Trunk Network Fault-Tolerance through Routing Table Updating in the Event of Trunk Failure

Mark C. Sinclair[1]

1. Introduction

Traditionally, research into circuit-switched (C-S) communication networks has been divided into two areas, reliability and performance, each making little reference to the other. More recently, however, research has focused on unification of the two. One possible choice for a combined performance and reliability measure is the network grade-of-service (NGOS), i.e. the overall probability of blocking, averaged over all node pairs, taking into account both the offered traffic between node pairs and the link (trunk group) availabilities. In a recent paper [1], two methods were given for the single-moment analysis of fixed-alternative-routing circuit-switched communications networks with reliable nodes but unreliable links. The second of these, *MKV_GOS*, employs a Markov model for an unreliable link, in combination with earlier work [2-6], so as to greatly reduce execution time compared to existing algorithms.

In the current paper, the *MKV_GOS* method is used as part of three heuristic routing table optimisation algorithms, which obtain routing tables such that the overall network grade-of-service is minimised under different call control rules. These algorithms are then used to demonstrate the improvements in trunk network fault-tolerance (assessed in terms of the network grade-of-service) that can be achieved through routing table updating in the event of link (trunk group) failure.

2. Network Description

The following six elements are used to describe the model of an unreliable trunk network used in [1] and also in this paper:

1. Routing Table
2. Call Control Rule
3. Network Configuration Table
4. Link Capacity & Availability Table
5. Busy-Hour Traffic Matrix
6. Loading

The Routing Table (e.g. Table 5 in Section 7 below) consists of an n x n matrix of entries, where n is the number of nodes in the network, except that the forward diagonal is empty. Entries are ordered sets of nodes, $(N_1,N_2,\ldots,N_m)$, such that the entry at position (S,T) represents links $(S\text{-}N_1,S\text{-}N_2,\ldots,S\text{-}N_m)$. These are searched in the given order to find a free circuit for a call arriving at S destined for T.

The Call Control Rule describes how the routing table will be used to find paths through the network. Four such routing disciplines can be analysed using the methods of [1]: Successive Office Control (SOC), Originating Office Control (OOC), Originating Office Control with Spill-Forward (OOC-SF) [2], and Tandem Node Matrix routing (TNM) [3]. In SOC, when a call reaches a given node, a free circuit will be searched for, trying the first-choice, and then the other links in order. If a free circuit is found, then control is passed to the next node over that link, otherwise the call is lost. In OOC, if a call is at its originating node, it will be routed as described for SOC. At a tandem (intermediate) node, however, only the first-choice link will be tried. If this link is unavailable, the call will not be lost, but will drop back to the originating node, where the next choice link will be tried. The call is only lost if the last choice link (and all previous choices of course!) from the originating node is unavailable. OOC-SF is intermediate between the two previous call control rules: it is identical to OOC unless one of the nodes in an entry in the routing table is marked as a spill-forward node. In that case, if a call reaches the spill-forward node, then call control is passed, and as far as routing is concerned, that node acts as the originating node. Such nodes are represented by a lower-case letter in the appropriate entry in the routing table. Both SOC and TNM are special cases of OOC-SF: SOC is OOC-SF with every tandem node marked as spill-forward, and TNM, which is only used in purely hierarchical networks, is described by Chan [3]. For a more detailed discussion, and examples, see Lin *et al.* [2].

[1] Department of Electronic Systems Engineering, University of Essex

The Network Configuration Table (e.g. Table 6) is an n x n array of link numbers, with an empty forward diagonal, the (S,T) entry being either the link number of the link connecting these two nodes, or -1 if there is no direct link.

The Link Capacity & Availability Table (e.g. Table 7) contains the capacities (i.e. number of circuits) and availabilities of the b links connecting the network.

The Busy-Hour Traffic Matrix (e.g. Table 8) is an n x n array, with an empty forward diagonal, of the busy-hour traffic in Erlangs between each node pair (S,T).

