Reduction of Routing Complexity in Telecommunication Networks by a Novel Multilayer Decomposition Method

Åke Arvidsson,* Wlodek Holender,† and Torgny Karlsson†

Abstract

Routing problems are often encountered when designing and managing telecommunication networks. Today, routing problems are affected by the growing sizes of networks, which increase the complexity, and by introduction of new services and technologies, which rise the demands. Rather than resorting to entirely heuristic algorithms and/or large data bases with off-line precomputed routing information for various situations, we propose a a new decomposition method whereby any routing algorithm is speeded up considerably, thus permitting the deployment of well founded routing algorithms even for real time purposes. In our paper, we present this new method in formal terms, and apply it to a real routing problem. Finally, investigating the performance of our fast implementation by comparing it to the optimal, non-accelerated solution, we find that considerable time savings can be made at a limited cost in terms of non-optimality of the final solution. It is also emphasised that in real-time applications with non-constant traffics, this nominal degradation might be more than compensated for by the prompt delivery.

Keywords: Routing complexity, Multilayer network decomposition.

1 Introduction

Routing problems in telecommunication networks refer to the process of selecting a path or set of paths through a graph, between an origin (ingress) and a destination (egress). Classical routing problems in circuit switched networks include selecting a series of incident links for a circuit, and in packet switched networks selecting a series of incident links for a connection in the virtual circuit approach, or the next node for a packet in the datagram approach. For packet switched services in the broadband integrated services digital network (B-ISDN) based on the asynchronous transfer mode (ATM), at least four routing problems occur: The design of the physical network, the configuration of virtual paths (VPs) on the physical network, the setting up of virtual channels (VCs) on the network of VPs, and the routing of packets on the VCs. Each of these routing problems has its own time scale: Physical networks are typically extended and rearranged over days or more, VP configurations over hours, VC set up over minutes and packets routed over seconds or less.

Inappropriate routing will lead to poor quality of service (QoS) and/or poor utilisation of network resources. Efficient routing is therefore an essential step towards cost-effective telecommunications services, where high QoS is achieved with a minimum of resources. Clearly, adequate routing means that routing plans must be updated as conditions change: The faster the changes, the faster and more frequent must the routing plans be reevaluated in order to keep routing up to date. Preferable time scales found in the literature are a few times per hour for VP networks, *e.g.* [2], and a few times per minute for VC networks, *e.g.* [17]. A major limitation to dynamic updating is, however, the complexity of the algorithms involved, which typically grows with the number of nodes N and the number of links M.

*Department of Telecommunications and Mathematics, University of Karlskrona/Ronneby, S-371 79 Karlskrona, Sweden. Telephone int+46 455 780 53, facsimile int+46 455 780 57, email akear@itm.hk-r.se.

†Formerly with the Department of Communication Systems, Lund Institute of Technology, Box 118, S-221 00 Lund, Sweden.

The world wide introduction of the B-ISDN based on ATM and continuous deployment of new fibre optic technology mean that networks are becoming larger and more complex while at the same time transmission rates are increasing. As for routing problems, the former means increasing complexity and the latter means that faster responses are required. This has encouraged a search for faster and more efficient ways to solve routing problems. These rely on various heuristics, on limiting the search space by predefining specific, restricted sets of permissible routes, or on decomposition.

Our work focuses on a new approach to multilayer decomposition and routing (MDR). By this method a large and complex routing problem is split into a number of smaller and simpler routing problems. This is achieved by decomposing the network into a number of suitably sized overlapping subnetworks, after which an abstract upper layer is created where the overlapping subnetworks are represented as an abstract network. The simplification process may then be repeated recursively on the abstract network as required to obtain acceptable network sizes. Besides reducing complexity and thereby computation times, MDR also simplifies parallel computation and decentralised decision making. Another advantage is that locally confined changes or light over all fluctuations may be handled within the concerned subnetworks rather than by costly global updates.

