European Journal of Operational Research 19 (1985) pp. 262-267
Received December 1982:
Revised December 1983 and February 1984.

Routing optimization in packet switching communication networks.

Z.Flikop 725 Greenbrier Dr. Apt.14 Bohemia, NY 11756 USA

Abstract. For routing assignments a special model and an optimization algorithm are proposed. The efficiency of the routing assessments is evaluated by the average value of the total cost of delays for all packets in the network. It is the objective function. The main idea is the traffic, which is transmitted from the source node to the destination node can be split between two or more logical paths. The minimum of the objective function can be found by varying the traffic on every path and simultaneously from all the source nodes to the destination notes. If this approach is applied, then the objective function is no separable and nonlinear. Because its shape is unknown in advance an adoptive nonlinear optimization algorithm is proposed. For evaluating its efficiency special set of test function has been used.

Keywords: Routing, communication, networks

1. Introduction

Nowadays complicated communication networks are developed and implemented. At the same time routing optimization for this networks is studied very intensively and literature in this field is vast, for example [1-5]. This article discusses the same problem, and a special method to design he mathematical models of the networks is described. The first model has been developed and implemented in computer language in 1970 and published in 1971 [6]. For optimization in the early version of the model, a method similar to [7] has been used. Later a more efficient optimization algorithm was created [8]. This article is dedicated to it generalize the approach, which can be used to simultaneously create and update the routing tables at all the nodes of the network. Before the approach is described, we analyze the network properties.

Usually a communication network consists of hosts and a transmission system [9]. The transmission system is a set of packet switches (notes) and channels which carry packets from the source nodes to the destination notes. The common situation is: The same note can serve different kinds of packets. As usual the same packet can be transmitted through the network by different paths. The total time of transmitting a packet from the resource note to the destination node dependents on the selected path. Particularly this time depends on the traffic intensity ratios for all the nodes which constitute the path. In turn, the traffic intensity ratio depends on the fraction of the traffic that is transmitted through the path (i.e. assignment all of the traffic flows within the network). Usually the delay in the network is the main factor used for evaluation the transmission system efficiency. Therefore the routing problem is a common problem during network development. To control the traffic assignments a special model may be used. This model must reflect all principal properties of the real network. For adaptation to the current situation of the network the model should be easily updated and have facilities needed for simulation and optimization the network. The main network property is the fact hat the communication network is a queuing network. These ideas are utilized in the approach described below.

1. Mathematical model

Assume that in transmission system every node needs some time for pocket service. It is a service time t_s . Because $t_s \neq 0$ a queue of packets can build up in every node. The time a packet to waits in the queue depends on an average value and distribution of t_s and an average value and distribution of interarrivail times. t_i Let $\lambda = 1/t_i$ and $\mu = 1/t_s$. Assume that from every source note to the destination note more than one path can be selected (i.e. traffic from the source to the destination node can be split). We will consider only the essential paths between every source and destination node. (An essential path does not contain all the nodes of any other path.) For instance, let us consider the network of Fig. 1.

