STRATEGIES FOR DYNAMIC NETWORK MANAGEMENT

Åke Arvidsson
Department of Communication Systems*
Lund Institute of Technology
Box 118
S-221 00 Lund
Sweden

Abstract

We study management of transport networks based on virtual paths, i.e. rearrangable end-to-end transport network "highways". Among its main advantages are found cost savings, network operation simplification and enhanced network management capabilities. Virtual paths networks are readily implemented in any network using the synchronous digital hierarchy and/or the asynchronous transfer mode.

Algorithms for virtual path designs are reviewed and found to have a few shortcomings. A new algorithm is therefore proposed and it is found to compare favourably with the algorithm providing the most similar features.

Applying the proposed algorithm to a large number of networks, we discuss operational considerations associated with reconfigurable networks and investigate simple traffic estimation procedures and their performance in this context. Finally, the choice of reconfiguration parameters such as updating frequency, and the impact of network design time are examined.

1 Reconfigurable Virtual Path Networks for ATM

1.1 Description

We are concerned with the part of a telecommunications network that often is referred to as the "transit network", figure 1, i.e. the part that carries calls between subscribers that belong to different local switching centres. It consists of transit switching systems interconnected by high capacity links, e.g. coaxial cables, micro wave links, satellite links or optical fibres.

For a number of reasons it may be desirable to enhance a basic, physical network by creating a higher layer which is logically independent of the underlying physical network. Such a network is formed by reserving a certain amount of transmission capacity on a series of links and cross connecting the reserved channels through possible, intermediate transit nodes. The combination of reserved capacities on links and possible cross connections is called a virtual path (VP). Referring to figure 1, a VP between nodes T1 and T4 could e.g. be formed by reserving equal amounts of transmission capacity on links L1, L3 and L5 and cross connecting the reserved channels through T2 and T5.

Two nodes being interconnected by a VP is, from the point of view of acceptance, routing and establishment of calls, essentially identical to having a single, direct route between them to which they have exclusive access. Figure 2 shows how the physical transit network of figure 1 can be transformed into a fully interconnected logical transit network by means of VPs.

The process of creating and/or rearranging a (logical) network of VPs between a number of end nodes, i.e. partitioning the transmission capacities of the links between VPs, is called capacity management [10], bandwidth management [21], bandwidth switching [1] or bandwidth control [30] and is performed at a network management centre. Since the

*The work reported herein was partly carried out at the Teletraffic Research Centre, The University of Adelaide, G.P.O. Box 498, Adelaide, South Australia 5001, Australia.

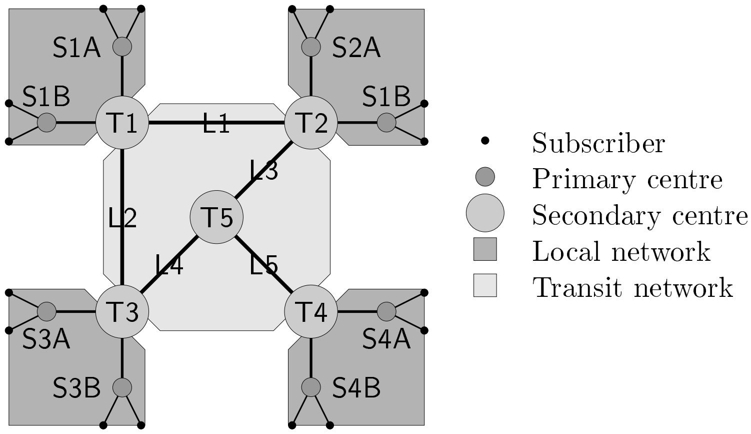


Figure 1: *Example of network.*

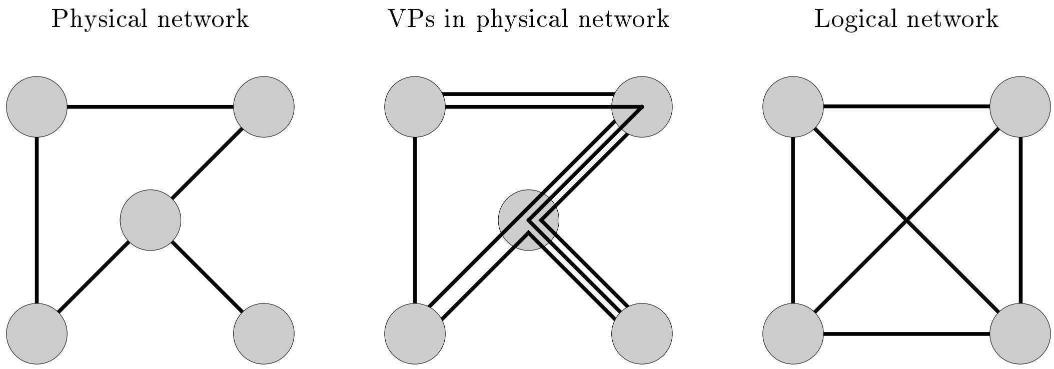


Figure 2: *Formation of VPs in the example network.*

optimality of a certain VP configuration is dependent on both link capacities and currently offered traffics, VP assignments must be re-evaluated in response to changes. The act of updating of VPs is referred to as dynamic capacity management (DCM). DCM can be performed either in advance or on demand. The former means that assignments are predefined and changed in an independent manner while individual call attempts control rearranging in the latter. We focus on the former alternative as this is the one most favoured by low transmission-to-processing costs [10, 11].

1.2 Motives

Some of the motives behind VP networks and DCM [2, 9, 10, 11, 29, 30] are

Network Costs: Due to continuous deployment of sophisticated, new services and successive introduction of increasingly complex signaling systems, switching tends to become more expensive while recent progress in optical fibre technology cause a relative decrease in high speed transmission costs. In particular, the marginal cost of high speed transmission tends to compare favourably to low speed transmission. With increasing relative costs of switching and decreasing relative costs of high speed transmission, the current trend is towards a network with a small number of switching systems interconnected through a sparse network of high capacity transmission links.

In a network with direct VPs between end nodes, switching functions such as acceptance, routing and establishment of calls are carried out directly between these nodes and the functions of transit nodes are reduced to one: Switching. Reducing the functionality of a transit switch opens up possibilities of considerable simplifications and hence cutting costs.

Clearly, possible savings are partly offset by reduced link utilisation, but studies have shown that substantial cost savings can be made. Moreover, current tendencies are expected to continue, whence a network structure with fewer nodes and a sparse grid of high speed transmission links is likely to prove an increasingly attractive alternative in the future.

Simplified Multiplexing: Questions relating to statistical multiplexing of bursty sources sharing a common medium are still largely unsolved. Existing proposals for call acceptance algorithms often include time consuming calculations for each link traversed.

Major difficulties encountered are the different statistical nature of various types of sources (e.g. voice, data and video) and their varying requirements on cell delay and loss. Both problems can be considerably simplified by grouping call requests into traffic classes (TCs) according to bandwidth and quality-of-service demands and assigning separate VPs for each TC. Class specific VPs means that multiplexing only takes place between sources of similar statistical properties and can be organised such that the particular delay and loss requirements for the class in question are met.

Faster Call Handling: VP networks offer faster call handling through simplified multiplexing and end-to-end paths. Simplified multiplexing, as outlined above, allows for simpler algorithms or tables which reduce call acceptance and processing times. End-to-end paths eliminate transit node processing hence the total processing time for a call bid is decreased. Moreover, signaling will be performed directly end-to-end and lessen signal transfer and processing times while the resulting, reduced signal load further contributes to speeding up call handling.

Improved Network Management Capabilities: VPs provide a quick means to reroute traffics in case of temporary overloads or equipment failures in that only the contents of the transit nodes' routing tables need to be modified. This applies not only to exceptions or emergencies but also to common, daily changes of busy hour areas.