The remainder of this paper is organised as follows: Section 2 is devoted to how the multilayer decomposition is implemented and used, and in Section 3 we discuss related work. The concept is applied to a problem of packet routing where we also discuss the details of a possible implementation in Section 4. Some numerical results follow in Section 5 and finally in Section 6 we give some conclusions and point at further research.

2 The Multilayer Approach

2.1 Decomposition

A graph model of a complex telecommunication network may contain a large number of nodes and links. The complexity of a routing problem in such a graph generally depends on the number of nodes N and the number of links M. Thus, to reduce the complexity we need to reduce this graph model to a simpler one with fewer nodes and links. An important requirement on such a reduction procedure is that it should preserve the essential properties of the network.

Our MDR-method means that we cover the graph with a number of overlapping subgraphs. An abstracted graph is then formed where the internal structure in each subgraph is represented by a logical node and the internal structure of the intersection between two subgraphs is represented by a logical link. Note that in this work we limit ourselves intersections of two networks only.

Figure 1 illustrates the concept. To abstract a network, it is partitioned into a number of overlapping subnetworks. The set of overlapping subnetworks is then abstracted to a network by abstracting subnetworks to nodes and intersections to links. The reverse process is called refinement. A network is refined into a set of overlapping subnetworks by refining the nodes to subnetworks and the links to intersections.

An abstracted graph can be further simplified by the same process of covering and abstracting in a recursive manner. The family of all abstracted graphs obtained is called a multilayer decomposition. We order the graphs in a hierarchical structure and say that the abstracted graph constitutes an upper layer to the original graph, Figure 1 shows the two layers L and $L+1$.

2.2 Routing

Routing decisions in a network represented as a multilayer decomposition are made sequentially for each layer starting from the highest abstraction level. The decisions on each layer are based on the result of some optimisation algorithm, and the results are used as preconditions to the routing decisions on the next lower layer.

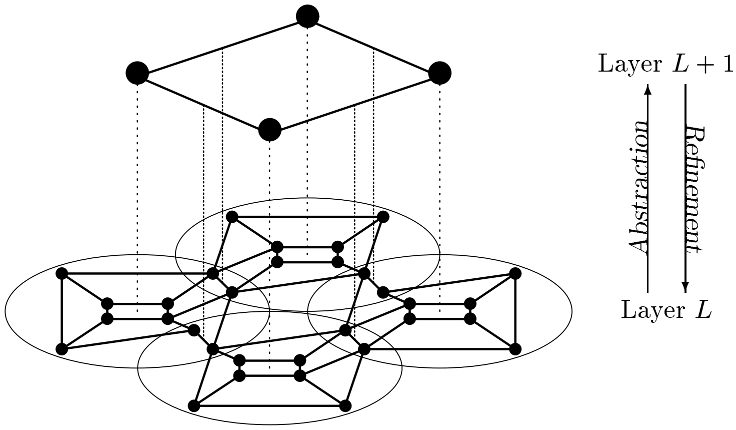

Figure 1: *Example of multilayer representation by abstraction/refinement.*

Complexity of routing algorithm	$N \log N$	N^2	$N^2 \log N$	N^3
Total complexity without decomposition	48	1024	1541	32768
Total complexity with decomposition	42	416	410	4064
Resulting gain from decomposition (%)	12	60	75	88

Table 1: *Complexity gains from applying multilayer routing in the example.*

For the example in Figure 1, we are faced with a routing problem of 32 nodes. By applying the decomposition shown in the figure, the problem is transformed to five smaller routing problems, one on layer $L + 1$ and four on layer L. The solution to the four node problem on layer $L + 1$ is used as input to the four layer L routing problems of ten nodes each.