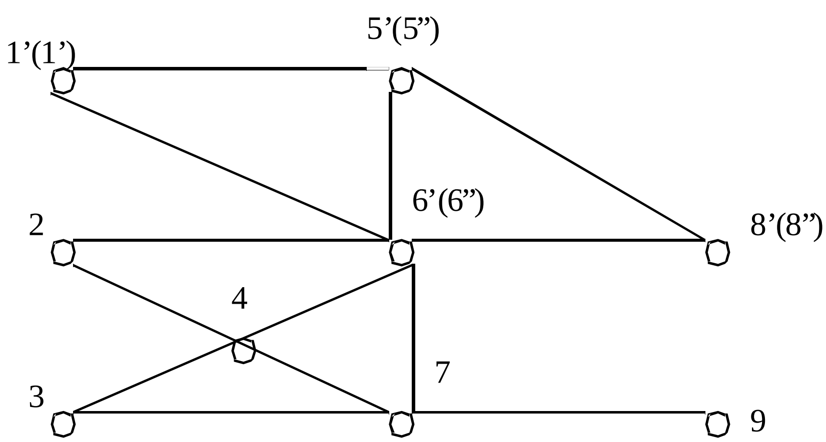

Fig. 1 Example network

This network has three source nodes (1, 2 and 3) and two destination ones (8 and 9). The nodes 2 and 3 generate only one kind of messages each, but then node 1 generates two kind of messages for the same destination node 8. (We will designate the messages as different if they have a different distribution of service time in any transmission node.) For the model development, one can use the idea of a logical path. By definition a logical path transmits only one kind of messages. Therefore, every physical path can contain one or more logical paths. For example, the path 158 contains the logical path 1'5'8' for the first kind of messages and the logical path 1’’5’’8’’ for the second kind of messages. The usual situation is when the same node belongs to different paths, i.e. it services different kind of messages (for example, the notes 4 and 6 of Fig.1)

Assume that note 1 generates the packet traffic for the note 8’ and $\Lambda_{1'8'}$ is the traffic rate for this notes. If the traffic can be split between two logical path 1’5’8’ and 1’6’8’ than

$$\Lambda_{1'8'} = \lambda_{1'5'8'} + \lambda_{1'6'8'} \qquad (1)$$

Let Ω_{AB} = {n | n is an essential logical path between the source node A and destination node B}, A,B $\in$ S, where S is the set of hosts in the network. If some source and destination notes are connected by only one essential path, than one or more nonessential logical path can be added. For the source node A and destination node B,

$$\Lambda_{AB} = \sum_{n \in \Omega_{AB}} \lambda_{AB}(n), \qquad (2)$$

where $\lambda_{AB}(n)$ is the fraction of the total traffic from A to B that has been assigned to the path *n*;

$$0 \le \lambda_{AB}(n) \le \Lambda_{AB} \qquad (3)$$

The entire network is

$$\Omega = \bigcup_{A,B \in S} \Omega_{AB} \qquad (4)$$

A static model with stationery traffic inputs and unchanging network configuration is considered. Under these conditions the average delay in path *n* is

$$\overline{T}_{AB}(n) = \sum_{z \in C_n} (\overline{t_{gz}} + \overline{t_{sz}}) + \overline{T_p}(n) \qquad (5)$$

Where

C_n = {*z* | *z* is a node, which belongs to the path *n*};

$\overline{t_{gz}}$ is the average value of the service time for the packets at the node *z* (for the packets are transmitted from A to B); the value of $\overline{t_{qz}}$ can be calculated in accordance with queuing theory [10];

$\overline{t_{qz}}$ is the average value of the service time for the *AB* packets at the note z: and

$\overline{T_p}(n)$ is there an average value of the total propagation delay for the path *n*.

The waiting time $\overline{t_{qz}}$ depends on the total traffic intensity ρ_z for the node z. For example the total traffic intensity for the note 6 of Fig. 1 is

$$\rho_6 = \lambda_{268}\mu_{268} + \lambda_{368}\mu_{368} + \lambda_{1'6'8'}\mu_{1'6'8'} + \ldots$$

Assume that

$$\overline{T_p}(n)\ \mathrm{p} \sum_{z\in C_n} (\overline{t_{qz}} + \overline{t_{sz}}) \qquad (6)$$

i.e. $$\overline{T_{AB}}(n) \equiv \sum_{z\in C_n} (\overline{t_{qz}} + \overline{t_{sz}}) \qquad (7)$$

Let the average value of the coast of the *n*-path utilization be.