A Means for Providing New Services: VPs can easily be set up between a number of access points of a customer to form a closed, private subnetwork under control of the user.

1.3 Implementation

The concept of VPs is also used in asynchronous transfer mode (ATM) cell addressing [2, 12, 23, 27, 31] and in the synchronous digital hierarchy (SDH) [8, 15, 24]. Both cases are easily mapped to the present network concept.

Cell addressing in ATM is performed through a 20 bit address in the user-to-network interface (UNI) and a 24 bit address in the network-to-network interface (NNI). Of these, the 12 least significant bits are referred to as the virtual circuit identifier (VCI) and the remaining ones, 8 and 12 for the UNI and NNI respectively, form the virtual path identifier (VPI). If VPs of a VP network are numbered, the VPI of each cell can be used to indicate the current VP and cells may be routed through the VP network only by means of the VPIs. Referring to figure 1, VCIs may be used in the local network and VPIs in the transit network.[1] Though transit switching is carried out on cell-by-cell basis, transit nodes are relieved from all functions but switching and smaller addresses mean smaller and cheaper switching networks.

Networks employing the plesiochronous digital hierarchy (PDH) [14] require full demultiplexing down to individual STM channels at every node before switching can take place, hence calls are demultiplexed and then again multiplexed at every intermediate node. A major aim of SDH is to reduce the need for repeated demultiplexing and multiplexing. SDH is based on virtual containers (VCs) of various sizes. A VC can contain lower order VCs or data, synchronous transfer mode (STM) channels or ATM cells. VCs may be switched "in bulk" and hence carried intact across a number of transit nodes. Such a chain of nodes is said to form a virtual path (VP). Only at the end of a VP is the VC demultiplexed and its contents examined: Lower order containers may either be further demultiplexed or refitted into other outbound containers, STM channels (ATM data) may be delivered to a call-by-call (cell-by-cell) switch which, in turn, either forwards it to the end user or to new, outbound VCs. A VP in the SDH sense is thus similar to the current network concept: A logical wide band, end-to-end channel in the transit network. SDH nodes, digital cross connects and add/drop multiplexers, are currently being developed or are already available [3, 4].

2 Algorithms for Dynamic Capacity Management

2.1 Existing Algorithms

We have found DCM algorithms or algorithms closely related to DCM published by Gopal et al. [18, 19], Herzberg [21], Evans [13], Gersht et al. [16, 17, 24] and Mase et al. [26]. Algorithms are also outlined by Hui et al. [23]. Finally, Mase et al. [27] discuss such algorithms in terms similar to those in [13, 16, 17, 23, 24] but without going into any detail. Summing up on these it is found that

- most algorithms explicitly or implicitly presume a linear relationship between the capacity of a VP and its call carrying capability; this is true only if the sources are smooth, e.g. voice, and/or sufficiently many of them are multiplexed together,[2]
- most algorithms explicitly or implicitly presume the existence of predefined paths for all VPs; selecting and storing predefined paths for every possible situation of overload and/or equipment failures is far from trivial, and
- some algorithms produce real valued solutions; VPs in PDH- or SDH-based networks must be determined as an integer number of circuits or virtual containers respectively.

We have developed a new algorithm that does not require a linear relationship between the capacity of a VP and its call carrying capability nor predefined paths and that produces integer valued solutions. Moreover, the optimisation function according to which VP capacities are assigned can be chosen arbitrarily. The algorithm is a heuristic and as such it does not guarantee that the final solution is a global optimum. On the other hand, the "optimality" guaranteed by some of the algorithms above is not entierly global but only applies only under the condition of a given set of paths.

[1] The 12 bit VCI restricts the number of circuits per path to $2^{12} = 4,096$ hence several VPI-numbers may refer to the same VP. For the same reason, a series of VPI-numbers may also be needed for calls that do not traverse the transit network in figure 1.

[2] For further discussions on this subject, refer to works on equivalent capacity etc., see for example [6, 7, 20, 22, 27] and references given therein.

2.2 Our Algorithm

Our algorithm is referred to as the automatic greedy algorithm[3] and exhibits some similarities to the algorithm developed by Gopal et al. [18, 19], though we permit several TCs between OD-pairs, non-linear equivalent bandwidths and search dynamically for routes. Letting t denote an arbitrary TC and o (d) an arbitrary node of origin (destination), its main steps can be described as follows:

1. For $t = 1, \ldots, T$, read the table that provides the relationship between capacities and circuits.
2. Read the number of nodes N, link capacities $C_{o,d}$ (expressed as units of capacity) and offered traffics $A_{t,o,d}$ (expressed as Erlangs).
3. Assign high, initially acceptable call loss levels $\alpha(t, o, d)$ for all TCs t and all OD-pairs o, d.
4. Find the shortest paths available to each traffic t, o, d.
5. Compute the gain achieved for each traffic t, o, d if one unit of capacity is added to its shortest path.
6. Compute the loss paid for each traffic t, o, d if one unit of capacity is added to its shortest path.
7. Find the traffic $t_{\max}, o_{\max}, d_{\max}$ that yields the highest gain/loss ratio.
8. If the highest gain/loss ratio is 0 then go to step 11, else proceed to step 9.
9. Assign one unit of capacity to the traffic $t_{\max}, o_{\max}, d_{\max}$ that provides the highest gain/loss ratio.
10. Go to step 4.
11. If low, ultimately acceptable loss levels $\alpha(t, o, d)$ have been reached for all traffics or all capacity has been assigned then stop, else proceed to step 12.
12. Reduce currently acceptable loss levels $\alpha(t, o, d)$ for all t, o, d.
13. Go to step 4.

It is observed that the algorithm tries to successively distribute the available capacity to VPs in such a way that a minimum amount of capacity is used and that maximum value is obtained for each unit of capacity, by using shortest paths and by chosing the VP with the highest gain/loss ratio respectively. The successive reductions of acceptable loss serve to ensure that reasonable fairness in grade of service is maintained among VPs and to make sure that assigned capacity will actually carry a reasonable amount of traffic. The algorithm will terminate when for every VP either (*i*) a final, predetermined, desirable loss level has been reached or (*ii*) no more capacity is available for VPs which still suffer from unacceptable losses.

The tables referred to in step 1, one for each TC, give the number equivalent of circuits $N(t, i)$ for TC t and $C = i$ units of capacity, $i = 1, 2, \ldots$. These tables are computed from traffic characteristics, grade-of-service demands, buffer space and acceptable loss, see e.g. [6, 7, 20, 22, 27]. The quantity "unit of capacity" is the integer unit in which capacity is

[3] "Automatic" refers to that the algorithm finds VP paths itself and "greedy" to that every step performed aims at achieving the highest immediate profit without taking long term consequences of a particular decision into account.

expressed: For PDH an integer number of 64 kb/s channels, for SDH the smallest virtual container used and for pure ATM networks a rate in cells/second large enough to carry a call of any TC.

Our implementation uses an initial setting of loss levels at step 3 of 50% and reduces it in step 11 to the ultimately acceptable level of 0.05% through two intermediate levels of 5% and 0.5% respectively.

Shortest paths in step 4 are determined using Floyd's algorithm [25] with the length associated to link o, d designed to find the shortest path in number of links traversed with preferential treatment to paths having more spare capacity left than other paths of equal length

$$\text{Length}(o, d) = \begin{cases} 1 + \dfrac{1}{{C'_{o,d}}^4} & C'_{o,d} > 0 \\ \infty & C'_{o,d} = 0 \end{cases} \tag{1}$$

where $C'_{o,d}$ denotes the remaining, not yet assigned capacity on link o, d.