The Loading is used to obtain different traffic levels. Each entry in the Busy-Hour Traffic Matrix is multiplied by the loading, and thus, for example, with a loading of 1.1, the traffic is increased by 10% over the nominal level. Unless otherwise stated, the loading is always set to 1.

3. Routing Table Optimisation

Using the *MKV_GOS* method for the necessary network analysis, three heuristic routing table optimisation algorithms, *RTE_OPT*, *RTE_OPTS*, and *RTE_OPTF*, were devised for C-S networks with unreliable links employing OOC, SOC and OOC-SF routing respectively. The three methods do not carry out an exhaustive search of all possible loop-free routing tables, and therefore cannot provide the optimum routing table for the chosen call control rule. They have been devised, however, to provide the best possible routing table within the assumptions made. The optimisation criterion used by all three methods is the minimisation of overall NGOS. The overall flow of control for all three methods is given in the flowchart below.

3.1. Directly Connected Node Pairs

In all three methods, the directly connected node pairs have their first-choice route set to the direct link. Thus, for example, node pairs (A,B) and (B,A) in the Example Network (Figure 3) would use link 0 as their first-choice route; the (A,B) routing table entry would be set to "B", and the (B,A) entry to "A". The assumption that the direct link is the best first-choice route would clearly be invalid if the direct link was either very failure prone or heavily loaded relative to other links. However, the assumption is a reasonable one in networks where all the links have approximately the same availability, and the load is not too unbalanced.

3.2. Nodes With Only One Link

If a node is directly connected to only one other node, then obviously that node must provide tandem service for all calls originating on the former node. For example, node A in the Example Network is only connected to one other node, node B. The node pairs (A,C), (A,D), (A,E), (A,F), (A,G), (A,H) and (A,I) must therefore use node B for all their traffic. In *RTE_OPT* all seven routing table entries would be set to "B", whereas in *RTE_OPTS* and *RTE_OPTF* entries of "b" would be made, the lower case indicating an SOC or spill-forward node. In *RTE_OPTF*, which could use either "B" or "b", the assumption is thereby made that, despite the resulting increase in overflow traffic, the spill-forward node ("b") would reduce the lost traffic from the singly-connected node (A), and, consequently, would also improve the overall NGOS.

In addition to making the routing table entries described above, all the corresponding node pairs are *marked*. This ensures that in the main loop of the methods no attempt is made to improve the routing table entries for node pairs with a singly-connected originating node. Obviously the routing table entries for such nodes cannot be improved.