$$\overline{Q_{AB}}(n) = R_{AB}(n)\overline{T_{AB}}(n)\overline{D_{AB}}(n) \qquad (8)$$

where $R_{AB}(n)$ represents the weight (cost) coefficient and $\overline{D_{AB}}(n)$ the average number of packets in paths *n*. Because [9]

$$\overline{D_{AB}}(n) = \mathrm{l}_{AB}(n)\overline{T_{AB}}(n).$$

it follows that

$$Q_{AB}(n) = R_{AB}(n)\,\mathrm{l}_{AB}(n)[\overline{T_{AB}}(n)]^2 \qquad (9)$$

Then, the average value of the cost for transmitting the packets from *A* to *B* is

$$\overline{Q_{AB}} = \sum_{n\in\Omega_{AB}} \overline{Q_{AB}}(n) \qquad (10)$$

The total average value of the cost of the network utilization is

$$\overline{Q} = \sum_{\Omega_{AB\in\Omega}} \sum_{n\in\Omega_{AB}} R(n)\,\mathrm{l}_{AB}(n)[\overline{T_{AB}}(n)]^2 \qquad (11)$$

The traffic flow assignment can be described by a vector:

$$V = \{\mathrm{l}_{12}(1), \mathrm{l}_{12}(2), ..., \mathrm{l}_{12}(h), ..., \mathrm{l}_{AB}(n), ..., \mathrm{l}_{YI}(g)\} \qquad (12)$$

Define the traffic flow assignment V^* to be optimal if

$$\overline{Q}(V^*) = \min_{\mathrm{l}_{AB}(n)} \overline{Q}(V). \qquad (13)$$

Because the function $\overline{t_{qz}}(\mathrm{r}_z)$ is nonlinear, the function $\overline{Q}(V)$ is nonlinear too; because the trffic intensity for every node depends on the traffic assignment for the whole network, the function $\overline{Q}(\mathrm{V})$ is a no separable function. We assume that $\overline{Q}(V)$ is a wellbehaved function admissible to nonlinear programming techniques. During optimization vector V may be changed as follows:

$$V_m = V_{m-1} + \Delta V. \qquad (14)$$

Where ΔV is the step size and m is the step number. Because of the communication network is a queuing network, the optimization procedure must be implemented within the feasible area, where the vector V_m provides a steady conditions for every note in the network.[10]:

$$\sum_{w \in \Omega_f} (\lambda_f(w)/\mu_f(w) < 1 \qquad (15)$$

with $\Omega_f = \{w \mid w$ is a path, which contains the f-node$\}$. Usually it is very difficult to find the initial feasible point for the optimization that provides the above conditions. To resolve the problem, one dummy path which contains only one node with $t=0$ and a very big value of $R_{AB}(n)$ may be added to a every Ω_{AB} in the network. This technique is similar to the penalty approach [10]. In the very beginning of the optimization procedure the whole traffic flow from a every source node can be assigned to the correspondent dummy path. The proposed method provides the condition (15) for the initial trial point. During optimization V_m must satisfy (2) and (15) for every step m. For this kind of model the shape of the objective function depends on the choices of $R_{AB}(n)$ and the current value of V_m. Because the shape is nonlinear and unknown in in advance an advanced optimization technique must be used. For the actual model an algorithm similar to that described below has been implemented.

1. Optimization algorithm

For the simplification goals it will be useful to change the description of V to

$$G_\tau = (X_{1\tau}, \ldots, X_{j\tau}, \ldots, X_{k\tau}) \qquad (16)$$

where $X_{j\tau}$ is the value of the j-th coordinate at τ – th step, $\tau = 1, 2, \ldots$

The optimization is divided into sub-algorithms. These sub-algorithms are linked and used depending on their efficiency. The first and main algorithm is based on utilization of a trial vector which has a random number of changeable variables during every realization. According to this algorithm the value of $X_{j\tau}$ is calculated by

$$X_{j\tau} = X_{j\Theta} + (\varepsilon_{j\tau})\Delta X_j \text{ if } Q(G_\Theta) \le (Q_{\tau-1}) \quad \text{and} \quad X_{j\tau} = X_{j\Theta+1} + (\delta_{j\tau})\Delta X_j \text{ if } Q(G_\Theta) > Q(G_{\tau-1}) = Q(G_{\tau+1}) \qquad (17)$$

where $Q(G_{\Theta+1})$ represents the value of the objective function after the $(\tau-1)$-th (last) successful step: the successful step is a step, that which provides the condition $Q(G_{\Theta}) \succ Q(G_{\tau-1}) = Q(G_{\Theta+1})$; $G_{\tau-1} = G_{\Theta+1}$ is a vector, which corresponds to this step;