In step 5, the gain $G(t, o, d)$ for t, o, d is the additional t-traffic that would be carried from o to d if one unit of capacity was added to its currently shortest path

$$G(t, o, d) = A_{t,o,d}[E_{N'_{t,o,d}}(A_{t,o,d}) - E_{N_{t,o,d}}(A_{t,o,d})] \tag{2}$$

where $E_N(A)$ is the Erlang-B formula, $N_{t,o,d}$ is the total number of circuits currently available to t, o, d and $N'_{t,o,d}$ is the total number of circuits that would be available to t, o, d if one unit of capacity was added to its shortest path. N is obtained from the assigned capacities through the tables read in step 1. Observing that circuits cannot be formed by joining capacities from several physically distinct routes, N and N' are determined for each route of the VP individually and then summed. Moreover, we define $G(t, o, d) \equiv 0$ if the capacity already assigned to t, o, d is enough to ensure that a certain degree of service $\alpha(t, o, d)$ has been achieved, that is if $E_{N_{t,o,d}}(A_{t,o,d}) < \alpha(t, o, d)$. We also set $G(t, o, d) \equiv 0$ if no more capacity is available to t between o and d.

The loss $L(t, o, d)$ associated to t, o, d addressed in step 6 is the sum of all gains that can be achieved at the same point and that require some of the capacity also requested by t, o, d:

$$L(t, o, d) = \sum_{t'=1}^{T} \sum_{o'=1}^{N-1} \sum_{d'=o'+1}^{N} I(\mathcal{L}_{t,o,d} \cap \mathcal{L}_{t',o',d'} \neq \emptyset)\, G(t', o', d') \tag{3}$$

where $\mathcal{L}_{t,o,d}$ is the set of links traversed by the shortest path for t, o, d and $I(\cdot)$ is an indicator function taking the value of 1 if its argument is true, otherwise 0.

2.3 Discussion

An obvious advantage with the proposed algorithm is its robustness. That is, unlike methods based on mathematical programming, it will remain stable and converge at the same speed for all types of non-linearities and discontinuities in gain and loss functions and irrespective of the ways in which routes for VPs are chosen. These properties leave full freedom to modify and extend the algorithm to meet particular demands such as

- Biased selection of routes: If, for technical or economical reasons, a biased selection of routes may be desirable this can readily be incorporated into the cost function (1) as penalties associated with certain links for some or all service TCs. Such penalties may be dependent on the capacity already seized on that link.

Algorithm	E_{Call}	$E_{\text{OD-pair}}$	U_{tot}	$P_{\text{OD-pair}}$	R_{tot}
A	39 (1.9)	59 (1.8)	64 (5.1)	(1.4)	5 (1.3)
B	25 (2.0)	5 (2.9)	0 (5.0)	(1.2)	3 (1.3)

Table 1: *Comparison of network performance using different DCM-algorithms.*

- A limitation to the number of distinct physical paths can be imposed by not allowing more than a prescribed number of distinct routes $R_{\max}$ for each traffic. For OD-pairs that already has $R_{\max}$ distinct routes, the shortest path is defined as the shortest of these.

- Predetermined routes: If, for some reason, predefined routes are preferred and available, automatic selection of routes can easily be inhibited by a modification similar to the one above.

- Profit maximisation: The gain $G(t, o, d)$ need not be just carried traffic but can, for example, be scaled both according to traffic TC, origin and destination, by multiplying (2) by a weight $w = w(t, o, d)$ based on e.g. revenues, distances or leasing cost for external lines.

- Traffic concentration: By defining a minimum allowable gain $G_{\min}$ can small and potentially uneconomical traffics be eliminated from a primary run. Such traffics can then be taken care of in a secondary run in which the VPs already established in the first run are seen as potential carriers, i.e. as virtual links. By thus carrying smaller traffics on top of larger ones can both be carried at a lower cost due to the non-linearities of the equivalent bandwidth and the Erlang B-formula.

2.4 Numerical Results

To investigate the power of the proposed algorithm (A), it was applied to a series of eight distinct networks, each consisting of ten nodes and subject to eight different traffic patterns, each summing up approximately 7,000 Erlangs. Details on the networks are found in appendix A. To enable comparisons to the comparable algorithm (B) of [18, 19], the number of TCs was set to one.[4] The predefined paths required by B were taken as the four most used ones found by A. The unit of capacity was set to 10 circuits for both algorithms.

Results are summarised in table 1. E_{Call} is the loss averaged over all calls in the network, $E_{\text{OD-pair}}$ is the loss averaged over all OD-pairs, U_{tot} is the mean carried traffic per seized unit of capacity and $P_{\text{OD-pair}}$ the mean number of distinct routes used per OD-pair. For each algorithm is given the number of times it produced the *best* result with the *mean* result over all 64 configurations within parenthesis. Loss is expressed in % and utilisation in carried Erlangs per unit of capacity. Finally, R_{tot} is the ratio between the virtual capacities of the rearranged networks to actual capacities of the physical ones averaged over all 8 networks.

The table suggests that, for the networks and traffic patterns considered, our algorithm results in a slightly better performance in terms of E_{Call} and even better in terms of $E_{\text{OD-pair}}$. Also, a somewhat higher degree of network utilisation U_{tot} is recorded and we observe equal savings in transmission capacity of about 30% R_{tot} by means of different VP arrangements for different traffic patterns.

[4] Results on two TCs, voice and frame relaying, are found in [6, 7].

3 Applying Dynamic Capacity Management

3.1 Operational Considerations

Changing physical routes and altering capacity assignments of VPs will introduce a need to rearrange calls in progress. New physical routes will result in calls having to be moved from one physical path to another. Rearranging, or repacking, will not be given any further attention here. Altering capacities, however, may result in VPs being forced to drop some calls. Such calls must either be rerouted over tandem nodes or prematurely cleared. Neither of these alternatives are very attractive: The former means increased demands on node processing and transmission capacity, while the latter is unacceptable from subscribers' point of view.

Our policy is to provide one-hop rerouting if this is possible. The alternative route is selected according to the Least Busy Alternative (LBA) strategy [28]: For each pair of trunks between the nodes in question is the highest utilisation computed after which the pair with the lowest maximum utilisation is chosen. Tandem routing over more than one node is prohibited in the interest of utilisation efficiency. Hence, if all two-hop paths are blocked, premature clearing is used as a last resort. Further, rerouting is combined with limited repacking so that, at every network updating point, rerouted calls are moved back to direct routes as far as possible.

3.2 Traffic Estimation

In a real environment, offered traffics are not known but must be estimated from forecasts and/or on-line measurements. This paper does not address traffic prediction and estimation explicitly, hence we restrict ourselves to giving some limited results on simple on-line measurement techniques and their performance. The focus on on-line measurements is motivated by a wish to device a fully automatic management system.

We distinguish between two methods of on-line traffic estimation: Arrival counting (AC) and carried traffic measurements (CT).[5] In AC, the number of call attempts N received during a measurement interval of length t_{M} is recorded from which the offered traffic A is estimated as $A = N/t_{\mathrm{M}}$. In CT is the carried traffic A', i.e. the number of busy circuits, averaged over similar interval from which A is computed by "backward computation", i.e. by solving for A in $A' = AE_N(A)$.

[5] More advanced alternatives include e.g. moving average, filtering and adaptive filtering, possibly including forecasting.