From this example it is immediately noted that multilayer decomposition simplifies the routing decision problem by decreasing the dimensionality of the search space. Table 1 gives an idea of the resulting complexity reduction for various routing algorithms. This is achieved at the cost of information loss and may result in deterioration of the quality of the obtained solution. The extent of this reduction depends on the specific choice of subnetworks, the topology of the networks, the traffic interest between various node pairs, *etc.*

3 Related Methods

3.1 Non Overlapping Subnets

The traditional approach to graph decomposition corresponds to non overlapping subnetworks. In this approach, networks are covered by non overlapping subnetworks, each of which is abstracted to a node in the adjacent higher layer [1, 3, 5, 6, 9, 8, 12, 13, 16]. One of the important consequences of the non overlapping approach is that abstract links are not well defined. To circumvent this problem, the concept of multiple links is usually suggested, Figure 2, which considerably reduces the complexity reduction achieved. Since complexity reduction is the main reason to deploy decomposition, multiple links seems to be an unsatisfactory approach.

3.2 Discussion

The overlapping graph decomposition method introduced here, provides a true link abstraction procedure. By including common switching and transmitting capabilities related to

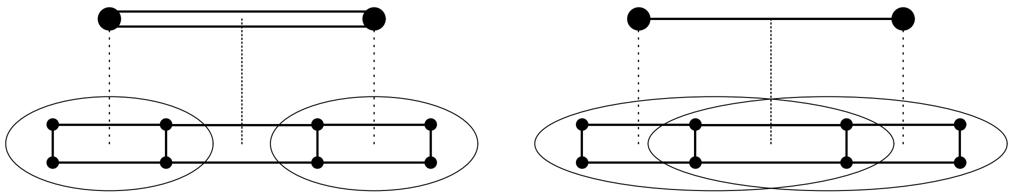

Figure 2: *Traditional decomposition by non overlapping subnets (left) and our approach by overlapping subnets (right).*

resource control *etc.*, we can achieve a meaningful definition of the logical link as an abstraction of interconnected subnetworks.

The removal of multiple link problem in our approach leads to a significant complexity reduction, since the network is represented by a "pure" graph model with single links connecting nodes at each layer. Pure graphs also mean that our approach can be used recursively. The latter allows for simplification to an arbitrary degree, as compared to the traditional approach of multiple links which comes to a halt after one simplification step.

On the other hand, the structural information about the network is lost during the abstraction process in our approach. This means that the refinement transformation in terms of routing results is not uniquely defined. A possible solution to this problem is shown in the following section.

4 Application to Data Traffic in ATM

4.1 Background

To illustrate our method in practice and to investigate its benefits, we consider a realistic problem, *viz.* data traffic in ATM.

It is well known that most kinds of data traffic exhibits burstiness and variations in many time scales, *e.g.* [4, 14, 15]. To efficiently match resources and demands in an ATM network carrying such traffic, various control mechanisms affecting both resources and demands must be applied on all of these time scales. One such mechanism routing.

We consider a physical network of links and nodes. The links are typically optical fibres and the nodes include ATM cross connects (ACCs), ATM switching systems (ASSs), and connectionless servers (CLSs) [7, 18]. The CLSs are attached to ASSs in a way similar to ordinary subscribers, and the ASSes are connected to one or more ACCs which in turn are directly interconnected. Users of packet switched services such as connectionless TCP/IP deliver packets to their local CLS either directly using *e.g.* a LAN interface, or via their local ASS using the normal ATM interface. The packets are then transported to the destination user via the CLS of the latter, possibly in a number of hops over intermediate CLSs, which act as intermediate nodes as in ordinary TCP/IP switching. The CLSs arrange their links as required by requesting VCs from their local ASSs. The ASSs in turn establish the VCs over VPs which are set up between various ASSes over one or more ACCs, Figure 3.

In such a network we are faced with a number of routing problems:

- to properly route the network of VPs subject to slow variations in demands, *e.g.* on the order of hours under the constraints of the current physical network;
- to properly route the network of VCs subject to moderate variations in demands, *e.g.* on the order of minutes under the constraints of the current VP network; and
- to properly route the packet flows subject to fast variations in demands, *e.g.* on the order of seconds under the constraints of the current VC network.