$\varepsilon_{j\tau} = 1$ if $0 \le \eta \le 0.5P$; $\varepsilon_{j\tau} = 0$ if $1 \ge \eta \ge P$ and $\varepsilon_{j\tau} = -1$ if $P \ge \eta \ge 0.5P$
(18)

with η = [0,1] a random number, *P*=[0,1] the probability that X_j will be changed for this step, and ΔX_j the step size for the j-th variable; and

$\delta_{j\tau} = 1$ if $X_{j\Theta+1} \succ X_{j\Theta}$; $\delta_{j\tau} = 0$ if $X_{j\Theta+1} = X_{j\Theta}$ and $\delta_{j\tau} = -1$ if $X_{j\Theta+1} \prec X_{j\Theta}$
(19)

For the optimization a random generator is used and one can see from (18) that the number of variables that will be changed during the τ-th step up is determined by *P*. During optimization the value of *P* can be changed depending on the efficiency of search. The described approach helps to adapt the optimization procedure to the current shape of the objective function If during the optimization two successful steps have been implemented, than the next value $X_{j\tau}$ is calculated by (17) with

$\delta_{j\tau} = [Q(G_{\Theta-1}) - Q(G_{\Theta+1})]/2[Q(G_{\Theta-1}) - 2Q(G_{\Theta}) + Q(G_{\Theta+1)})]$ if $X_{j\Theta+1} \succ X_{j\Theta}$;
$\delta_{j\tau} = 0$ if $X_{j\Theta+1} = X_{j\Theta}$
$\delta_{j\tau} = -[Q(G_{\Theta-1}) - Q(G_{\Theta+1})]/2[Q(G_{\Theta-1}) - 2Q(G_{\Theta}) + Q(G_{\Theta+1)})]$ if $X_{j\Theta+1} \prec X_{j\Theta}$
(20)

Then, the optimization procedure continues with $\Delta X'_j = 2\Delta X_j$. This technique helps to accelerate the search when the shape it is a parabola. Sometimes the shape changes very sharply or very slowly. In this situation it is expedient to cut of the acceleration in (20) and continue the optimization procedure with a double step. For testing purposes the next inequality may be used:

$\Psi'' \ge [Q(G_{\Theta-1}) - 2Q(G_{\Theta}) + Q(G_{\Theta+1)})] > \Psi''$
(21)

where Ψ'' is the threshold to cut off acceleration if the shape changes very slowly and Ψ'' if the shape changes very sharply.

If after few successive successful steps an unsuccessful steps follows, then coordinates for the next step are calculated according to

$X_{j\tau} = X_{j\Theta+1} - \Delta X_j$ if $\{\{\, X_{j\Theta-1} - X_{j\Theta} < 0$ and $X_{j\Theta} - X_{j\Theta+1} \geq 0 \,\}$ or $\{\, X_{j\Theta-1} - X_{j\Theta} \geq 0$ and $X_{j\Theta} - X_{j\Theta+1} > 0 \,\}\}$;

$X_{j\tau} = X_{j\Theta+1}$ if $X_{j\Theta-1} = X_{j\Theta} = X_{j\Theta+1}$;

$X_{j\tau} = X_{j\Theta+1} + \Delta X_j$ if $\{\{\, X_{j\Theta-1} - X_{j\Theta} > 0$ and $X_{j\Theta} - X_{j\Theta+1} \leq 0 \,\}$ or $\{\, X_{j\Theta-1} - X_{j\Theta} \leq 0$ and $X_{j\Theta} - X_{j\Theta+1} < 0 \,\}\}$ (22)