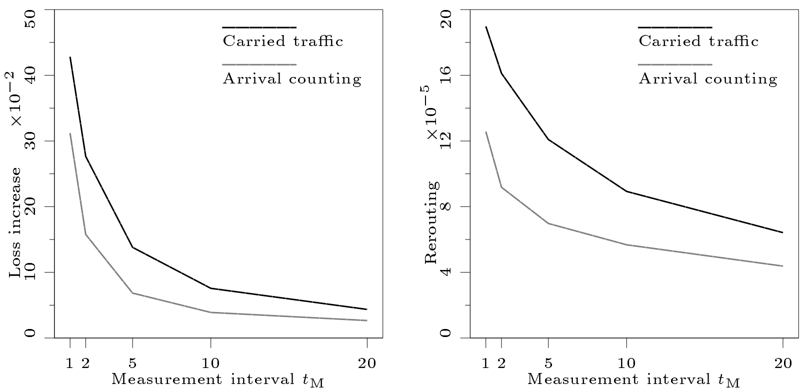


Figure 3: *Network performance for various measurement intervals and estimation methods under constant load.*

Normalising time to the mean call holding time, we have simulated the same eight networks and traffic configurations as above during 300 time units each for $t_{\mathrm{M}} = 1, 2, 5,$ 10 and 20 time units. Assuming an ideal execution time $t_{\mathrm{E}} = 0$ for the DCM algorithm, reconfigurations were performed immediately upon completion of measurement periods. A constant updating interval $t_{\mathrm{U}} = 20$ was used throughout the simulations.

Figure 3 shows observed performance in terms of lost calls (increase relative to an infinite measurement interval) and fraction of calls subject to rerouting. We immediately observe that AC performs superior to CT, irrespective of the length of the observation interval. It is also clear short observations intervals result in increased loss and rerouting. The intuitively obvious explanation is that short observation intervals results in inaccurate estimates. The diagrams further indicate that $t_{\mathrm{M}} = 10$ seems to yield a reasonable accuracy.

3.3 Time Scale

3.3.1 Constant Traffic

To get an idea of suitable time scales for network reconfigurations, the same networks as above were simulated during 300 time units for $t_{\mathrm{U}} = 1, 2, 5, 10$ and 20, both for the ideal execution time $t_{\mathrm{E}} = 0$ and the two non-ideal ones $t_{\mathrm{E}} = 1$ and 10 respectively. Guided by previous results, we chose AC as our estimation policy and $t_{\mathrm{M}} = 10$ as our estimation interval. Results are depicted in figure 4.

The three loss curves in the right diagram differ considerably: $t_{\mathrm{E}} = 0$ means *lower* loss for more frequent updating, $t_{\mathrm{E}} = 1$ results in loss appearing to be *independent* of updating and $t_{\mathrm{E}} = 10$ responds to frequent updating by *higher* loss.

To explain this, recall that the idea of frequent updating is to redimension VPs by shifting capacity between them as load levels vary stochastically. The faster and the more frequent new measurements are transformed into network designs, the closer will VPs follow these variations. On the other hand, every network update will result in some calls being subject to rerouting and the more frequent the updates, the more calls will be rerouted, left diagram in figure 4. While an ordinary call goes end-to-end on a single virtual link, a rerouted call traverses two virtual links thereby reducing the total number of calls that can be carried by the network. Hence, frequent updating becomes advantageous only if the gain from network matching is large enough to compensate for the loss associated with rerouting. Clearly, this is the case for $t_{\mathrm{E}} = 0$ while for $t_{\mathrm{E}} = 1$ gain and loss equal and for $t_{\mathrm{E}} = 10$ gain cannot make up for loss.

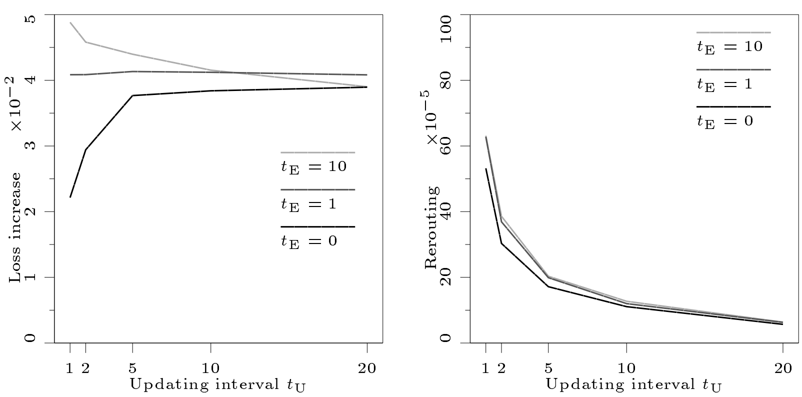


Figure 4: *Network performance for various updating intervals and execution times under constant load.*

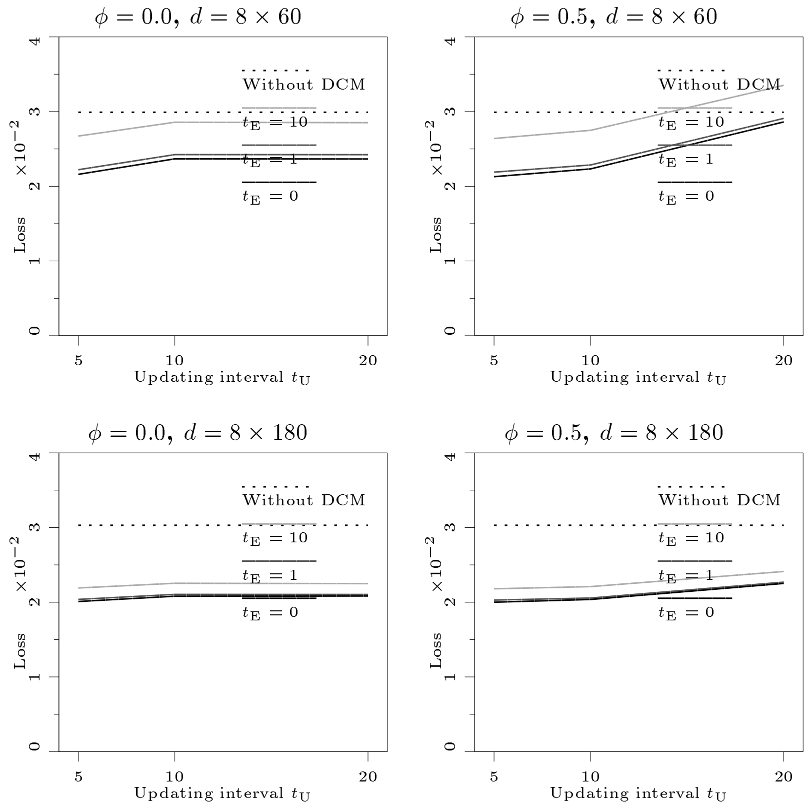


Figure 5: *Network loss for various updating intervals and execution times under load varying step wise.*

In other words, frequent updating will reduce network loss, but the improvement disappears quickly with increasing execution time and it is counteracted by the associated rerouting. It is concluded that DCM is of no use to meet short term fluctuations as it can provide gain for constant traffics only if t_U and t_E both are very small and then at a high processing cost (large number of calls rerouted).

3.3.2 Variable Traffic

Recalling the reuse factor R in table 1, it is clear that considerable savings can be made for variable traffics. To study this, simulations were conducted in which the eight physical networks above were subject to each of its eight distinct traffic patterns for a period of 60 (180) time units per pattern resulting in durations d of $d = 8 \times 60$ and $d = 8 \times 180$ respectively: Traffic 1 during 0–60 (0–180), traffic 2 during 60–120 (180–360) and so on until finishing by traffic 8 during 420–480 (1260–1440).

Based on the negative results for frequent updating, it was decided to omit the shortest updating times, $t_U = 1$ and 2 respectively, while $t_U = 5$, 10 and 20 were retained. It was also decided to let network updates start either "in phase" ($\phi = 0.0$) or "out of phase" ($\phi = 0.5$) with traffic changes, e.g. for $t_U = 5$ and $d = 8 \times 60$ updates commence at 0, 5, 10,... with traffic changes taking place at 0, 60,... and at 2.5, 7.5, 12.5,... still with traffic changes at 0, 60,... for $\phi = 0.0$ and $\phi = 0.5$ respectively.