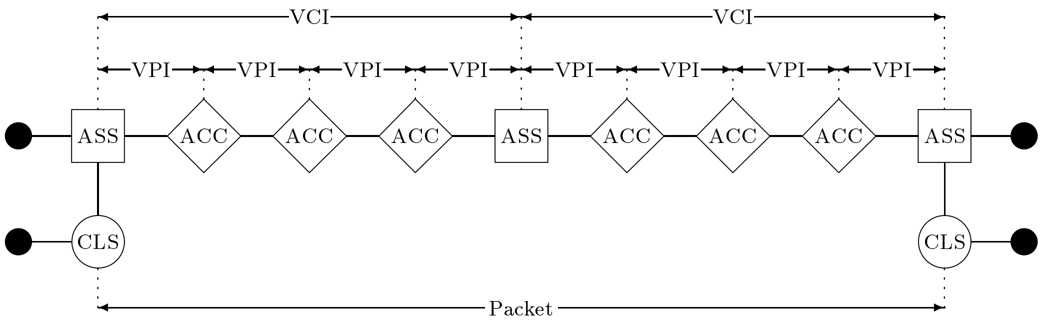

Figure 3: *Addressing on a link between two CLSs for connectionless traffic in ATM networks.*

This means that the CLSs are interconnected by a backbone network of VPs which designed to suit the long term traffic needs and modified in accordance with variations in these needs. The VP network supports a network of VCs which is configured to suit the medium term traffic needs and modified as required to match medium term requirement variations. The CLSs take care of the packet switching protocol, hence two packets referring to the same session need not go over the same physical route, VP, or VC, but the packet flow can be optimised with respect to short term traffic needs, *i.e.* with respect to current loads on servers (CLSs) and links (incident VCs).

Clearly, the faster the time scale of the routing problem, the more important is the speed of the routing algorithm employed. We treat this problem in a general sense and present a method to speed up any routing algorithm, irrespective of the problem considered. Hence, although the high complexity and real time demands of B-ISDN/ATM-networks are the fundamental motivation for this work, neither the rates nor hte extent of traffic variations are visible in the abstraction/refinement steps presented here, and we may for simplicity present the concepts under given resources and demands.

4.2 Problem

We consider a packet switched network with N nodes (CLSs), M links (VCs) the current transmission capacities of which described by a capacity matrix C

$$C = \begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,N} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ c_{N,1} & c_{N,2} & \cdots & c_{N,N} \end{pmatrix}$$

where $c_{o,d}$ is the unidirectional transmission capacity in bits per second from a node o to another node d. With respect to our application, these capacities are the result of the latest solution to the VC routing problem. If no direct physical link exists from o to d we set $c_{o,d} = 0$. A path between an origin o and a destination d is defined as an ordered set of nodes interconnected by direct physical links. We also introduce a traffic matrix ,

$$, = \begin{pmatrix} \gamma_{1,1} & \gamma_{1,2} & \cdots & \gamma_{1,N} \\ \gamma_{2,1} & \gamma_{2,2} & \cdots & \gamma_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{N,1} & \gamma_{N,2} & \cdots & \gamma_{N,N} \end{pmatrix}$$

where $\gamma_{o,d}$ is the total rate of packets per second which users at node o wish to transmit to users at node d. In our application these are the current rates which typically may be estimated by counting over a small time window, or equivalently by filtering through a suitable band pass filter, at the user interfaces in all nodes. The length of a packet is supposed to follow a negative exponential distribution with a fixed mean of B bits per packet. Finally we also introduce a flow vector Λ

$$\Lambda = \begin{pmatrix} \lambda_1 & \lambda_2 & \dots & \lambda_M \end{pmatrix}$$

where λ_m denotes the load on link $m = 1, \dots, M$.

