Yugoslav Journal of Operations Research
4 (1994), Number 1, 43–66

A STUDY OF ROUTING AND FAIRNESS ALGORITHMS IN VOICE–DATA NETWORKS

Ioannis GRAGOPOULOS, Fotini–Niovi PAVLIDOU

Aristotle University of Thessaloniki Dept. of Electrical Engineering
P.O. Box 1641, 54006 Thessaloniki, Greece

Abstract: A comparative study on mixed dynamic routing techniques is presented. Both virtual circuit and datagram techniques are applied on voice–data networks and an extended analysis of the different performance parameters is made through adequate simulation techniques. Fairness strategies are also examined and the obtained results are discussed and evaluated.

Keywords: Routing, datagram, voice–data network.

1. INTRODUCTION

Routing algorithms have been strongly investigated during the past two decades since they constitute an important issue in the design of communications networks, affecting several performance measures of interest. By routing we mean a set of decisions in order to provide the best collection of paths between source and destination, given the traffic requirements and the network configuration.

Two broad classes of routing techniques have been applied to communications networks, that is, 1) circuit–switching hierarchical or nonhierarchical routing [11, 4, 9] used mainly on traditional telephone networks and 2) packet–switching techniques used on data networks [6, 16, 17]. Algorithms of both categories can further be classified as centralized, or decentralized ones according to the way paths are established or they can be classified as static or dynamic/adaptive according to the way paths are changed or retained. The mathematical background of all these techniques includes heuristics [14] as well as optimization techniques [5, 1, 13] with the purpose of studying the main performance criteria, which are the average packet delay, the blocking probability and the throughput (transmitted load) as functions of the offered load. Sometimes fairness criteria may be examined, too, for the fair servicing of all subscribers on the network.

In recent years we have seen increased research activities towards hybrid routing techniques appropriate for mixed voice–data networks, but the studies concentrate mainly on voice–data nonhierarchical fully connected, symmetric networks [19] using deterministic alternate routing. The ATM (Asynchronous Transfer Mode) has been also accepted as a very promising technique for integrated services broadband networks [12, 8]. On the other hands as far as we know there are no comparative studies on hybrid dynamic routing techniques suitable for the currently mixed media networks.

The contribution of this paper consists of the application of both the virtual circuit and datagram techniques on voice–data networks and the extended analysis of the different performance parameters through adequate simulation techniques.

In the second section the mathematical formulation of the problem is given. The datagram algorithm, used for data traffic, is briefly described since it has been well analyzed in previous papers. Following, a virtual circuit finding (VCF) algorithm is proposed which results in a set of virtual circuits of any source node to a given destination, the VC's being sorted in ascending order of the number of hops to the destination. This algorithm constitutes the core of a modified flow deviation (FD) technique used for the voice traffic.

The mixed routing problem is then introduced, defining the different possible resource allocation strategies which can be selected. Next, the fairness and flow control algorithm proposed in [7] is applied to the voice traffic, after some necessary modifications.

In the third section, the performance analysis of the network is studied in case that the previous algorithms were all concurrently applied on the network and the influence of the different performance parameters on the optimisation procedure is thoroughly examined. Finally, conclusion remarks and suggestions for further work are given in the last section of the work.

2. PROBLEM FORMULATION

We assume a network of N nodes and L directed, full–duplex links. The ordered pair (i, k) denotes the directed link from node i to node k. We are interested in the case of two kinds of traffic on the network, that is, voice traffic routed with virtual circuit techniques and burst data traffic routed with datagrams. Our aim is to choose the proper algorithms for both kinds of traffic and run them simultaneously on the network investigating their interactions. We describe in the following the algorithms applied, all of which are based on optimization nonlinear programming methods.

Datagram Traffic

A multicommodity flows model and especially a gradient projection technique has been chosen for the modeling of datagram routing. This is the well–known model of Newton's method based on the use of the second derivatives of the objective function

[1]. The choice of the algorithm has been made because of its robust description and the improved rate of convergence near the optimum point.

In the algorithm, the studied measure (the total mean packet delay on the network D_T^d) is given by

$$D_T^d = \sum_{(i,k)\in L} D_{ik}^d(f_{ik}^d) \tag{1}$$

under the constraints $$\sum_k f_{ik}^d - \sum_l f_{li}^d = r_i^d \quad \forall i \tag{2}$$

$$f_{ik}^d \ge 0 \quad \text{and} \quad f_{ik}^d \le C_{ik} \tag{3}$$

Where $D_{ik}^d(f_{ik}^d)$ is the average delay on link (i, k).

f_{ik}^d is the total traffic on link (i, k) in packets/sec.

C_{ik} is the capacity of the link (i, k) in kbits/sec.

r_i^d is the expected traffic entering the network at node i and destined to node j (packets/sec).

The uppercase index d (D^d, f^d) denotes the datagram traffic. For simplicity of notation we shall use it only in the presentation of D_{ik}^d and D_T^d.

The algorithm is described for only one destination node. Its application for many destinations is achieved by a simple repetition of the algorithm.

Let t_i be the total incoming traffic at node i (entering the network or transit from neighbor nodes) that is

$$t_i = r_i + \sum_m f_{mi}$$

We define $\phi_{il} = f_{il}/t_i$ the fraction of flow t_i that travels on link (i, l). These variables are used as the independent variables in the model studied.

In the mathematical analysis we need the first and second derivatives of D_{ik} with respect to ϕ_{ik} and r_i which are given by [3].

$$\frac{\partial D^d}{\partial \phi_{ik}} = t_i\,\delta_{ik} \tag{4}$$

$$\frac{\partial D^d}{\partial r_i} = \sum_k \phi_{ik}\,\delta_{ik} \tag{5}$$

where $$\delta_{ik} = \dot{D}_{ik}^d + \frac{\partial D^d}{\partial r_k} \tag{6}$$

and $$\frac{\partial^2 D^d}{\partial \phi_{ik}^2} = t_i^2\left[\ddot{D}_{ik}^d + R_k\right] \tag{7}$$

with $R_k = \sum (\phi_{lm})^2 \ddot{D}^d_{lm} + \left[\sum_{m'} \phi_{lm'} \delta_{lm'}\right]\left[\sum_{m} \frac{\phi_{lm}}{\delta_{lm}} R_m\right]$ (8)

$\dot{D}^d_{ik}$ and $\ddot{D}^d_{ik}$ being the first and second derivatives of D^d_T with respect to f_{ik}.

Our aim is to minimize D^d_T and this is achieved by Newton's method iteratively. At each iteration k the new values of the independent variables ϕ_{il} are given by

$$\phi^k_{il} = \max\left\{0,\ \phi^{k-1}_{il} - \frac{\lambda(\delta_{il} - \mu_i)}{t_i(\ddot{D}^d_{il} + R_l)}\right\} \tag{9}$$

where λ is a stepsize studied in detail in the following and is μ_i a Lagrange multiplier determined by

$$\sum \max\left\{0,\ \phi^k_{il} - \frac{\lambda(\delta_{il} - \mu_i)}{t_i(\ddot{D}^d_{il} + R_l)}\right\} = 1 \tag{10}$$

In our applications the frequently used formula of $M/M/1$ has been examined for $D^d_{ik}(f_{ik})$, that is

$$D^d_{ik} = \frac{f_{ik}}{\mu C_{ik} - f_{ik}} \tag{11}$$

with $1/\mu$ being the message length in bits per second.

Furthermore another choice with more user oriented performance has been proposed [10] and this is

$$D^d_{ik} = \log \frac{1}{1 - f_{ik}/C_{ik}} \tag{12}$$

The numerical results of both applications are given in the next section.

Virtual Circuit Traffic

For the case of voice traffic the gradient projection approach can be used as well [17] but the combination of these two algorithms has been found to be very cumbersome and slow so, we apply a modification of the already known flow deviation algorithm [5]. "Modification" denotes that our model is working on the path flows instead of link flows. In order to achieve more balanced traffic on the network we used a shortest path circuit through any node neighbor to the one examined. Our aim is now to minimize the mean virtual circuit delay D^v_T which is the sum of the delays of all the virtual circuits. The link delay D^v_{ik} is given again by Equations (11,12) where f_{ik} is now the flow on the link (i,k) due to all the VC's which include this node in their paths. The core of this FD algorithm is a Virtual Circuit Finding Algorithm, which finds and sorts the VC' s in the network.

Virtual Circuit Finding (VCF) Algorithm

step 1 Define the destination node j.

step 2 Create the VC' s tree as follows: The root of the tree (level 1) is the destination node. Assign the neighbors nodes of the destination node to the second level. Continuing similarly assign the neighbors of the neighbors to the next level. The depth of the tree gives the longest VC with respect to the number of hops.

step 3 Set the origin node $i = 1$.

step 4· Set the level number $l = 2$.

step 5 Starting from level l try to find node i in this level. After finding this node, descend the tree down to the root. All the nodes found on this route constitute a VC with origin node i and destination node j. The number of node' s i VCs with hops l–1 is equal to the times we encountered node i in level l.

step 6 Set $l = l + 1$ and go to step 5.

step 7 Set $i = i + 1$ and go to step 4.

The result of the algorithm is a set of all the VC' s between any source–destination pair ordered according to their hops length. The algorithm is described only for one destination node as indicated in the Diagram 1. (The square boxes represent source codes, the circles the results values and the squares with bent edges denote various program parameters).

Modified Flow Deviation Algorithm

step 1 Find all the VC' s to a destination j using the previous VCF algorithm.

step 2 Find an initial feasible flow and a value of D_T^V using an algorithm of [5]. Set K=0.

step 3 Find the first derivative of D_{ik}^V with respect to f_{ik}. This is the link length. The length of the VC is the sum of its link length. Define the shortest–path virtual circuit (with respect to the number of hops), for any source node, through every one of its neighbor nodes. Thus, for any source node we have not only one shortest–path virtual circuit, but one for every attached node achieving a more balanced traffic on the network.

step 4 Find the shortest–path virtual circuit (with respect to the link length) for any source–destination node.

step 5 i) Set node $i = 1$.

ii) For the source node i use the flow deviation method,

$$f_{ik} = f_{ik} + \lambda(\bar{f}_{ik} - f_{ik})$$

to deviate an amount of flow from all its virtual circuits to its shortest-path circuit, where $\bar{f}_{ik}$ is the shortest path flow.

iii) Find again the shortest-path virtual circuit (with respect to the link length) for any source-destination node, using subroutines of step 3 and step 4 without exiting step 5.

iv) Set $i = i + 1$. If $i = N$ go to step 6, else go to 5i).

step 6 Find the new D_T^V. If $D_T^{V^{(K)}} - D_T^{V^{(K+1)}} < \varepsilon$, where ε is a small positive number, stop, otherwise set $K = K + 1$ and go to step 3.

The input and output data of the FD algorithm are given into the Diagram 2.