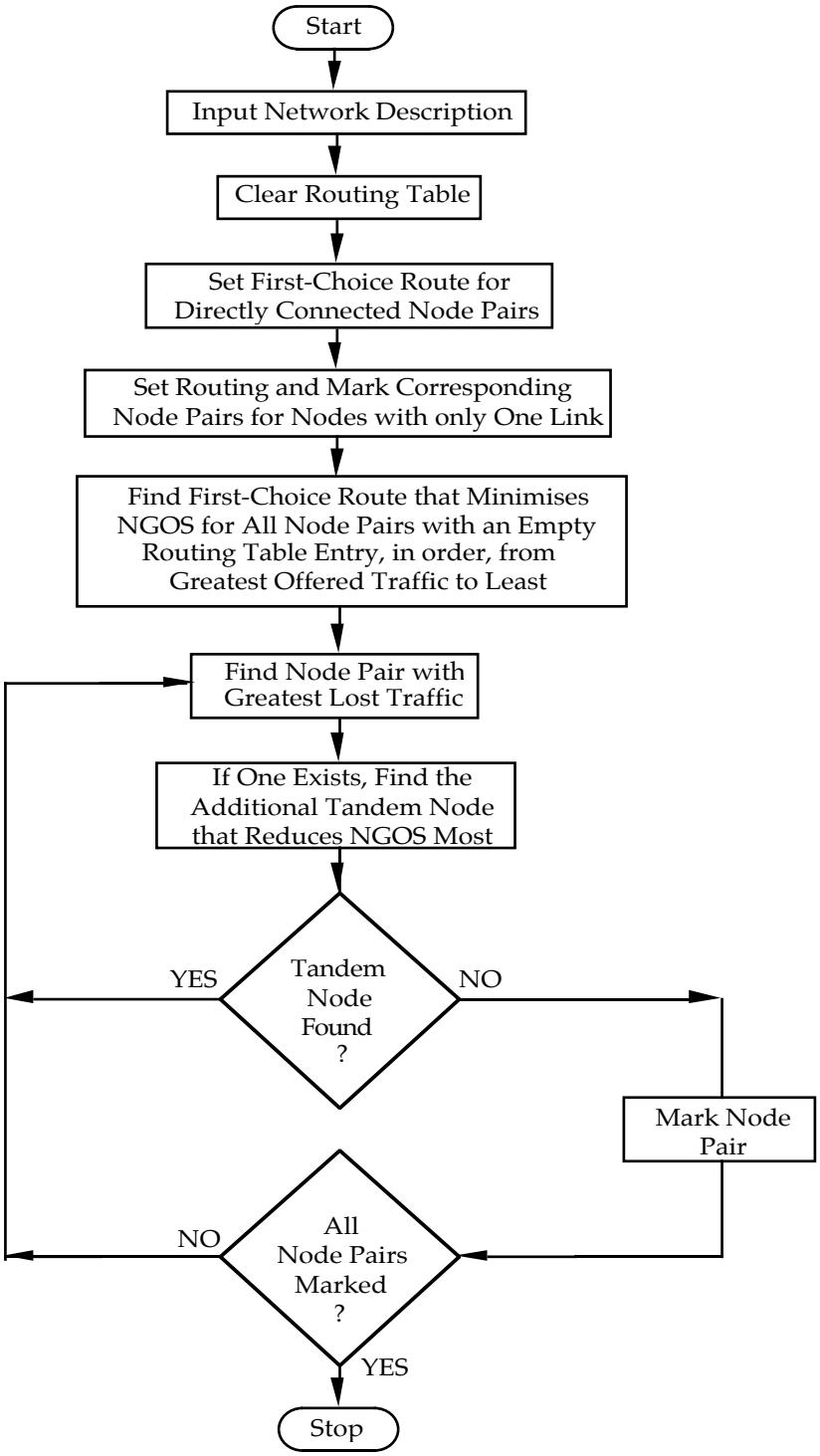
3.3. First-Choice Routes

The next part of the methods, represented by a single box in the flowchart, sets the first-choice routes for all those node pairs whose routing table entries are still empty i.e. not covered by Sections 3.1 and 3.2.

The node pair with the largest offered traffic, amongst those with an empty routing table entry, is found. The first-choice route is set to each of the nodes directly connected to the originating node in turn, and for each one the overall NGOS found using the *MKV_GOS* method. *RTE_OPT* enters the nodes as OOC, *RTE_OPTS* as SOC, and *RTE_OPTF* tries each node as first OOC and then spill-forward. For example, node pair (E,C) in the Example Network would have nodes B, D and G tried as first-choice route (i.e. "B", "D" and then "G" for *RTE_OPT*; "b", "d" and then "g" for *RTE_OPTS*; and "B", "b", "D", "d", "G" and then "g" for *RTE_OPTF*). That route which produces the lowest NGOS (discounting, of course, entries, such as "G" or "g" here, that cause the network to loop) is entered as the permanent first-choice route for that node pair. The methods then select from the remaining node pairs the one with the largest offered traffic, and so on, until all node pairs have their first-choice routes established.

The key assumption here is that setting the remaining node pairs' first-choice routes one by one, in order of offered traffic, will still provide reasonable results compared to exhaustive search through all possible (loop-free) routing tables.