Our routing problem consists of for every origin-destination pair (o, d) finding the optimal set of paths to be used, and the optimal proportion of the total traffic $\gamma_{o,d}$ that should be sent over each of these. Optimal is here taken in the sense of minimum expected delay for an arbitrary packet W

$$W = \sum_{m=1}^{M} \frac{\lambda_m}{\gamma(c_m/B - \lambda_m)}$$

where $\gamma = \sum_{o,d} \gamma_{o,d}$ and c_m is the capacity of link m. A straight forward solution to this problem is given in [11] as the flow deviation method. We will apply these results here in order to evaluate the impact of our scheme with respect speed in terms of faster computation and the sub-optimality in terms of increased W.

4.3 Multilayer Decomposition and Routing

For reasons of simplicity we restrict ourselves to two layers, *i.e.* to one abstract layer. Applying our strategy, the network is decomposed to a number of overlapping subnetworks. The subnetworks will constitute the logical nodes and the intersections will constitute the logical links in the upper layer. Once the capacities and traffics are known in for a layer, the flow deviation algorithm can immediately be used to solve to the corresponding routing problem. The result is then carried on down and converted to solve the routing problems for the subnetworks, after which the final solution is obtained by combining the results for the two layers. Summing up, the basic steps are

1. identify the set of overlapping subnetworks to be used,
2. compute the abstract network and traffics,
3. solve the routing problem on the upper layer,
4. compute the refined networks and traffics,
5. solve the routing problems on the lower layer, and
6. combine the solutions for the two layers.

4.3.1 Subnetwork Definition

The choice of subnetwork is a matter of clustering nodes. The sizes of the subnetworks determine the gain obtained by our approach, and the degree of suboptimality in the final result depends on exact choice of members in each subnetwork.

In our numerical studies below, we have largely ignored the suboptimality problem, but aimed at maximal complexity gain. The latter is achieved if all routing problems have the same search space, *i.e.* if all subnetworks have the same number of nodes. Clearly, for practical purposes, both suboptimality and complexity should be taken into account. Although this is an important topic, we only mention the heuristic observation that suboptimality may be reduced by keeping nodes with a large mutual traffic interest within the same subnetwork, and leave the details for further study.

Let $\mathcal{N}$ be the set of all nodes where o and d are two distinct, arbitrary nodes. Also let a similar notation with primes refer to the abstracted network, hence $\mathcal{N}'$ is the set of all abstract nodes (subnetworks) where o' and d' are two distinct, arbitrary abstract nodes (subnetworks). Also let N' and M' be the number of abstract nodes (subnetworks) and abstract links (intersections) respectively.

4.3.2 Abstracting Links

Let the nodes of intersections between two subnetworks be called common nodes with respect to these networks, and let nodes that uniquely belongs to one subnetwork be called proprietary nodes.

We define the capacity $c'_{o',d'}$ of an abstract link (intersection) from one abstract node (subnetwork) o' to another abstract node (subnetwork) d' as the minimum of (i) the total capacity from proprietary nodes of o' into the common nodes in the intersection and (ii) the total capacity from common nodes in the intersection to proprietary nodes of d',

$$c_{o',d'} = \min \left(\sum_{\substack{o \in \mathcal{P}(o') \\ d \in \mathcal{C}(o',d')}} c_{o,d}, \sum_{\substack{o \in \mathcal{C}(o',d') \\ d \in \mathcal{P}(d')}} c_{o,d} \right)$$

where $\mathcal{P}(o')$, $\mathcal{P}(d')$, and $\mathcal{C}(o',d')$ denote the set of proprietary nodes with respect to o' and d' and the set of common nodes with respect to o', d' respectively.

4.3.3 Abstracting Traffics

Let the traffic between two nodes in different subnetworks be called external traffic, and traffic between two nodes in the same subnetwork be called internal traffic.