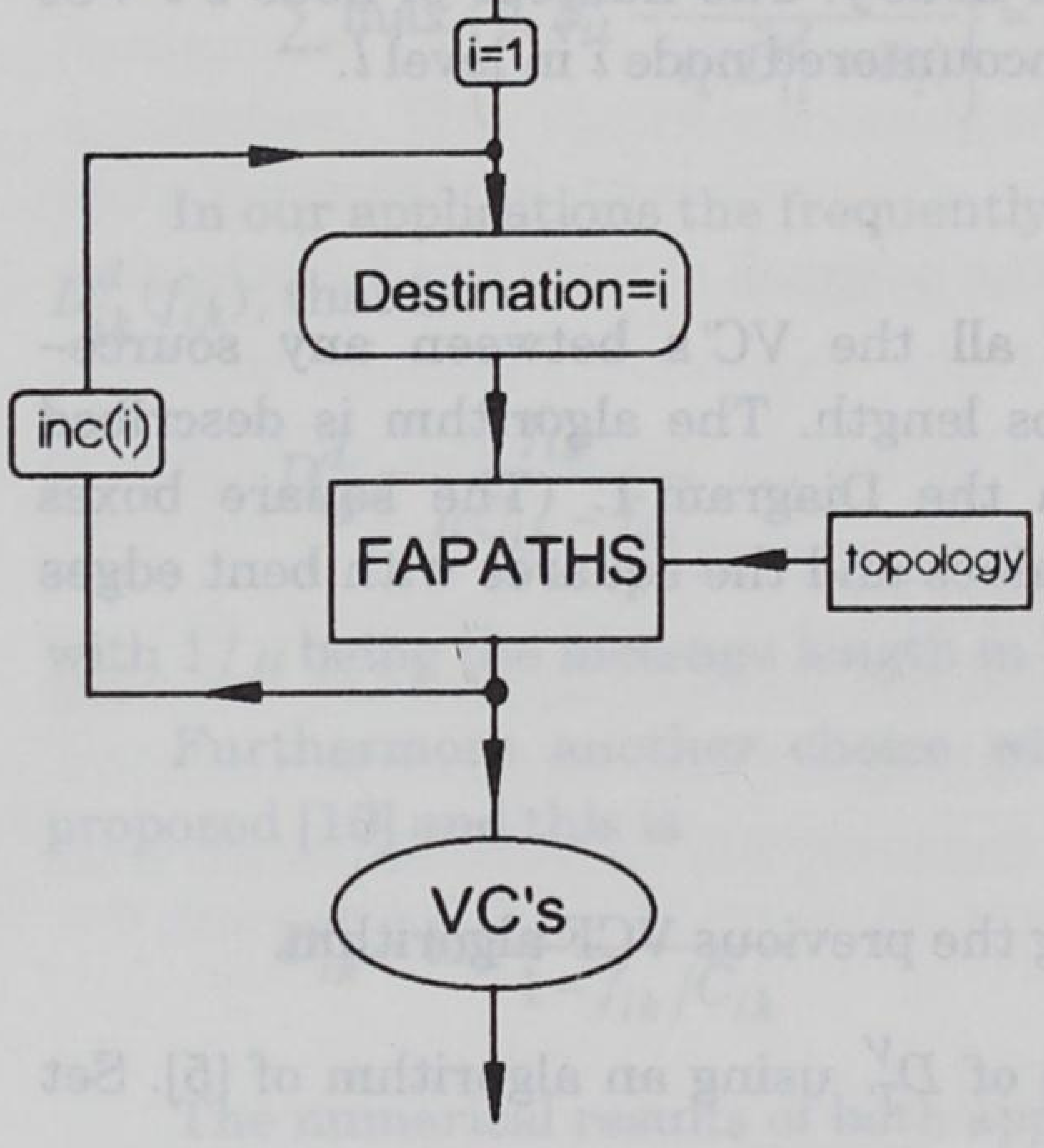

Diagram 1. Flow chart of VCF algorithm

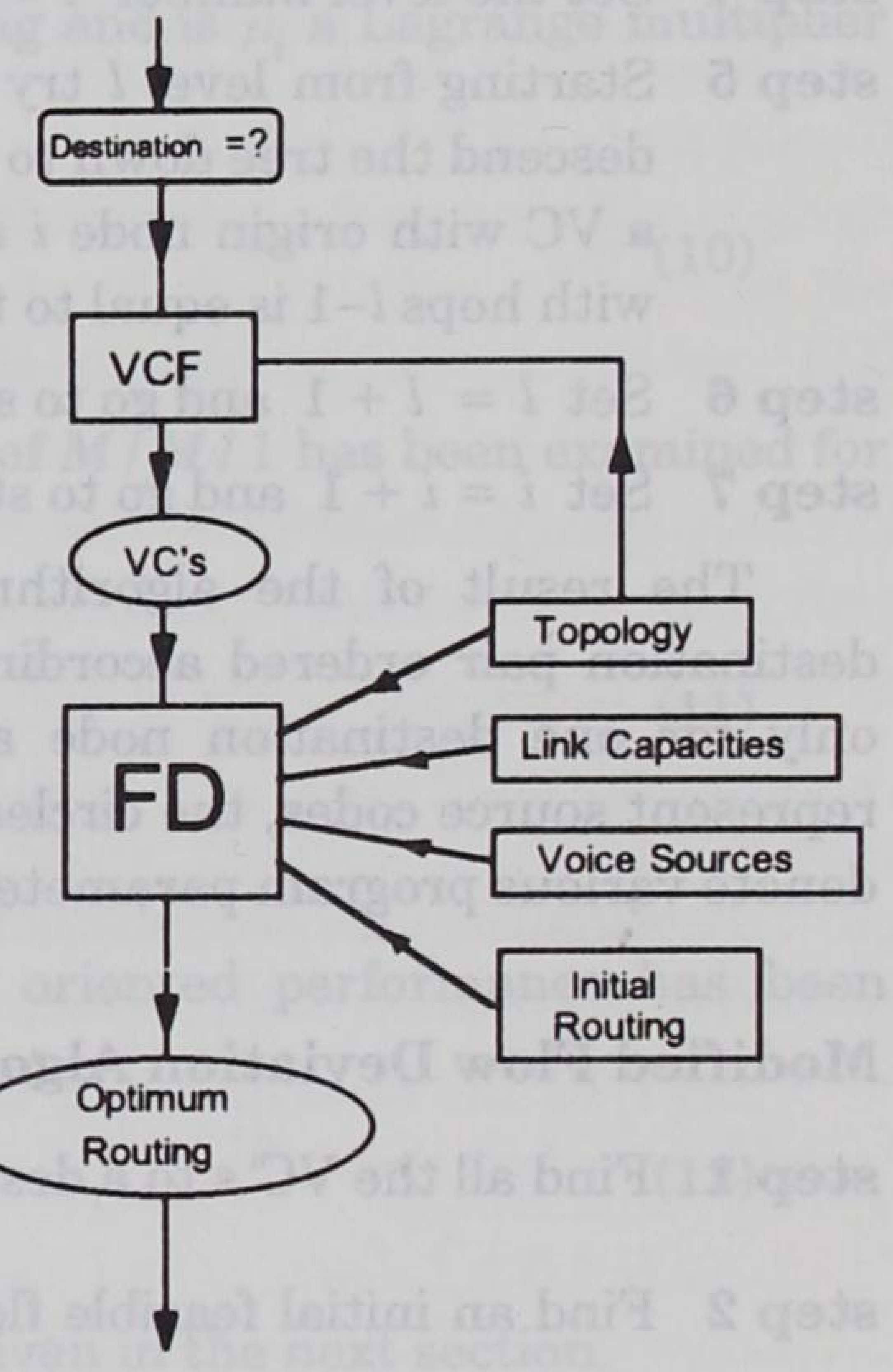

Diagram 2. Flow chart of MFD algorithm

Mixed Routing

The above techniques must be used simultaneously on the network. Since both kinds of traffic share the same network resources a sharing scheme must be defined. The usually applied capacity allocation schemes are: 1) A totally fixed sharing scheme where every kind of traffic occupies constantly a well-defined portion of the link capacity and if this is full, the new packets are blocked; this scheme is simple but not effective, 2) A totally free movable boundary scheme with nonpreemptive or preemptive priority of voice traffic. This is the real routing scheme but the nonpreemptive case is very difficult to analyze. So, in the following we examine the preemptive case and some special cases of the nonpreemptive one.

In our study we have adopted the second model. Throughout the work we pay more attention to the voice traffic since it is more regular in operation, with long sessions, while datagram is a bursty, short duration, traffic. Under the sharing schemes the data delay Equation (11) is modified in

$$D_{ik}^{d} = \frac{1}{\mu\left[C_{ik} - f_{ik}^{v}\right] - f_{ik}^{d}} \qquad (13)$$

and the Equation (12) is changed in a similar manner.

Since f_{ik}^{v} and f_{ik}^{d} are independent processes their crossderivatives are taken to be zero. So the form of datagram formulas (4–10) remains the same.

In order to achieve a tractable analysis of the mixed routing case we assume that the input of the datagram traffic is made at semirandom points during the FD iterations. Actually, the datagram inputs are entered to the network every five iterations of the FD algorithm. Under this approximation we study by simulation the common performance of the two algorithms. In Diagram 3 we can see how the cooperation of the two algorithms (FD, Datagram) is achieved.

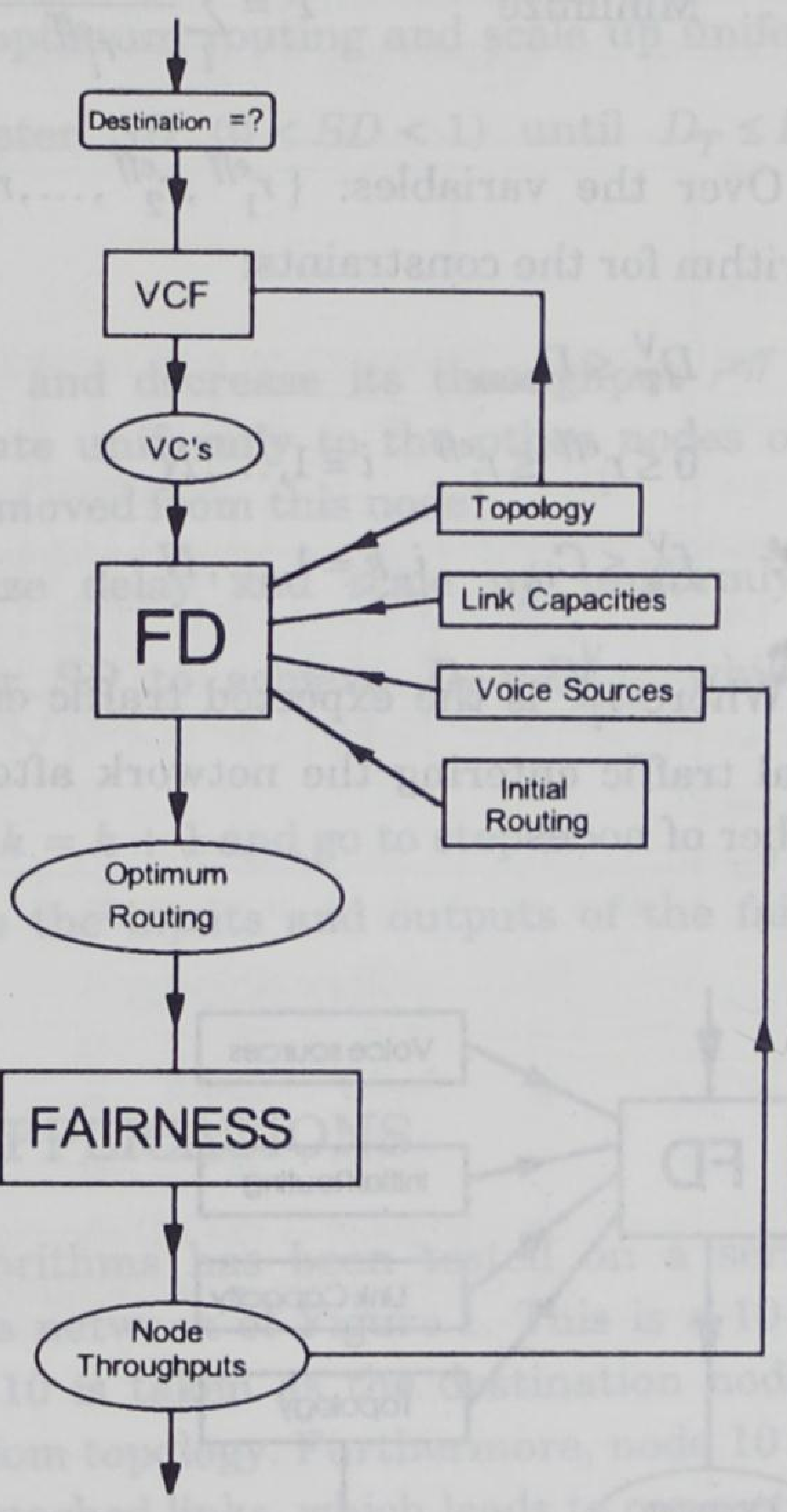

Diagram 3.

Fairness and Flow Control

Recently a few contributions [18, 15, 7] have been made towards achieving integration of routing and flow control for the purpose of avoiding deadlocks and congestion through optimum resource allocation. In a mixed routing network the flow control and fairness issues are very important for its proper operation. Since we have decided to pay attention to voice traffic, we incorporated a fairness technique in the virtual circuit algorithm.

The model adopted [7] aims to a direct input rate limitation in order to support equal node throughputs and equal delays for the different paths, holding the total mean delay of the network under a maximum value ($D_{\max}$). The problem of "fair" routing and flow control is stated in [7]:

$$\text{Minimize} \qquad P = \sum_i \frac{r_i^V - r_i^{eff}}{r_i^{eff}}$$

Over the variables: $\{ r_1^{eff}, r_2^{eff}, \ldots, r_N^{eff} \}$ and taking into consideration the FD algorithm for the constraints:

$$D_T^V \le D_{\max}$$

$$0 \le r_i^{eff} \le r_i^{eff} \quad i = 1, \ldots, N$$

$$f_{ik}^V \le C_{ik} \quad i, k = 1, \ldots, N$$

Where r_i^V is the expected traffic entering the network at node i and r_i^{eff} is the actual traffic entering the network after the flow control mechanism and N is the number of nodes.

The necessary condition for the minimization of [7] is to find equal values for voice traffic (for simplicity we shall drop the index V) for all

$$\nabla i = \frac{(r_i^{eff})^2 w_i}{r_i}$$

where $\quad w_i = \dfrac{\partial D_T^V}{\partial r_i^{eff}}$

The quantity w_i in some sense indicates how much congestion will be caused in the network by an increase in node's i throughput. It can be seen from [7] that transferring traffic from a node j to a node k, the objective function P can always be improved when ∇j > ∇k.

The algorithm proposed in [7] was incorporated in our FD algorithm. The main steps for the iteration k are the following.

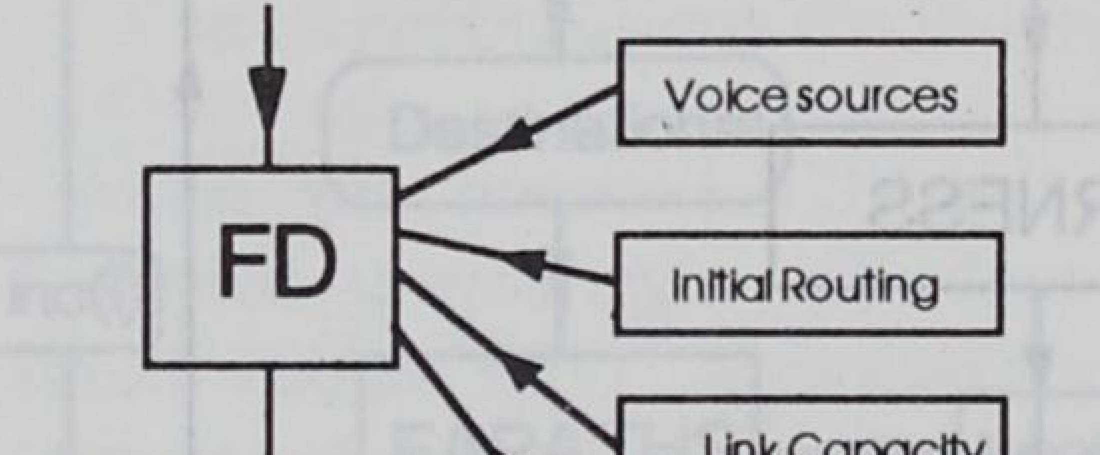

Diagram 4.

step 1 Set $k = 0$. Let $r_i^{eff^{(0)}}$ be the initial throughputs of all nodes in the network.

step 2 Use the FD algorithm to obtain an optimum routing and scale up uniformly the throughputs r_i^{eff} by a parameter SD $(0 < SD < 1)$ until $D_T \leq D_{\max}^V$. Compute $P^{(0)}$.

step 3 Compute w_i and ∇_i.

step 4 Find the node with the highest ∇, and decrease its throughput r^{eff} by a parameter SS $(0 < SS < 1)$. Distribute uniformly to the other nodes of the network, the throughput that was removed from this node.

step 5 Use the FD algorithm to minimize delay and scale up uniformly the throughputs r_i^{eff} by the parameter SD to achieve $D_T = D_{\max}^V$ while the routing is optimum. Compute $P^{(k)}$.

step 6 If $P^{(k)} - P^{(k-1)} < \varepsilon$ stop, otherwise put $k = k + 1$ and go to step 3.