Flowchart



3.4. Alternative Routes

Once every node pair has its first-choice route set, the methods proceed to tackle alternative routing. The node pairs are examined to find the unmarked node pair with the greatest lost traffic. All the nodes that are directly connected to the originating node, and not already used in the node pair routing table entry, are appended to that entry in turn. As in Section 3.3 above, *RTE_OPT* tries each such node as OOC,

RTE_OPTS as SOC, and *RTE_OPTF* as both OOC and spill-forward. For example, node pair (C,E) in the Example Network, with an existing routing table entry of "B", would have entries "BD" and "BF" tried by *RTE_OPT*; with an existing entry of "BD", however, *RTE_OPT* would have only the one choice of "BDF". The entry that reduces the current network NGOS by the largest amount, without causing the network to loop, is adopted as the new routing table entry for that node pair. If no suitable tandem node can be found, however, then the node pair is marked as unsuitable for further attention. The methods then look for the unmarked node pair with the greatest lost traffic, and so on, until no unmarked node pairs remain.

The use of marking was originally introduced simply to indicate those node pairs that had been selected by the methods, but for which an additional node that reduced the NGOS without looping could not be found. As soon as any routing table entry was successfully changed, all marked node pairs were set to unmarked once again. It was found, however, that node pairs that had once been marked, and then subsequently unmarked, hardly ever had their routing table entries successfully updated. Nevertheless, they contributed significantly to the execution time of the methods, as they were often marked and unmarked repeatedly, with the NGOS recalculated a number of times each cycle.

The key assumption in this section is almost identical to that in Section 3.3; that the selection of alternative routes, one additional tandem node for one node pair at a time, in lost traffic order, will nevertheless produce reasonable results compared to exhaustive search through all possible (loop-free) routing tables.

3.5. Results for the Example Network

As an illustration of the results of the routing table optimisation methods, all three were run on the Example Network, with the link availabilities set to 0.98. Tables 1 to 3 give the routing tables obtained by *RTE_OPT*, *RTE_OPTS*, and *RTE_OPTF* respectively.

The overall NGOS results, using the various routing tables above, were: OOC 0.0252, SOC 0.0217, and OOC-SF 0.0158. For comparison purposes, Table 5 shows a direct routing table comprising either direct links, where possible, or tandem nodes chosen by inspection; the overall NGOS obtained using this routing table was 0.0781.

Table 1 OOC Routing Table

From \ To	A	B	C	D	E	F	G	H	I
A		B	B	B	B	B	B	B	B
B	A		CD	DCE	ED	CDE	EDC	DCE	DCE
C	DB	BD		DBF	BD	FD	DB	FDB	FDB
D	BE	BECF	CBFE		EBCF	FCB	EB	FC	FC
E	BD	BD	BD	DB		DB	G	DB	DB
F	CD	CD	CD	DC	CD		CD	H	I
G	E	E	E	E	E	E		E	E
H	F	F	F	F	F	F	F		F
I	F	F	F	F	F	F	F	F	

Table 2 SOC Routing Table

From \ To	A	B	C	D	E	F	G	H	I
A		B	b	b	b	b	b	b	b
B	A		C	Dc	E	cde	e	dc	dc
C	db	B		D	bd	F	db	f	f
D	be	Bc	Cb		Eb	Fc	eb	fc	fc
E	b	Bd	bd	Db		d	G	db	db
F	cd	cd	Cd	Dc	cd		cd	H	I
G	e	e	e	e	E	e		e	e
H	f	f	f	f	f	F	f		f
I	f	f	f	f	f	F	f	f	

Table 3 OOC-SF Routing Table

From \ To	A	B	C	D	E	F	G	H	I
A		B	b	b	b	b	b	b	b
B	A		CD	DCE	ED	CdE	EDC	DCE	DCE
C	DB	BD		DBF	Bd	FD	DB	FDB	FDB
D	BE	BCF	CBFE		EB	FC	EB	FC	FC
E	BD	Bd	BD	DB		DB	G	DB	DB
F	CD	CD	CD	DC	CD		CD	H	I
G	e	e	e	e	E	e		e	e
H	f	f	f	f	f	F	f		f
I	f	f	f	f	f	F	f	f	