Also, all simulations employed the same estimation concept as above (AC, $t_M = 10$) and the same values of t_E, $t_E = 0$, 1 and 10 respectively. To compare network performance with and without DCM, we also designed a network based on the average of all eight traffics which used for all traffics throughout the simulation. Results are illustrated in

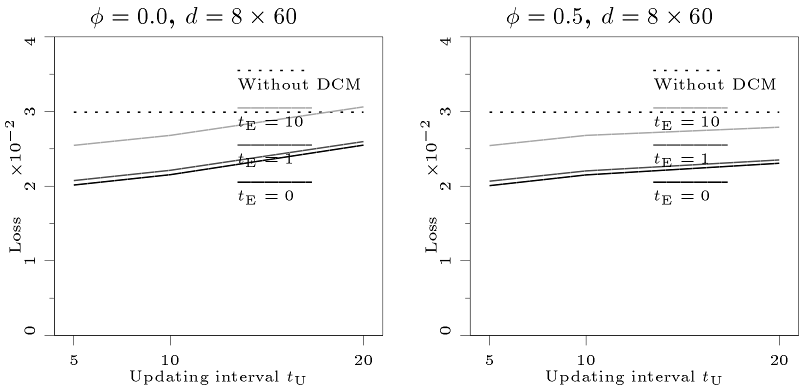


Figure 6: *Network loss for various updating intervals and execution times under load varying in smoothed steps.*

figure 5.

Comparing results, it is noted that losses for $t_E = 0$ and $t_E = 1$ are almost the same while losses for $t_E = 10$ are distinctively higher. For the case of slower changes, $d = 8 \times 180$, all values of t_E, t_U and ϕ tried result in a significant improvement as compared to the case where DCM is not used. For faster changes, $d = 8 \times 60$, we observe that a combination of infrequent updating, $t_U = 20$, and unfortunate updating moments, $\phi = 0.5$, actually results in a network that is inferior to the static one dimensioned for the average traffic!

It should, however, be kept in mind that these results apply to a network in which *all* traffics change in *zero time*, a rather unlikely event in a real network. For comparison, the runs with $d = 8 \times 60$ above were repeated with the only modification that traffics do not change in a single step but gradually during 5 time units. Results show, figure 6, that only the slight smoothening of traffic changes means that neither phase ϕ nor execution time t_E have such a strong impact on network performance.

4 Conclusions

VPs and DCM are two important issues in a broadband network based on ATM and/or SDH. We have presented an algorithm for DCM and examined its performance under various traffic estimation strategies and intervals and for different updating intervals and algorithm execution times.

It has been concluded that, with reasonable parameter values (figure 4), DCM is not suitable to exploit variations in utilisation caused by natural stochastic fluctuations in what essentially is a steady state mode of operation. This is explained by their fast rate and random behaviour: Variations occur faster than they can be detected and used for network design and their random nature means that results have very limited validity in time.

DCM is, however, most useful for rearranging networks as the actual offered load varies: For the networks considered, we have found that without DCM about 30% extra capacity would be required to provide the same degree of service as with DCM (table 1) or that the degree of service was improved by 30% by using DCM (figures 5 and 6).

These values were obtained for a simple traffic measurement strategy (figure 3), i.e. by counting the number of incoming calls during 10 mean call holding times, and for updating intervals of up to 20 mean call holding times under the assuption that traffic patterns last for at least 60 mean call holding times.

5 Further Work

The results presented above only represent a sample of important issues related to DCM algorithms and application strategies. Further areas include, but are not limited to,

Alternate routing: Alternate routing, possibly combined with state protection, has long been employed as a way of coping with varying load in circuit switched networks, see e.g. [5] for references. A most interesting topic is to compare DCM to alternate routing and to a DCM-network on which alternate routing is used.

Speed improvement: The present work has shown that the speed of the updating algorithm is important to network performance: The faster the algorithm, the lower the loss. Speed can be improved e.g. by reducing the work per cycle by less frequently searching for new paths or by reducing the number of cycles by assigning capacity in larger units during the early stages. These and other ways all need to be evaluated. Another variant to investigate is to make performance less sensitive to speed e.g. by providing more recent traffic estimates at certain points during the computation.

Layered DCM: From the point of view of multiplexing efficiency and utilisation, high capacity VPs with large traffics are superior to low capacity VPs with small traffics. Hence it may be desirable to merge several small traffics to one large traffic that can be carried by one high capacity VP. Such a traffic can be formed by merging different TCs between the same OD-pair, one TC between a number of OD-pairs or both. Either of these means that internal sub-VPs must be formed within the VP between TCs, OD-pairs or both respectively. Algorithms for such strategies and possible gains require further studies.

Improved traffic estimation: Traffic estimates are a key to efficient DCM. Algorithms for this, their parameters and their performance also require further attention.

References

[1] Addie, R. & Warfield, R.: *Bandwidth Switching and New Network Architectures*, paper no. 2.3iiA.1 in Proc. 12th Int. Teletraffic Cong., Torino, Italy (1988).

[2] Addie, R., Burgin, J. & Sutherland, S.: *Information Transfer Protocols for the Broadband ISDN*, paper no. 22.6 in Proc. IEEE Globecom '88, Hollywood, Florida, U.S.A. (1988).

[3] Ahmed, H., Brégi, P., Haag, J.-P. & Wiest, B.: *Synchronous Transmission Systems and Cross Connects for Broadband Networks*, Electr. Commun., vol. 64, no. 2/3, pp. 194–203 (1990).

[4] Andersson, J.-O., Improta, S. & Evangelisti, G.: *Digital Cross-Connect Systems — A Tool for Managing the Transport Network*, paper no. C6.5 in Proc. XIIIth Int. Switching Symp., Stockholm, Sweden (1990).

[5] Arvidsson, Å.: *Priorities in Circuit Switched Networks*, Dis. thesis, Department of Communication Systems, Lund Institute of Technology, Lund, Sweden (1990).

[6] Arvidsson, Å.: *A Study on Statistical Multiplexing and Dynamic Capacity Management in Voice/Frame Relaying Networks*, Report no. 7/92, Teletraffic Research Centre, The University of Adelaide, Adelaide, South Australia, Australia (1992).

[7] Arvidsson, Å.: *On Dynamic Capacity Management in Voice/Frame Relaying Networks*, paper no. 2.3 in Proc. 10:th Nordic Teletraffic Sem., Århus, Denmark (1992).

[8] Astani, K., Harrison, R. & Ballart, R.: *CCITT Standardization of Network Node Interface of Synchronous Digital Hierarchy*, IEEE Commun. Mag., vol. 28, no. 8, pp. 15–20 (1990).

[9] Brungard, D., Grotjohann, H. & Kallenberg, P.: *Impact of New Transmission Technologies on the Network Architecture, Network Management and Economics*, paper no. A1.2 in Proc. XIIIth Int. Switching Symp., Stockholm, Sweden (1990).

[10] Burgin, J.: *Management of Capacity and Control in Broadband ISDN*, Int. J. of Dig. and Ana. Cabled Sys., vol. 2, pp. 155–165 (1989).

[11] Burgin, J.: *Broadband ISDN Resource Management*, Computer Netw. and ISDN Syst., vol. 20, no. 1–5, pp. 323-331 (1990).

[12] Cheung, N.: *The Infrastructure for Gigabit Computer Networks*, IEEE Commun. Mag., vol. 30, no. 4, pp. 60–68 (1992).

[13] Evans, S.: *A Mathematical Model and Related Problems of Optimal Management and Design in a Broadband Integrated Services Network*, J. Australian Math. Soc., ser. B, vol. 31, part 2, pp. 150–175 (1989).