In the flow chart of Diagram 4 we give the inputs and outputs of the fairness algorithm.

3. NUMERICAL APPLICATIONS

The performance of the proposed algorithms has been tested on a series of network designs. We give the results for the network of Figure 1. This is a 10–node network with $C_{ik} = 64$ Kbits/sec. The node 10 is taken as the destination node. We select this network design because of its random topology. Furthermore, node 10 is the worst case of destination because of its few attached links, which leads to congestion to the right part of the network.

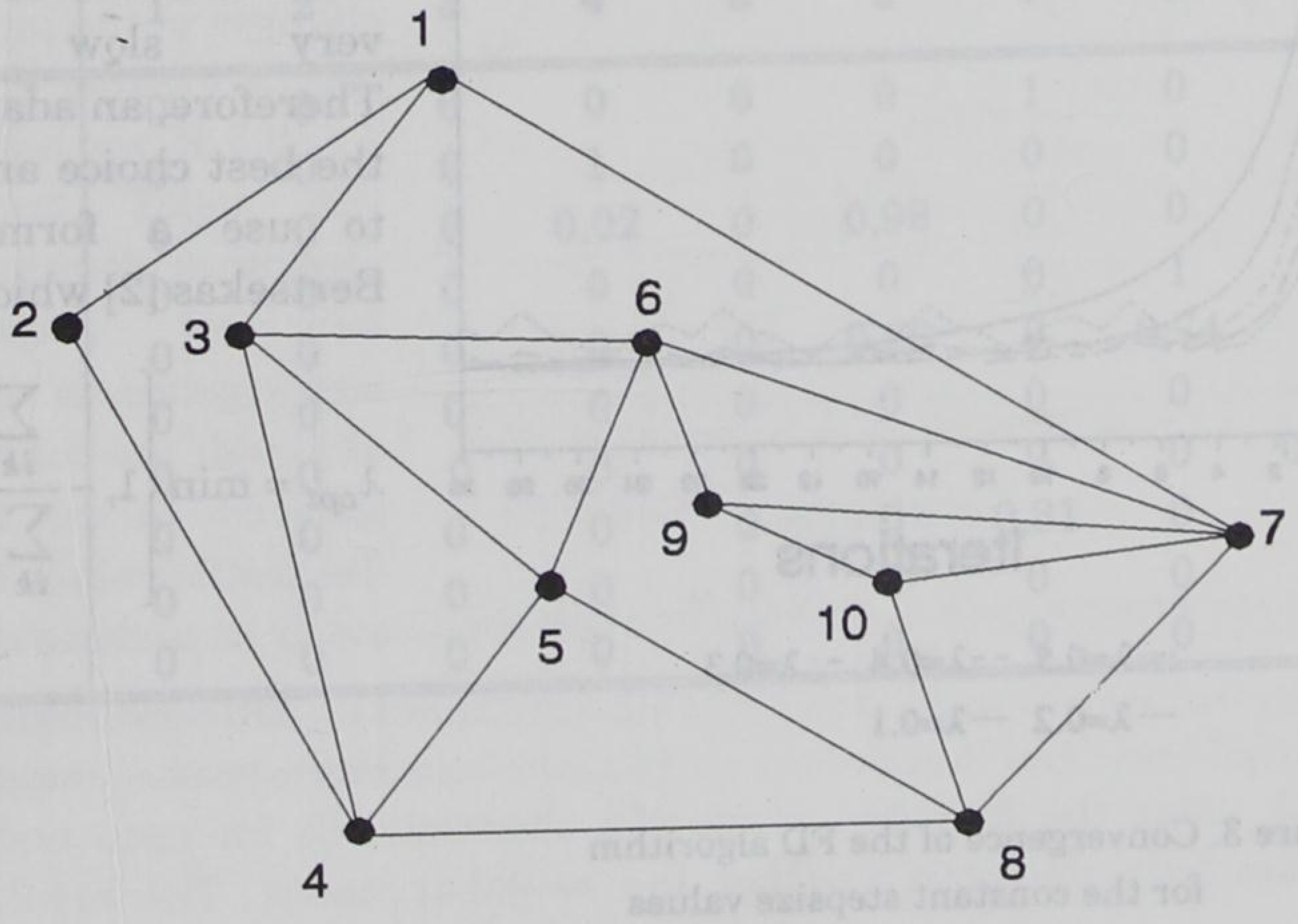

Figure 1. The applications network

Stepsize selection

Since the performance of optimization algorithms is greatly affected by the proper selection of the stepsize λ, we study first the stepsize behavior in three cases:

a) voice traffic only,

b) voice traffic with abrupt changes in the incoming flows

c) mixed routing of virtual circuits and datagrams.

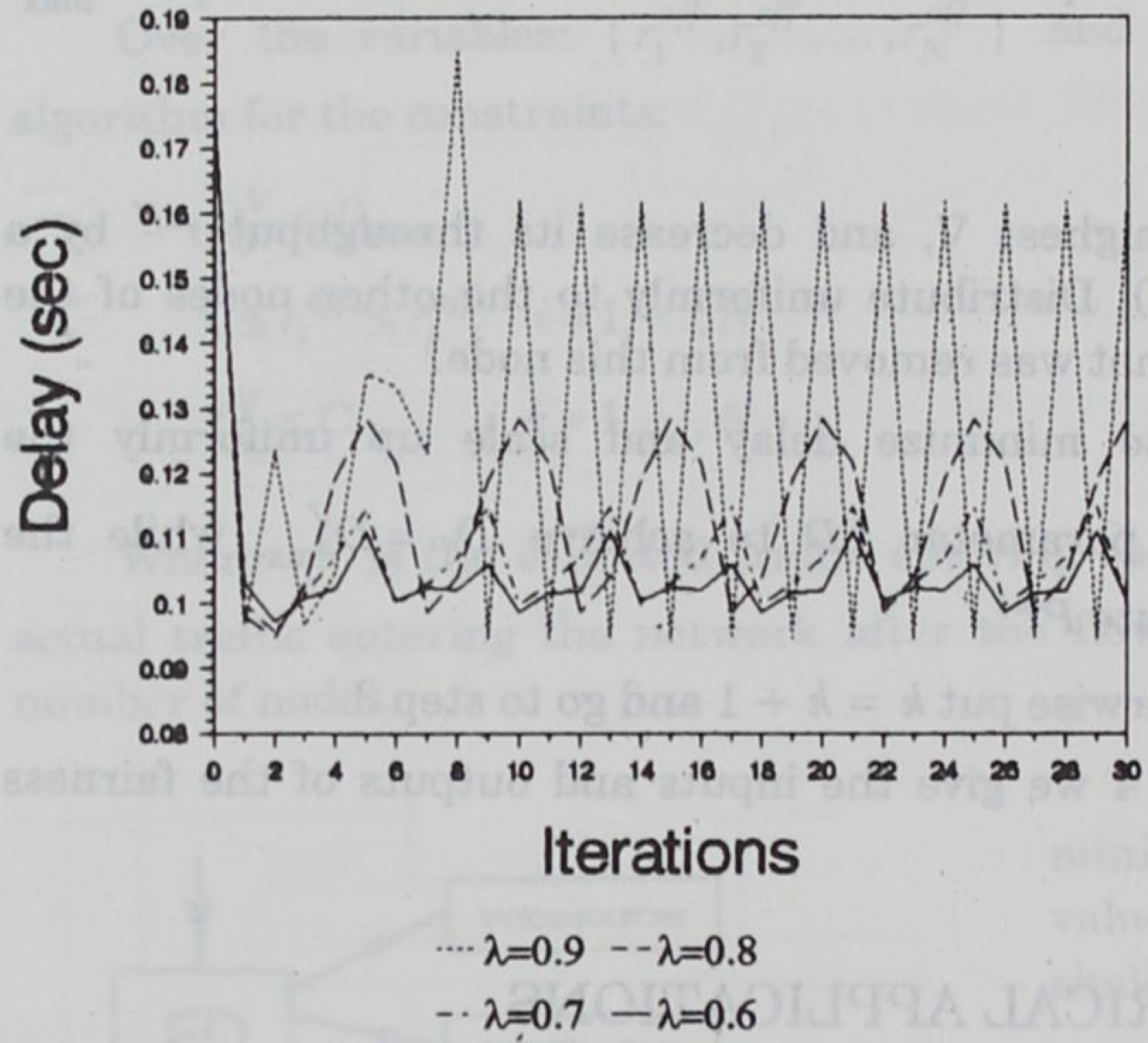

Figure 2. Convergence of the FD algorithm for the constant stepsize values

In Figures 2 and 3 are given the results for the modified FD algorithm, with voice traffic only on the network for various constant values of the stepsize in the real space (0, 1).

In Table 1 we give the voice inputs in packets/sec. The initial and the final routing patterns are given in Table 2. We observe that a large stepsize is desirable in the first steps of the algorithm since it proceeds faster to the minimum point but near the optimum point it leads to oscillations and convergence is almost impossible. On the other hand a small stepsize leads to very slow convergence. Therefore, an adaptive stepsize is the best choice and we preferred to use a formula given by Bertsekas [2] which is

$$\lambda_{opt} = \min\left\{1, -\frac{\sum_{ik}(\bar{f}_{ik} - f_{ik})\dot{D}_{ik}^{V}}{\sum_{ik}(\bar{f}_{ik} - f_{ik})^{2}\ddot{D}_{ik}^{V}}\right\}$$

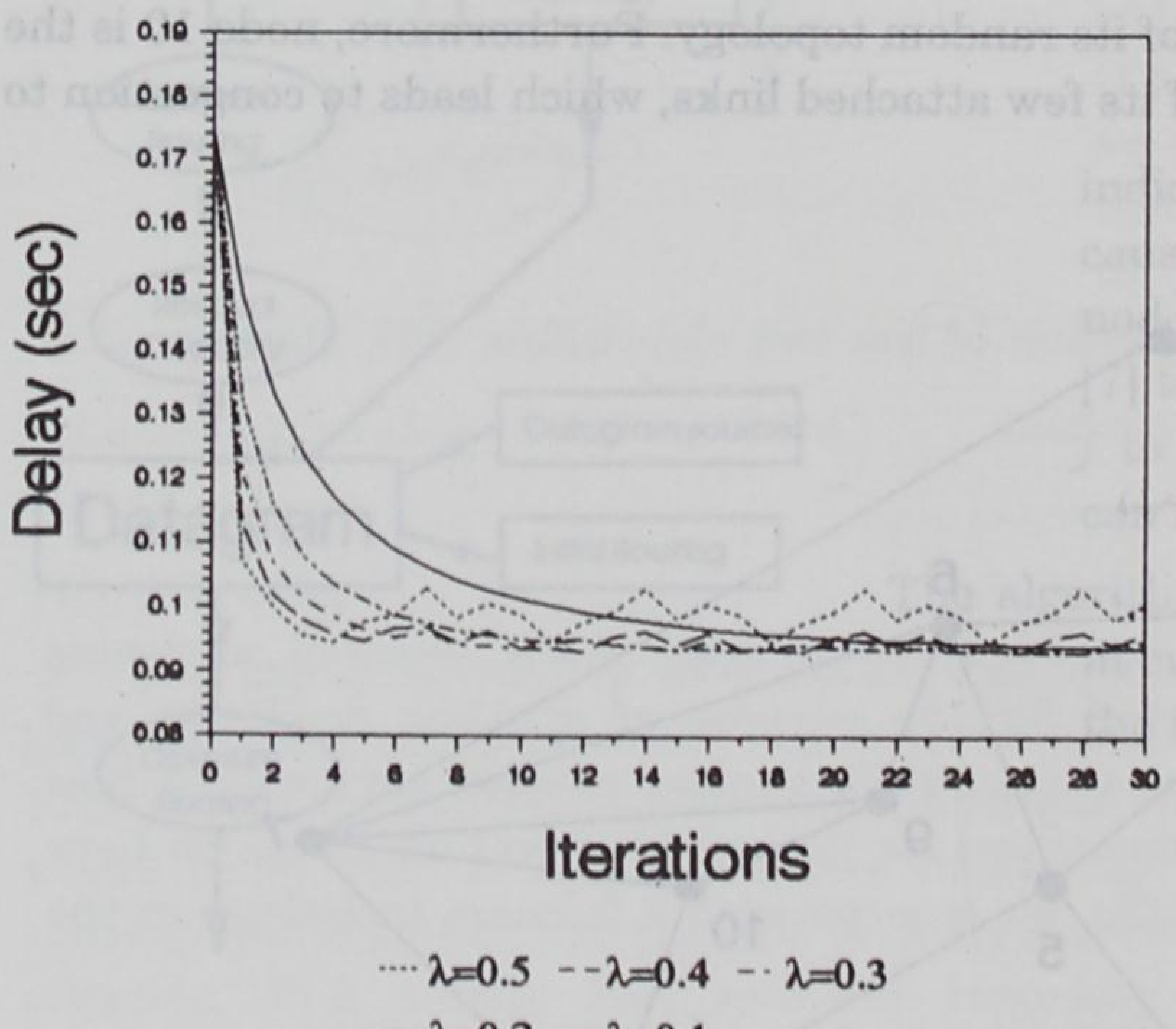

Figure 3. Convergence of the FD algorithm for the constant stepsize values

Table 1. Input voice traffic in packets/sec

Nodes	1	2	3	4	5	6	7	8	9	10
Voice input in packets/sec	9	15	9	15	7	10	10	10	8	0

Table 2. Percentage of traffic exchanged between any source-destination pair

Initial Routing Pattern

Destination Source	1	2	3	4	5	6	7	8	9	10
1	0	0,2	0,3	0	0	0	0,5	0	0	0
2	0	0	0	1	0	0	0	0	0	0
3	0	0	0	0,4	0	0,6	0	0	0	0
4	0	0	0	0	0	0	0	1	0	0
5	0	0	0,3	0,3	0	0,2	0	0,2	0	0
6	0	0	0	0	0	0	0,4	0	0,6	0
7	0	0	0	0	0	0	0	0	0,2	0,8
8	0	0	0	0	0	0	0,4	0	0	0,6
9	0	0	0	0	0	0	0	0	0	1
10	0	0	0	0	0	0	0	0	0	0