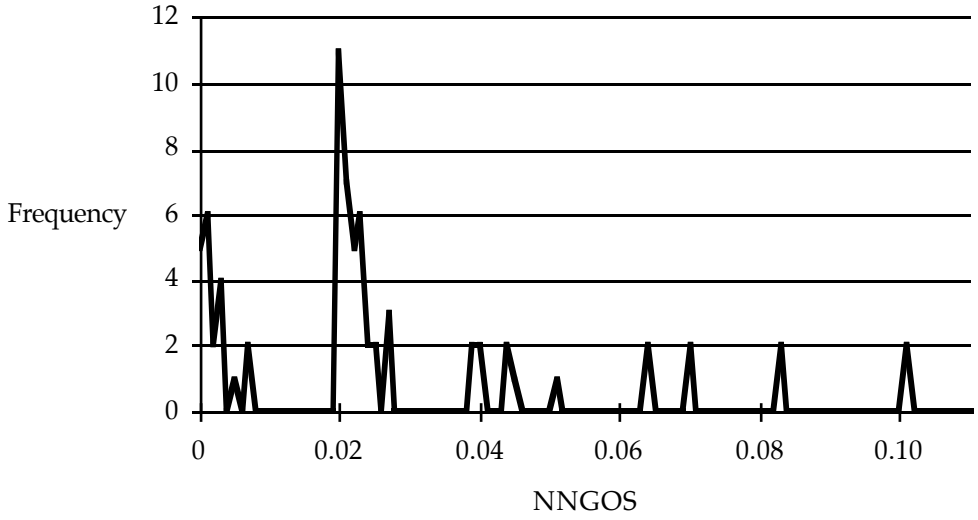
3.6. Discussion

The use of the three heuristic routing table optimisation algorithms described in this paper would not have been feasible, even on networks as small as the Example Network, without their incorporation of the *MKV_GOS* method for network analysis [1]. This allowed, for example, the optimisation of the Example Network using *RTE_OPTS* in twelve hours on an IBM PC/XT compatible; whereas using the slower *UNRELGOS* network analysis method from [1] would have required nearly three weeks. As was shown in [1], the advantage of the *MKV_GOS* method becomes increasingly apparent for larger networks.

It has been observed [7] that the use of NGOS alone for network optimisation can penalise nodes with relatively low offered traffic. NGOS minimisation was, however, used for *RTE_OPT*, *RTE_OPTS*, and *RTE_OPTF* as subsequent network studies [8] were made considerably simpler by the use of a single parameter to reflect both the performance and reliability. Other approaches, such as those used by Lin *et al.* [9,10], could be used after NGOS minimisation if node-to-node grade-of-service (NNGOS) imbalances became too great. Lin *et al.*'s algorithm takes an existing routing table (SOC [9], OOC [10]) and attempts to equalise NNGOS imbalances in such a way as to bring all the NNGOS values below a supplied maximum, BPMAX.

By way of illustration, Figure 1 shows the frequency distribution of the overall NNGOS for the Example Network, optimised using OOC-SF routing at link availabilities of 0.98.