[14] CCITT Recommendations G.702–G.704: *Digital Hierarchies*, CCITT, ITU, Geneva, Switzerland.

[15] CCITT Recommendations G.707–G.709: *Synchronous Digital Hierarchy*, CCITT, ITU, Geneva, Switzerland.

[16] Gersht, A. & Shulman, A.: *Optimal Routing in Circuit Switched Networks*, IEEE Trans. on Commun., vol. 37, no. 11, pp. 1203–1211 (1989).

[17] Gersht, A. & Kheradpir, S.: *Integrated Traffic Management in SONET-Based Multi-Service Networks*, pp. 67–72 in Proc. 13th Int. Teletraffic Cong., Copenhagen, Denmark (1991).

[18] Gopal, G., Kim, C.-K. & Weinrib, A.: *Dynamic Network Configuration Management*, paper no. 302.2 in IEEE Int. Conf. on Commun., Atlanta, Georgia, U.S.A. (1991).

[19] Gopal, G., Kim, C.-K. & Weinrib, A.: *Algorithms for Reconfigurable Networks*, pp. 341–347 in Proc. 13th Int. Teletraffic Cong., Copenhagen, Denmark (1991).

[20] Guérin, R., Ahmadi, H. & Naghshineh, M.: *Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks*, IEEE J. Sel. Areas in Commun., vol. 9, no. 7, pp. 968–981 (1991).

[21] Herzberg, M.: *Network Bandwidth Management — A New Direction in Network Management*, pp. 218–225 in Proc. 6th Australian Teletraffic Sem., Wollongong, New South Wales, Australia (1991).

[22] Hong, D. & Suda, T.: *Congestion Control and Prevention in ATM Networks*, IEEE Netw. Mag., vol. 5, no. 4, pp. 10–16 (1991).

[23] Hui, J., Gursoy, M., Moayeri, N. & Yates, R.: *A Layered Broadband Switching Architecture with Physical or Virtual Path Configurations*, IEEE J. Sel. Areas in Commun., vol. 9, no. 9, pp. 1416–1426 (1991).

[24] Kheradpir, S., Gersht, A. & Stinson, W.: *Performance Management in SONET-Based Multi-Service Networks*, paper no. 39.3 in Proc. IEEE Globecom '91, Phoneix, Arizona, U.S.A. (1991).

[25] Kleinrock, L.: *Queueing Systems, Volume II: Computer Applications*, John Wiley & Sons, New York (1976).

[26] Mase, K. & Imase, M.: *An Adaptive Capacity Allocation Scheme in Telephone Networks*, IEEE Trans. on Commun., vol. 30, no. 2, pp. 354–359 (1982).

[27] Mase, K. & Shioda, S.: *Real-Time Network Management for ATM Networks*, pp. 133–140 in Proc. 13th Int. Teletraffic Cong., Copenhagen, Denmark (1991).

[28] Mitra, D. & Gibbens, R.: *State-Dependent Routing on Symmetric Loss Networks with Trunk Reservations. II: Asymptotics, Optimal Design*, Ann. Op. Res., vol. 35, no. 1–4, pp. 3–30 (1992).

[29] Moondra, S.: *Impact of Emerging Switching-Transmission Cost Tradeoffs on Future Telecommunications Network Architectures*, IEEE J. Sel. Areas in Commun., vol. 7, no. 8, pp. 1207–1218 (1989).

[30] Ohta, S. & Sato, K.-I.: *Dynamic Bandwidth Control of the Virtual Path in an Asynchronous Transfer Mode Network*, IEEE Trans. on Commun., vol. 40, no. 7, pp. 1239–1247 (1992).

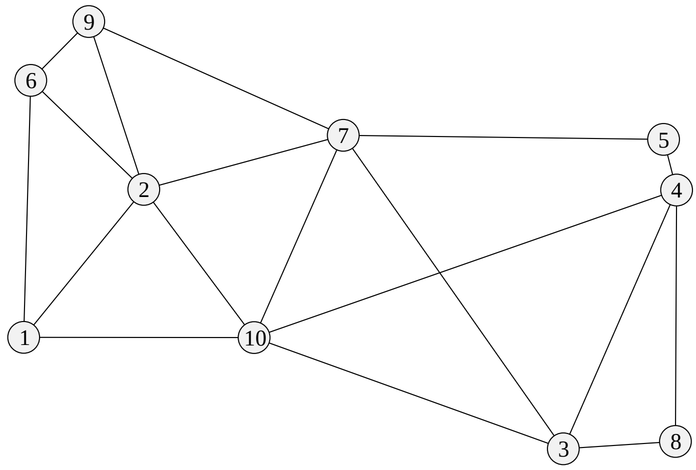
[31] Sato, K.-I., Ohta, S.. & Tokizawa, I.: *Broad-Band ATM Network Architecture Based on Virtual Paths*, IEEE Trans. on Commun., vol. 38, no. 8, pp. 1212–1222 (1990).

A Test Networks

The set of test networks used was obtained by artificially generating a series eight of voice networks each consisting of $N = 10$ nodes scattered at random over a fixed rectangular area. Nodes were interconnected through a network of links in such a way that no node would be cut off from the rest of the network in case of a failure in any other node. Transmission capacities between OD-pairs were assigned in units of 640 kb/s links, with the probability of obtaining any number 1,. . . ,36 of links being equal to 1/36. All traffics were routed along the shortest paths in terms of number of links traversed. Offered traffics were initially engineered to give a loss of exactly 1% between every OD-pair, given the assigned paths and capacities.

To reproduce typical busy hour behaviour of networks, eight variations on the basic traffic patterns were generated as follows: First, coordinates of a "hot spot" centre were then chosen at random with equal probability over the entire area. An imaginary circle was then drawn around this point and all nodes within the circle were assumed to experience busy hour. The radius of the circle was successively increased until at least three nodes were within the busy hour zone. Traffics were then modified randomly within a lower and an upper bound with uniform probability throughout the range. Hence, traffics between two busy hour nodes were increased by 10–30%, traffics between a busy hour node and a normal node were reduced by 10–30% and traffics between two normal nodes were reduced by 30–50%.

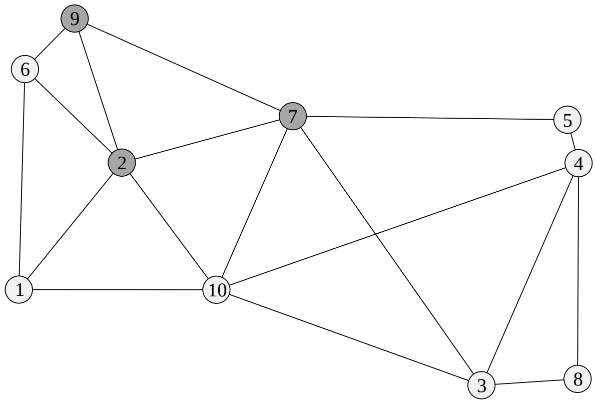
The following illustrations provide full information on one of these networks: Figure 7 shows the basic network, link capacities C (in units of 10 circuits) and average traffics A and figures 8–15 show the corresponding hot spot configurations and associated traffic values.