Final Routing Pattern

Destination Source	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	1	0	0	0
2	0	0	0	1	0	0	0	0	0	0
3	0	0	0	0,02	0	0,98	0	0	0	0
4	0	0	0	0	0	0	0	1	0	0
5	0	0	0	0	0	0,26	0	0,74	0	0
6	0	0	0	0	0	0	0	0	1	0
7	0	0	0	0	0	0	0	0	0,97	0,03
8	0	0	0	0	0	0	0,81	0	0	0,19
9	0	0	0	0	0	0	0	0	0	1
10	0	0	0	0	0	0	0	0	0	0

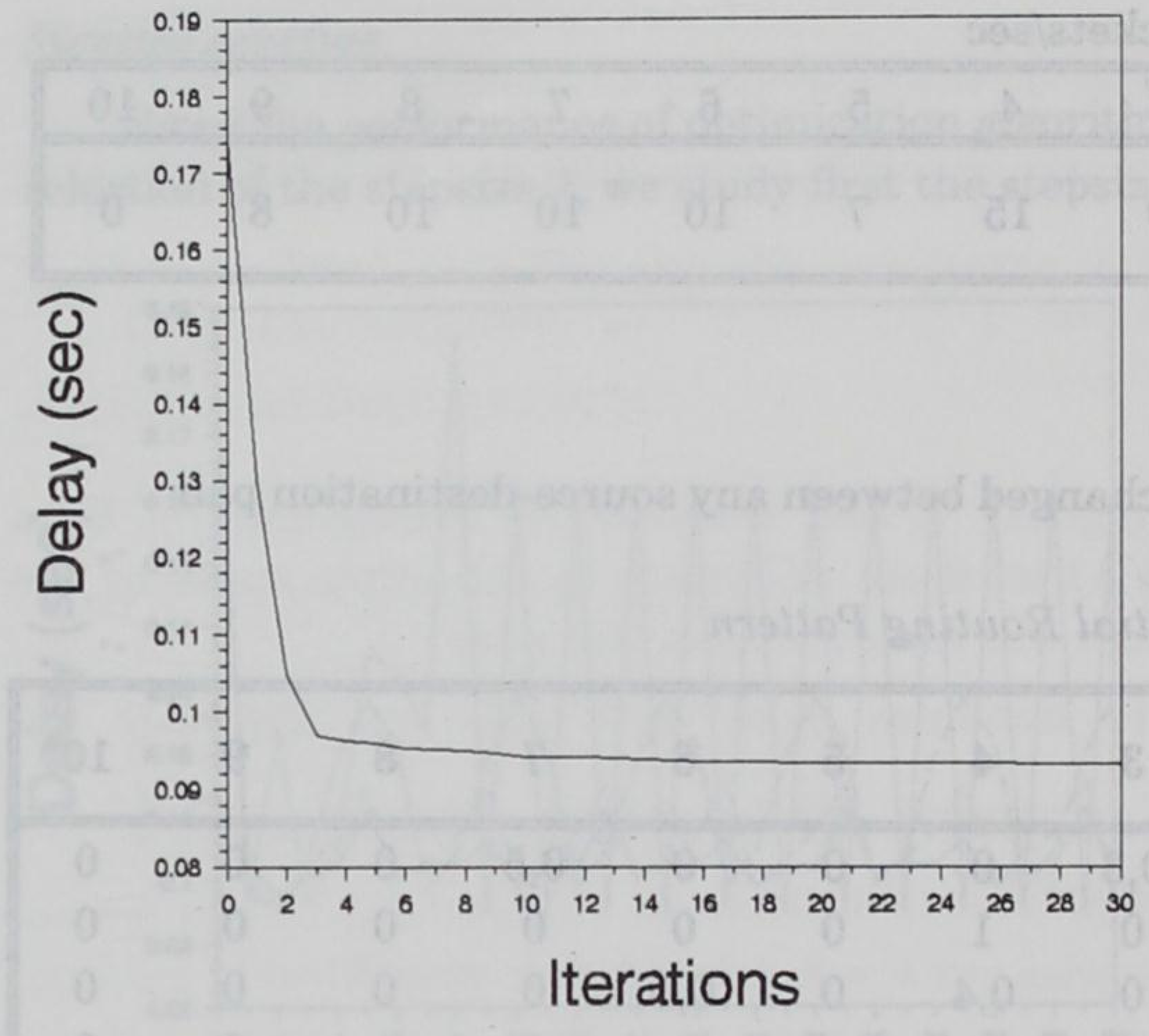

Figure 4. The convergence of the FD algorithm with an adaptive stepsize λ

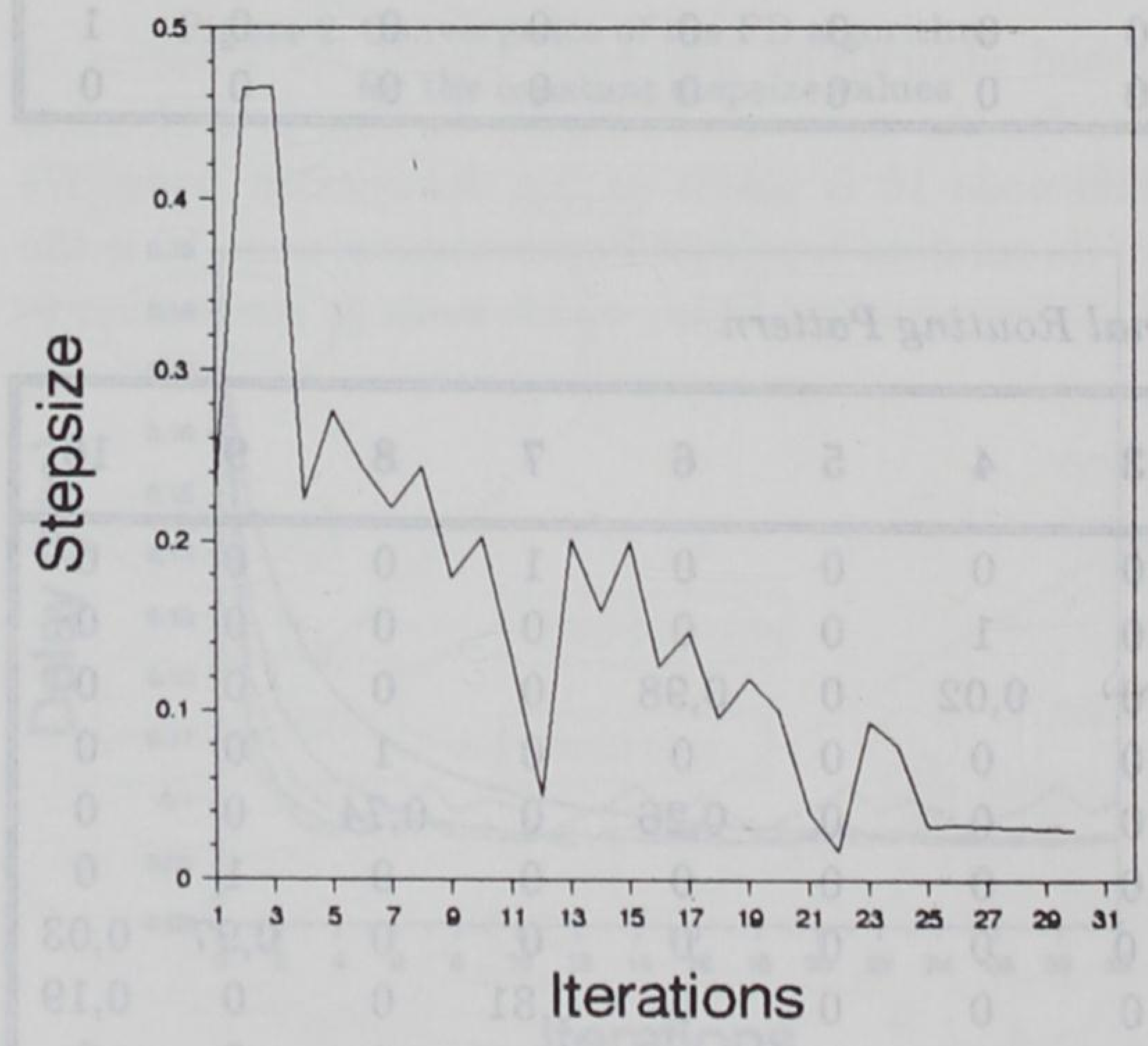

Figure 5. A convergence study of the adaptive stepsize λ

In Figure 4 the FD performance is given with the adaptive stepsize. We note the performance stability as well as the rapid convergence of this choice. In Figure 5 we give the variation of the values of λ through the iterations of the FD algorithm. As it is expected these values converge to an optimum value in the final iterations of the FD algorithm.

In order to study the correlation of the stepsize choice with the traffic conditions on the network, we studied the given network under very light traffic conditions (Figure 6). We observe that in the case of light traffic a constant value of $\lambda = 0.5$ gave quite the same results with the adaptive stepsize, but in heavy traffic the results are those given by Figures 2,3 and 4.

The next investigation concerned a real event on the telecommunications networks. The rapid and continuous changes of the incoming flow. We see in Figures 7 and 8 that for $\lambda = 0.5$ if we have variations in the input traffic the FD algorithm diverges at all, while for an adaptive stepsize convergence is always achieved although the convergence time is affected. In the Table 3 we give the traffic values for the previous study in packets/sec.

Our modified FD algorithm has been compared with a simpler FD, using the first two, three etc. shortest VCs for each node i taking into consideration only the hops and not the neighbor nodes. The results are given in

Figure 9 where we note that our modified algorithm has a better performance when adopting an adaptive stepsize.

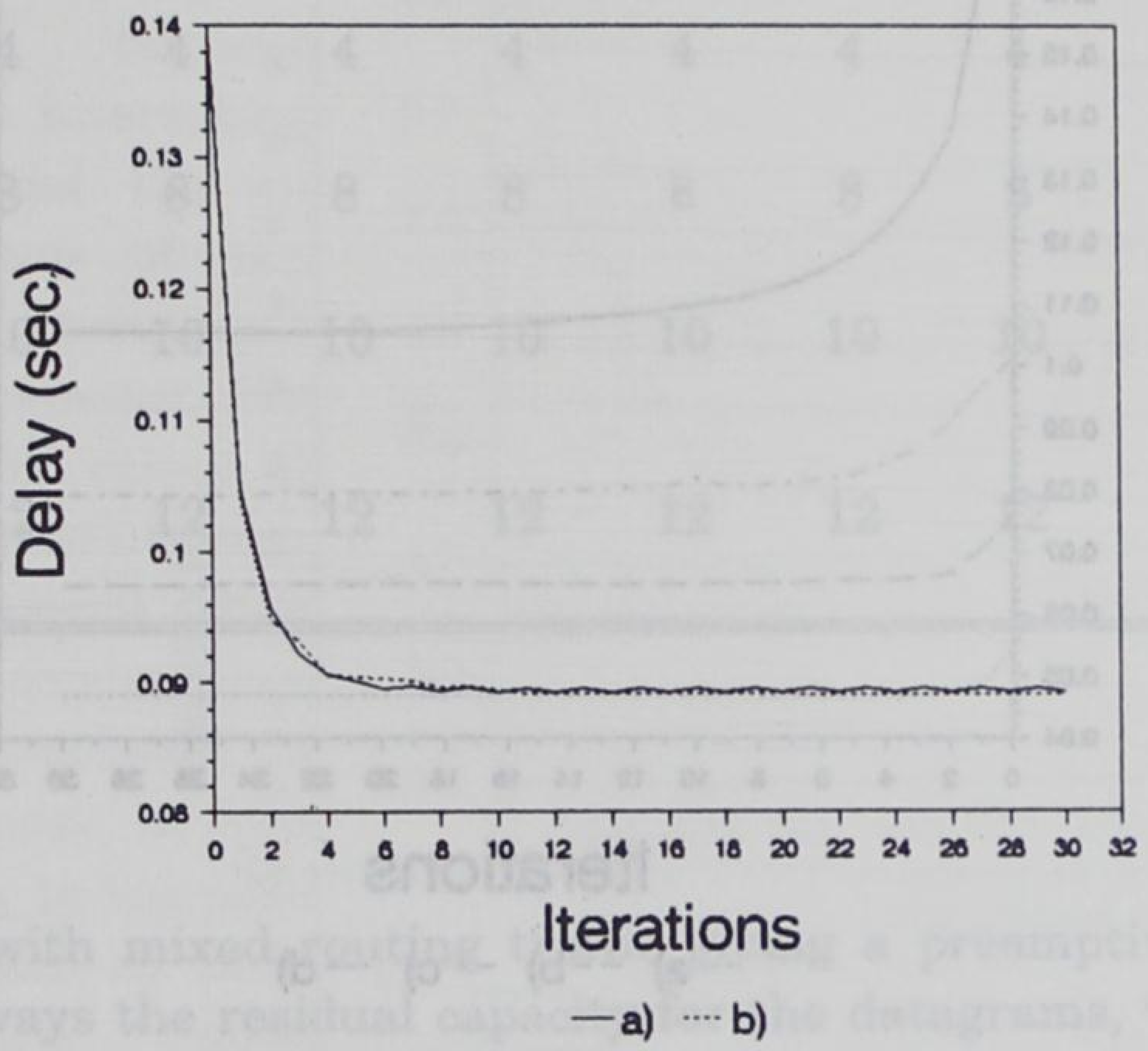

Figure 6. A study of the stepsize selection under very light traffic situation on the network. a) λ=0.5 b) adaptive λ