Figure 1 Frequency Distribution of NNGOS



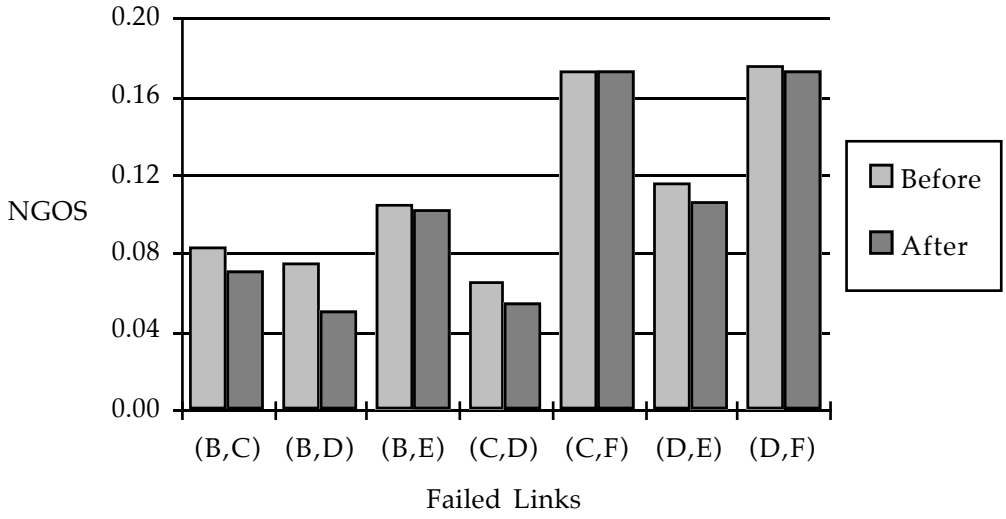
The overall NGOS corresponding to the results in Figure 1 is 0.0158, but the mean NNGOS is 0.0256 with standard deviation 0.0234. Clearly, while most of the values fall either below or immediately around the mean, a few are considerably higher.

4. Routing Table Updating in the Event of Link Failure

One of the possible techniques that might be used to improve trunk network fault-tolerance is the automatic (or pre-calculated manual) adjustment of the node routing tables in the event of link failure. In this paper, only the failure of links that do not disconnect the network are considered, as these offer the best opportunities for improvement. Results for the Example Network, using OOC-SF routing and link availabilities of 0.98, follow.

There are seven links whose failure does not disconnect the Example Network, namely (B,C), (B,D), (B,E), (C,D), (C,F), (D,E) and (D,F). Figure 2 shows the NGOS corresponding to each link failure both before and after re-optimisation with the appropriate link failure assumed. The results given are for overall NGOS conditioned on the appropriate link failure, i.e. NGOS allowing for the unavailability of the remaining links, rather than simply the NGOS for the single failure-and-repair state. For comparison purposes, recall that the unconditioned NGOS is 0.0158, optimised at link availabilities of 0.98.

Figure 2 NGOS Conditioned on Link Failure both Before and After Re-optimisation



Clearly, while re-optimisation has little effect on conditional NGOS in the case of some link failures, quite substantial gains are made in others. For example, re-optimisation in the event of a link (D,F) failure improves the NGOS by only 1%, whereas for link (B,D) the NGOS is improved by 33%. Some caution must be exercised, however, in dismissing re-optimisation for some link failures on the grounds of conditional NGOS values alone. If link (D,F) fails, four node pairs are disconnected, and while re-optimisation only improves NGOS by 1%, it nevertheless restores communication between all node pairs.

Another observation of interest is the differing degree to which the various link failures degrade the network. Comparing values before failure and after re-optimisation, the loss of link (D,F) increases the NGOS by some 997%, whereas that of link (B,D) causes only a 220% increase. Examination of the Example Network confirms the reasonableness of this result: link (D,F) failure all but isolates nodes F, H and I, whereas losing the centrally located link (B,D) can be easily handled by traffic redistribution.

In order to provide a detailed comparison, Table 4 shows the routing table for the Example Network, optimised by *RTE_OPTF*, conditional on link (B,D) having failed, and with link availabilities at 0.98. The corresponding unconditional routing table is given in Table 3. The NGOS obtained using Table 3, after link (B,D) failure, is 0.0752; re-optimisation resulting in Table 4 yields 0.0505.

