad \text{where} \end{aligned}$$

$$\mathbf{x} = (W_1, W_2, \phi_{11}, \phi_{21}, \phi_{12}, \phi_{22})^T$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

$$\mathbf{b} = (1,1)^T$$

$$\mathbf{l} = (1,1,0,0,0,0)^T$$

$$\mathbf{u} = (5,5,1,1,1,1)^T$$

Let $\mathbf{x}^0 = (1,1,0.5,0.5,0.5,0.5)^T$ be the initial value. Obviously it satisfies the constraints. We computed

Table 2: Service times

server station i	service time	
	$1/\alpha_{i1}\mu_i$	$1/\alpha_{i2}\mu_i$
1	0.746	0.556
2	0.435	0.695
3	0.893	0.414
4	0.311	1.236
5	0.667	0.367
6	0.285	0.585
7	0.226	0.267

$F(\mathbf{x}^0) \approx 0.66577$, and $\mathbf{g}(\mathbf{x}^0) \approx (0.1183, 0.1263, -0.7081, -0.6571, -0.6050, -0.6929)^T$. Then the MINOS begins to search. After one iteration, we get a new value of $\mathbf{x}^1 \approx (1.7099, 1.7580, 0.3471, 0.6529, 0.7637, 0.2363)^T$ with $F(\mathbf{x}^1) \approx 0.76032$. This is better than the previous value. After total 12 iterations the search converged to an optimal point. The size of reduced gradients was reduced to 10^{-5}, which is used as the convergence test. The optimal point is $\mathbf{x}^{12} = (1.8253, 2.1696, 0.0000, 1.0000, 0.1062, 0.8938)^T$, and objective function is $F(\mathbf{x}^{12}) = 0.79910$. By rounding the non-integer window size components, we obtain $\mathbf{x}^{12*} = (2, 2, 0.0000, 1.0000, 0.1062, 0.8938)^T$, and $F(\mathbf{x}^{12*}) = 0.796830530$. The total number of evaluations of $F(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ is 22.

As this network is small, by enumerating all the possible values in the range with a step size for the window variable being 1, and that for routing variable being 0.05, we can obtain the true optimal value $\mathbf{x}^* = (2, 2, 0.00, 1.00, 0.10, 0.90)^T$, and $F(\mathbf{x}^*) = 0.796837866$. Comparing our result, we see that they are very close to the exact values.

7 Evaluation of The Method

This section discusses the operational evaluation of the method, that is, the convergence of the method, the time complexity, the storage requirement, and the accuracy of the results.

From the examples, several consequences can be implied. First, we made the computation with different initial values. When one of the variables in the initial vector is 0, for instance, $\mathbf{x}^0 = (1, 1, 1.0, 0.0, 1.0, 0.0)^T$, MINOS refused to search since $\mathbf{g}(\mathbf{x})$ at these points is not correct, i.e., it is singular at these points. Except for these singular points, all other initial values lead to the same optimal point, with a little difference on the performance of the method. For instance, when

$\mathbf{x}^0 = (1, 1, 0.4, 0.6, 0.4, 0.6)^T$ the total number of iterations is 11 and total number of evaluations of objective function is 21. Except for those singular initial values, we feel that this method converges with respect to the optimal search of $\mathbf{x}^*$.

We also looked into the matrix **A** mentioned in Section 5 for some iterations, which is the key to compute the gradient vector. All the instances have shown that the main diagonal elements are much larger than the other elements in the matrix. We have not encountered any instance of singularity of this matrix during our computations. Therefore, our method is numerically stable.

For the other aspects of the method, since this is the first attempt to attack the problem, there is no other existing method, except for an enumeration method which just enumerates all points in the variable range, to compare with. Compared with the enumeration method, our method is obviously much better. For instance, in the example, we used a step size 1 for window variables and a step size 0.05 for routing variables to obtain the points to enumerate. Therefore, the enumeration method requires $5 \times 5 \times 20 \times 20 = 10000$ evaluations. And ours need only 22 evaluations. For other examples, we can claim similar improvement. A storage requirement of order $O(R^2)$ is required by our method, in addition to the storage required by the AMVA method and by MINOS.

To discuss the accuracy aspect, we need to analyze every component of the method. After introducing the optimization problem in Section 4, we relaxed the integer constraints and proposed to use rounding scheme for the resulting window variables. Obviously, this may make resulting point away from the optimal. The second source of the non-accuracy may come from the AMVA method since it uses an iterative method to obtain the performance measures which are used to evaluate the objective function. The other two possible sources might be either our procedures to compute the gradient vectors and Hessian matrix or the Murtagh and Saunders' algorithm.

The program MINOS is used in many places and for different problems. It is believed that it is one of the best in the current nonlinear optimization field. However, its performance depends on procedures used to evaluate the objective function and the gradient vector, as we mentioned before. Our derivation and the computation of the gradient vector shown in Section 5 are exact. Hence the non-accuracy of our method would be determined by either the AMVA method, or the rounding scheme for window variables. It is difficult to analyze the non-optimality due to relaxation of integer constraints. The empirical results are needed. From the examples we have done, the accuracy aspects are not a problem.

8 Conclusion

In this paper we formulated the network control as a performance optimization problem, based on a multi-chain, closed queueing network model. This model, utilizing entry-to-exit routing as a routing strategy and window mechanism for flow control and congestion control, considers the three components of network control simultaneously. The novel feature of our method to solve the optimization problem is that the objective function can be expressed by a set of equations, utilizing the special structure of the model and the AMVA method. The first and second derivatives of the objective function are computed by solving a series of simultaneous linear equations. After the necessary quantities, specifically, the values of objective function, gradient vector, and Hessian matrix, are obtained, many of the existing efficient methods, for example, those in [19, 17], and some software packages, for example, MINOS, for nonlinear optimization problems can be utilized.