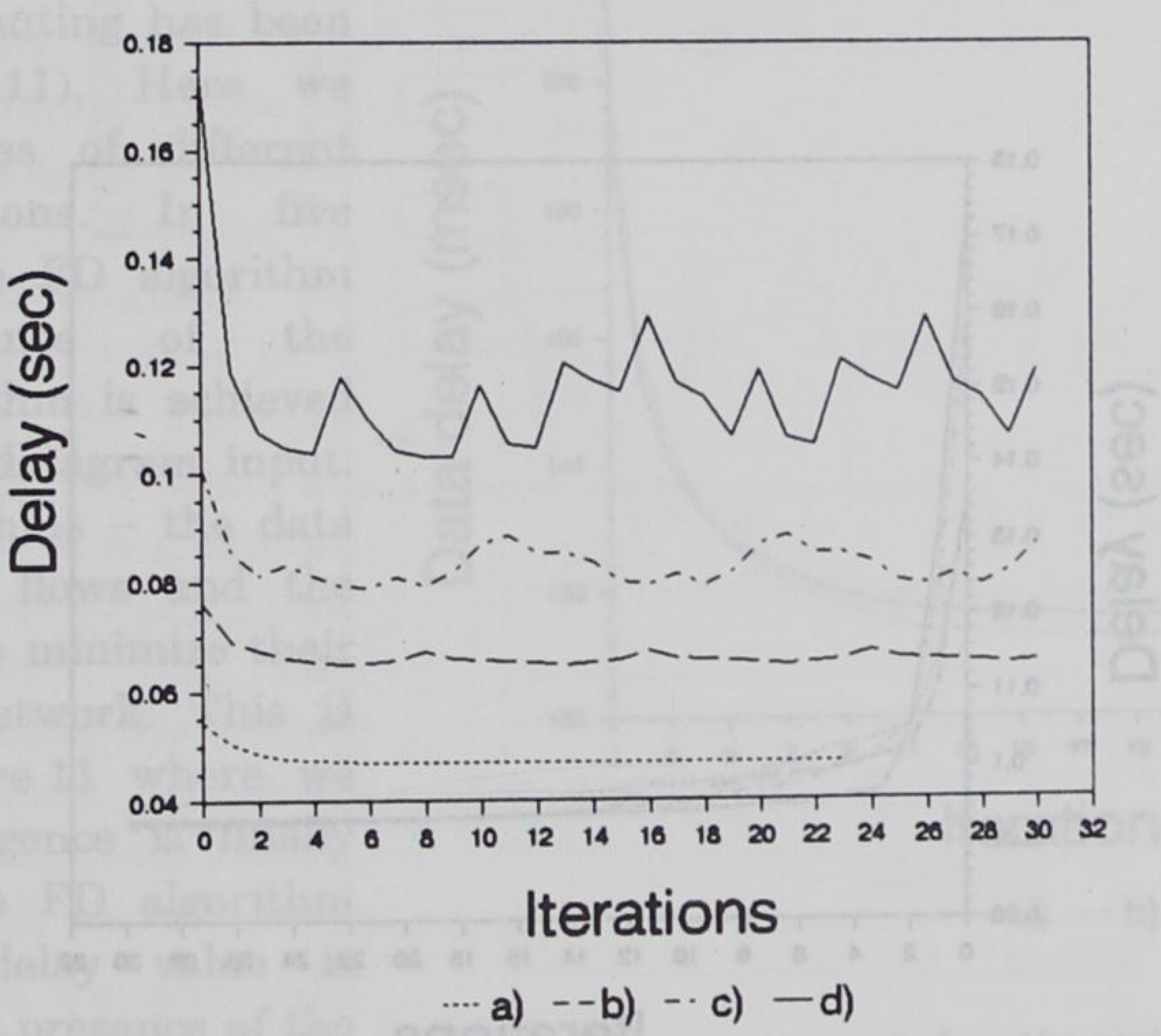

Figure 7. An FD performance with a constant stepsize λ=0.5, in the case of input flow changes: a) light input load b) medium input load c) heavy input load d) very heavy input load

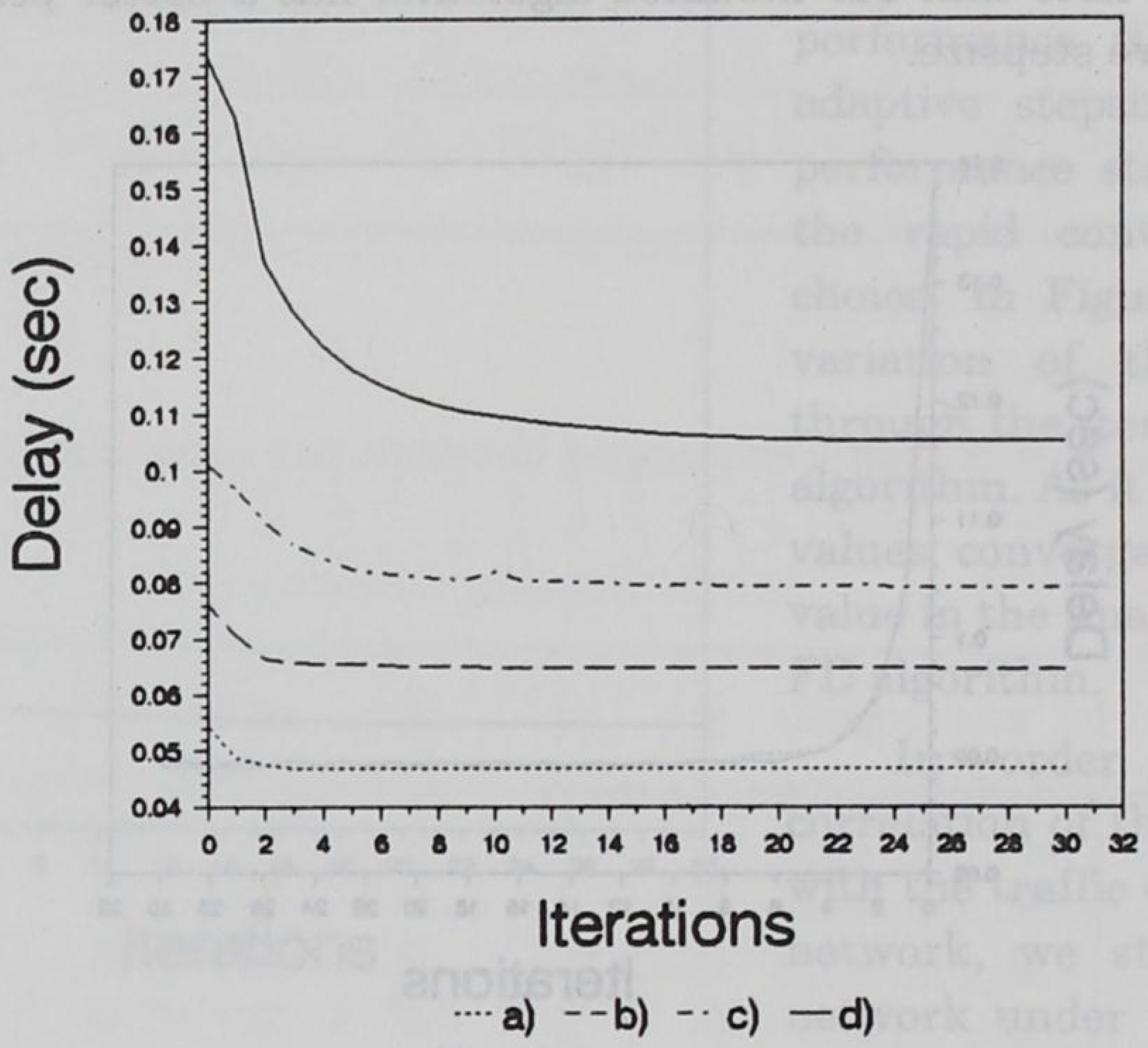

Figure 8. An FD performance with an adaptive stepsize in the case of input flow changes:
a) light input load b) medium input load
c) heavy input load d) very heavy input load

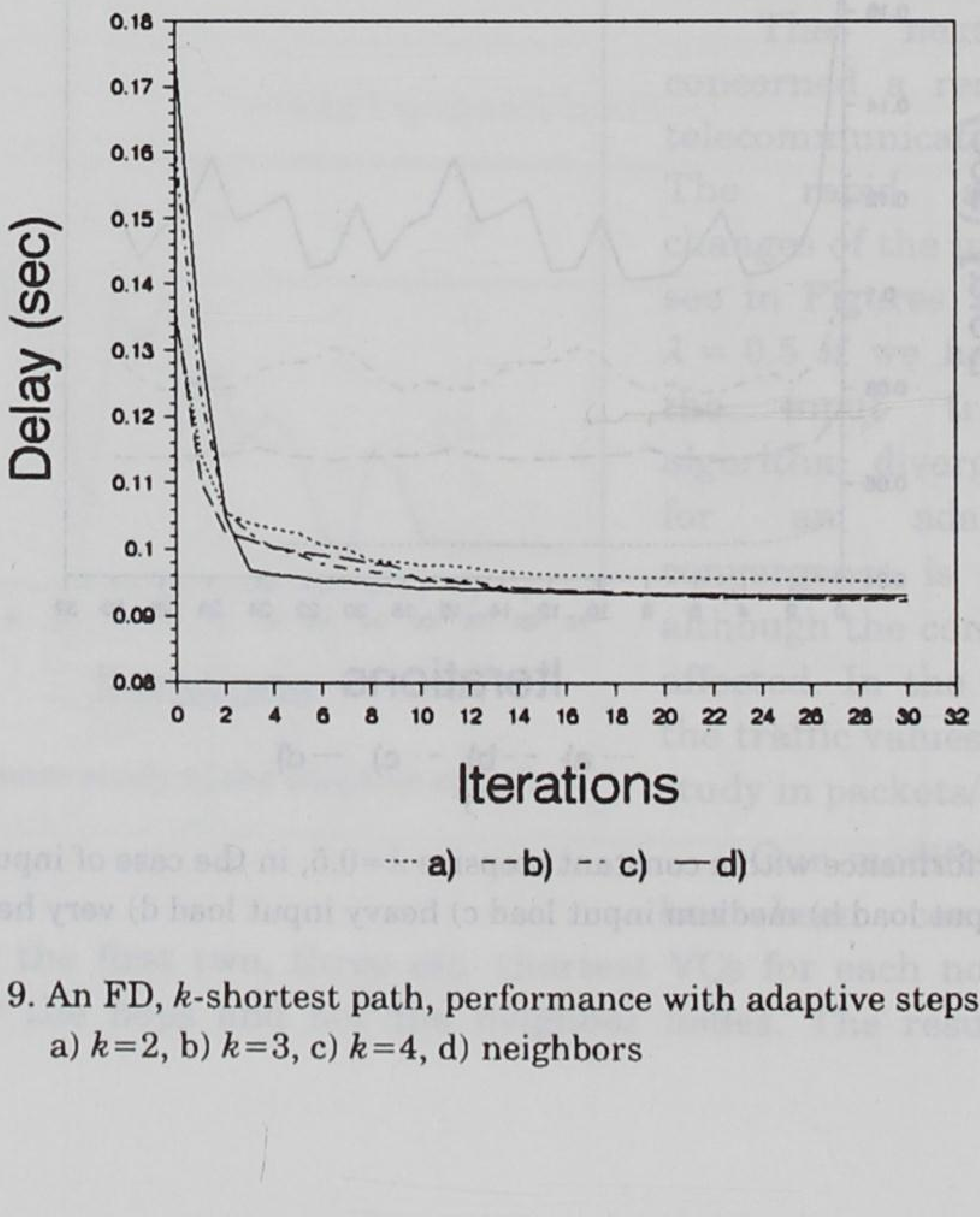

Figure 9. An FD, k-shortest path, performance with adaptive stepsize λ
a) $k=2$, b) $k=3$, c) $k=4$, d) neighbors

Table 3. Voice inputs in packets/sec for different load conditions

Nodes	1	2	3	4	5	6	7	8	9	10
r_i for light input load	4	4	4	4	4	4	4	4	4	0
r_i for medium input load	8	8	8	8	8	8	8	8	8	0
r_i for heavy input load	10	10	10	10	10	10	10	10	10	0
r_i for very heavy input load	12	12	12	12	12	12	12	12	12	0

Mixed routing traffic

When dealing with mixed routing traffic giving a preemptive priority to voice traffic and using always the residual capacity for the datagrams, we note (Figure 10) that the speed of convergence and the final value are better again for the adaptive stepsize. For datagrams we use the constant value $\lambda = 0.5$. Voice and datagram inputs are given in Table 4.

In the sequence a more realistic case of mixed routing has been studied (Figure 11). Here we have three cases of different datagram sessions. In five iterations of the FD algorithm after convergence of the datagram algorithm is achieved we have a new datagram input. Both the algorithms – the data multicommodity flows and the voice FD – try to minimize their delay on the network. This is obvious in Figure 11 where we see that convergence is finally achieved for the FD algorithm although the delay value is affected from the presence of the datagrams. We observe that in case b the convergence of datagram routing is faster since the voice session is smaller, and more datagrams are serviced in the sixty iterations cycle. The adaptive stepsize rule is always used.