When link (B,D) fails, using Table 3, three node pairs are disconnected: (F,A), (H,A) and (I,A). The corresponding path-loss sequences [1] are {P_1(FCDBA), P_2(FDBA), L_1(FL)} for (F,A); {P_1(HFCDBA), P_2(HFDBA), L_1(HFL), L_2(HL)} for (H,A); and {P_1(IFCDBA), P_2(IFDBA), L_1(IFL), L_2(IL)} for (I,A). In all three cases, both the alternative success paths contain the failed link.

Table 4 OOC-SF Routing Table with link (B,D) failed

From \ To	A	B	C	D	E	F	G	H	I
A		B	b	b	b	b	b	b	b
B	A		C	CE	E	CE	E	CE	CE
C	B	BD		DF	BD	FD	DB	FD	FD
D	CE	EC	CF		EC	FC	E	FC	FC
E	BD	B	BD	DB		DB	G	DB	DB
F	Cd	Cd	CD	DC	Dc		CD	H	I
G	e	e	e	e	E	e		e	e
H	f	f	f	f	f	F	f		f
I	f	f	f	f	f	F	f	f	

Table 4 differs from Table 3 in 29% of its entries, requiring the routing to be changed at five nodes: B, C, D, E and F. While this may seem a large number of changes to make, if only the entries corresponding to the failed link are removed then a routing table loop results. Similarly, a few attempts were made to find, by inspection, a loop-free routing table that required changes at less than five nodes, but without success.

In conclusion then, if fixed alternative routing is already used in a trunk network, and mechanisms exist or can be easily installed to permit rapid alteration of the node routing tables, then the use of routing table updating in the event of link failure will confer major benefits at minimal cost. However, if this approach is used, then clearly additional techniques will have to be developed to find an appropriate loop-free change sequence between the standard and conditional routing tables.

5. Acknowledgements

The research on which the paper is based was undertaken at the Dept. of Electronic Systems Engineering, University of Essex, initially as part of a project undertaken for the degree of M.Sc. in Telecommunication and Information Systems, sponsored under a SERC Industrial Studentship by GPT Limited. The author is currently employed as British Telecom Lecturer in Communications Network Modelling at the Dept. of Electronic Systems Engineering, University of Essex.

The author would like to take this opportunity to thank Prof. Charles Hughes, formerly as M.Sc. supervisor, and now as colleague, whose suggestions and encouragement have proved invaluable.

6. References

1 Sinclair, M.C.
"Single-Moment Analysis of Unreliable Trunk Networks: Two Methods for the Network Grade-of-Service"
Proc 8th UK Teletraffic Symposium, Beeston, Apr 91, p8.1-11

2 Lin, P.M., Leon, B.J. & Stewart, C.R.
"Analysis of Circuit-Switched Networks Employing Originating-Office Control with Spill-Forward"
IEEE Trans Commun vCOM-26 n6 Jun 78 p754-765

3 Chan, W.S.
"Recursive Algorithms for Computing End-to-End Blocking in a Network with Arbitrary Routing Plan"
IEEE Trans Commun vCOM-28 n2 Feb 80 p153-164

4 Bonaventura, V., Cacopardi, S., Decina, M. & Roveri, A.
"Service Availability of Communication Networks"
Proc 1980 Nat Telecommun Conf, Houston, Dec 80, p15.2.1-15.2.6

5 Li, V.O.K. & Sylvester, J.A.
"Performance Analysis of Networks with Unreliable Components"
IEEE Trans Commun vCOM-32 n10 Oct 84 p1105-1110

6 Shier, D.R.
"A New Algorithm for Performance Analysis of Communication Systems"
IEEE Trans Commun vCOM-36 n4 Apr 88 p516-519

7 Grandjean, C.
"Call Routing Strategies in Telecommunication Networks"
Elec Comm v42 n3 1967 p380-391

8 Sinclair, M.C.
"Strategy for a Fault-Tolerant Trunk Network"
MSc. Thesis, E.S.E., University of Essex, Mar 90

9 Lin, P.M., Leon, B.J. & Stewart, C.R.
"System Control for Circuit-Switched Communications"
Proc 7th Annual Southeastern Symposium on System Theory, Auburn, Alabama, Mar 75, p256-260