This approach uses the idea that the current shape resembles a ravine. If the step is unsuccessful, than the procedure continues by rule (17). We will denote every trial value of P as $P_\beta\ (\beta = 1,2...)$. During optimization the two last values of P_β, which permit us to find one or more successful points, are memorize as $P_{\alpha-1}$ and P_α. When one or more successful steps follow a set of successive unsuccessful steps and their number exceeds a particular threshold $N_\sigma\ (\sigma = 1,2,...)$ the next value of P_β is calculated by

$$P_\beta = P_\alpha + \Delta P \quad \text{if} \quad P_\alpha = P_{\alpha-1} + \Delta P \quad \text{and} \quad P_\beta = P_\alpha - \Delta P \quad \text{if} \quad P_\alpha = P_{\alpha-1} - \Delta P$$
(23)

This approach helps us to use the accumulated information about shaped behavior. If after N_σ steps the value of P_β does not permit any successful steps, than the next value of $P_{\beta+1}$ is calculated by

$$P_{\beta+1} = P_\alpha + (-1)^\gamma (\Delta P)^\gamma, \gamma = 1,2,...,$$
(24)

where γ is incremented by 1 every time if no successful step was done. As a result, the value of P_β is changed periodically by $\gamma(\Delta P)$. This procedure continues until either one or more successful steps will be done (then we can assume $\gamma = 1$ and $P_\beta = P_\alpha$ or whole set of P will be tested. Then the procedure is switched to the value of P_α which provides the last successful step; but for the condition

$$\gamma = 1, \quad \Delta X_{j\pi} = \Delta X_{j\pi-1} / \varpi \qquad \varpi > 1$$
(25)

where π is the number of changes of ΔX.

The it is often the case that in an area close to the extremum point, the shape is almost flat and it is very difficult to find a successful point. In this situation the area around the last successful point needs to be tested carefully. How carefully depends on the value of N_σ. The large N_σ the more careful a search will be; on other hand the less often the value of

P will be changed the more time will be needed for the optimization. To create the fair conditions for all optimization stages the value of N_s can be controlled by

$$(N_\alpha / L_\alpha)^{N_s} \geq M = \text{constant},$$
(26)

where N_s is the number of successful steps realized with P_α during the α -th stage of optimization; L_α is the total number of steps at the same stage. Each stage of the search continues until the inequality (26) is valid. As a criterion to terminate the optimization, the inequality $X_{jp} \geq \varphi$ = constant (for example) may be used. The same algorithm can be utilized if the model is employed in real time mode. In this mode the optimization is a continuous procedure and information about network statues is updated currently

This optimization algorithm has been tested at this special set of test objective functions. The functions included in the set represent a wide spectrum of possible shapes. In [11,12] one can find a detailed description of each function and its shape. The universality of the search may be determined by the number of tests functions, for which the extreme points or their close neighborhood can be found. The efficiency of the optimization algorithm is evaluated by the number of steps needed to accomplish the search. The test results (see the Appendix) have shown that the proposed algorithm is universal and effective for all the test function. In addition, during the routing optimization the time needed to find $\lambda_{AB}^{\min Q(V)}(n)$ depends on the distance between the initial point and the point that provide the optimal value of Q, the choices of $R_{AB}(n)$, the network configuration, the service and interarrival rates.

The model developed here will be useful in the study of the stability and sensitivity properties of the network as well as in optimization. The first property shows to what limits the values of the network parameters can be changed upon the condition that the values dependable parameters will stay within described limits. The second property shows how the speed of variation of some parameters depend on the speed of variation of the other parameters. This information can be used together with information about the dynamic properties of the network for the calculation of different kind of thresholds, for the control and optimization procedures and for network improvement.

References.

[1] Gerla, M. Kleinrock, L., “Topological design of distributed computer networks”, *IEEE Transactions on Communications* 25 (1977) 48-60.
[2] Gerla, M. Cantor, D.G., “Optimal routing in a packet switched computer network” , IEEE Transactions on Computers 23(1974) 1062-1069.
[3] Agnew, C., “On the optimality of adaptive routing algorithms”, in :*Conf. Rec. Nat. Telecommun.*, 1974, 1021-1025.
[4] Gallager, R.G., “A minimum delay routing algorithm using distributed computations”, *IEEE Transactions on Communications* (1977) 73-75.