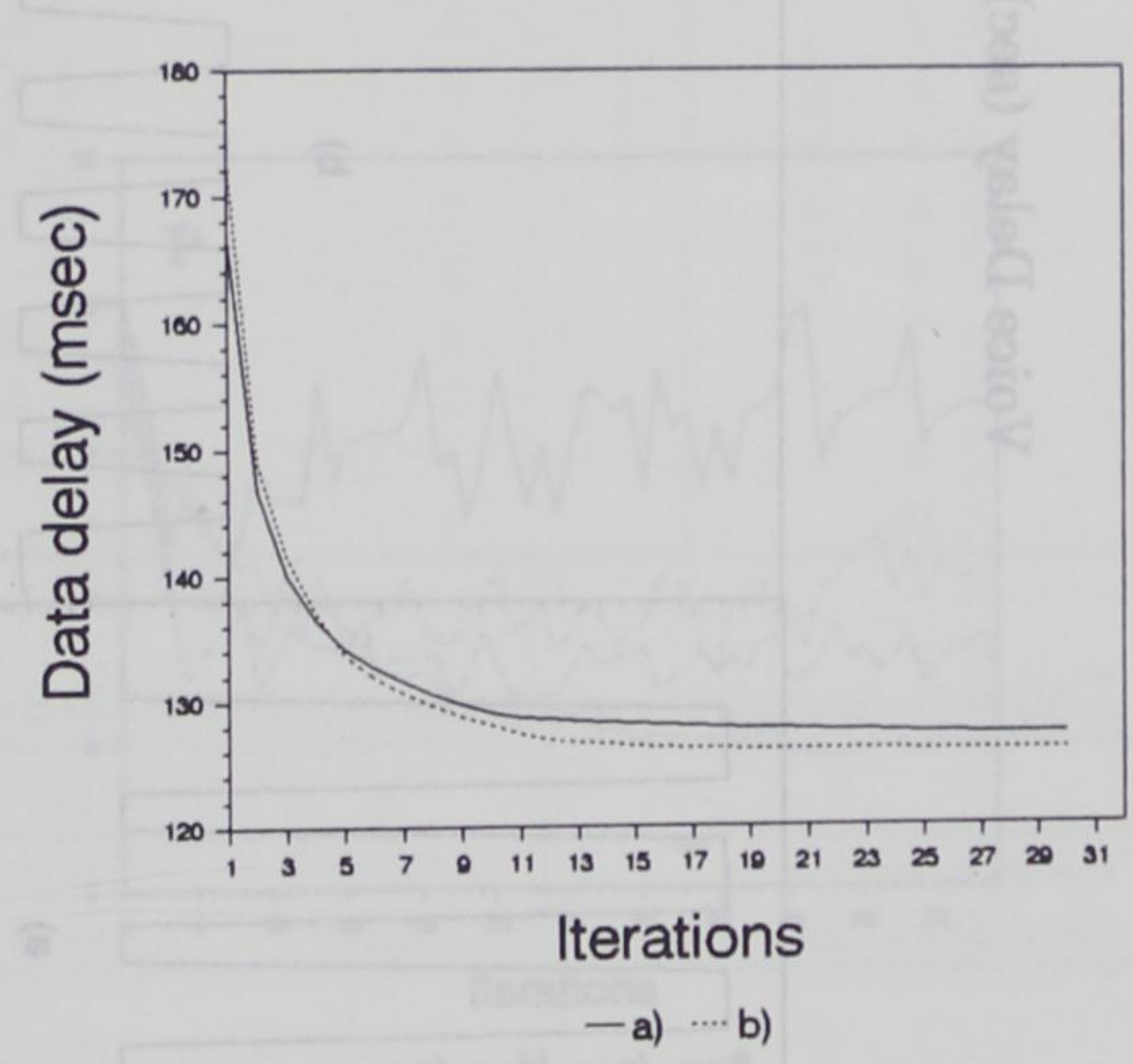

Figure 10. Datagrams routed on the residual capacity with a) λw=0.1, b) λw adaptive, while λd=0.5

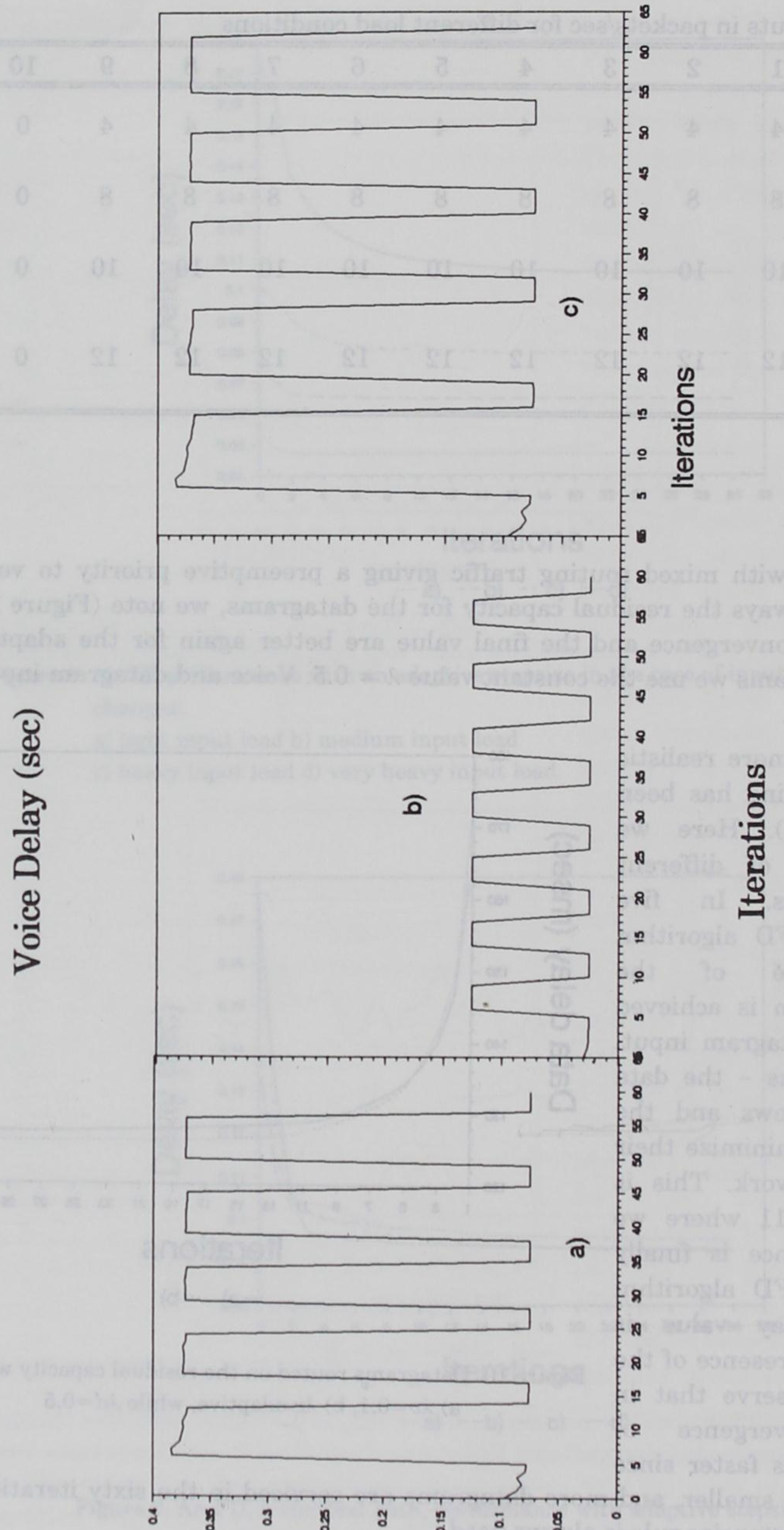

Figure 11. Mixed routing with different message lengths
a) $1/\mu v = 1000$ bits/mess, $1/\mu d = 250$ bits/mess,
b) $1/\mu v = 500$ bits/mess, $1/\mu d = 250$ bits/mess,
c) $1/\mu v = 1000$ bits/mess, $1/\mu d = 500$ bits/mess

Fairness criteria

The simulation results obtained from fairness applications are very interesting too. In Figures 12 and 12c we give the convergence of the fairness algorithm for different values of the *ss* parameter. We observe that for small values of the *ss* factor convergence is very slow but always achieved while for large values of *ss* convergence may be lost. As mentioned in Section II, the purpose of the fairness algorithm is to increase and equalize (as mach as it is possible) all the node throughputs. The initial and final throughputs are given in Table 5 for various values of *SS*.

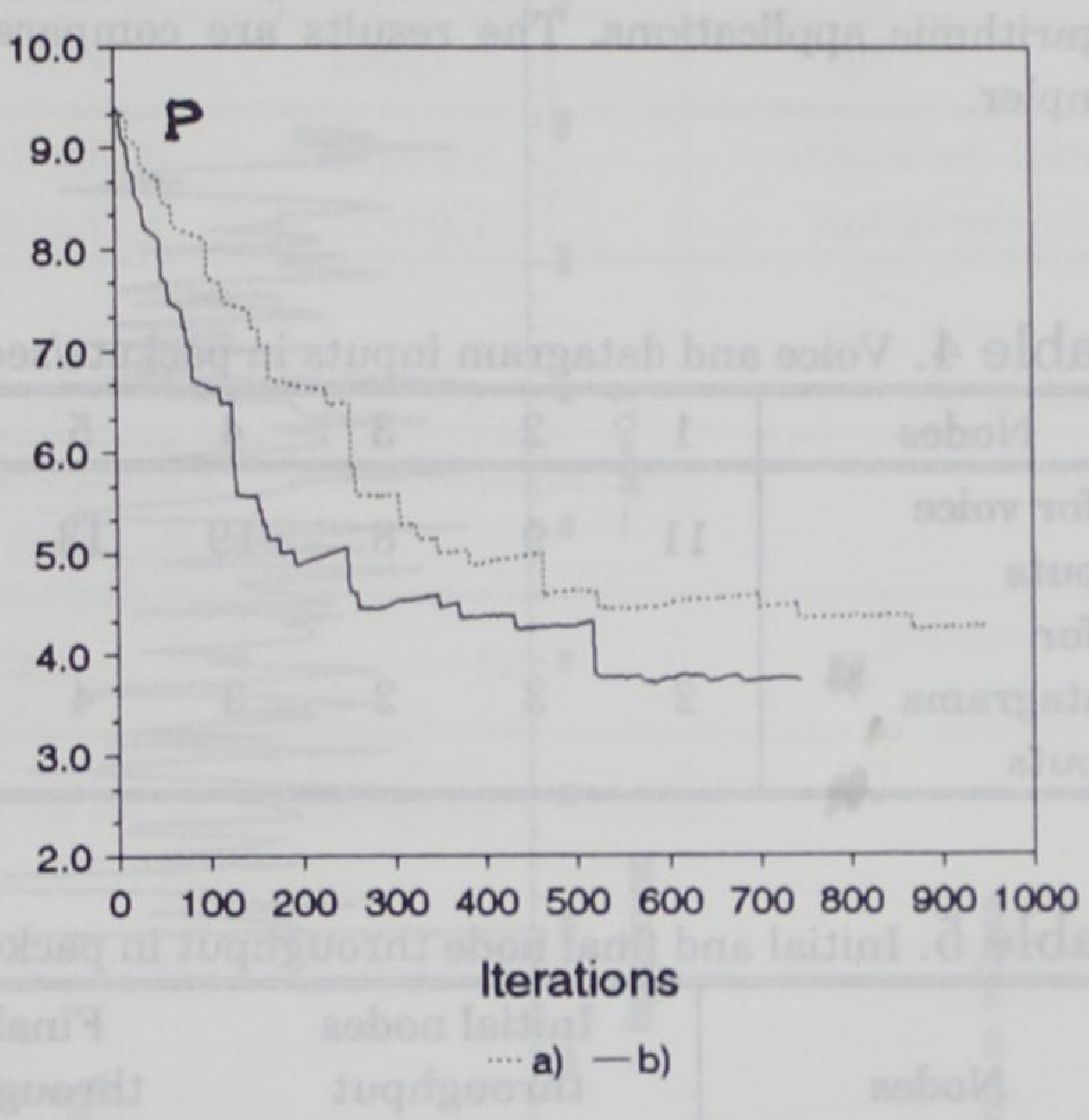

Figure 12. Convergence of the fairness algorithm with a) $ss=0.005$, b) $ss=0.01$

In Figure 13 there is a study on the oscillations around the value $D_{\max}$ for different values of the *ss* factor. In Figure 14 there is the influence of r_i changes in the algorithm convergence. In Table 6 we give the values of *i* before and after the fairness application for different values of r_i. In Figure 15 there is a similar study on the values of $D_{\max}$. We observe, in Table 7, that the lower the $D_{\max}$ value the better the fairness results obtained. Furthermore initial delay values are greatly affected, too.

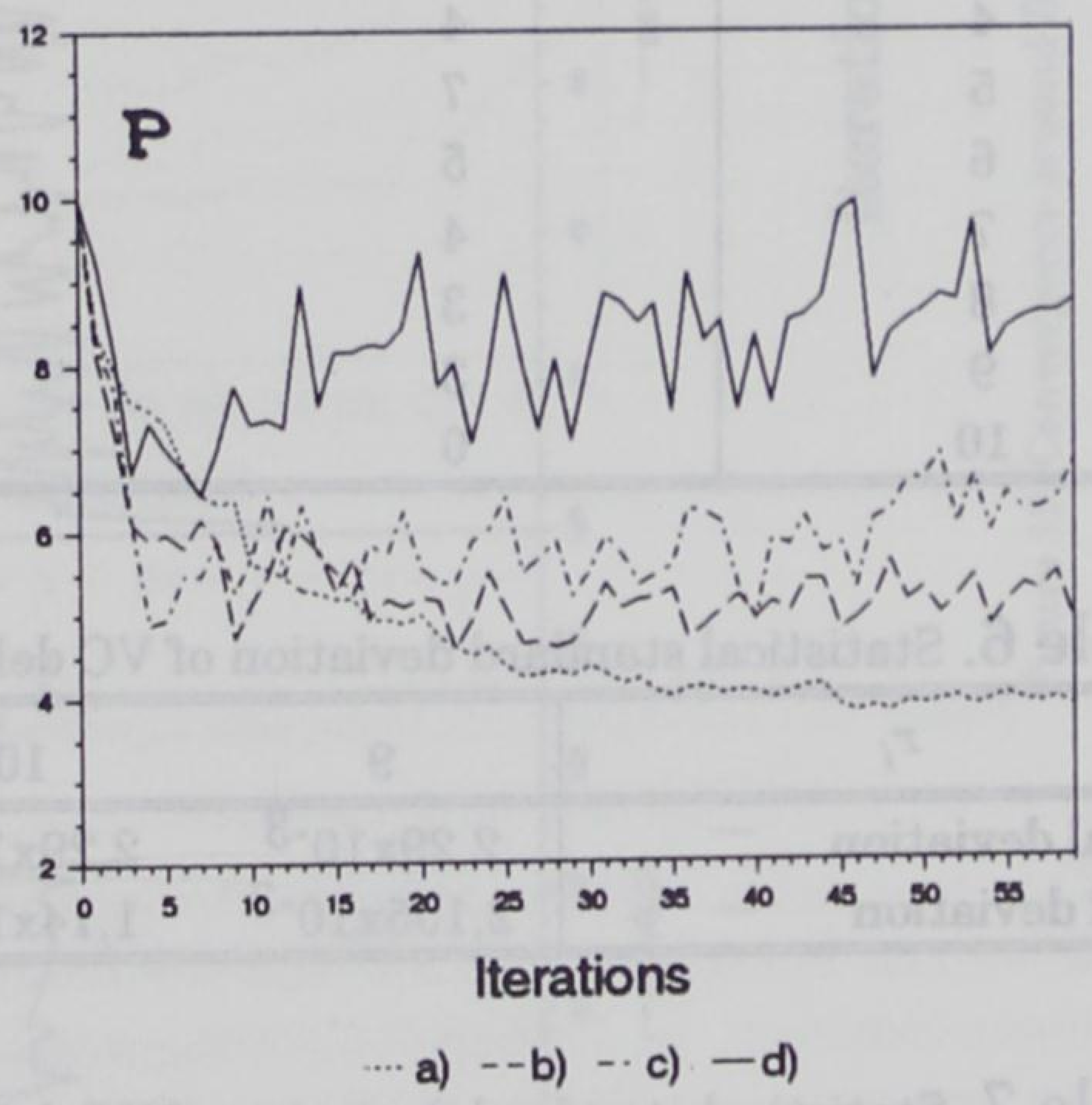

Figure 12c. Convergence of the fairness algorithm with a) $ss=0.1$, b) $ss=0.4$ c) $ss=0.5$, d) $ss=0.7$

In Figure 16 we give the influence of the *SD* parameter on the network performance. A large *SD* gives better and more rapid fairness results as it is seen and from Table 8. Since the fairness algorithm running is cumbersome and slow we tried the second delay formula given by Equation 12 to obtain more user oriented results. In Table 9 we give the virtual circuits delays for the FD, fairness and the logarithmic applications. The results are comparable while the logarithmic case is simpler.