10 Lin, P.M., Leon, B.J. & Stewart, C.
"Performance Analysis and Routing Table Updating for Networks Employing Originating-Office Control"
Proc Natl Telecomm Conf Dec 75 p8.1-8.5

7. Appendix: The Example Network

Figure 3

A Circuit-Switched Communications Network

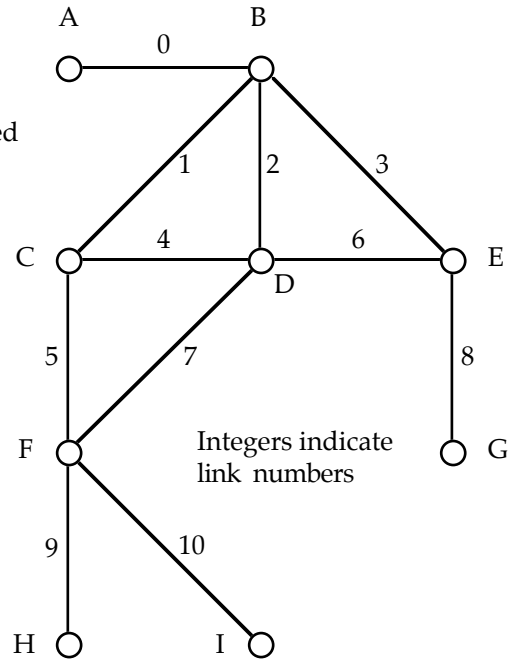


Table 5 Routing Table

From \ To	A	B	C	D	E	F	G	H	I
A		B	B	B	B	B	B	B	B
B	A		C	D	E	C	E	D	D
C	B	B		D	D	F	D	F	F
D	B	B	C		E	F	E	F	F
E	B	B	D	D		D	G	D	D
F	C	C	C	D	D		D	H	I
G	E	E	E	E	E	E		E	E
H	F	F	F	F	F	F	F		F
I	F	F	F	F	F	F	F	F	

Table 6 Network Configuration Table

From \ To	A	B	C	D	E	F	G	H	I
A		0	-1	-1	-1	-1	-1	-1	-1
B	0		1	2	3	-1	-1	-1	-1
C	-1	1		4	-1	5	-1	-1	-1
D	-1	2	4		6	7	-1	-1	-1
E	-1	3	-1	6		-1	8	-1	-1
F	-1	-1	5	7	-1		-1	9	10
G	-1	-1	-1	-1	8	-1		-1	-1
H	-1	-1	-1	-1	-1	9	-1		-1
I	-1	-1	-1	-1	-1	10	-1	-1	

Table 7 Link Capacity & Availability Table

Link Number	Capacity	Avail.
0	30	1.0
1	30	1.0
2	30	1.0
3	30	1.0
4	30	1.0
5	30	1.0
6	30	1.0
7	30	1.0
8	30	1.0
9	30	1.0
10	30	1.0

Table 8 Busy-Hour Traffic Matrix

From \ To	A	B	C	D	E	F	G	H	I
A		0.94	0.94	1.88	0.94	0.94	0.47	0.47	0.47
B	0.94		1.88	3.76	1.88	1.88	0.94	0.94	0.94
C	0.94	1.88		3.76	1.88	1.88	0.94	0.94	0.94
D	1.88	3.76	3.76		3.76	3.76	1.88	1.88	1.88
E	0.94	1.88	1.88	3.76		1.88	0.94	0.94	0.94
F	0.94	1.88	1.88	3.76	1.88		0.94	0.94	0.94
G	0.47	0.94	0.94	1.88	0.94	0.94		0.47	0.47
H	0.47	0.94	0.94	1.88	0.94	0.94	0.47		0.47
I	0.47	0.94	0.94	1.88	0.94	0.94	0.47	0.47	