The evaluation of the method considering the time complexity, storage requirements, and the accuracy of the result is discussed. However, since this is the first attempt to optimize both window and routing variables at the same time, there is no other existing method that can be compared for the accuracy of the results. The inaccurate results would be caused by AMVA, rounding of window sizes due to the integer constraints.

The performance of the network using such a control strategy will depend on various other factors such as the physical routes available [8], etc., which need further study. It is likely that such a performance evaluation will be carried out by simulation, because of the complexity of the problem and the current research status in this area.

Acknowledgement

We would like to thank Mr. Ping Qian for his help in numerical methods of linear equations. We are also grateful to Professor P. J. Schweitzer for the useful discussions during the early stage of the problem formulation.

References

[1] I.F. Akyildiz and G. Bolch, "Throughput and response time optimization in queueing network models of computer", *Proc. of the IFIP TC 7/WG 7.3 International Seminar on Performance of Distributed and Parallel Systems*, Kyoto, Japan, Dec. 7-9, 1988, pp. 251-269.

[2] I.F. Akyildiz and R. Shonkwiler, "Simulated annealing for throughput optimization in communication networks with window flow control", *IEEE Internation Conference on Communications*, Atlanta, GA, April, 16-19, 1990, pp. 330.4.1-330.4.8.

[3] Y. Bard, "Some extension to multiclass queueing network analysis", *Proc. of the Performance 79*, Feb. 1979, Vol. 1.

[4] K.M. Chandy and D. Neuse, "Linearizer: A heuristic algorithm for queueing network models of computing systems", *Comm. ACM* **25** (2) (1982), pp. 126-134.

[5] J.J. Dongarra, J.R. Bunch, C.B. Moler, and G.W. Stewart, *LINPACK User's Guide, SIAM*, 1984.

[6] B. Enshayan, "ALPHA transport", *Ninth Data Communication Symposium*, Whistler Mountain, British Columbia, Sept. 9-12, 1985, pp. 146-154.

[7] P.H. Enslow, Jr. and Kongxun Wang, "Study of ALPHA protocol", *Report GIT-ICS-89/33*, School of ICS, Georgia Inst. of Tech., Sept. 1989.

[8] P.H. Enslow, Jr., I.F. Akyildiz, and Kongxun Wang, "A new scheme to maintain routes in entry-to-exit routing strategy", *4th Int. Conf. on Data Comm. Syst. and Their Perf.*, Barcelona, Spain, June 20-22, 1990, pp. 380-395.

[9] P.H. Enslow, Jr., I.F. Akyildiz, and Kongxun Wang, "Performance optimization of packet-switched networks with integrated routing, flow control, and congestion control", *Report GIT-ICS-90/36*, School of ICS, Georgia Inst. of Tech., Sept. 1990.

[10] M. Gerla and L. Kleinrock, "Flow control: A comparative survey", *IEEE Trans. on Comm.*, Vol. COM-28, No. 4, pp. 553-574, Apr. 1980.

[11] M. Gerla and P.O. Nilsson, "Routing and flow control interplay in computer networks", *Proc. 5th ICCC*, Atlanta, GA, Oct. 27-30, 1980, pp. 85-89.

[12] P.E. Gill, W. Murray, and M.H. Wright, *Practical Optimization*, Academic Press, 1981.

[13] P.E. Gill, W. Murray, M.A. Saunders, and M.H. Wright, "Procedures for optimization problems with a mixture of bounds and general linear constraints", *ACM Trans. Math.*, Vol. 10, No. 3, pp. 282-298, 1984.

[14] D.W. Glazer and C. Tropper, "A new metric for dynamic routing algorithms", *IEEE Trans. on Comm.*, Vol. COM-38, No. 3, pp. 360-367, Mar. 1990.

[15] H. Heffes, "Performance analysis, traffic engineering and congestion controls for ISDN systems", *IEEE GLOBECOM*, pp. 14.5.1-14.5.7, 1987.

[16] H. Kobayashi and M. Gerla, "Optimal routing in closed queueing networks", *ACM Trans. on Computer Systems*, Vol. 1, No. 4, pp. 294-310, 1983.

[17] R.E. Marsten and D.F. Shanno, "Conjugate-gradient methods for linearly constrained nonlinear programming", *Report 79-13*, Department of Management Information Systems, University of Arizona, Tucson, Arizona, 1979.

[18] B.A. Murtagh and M.A. Saunders, "MINOS 5.1 User's guide", *Report SOL 83-20R*, Department of Operations Research, Stanford University, California, Dec. 1983, revised Jan. 1987.

[19] B.A. Murtagh and M.A. Saunders, "Large-scale linearly constrained optimization", *Math. Prog.* **14** (1978), pp. 41-72.

[20] M. Reiser, "A queueing network analysis of computer communication networks with window flow control", *IEEE Trans. on Comm.*, Vol. COM-27, No. 8, pp. 1199-1209, Aug. 1979.

[21] H. Rudin and H. Mueller, "Dynamic routing and flow control", *IEEE Trans. on Comm.*, Vol. COM-28, No. 7, pp. 1030-1039, July. 1980.

[22] P.J. Schweitzer, "Approximate analysis of multiclass closed networks of queues", *Int. Conf. Stochastic Control and Optimization*, Amsterdam, 1979.

[23] G.H. Thaker and J.B. Cain, "Interactions between routing and flow control algorithms", *IEEE Trans. on Comm.*, Vol. COM-34, No. 3, pp. 269-277, Mar. 1986.

[24] Kongxun Wang, "Performance optimization with integrated consideration of routing, flow control, and congestion control in packet-switched networks", *Ph. D. dissertation*, Georgia Inst. of Tech., Jan. 1991.