Table 4. Voice and datagram inputs in packets/sec

Nodes	1	2	3	4	5	6	7	8	9	10
r_i for voice inputs	11	9	8	19	13	8	9	6	6	0
r_i for Datagrams inputs	2	3	2	3	4	2	1	1	1	0

Table 5. Initial and final node throughput in packets/sec

Nodes	Initial nodes throughput	Final nodes throughput for ss=0,2	Final nodes throughput for ss=0,01
1	4	9,29	8,75
2	5	6,94	7,32
3	6	8,94	7,97
4	4	8,3	8,06
5	7	7,35	8,84
6	5	9	10,93
7	4	10,89	10,7
8	3	8,37	10,34
9	3	10,97	10,99
10	0	0	0

Table 6. Statistical standard deviation of VC delays as a function of r_i

r_i	9	10	11	13
Initial deviation	$2{,}29\times10^{-3}$	$2{,}29\times10^{-3}$	$2{,}29\times10^{-3}$	$2{,}29\times10^{-3}$
Final deviation	$2{,}155\times10^{-3}$	$1{,}14\times10^{-3}$	$1{,}78\times10^{-3}$	$1{,}082\times10^{-3}$

Table 7. Statistical standard deviation of VC delays as a function of D_{max}

D_{max}	55 msec	59 msec	61 msec
Initial deviation	$2{,}29\times10^{-3}$	$2{,}29\times10^{-3}$	$2{,}29\times10^{-3}$
Final deviation	$0{,}71\times10^{-3}$	$1{,}54\times10^{-3}$	$1{,}68\times10^{-3}$

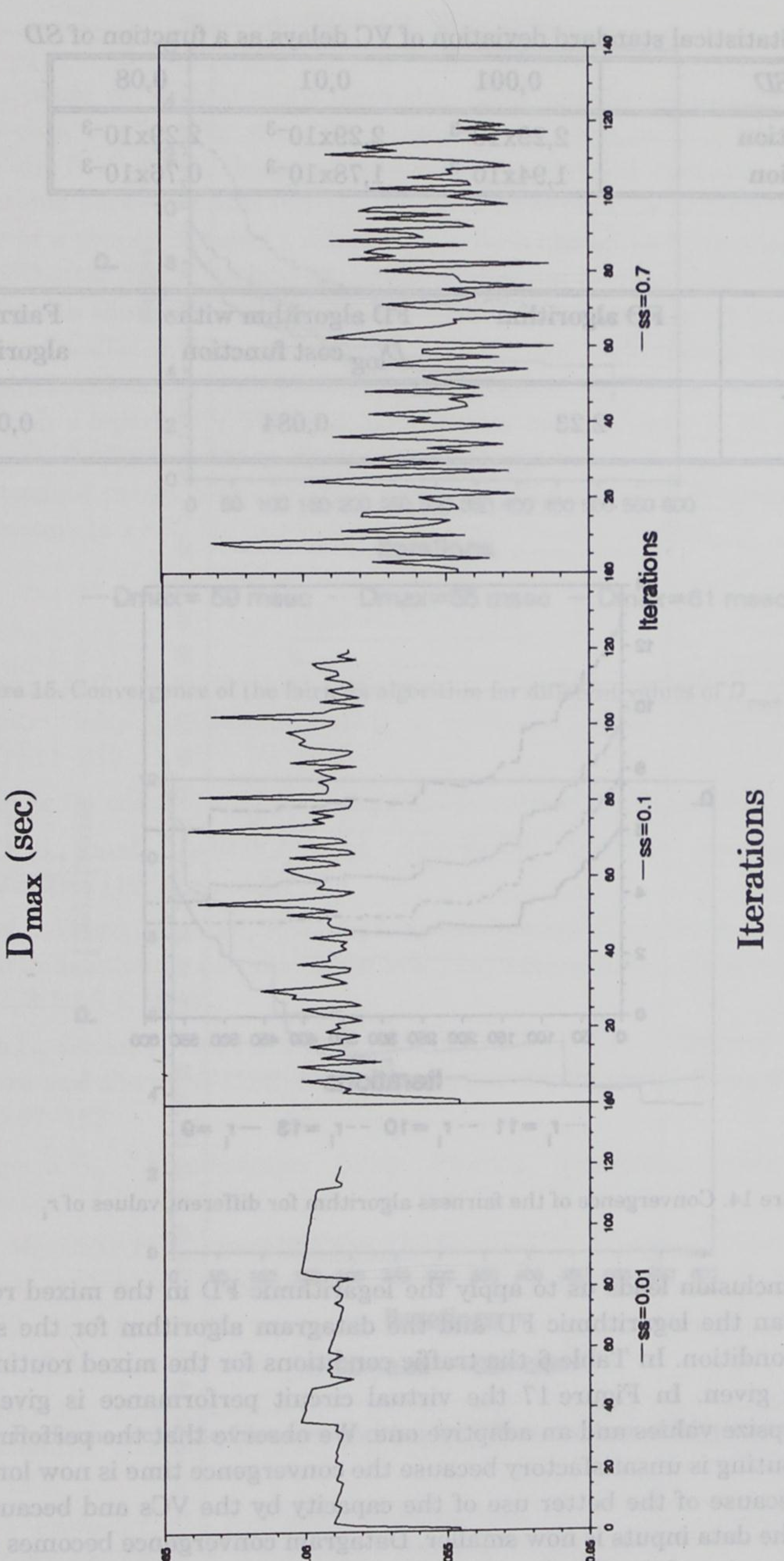

Figure 13. Convergence around D_{max} for different *ss* values

Table 8. Statistical standard deviation of VC delays as a function of *SD*

SD	0,001	0,01	0,08
Initial deviation	$2{,}29\text{x}10^{-3}$	$2{,}29\text{x}10^{-3}$	$2{,}29\text{x}10^{-3}$
Final deviation	$1{,}94\text{x}10^{-3}$	$1{,}78\text{x}10^{-3}$	$0{,}76\text{x}10^{-3}$

Table 9.

	FD algorithm	**FD algorithm with $D_{\log}$ cost function**	**Fairness algorithm**
Deviation of VC delays	2,23	0,084	0,07

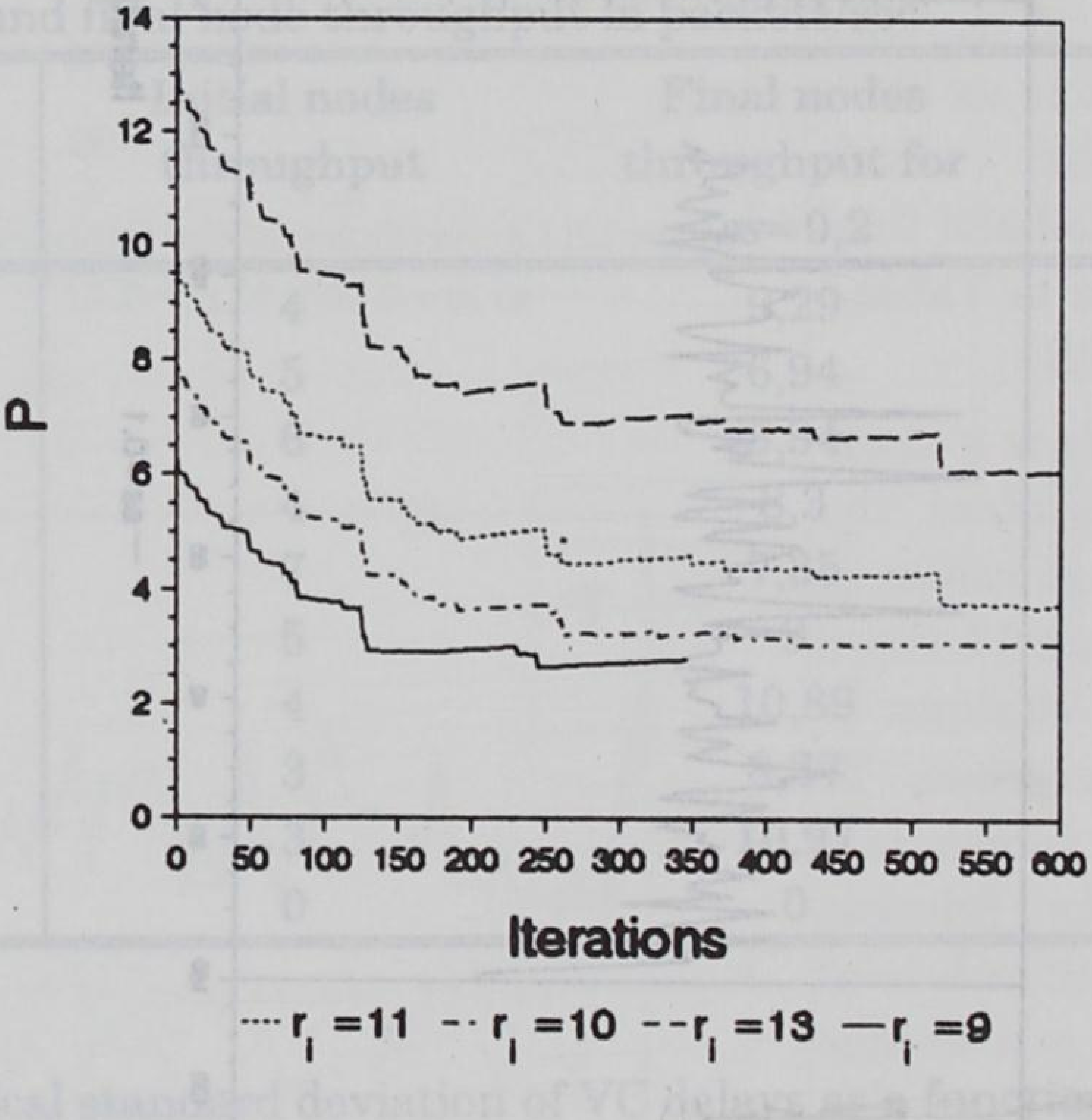

Figure 14. Convergence of the fairness algorithm for different values of r_i

This conclusion leads us to apply the logarithmic FD in the mixed routing case, that is we ran the logarithmic FD and the datagram algorithm for the semirandom data input condition. In Table 6 the traffic conditions for the mixed routing case with fairness are given. In Figure 17 the virtual circuit performance is given with two constant stepsize values and an adaptive one. We observe that the performance of the datagram routing is unsatisfactory because the convergence time is now longer. This is explained because of the better use of the capacity by the VCs and because the total number of the data inputs is now smaller. Datagram convergence becomes slower near the optimum point of the VC algorithm. Therefore, a trade–off analysis must be made before the fairness mixed routing decisions are taken.