OD	C	OD	C	OD	C	OD	C	OD	C	OD	C
1– 2	96	1– 6	43	1–10	132	2– 6	59	2– 7	164	2– 9	77
2–10	72	3– 4	57	3– 7	195	3– 8	94	3–10	80	4– 5	65
4– 8	18	4–10	70	5– 7	90	6– 9	6	7– 9	132	7–10	34

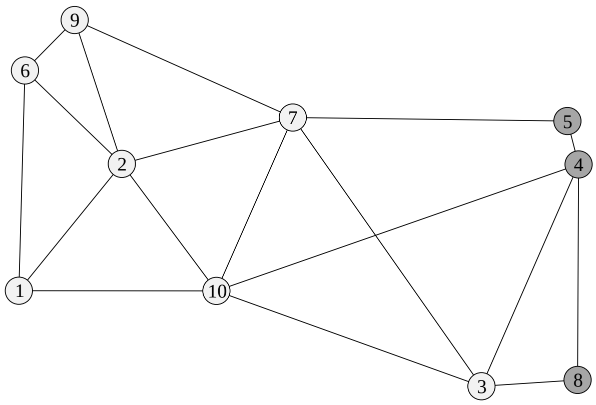
OD	A	OD	A	OD	A	OD	A	OD	A
1– 2	334.54	1– 3	225.20	1– 4	167.81	1– 5	284.87	1– 6	41.29
1– 7	118.47	1– 8	216.32	1– 9	130.54	1–10	204.11	2– 3	227.10
2– 4	85.07	2– 5	5.11	2– 6	75.94	2– 7	340.40	2– 8	57.41
2– 9	235.72	2–10	256.27	3– 4	90.12	3– 5	57.24	3– 6	195.80
3– 7	149.17	3– 8	4.55	3– 9	191.19	3–10	176.67	4– 5	332.52
4– 6	86.70	4– 7	133.02	4– 8	106.25	4– 9	206.07	4–10	102.44
5– 6	71.01	5– 7	221.46	5– 8	38.80	5– 9	224.32	5–10	159.57
6– 7	83.70	6– 8	26.54	6– 9	41.45	6–10	217.71	7– 8	225.95
7– 9	278.50	7–10	304.11	8– 9	215.15	8–10	48.10	9–10	295.28

Figure 7: *Basic network configuration.*



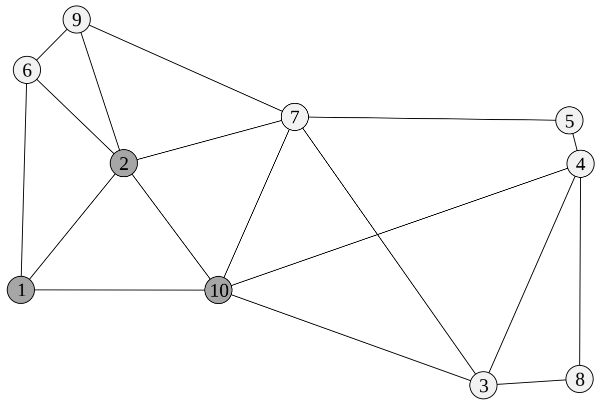
OD	A	OD	A	OD	A	OD	A	OD	A
1– 2	366.5	1– 3	200.4	1– 4	126.8	1– 5	254.5	1– 6	39.5
1– 7	122.5	1– 8	195.6	1– 9	145.0	1–10	153.8	2– 3	241.6
2– 4	93.2	2– 5	4.7	2– 6	86.6	2– 7	413.6	2– 8	58.0
2– 9	286.9	2–10	278.9	3– 4	80.6	3– 5	46.5	3– 6	191.7
3– 7	155.7	3– 8	4.2	3– 9	233.7	3–10	142.1	4– 5	294.6
4– 6	71.4	4– 7	148.3	4– 8	87.1	4– 9	244.8	4–10	85.5
5– 6	64.8	5– 7	223.8	5– 8	34.9	5– 9	232.6	5–10	141.0
6– 7	91.3	6– 8	26.0	6– 9	53.9	6–10	218.5	7– 8	225.9
7– 9	372.2	7–10	326.7	8– 9	247.7	8–10	43.2	9–10	320.8

Figure 8: *Traffic configuration 1.*



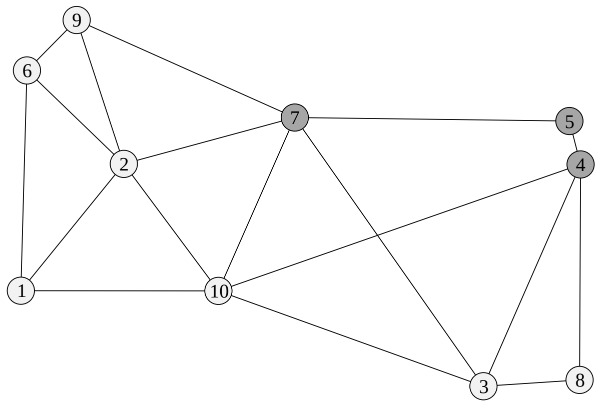
OD	A	OD	A	OD	A	OD	A	OD	A
1– 2	292.6	1– 3	199.6	1– 4	175.0	1– 5	318.1	1– 6	38.9
1– 7	108.7	1– 8	238.7	1– 9	103.2	1–10	188.2	2– 3	209.4
2– 4	82.7	2– 5	5.1	2– 6	67.8	2– 7	298.9	2– 8	63.7
2– 9	199.2	2–10	207.9	3– 4	100.1	3– 5	59.3	3– 6	174.4
3– 7	128.9	3– 8	5.4	3– 9	166.7	3–10	160.4	4– 5	379.8
4– 6	89.0	4– 7	139.1	4– 8	135.2	4– 9	206.3	4–10	113.4
5– 6	82.8	5– 7	224.7	5– 8	49.8	5– 9	225.0	5–10	177.2
6– 7	71.0	6– 8	29.4	6– 9	38.3	6–10	211.2	7– 8	245.2
7– 9	231.9	7–10	248.1	8– 9	227.9	8–10	52.8	9–10	271.6

Figure 9: *Traffic configuration 2.*



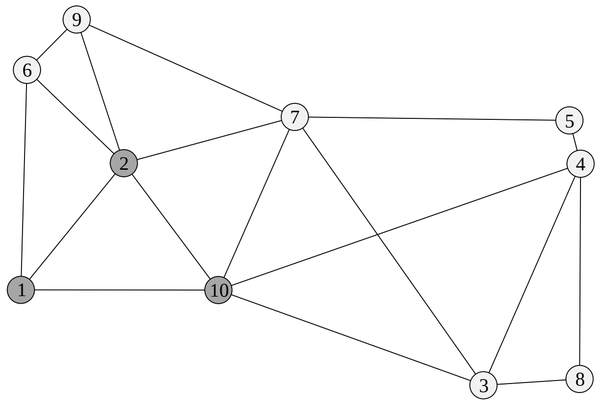
OD	A	OD	A	OD	A	OD	A	OD	A
1– 2	420.2	1– 3	254.0	1– 4	188.0	1– 5	322.5	1– 6	48.3
1– 7	120.2	1– 8	236.5	1– 9	149.3	1–10	265.8	2– 3	226.3
2– 4	89.8	2– 5	5.5	2– 6	84.3	2– 7	322.4	2– 8	57.7
2– 9	264.9	2–10	296.3	3– 4	80.5	3– 5	48.4	3– 6	204.2
3– 7	135.3	3– 8	4.6	3– 9	190.1	3–10	178.6	4– 5	263.5
4– 6	87.1	4– 7	102.5	4– 8	89.7	4– 9	167.8	4–10	98.3
5– 6	69.5	5– 7	182.4	5– 8	32.5	5– 9	217.3	5–10	180.1
6– 7	70.2	6– 8	26.7	6– 9	42.5	6–10	253.1	7– 8	215.4
7– 9	275.0	7–10	299.8	8– 9	208.0	8–10	52.4	9–10	352.8