[5] Segal, A. “The modeling of adaptive routing in data communication networks” *IEEE Transactions on Communications* 25 (1977) 85-96.
[6] Flikop, Z “Optimal assignment of traffic flows into hierarchical communication network”, *Matematithceskie metody issledovania sistem*,
Kiev, 1971, 249-255
[7] Flikop, Z. “Random search as a means for queuing network optimization”, *Kibernetica* 4 (1969) 75-79.
[8] Flikop, Z. Dolotov, A., “Method of the system optimization”, Voprosy ekonomiki I upravlenia v stroitelnom proizvodctve”, Kiev, 1978.
[9] Tanenbaum, A., “Computer Networks” , Prentice-Hall, Englewood Cliffs, NJ. 1981.
[10] Wagner, H., *Principles of Operation Research*,. Prentice-Hall, Englewood Cliffs, NJ. 1969.
[11] Zakcharov, V., “Ten test functions for the optimization”, *Avtomatica i vitchilitelnaia technical* 6 (1974).
[12] Lazarev, I., Valujsky, V., “Experimental comparison the random optimization algorithms”, *Avtomatica i vitchilitelnaia technical* 2 (1974).

Appendix

No.	Objective functions, constrains, initial points	Total # of steps	No.of success steps	Value obtined
1	$Q=\sum_{i=1}^{10} X_i^4, X^0=(1,...,1)$	358	80	10^{-7}
2	$Q=(X_1-X_2)^2+1/9(X_1+X_2-10)^2, X^0=(0,-1)$	47	20	0.0
3	$Q=\sum_{i=1}^{10}(1/i)X_i^2, X^0=(1,...,1)$	750	149	10^{-7}
4	$Q=\sum_{i=1}^{10} X_i^2+(\sum_{i=1}^{10}(i^{0.5})X_i^2)^2+(\sum_{i=1}^{10}(i^{0.5})X_i)^4, X^0=(0,1,...0,1)$	1000	162	10^{-7}
5	$Q=(X_1-10X_2)^2+5(X_3-X_4)^2+(X_2-2X_3)^4+10(X_1-X_4)^4, X^0=(3,-1,0,1)$	681	296	0.041
6	$Q=10^{-4}(X_1-3)^2-(X_2-X_1)+\exp[20(X_2-X_1)], X^0=(0,-1)$	25	15	0.19978
7	$Q=100(X_2-X_1^2)^2+(1-X_1)^2, X^0=(1.2,1)$	683	348	0.057
8	$Q=100(X_2-X_1^3)^2+(1-X_1)^2, X^0=(1.2,1)$	683	364	0.037
9	$Q=100(X_3-10\mathrm{b})^2+(q-1)^2+X_3^2, \mathrm{b}=arctg(X_2/X_1), q=(X_1^2+X_2^2)^{0.5}, X^0=(-1,1,1)$	681	296	0.041
10	$Q=20(X_2-X_1^2)^2+(1-X_1)^2, X^0=(1.2,1)$	2550	1167	10^{-7}
11	$Q=1-\exp[-10^{-4}\sum_{i=1}^{5} X_i^2], X^0=(23.3,...,23.3)$	174	48	10^{-7}
12	$Q=(X_1+5)^2/(10+X_2^2)$ at $X_2^2-[\sin(0.5X_1+\mathrm{p})^2+1.05]^2\le 0, X^0=(14,0.1)$	55	34	0.0
13	$Q=\sum_{i=1}^{10} 5*10^{-4}(X_i^4+1000)^2$ at $X_j-\sum_{i=1, i\ne j}^{10} 3*10^{-4}X_i\ge 0,$ X^0=(2.797,11,19,2.797,11,19,2.797,11,19,…,11,19)	309	79	5007.2
14	$Q=\sum_{i=1}^{50} i^2(\sum_{j=1}^{i} jX_j^2-0.81)^2+(X_1-0.9)^2+\sum_{i=2}^{50} i; X^0=(-1,0,...0)$	8000	1478	0.0