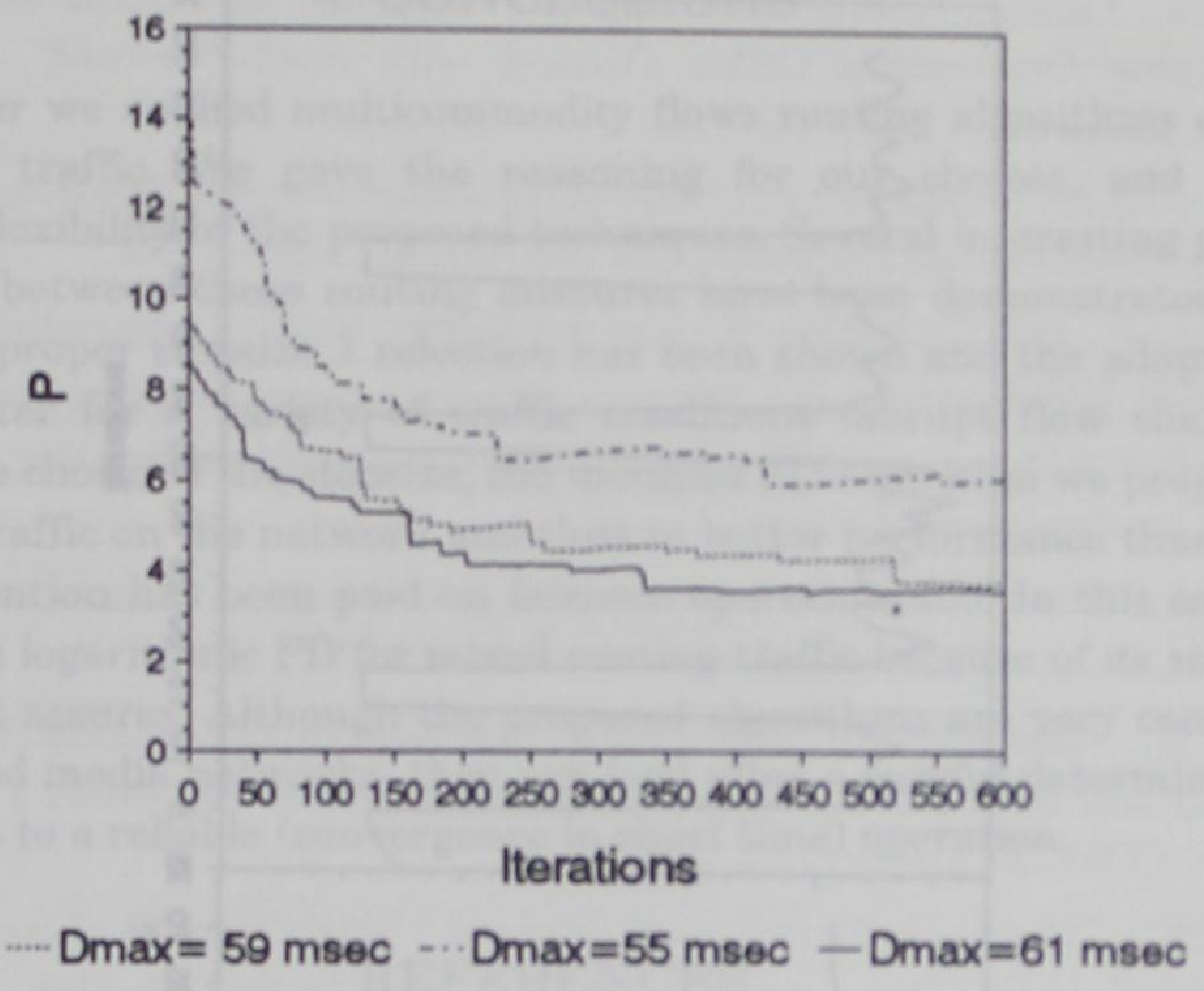

Figure 15. Convergence of the fairness algorithm for different values of D_{max}

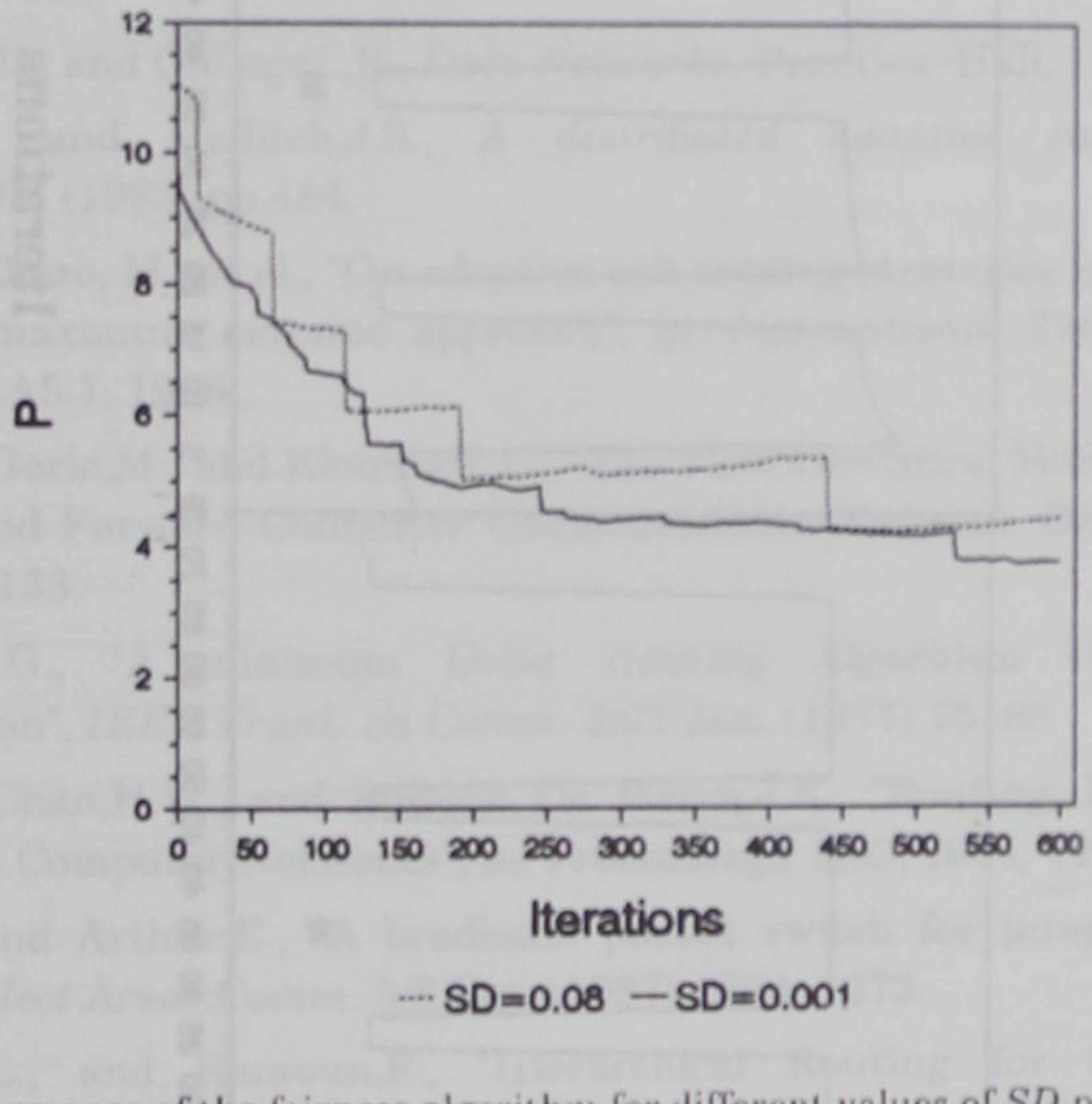

Figure 16. Performance of the fairness algorithm for different values of SD parameters

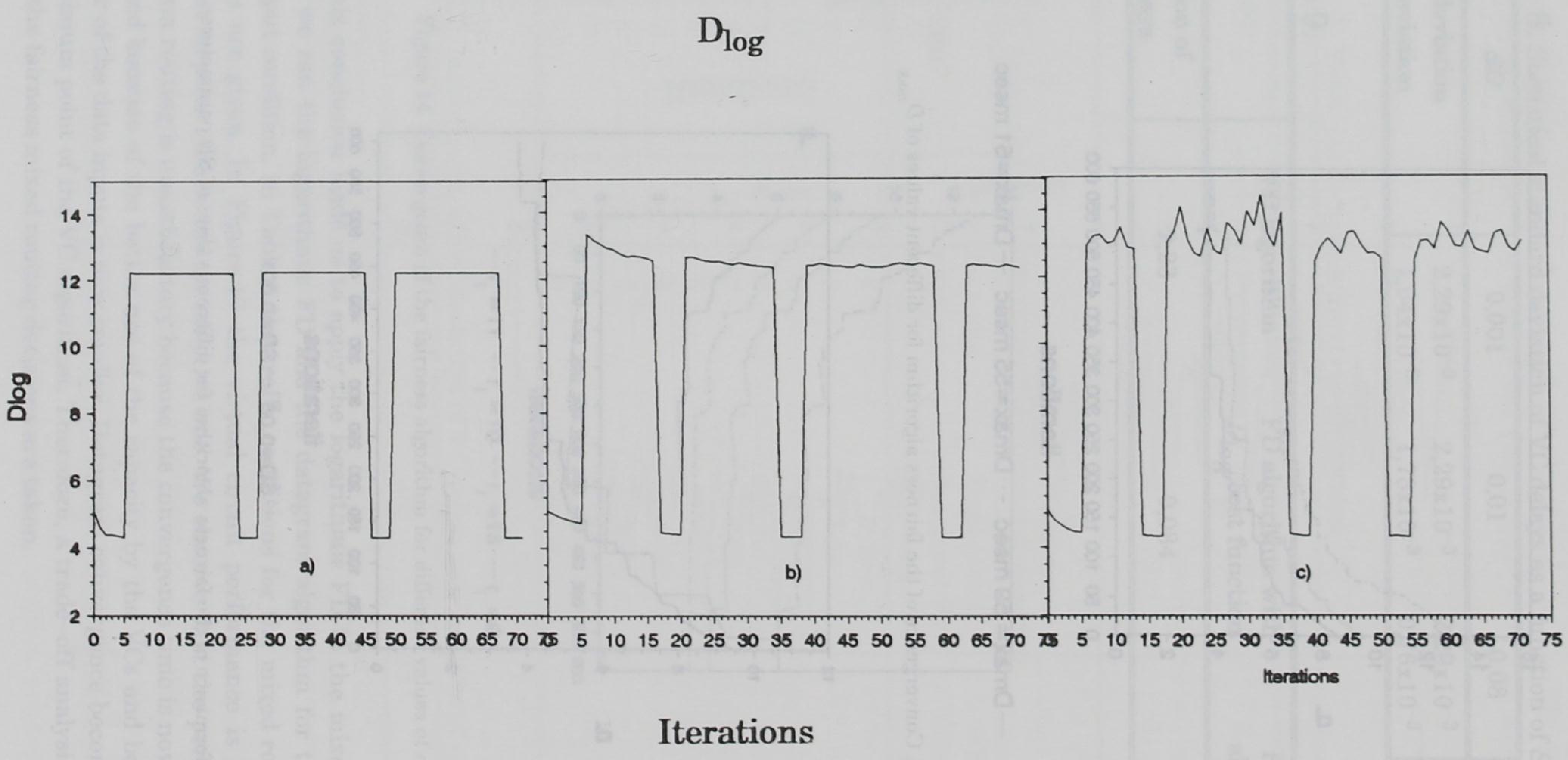

Figure 17. An logarithmic FD performance in the mixed routing case for various stepsize λ values:

a) adaptive λ,
b) λ=0.1,
c) λ=0.3

4. CONCLUSIONS

In this paper we applied multicommodity flows routing algorithms on networks with voice–data traffic. We gave the reasoning for our choices, and studied the robustness and flexibility of the proposed techniques. Several interesting properties of the interactions between these routing mixtures have been demonstrated. The great importance of a proper stepsize λ selection has been shown and the adaptive stepsize found to be better for a variety of traffic conditions (abrupt flow changes, mixed routing). For this choice of the stepsize, the modified FD algorithm we propose leads to move balanced traffic on the network and thus to better performance than any simple FD. A great attention has been paid on fairness operation, too. In this case, we have chosen to apply a logarithmic FD for mixed routing traffic because of its simplicity and the fair results it assures. Although the proposed algorithms are very complicated for the current mixed media networks, they can lead after a careful determination of the main parameters to a reliable (convergence in short time) operation.

REFERENCES

[1] Bertsekas,D.P., Gafni,E., and Gallager,R.G., "Second derivative algorithms for minimum delay distributed routing in networks", *IEEE Trans. on Comm.* 32 (1984) 911–919.

[2] Bertsekas,D., and Gallager.,R., *Data Networks*, Prentice–Hall, 1987, pp. 388.

[3] Chen,M.S., and Meditch,J.S., *A distributed adaptive routing algorithm*, GLOBECOM (1983) pp.484.

[4] Dziong,Z. Dioro, M., et al., "On adaptive call routing strategies in circuit–switched networks maximum revenue approach", in: *International Teletraffic Congress*, vol. 12, 3.1.A5.1, 1988.

[5] Fratta,L., Gerla,M., and Kleinrock,L., "The Flow Deviation Method: An Approach to Store and Forward Computer Communication Network Design", *Networks* 3 (1973) 97–133.

[6] Gallager,R.G., "A minimum Delay Routing Algorithm Using Distributed Computation", *IEEE Trans. on Comm.* 25/1 Jan. (1977) 73–85

[7] Gerla,M., Chan,H.W., and Boisson De Marca.J.R., "Routing, Flow Control and Fairness in Computer Networks", in: *Proceedings 1CC*, 1984, 1272–1276.

[8] Hui,J.Y., and Arthur,E., "A bradband packet switch for integrated transport", *IEEE J. Select Areas Comm.* 5/8 Oct. (1987) 1264–1273.

[9] Kleinrock,L., and Kamoun,F., "Hierarchical Routing for Large Networks: Performance Evaluation and Optimization", *Computer Networks 1* (1977) 155–174.

[10] Kuo,F., *Protocols and techniques for data communication networks*, Prentice–Hall, 1981, pp. 122.

[11] Lin,P.M., Leon,B.J., and Stewart,C.R., "Analysis of Circuit–switched Networks Employing Originating Office Control with Spill–Forward", *IEEE Trans. on Comm.* 26/6 June (1978) 754–765.

[12] Pattarvina,A., "Multichannel Bandwidth allocation in a broadband packet switch", *IEEE J. Select Areas Commun.* 6/9 Dec. (1988) 1489–1499.

[13] Pavlidou,F., and Kouris,S., "Second derivative routing algorithms", *Electron. Lett.* 22 (1986) 749–750.

[14] McQuillan,J.M., et al., "The new Routing Algorithm for the ARPAnet", *IEEE Trans. on Comm.* 28/5 (1980) 711–719.

[15] Rinde,J., "Routing and flow Control in a Centrally Directed Network", in: *Nat. Comput. Conf. AFIPS Conf. Proc.* vol. 38, 1979, 211–216.

[16] Rudin,H., "On Routing and Delta Routing: A Taxonomy and Performance Comparison of Techniques for Packet–Switched Networks", IEEE Trans. on Comm. COM–24/1 Jan. (1976) 43–59.

[17] Segall,A., "Optimal Distributed Routing for Virtual Line–Switched Data Networks", *IEEE Trans. on. Comm.* 27/1 Jan. (1979) 201–209.

[18] Wong,J.W., Sauve,J.P., and Field,J.A., " A study of fairness in Packet– Switching Networks", *IEEE Trans. on. Comm.* 30/2 Feb. (1982) 346–353.

[19] Yum,T.K., and Schwartz,M., "Packet–Switched Performance with Different Circuit Switched Routing Procedures in Nonhierarchical Integrated Circuit–Switched and Packet–Switched Networks", *IEEE Trans. on Comm.* 35/3 (1987) 362–366