Figure 10: *Traffic configuration 3.*



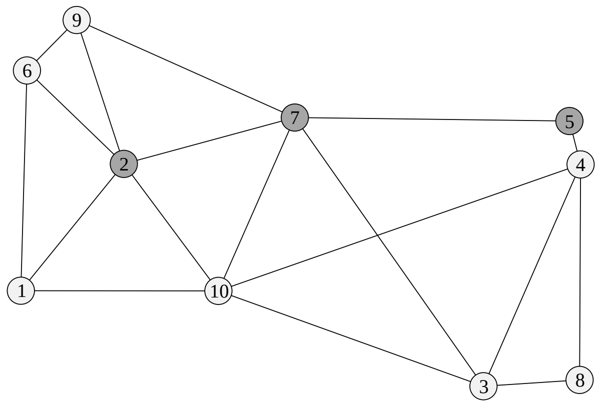
OD	A	OD	A	OD	A	OD	A	OD	A
1– 2	279.1	1– 3	198.0	1– 4	165.7	1– 5	267.9	1– 6	37.8
1– 7	131.7	1– 8	189.2	1– 9	107.5	1–10	170.7	2– 3	194.1
2– 4	79.3	2– 5	4.9	2– 6	65.5	2– 7	345.9	2– 8	51.6
2– 9	228.4	2–10	225.3	3– 4	99.9	3– 5	61.2	3– 6	186.8
3– 7	175.1	3– 8	3.7	3– 9	168.2	3–10	157.7	4– 5	415.8
4– 6	104.7	4– 7	175.3	4– 8	114.4	4– 9	219.7	4–10	104.6
5– 6	71.8	5– 7	269.6	5– 8	39.4	5– 9	258.9	5–10	149.8
6– 7	89.0	6– 8	24.0	6– 9	36.8	6–10	196.7	7– 8	234.7
7– 9	318.1	7–10	351.5	8– 9	188.2	8–10	42.1	9–10	238.4

Figure 11: *Traffic configuration 4.*



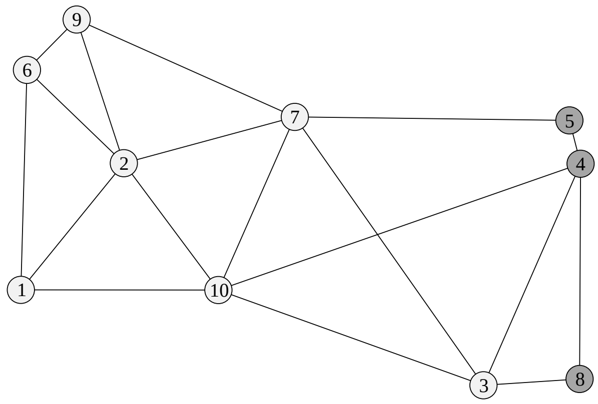
OD	A	OD	A	OD	A	OD	A	OD	A
1– 2	406.7	1– 3	252.4	1– 4	178.7	1– 5	272.3	1– 6	47.2
1– 7	117.9	1– 8	232.2	1– 9	153.6	1–10	248.3	2– 3	260.1
2– 4	86.4	2– 5	5.3	2– 6	82.0	2– 7	369.4	2– 8	57.3
2– 9	243.0	2–10	313.8	3– 4	80.3	3– 5	50.3	3– 6	169.4
3– 7	148.2	3– 8	3.9	3– 9	191.6	3–10	215.2	4– 5	299.6
4– 6	83.3	4– 7	111.3	4– 8	92.3	4– 9	181.2	4–10	111.0
5– 6	58.5	5– 7	184.1	5– 8	30.2	5– 9	202.1	5–10	152.7
6– 7	68.7	6– 8	27.5	6– 9	41.1	6–10	238.7	7– 8	204.9
7– 9	238.9	7–10	338.1	8– 9	217.5	8–10	51.6	9–10	319.6

Figure 12: *Traffic configuration 5.*



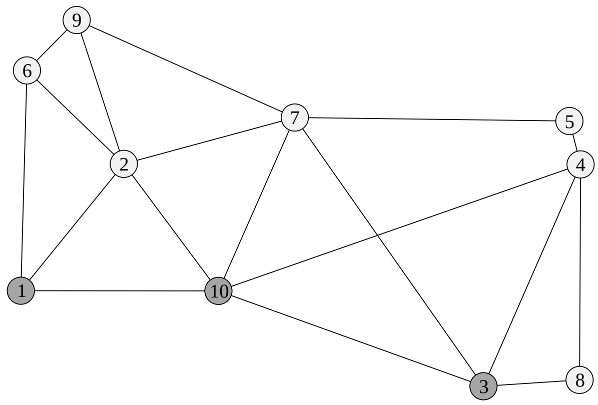
OD	A	OD	A	OD	A	OD	A	OD	A
1– 2	332.8	1– 3	196.4	1– 4	156.4	1– 5	276.8	1– 6	36.8
1– 7	129.4	1– 8	184.8	1– 9	111.8	1–10	153.2	2– 3	227.9
2– 4	93.5	2– 5	5.7	2– 6	80.8	2– 7	393.0	2– 8	63.1
2– 9	257.7	2–10	242.8	3– 4	80.2	3– 5	63.1	3– 6	199.2
3– 7	154.7	3– 8	4.0	3– 9	169.7	3–10	155.1	4– 5	317.6
4– 6	81.4	4– 7	129.4	4– 8	93.6	4– 9	187.9	4–10	95.9
5– 6	76.4	5– 7	271.3	5– 8	37.0	5– 9	243.6	5–10	155.7
6– 7	106.9	6– 8	24.8	6– 9	35.4	6–10	182.3	7– 8	224.2
7– 9	282.0	7–10	324.7	8– 9	197.7	8–10	41.3	9–10	270.4

Figure 13: *Traffic configuration 6.*



OD	A	OD	A	OD	A	OD	A	OD	A
1– 2	259.0	1– 3	195.6	1– 4	169.4	1– 5	281.2	1– 6	36.2
1– 7	90.2	1– 8	227.9	1– 9	128.6	1–10	187.6	2– 3	195.6
2– 4	83.1	2– 5	5.1	2– 6	62.0	2– 7	278.2	2– 8	57.0
2– 9	221.1	2–10	224.9	3– 4	99.7	3– 5	64.1	3– 6	181.8
3– 7	127.9	3– 8	5.2	3– 9	193.1	3–10	173.4	4– 5	402.8
4– 6	99.0	4– 7	147.5	4– 8	141.7	4– 9	239.7	4–10	102.2
5– 6	78.8	5– 7	229.0	5– 8	52.0	5– 9	235.9	5–10	158.6
6– 7	86.6	6– 8	28.3	6– 9	39.7	6–10	175.0	7– 8	243.6
7– 9	264.0	7–10	246.1	8– 9	227.0	8–10	50.9	9–10	286.4

Figure 14: *Traffic configuration 7.*



OD	A	OD	A	OD	A	OD	A	OD	A
1– 2	319.4	1– 3	305.1	1– 4	182.4	1– 5	285.7	1– 6	45.6
1– 7	127.1	1– 8	225.7	1– 9	145.4	1–10	265.3	2– 3	261.7
2– 4	72.6	2– 5	4.5	2– 6	78.5	2– 7	301.7	2– 8	50.9
2– 9	184.6	2–10	260.2	3– 4	99.6	3– 5	65.0	3– 6	258.8
3– 7	167.6	3– 8	5.4	3– 9	216.4	3–10	230.9	4– 5	286.5
4– 6	77.6	4– 7	110.9	4– 8	96.2	4– 9	201.2	4–10	108.6
5– 6	65.4	5– 7	186.7	5– 8	34.7	5– 9	179.1	5–10	161.5
6– 7	85.8	6– 8	25.6	6– 9	43.9	6–10	266.1	7– 8	213.8
7– 9	245.9	7–10	297.8	8– 9	207.2	8–10	50.5	9–10	302.3

Figure 15: *Traffic configuration 8.*