We define the traffic $\gamma'_{o',d'}$ of from one abstract node (subnetwork) o' to another abstract node (subnetwork) d' as the sum of (i) all traffics from proprietary nodes of o' to all proprietary nodes in d', (ii) half the traffics from proprietary nodes of o' to all common nodes in d', (iii) half the traffics from common nodes of o' to all proprietary nodes in d', and (iv) a quarter of the traffics from common nodes of o' to all common nodes in d'

$$\gamma_{o',d'} = 1 \sum_{\substack{o \in \mathcal{P}(o') \\ d \in \mathcal{P}(d')}} \gamma_{o,d} + \frac{1}{2} \sum_{\substack{o \in \mathcal{P}(o') \\ d \in \mathcal{C}(d')}} \gamma_{o,d} + \frac{1}{2} \sum_{\substack{o \in \mathcal{C}(o') \\ d \in \mathcal{P}(d')}} \gamma_{o,d} + \frac{1}{4} \sum_{\substack{o \in \mathcal{C}(o') \\ d \in \mathcal{C}(d')}} \gamma_{o,d}$$

where $\mathcal{C}(n')$ denotes the set of nodes which are common to an abstract node (subnetwork) n' and any other abstract node (subnetwork). The factors 1, 1/2, 1/2 and 1/4 come from our definition that

- the external traffic from a proprietary node bound for another proprietary node terminates in the corresponding abstract node;
- the external traffic from a proprietary node bound for a common node is split equal with respect to destination between the abstract nodes in question;
- the external traffic from a common node bound for a proprietary node is split equal with respect to origin between the abstract nodes in question; and
- the external traffic from a common node bound for another common node is split equal with respect to both origin and destination between the abstract nodes in question.

4.3.4 Upper Layer Routing

The upper layer routing problem can now be solved by the flow deviation method using the matrices , $' = [\gamma'_{o',d'}]$, $C' = [c'_{o',d'}]$ with respect to the flow vector $\Lambda' = [\lambda'_{m'}]$ to minimise

$$W' = \sum_{m'=1}^{M'} \frac{\lambda'_{m'}}{\gamma'(c'_{m'}/B - \lambda'_{m'})}$$

Let $\beta'_{o',d'}(n'_1, n'_2)$ be the fraction of the abstract traffic $\gamma'_{o',d'}$ which, according to the optimal upper layer solution, flows through the abstract nodes n'_1 and n'_2.

4.3.5 Refining Links

Let the nodes of an abstract node (subnetwork) n' be called internal nodes with respect to n', and the abstract nodes (subnetworks) to which n' is directly connected be represented by external nodes, and let $\mathcal{I}(n')$ and $\mathcal{E}(n')$ denote the set of internal and external nodes with respect to n' respectively. Also let $\hat{n}''$ denote the neighbouring subnetwork represented by an external node n''.

We identify three classes of links: (i) internal links between two internal nodes in n', (ii) outbound links from an internal node of n' to an external one, and (iii) inbound links from an external node of n' to an internal one. Omitting the formalities of node numbering within subnetworks we write the capacities $c''_{o'',d''}$ as

$$c''_{o'',d''} = c_{o,d}$$

i.e. as the original links for $o'', d'' \in \mathcal{I}(n')$,

$$c''_{o'',d''} = \sum_{\delta \in \mathcal{P}(\hat{d}'')} c_{o,\delta}$$

i.e. as the sum of all outbound links with respect to $\hat{d}''$ for $o'' \in \mathcal{I}(n'), d'' \in \mathcal{E}(n')$, and

$$c''_{o'',d''} = \sum_{\omega \in \mathcal{P}(\hat{o}'')} c_{\omega,d}$$

i.e. as the sum of all inbound links with respect to $\hat{o}''$ for $o'' \in \mathcal{E}(n'), d'' \in \mathcal{I}(n')$.

Note that external nodes are artificial constructions and can not be used as transit nodes for traffics within the subnetwork. This also means that they do not contribute to the complexity of subnetwork routing problems.

4.3.6 Refining Traffics

We identify four classes of traffics: (i) internal traffics between two internal nodes in n', (ii) outbound traffics from an internal node of n' to an external one, (iii) inbound traffics from an external node of n' to an internal one, and (iv) transit traffic between two external nodes to n'. Omitting the formalities of node numbering within subnetworks as before we write the traffics $\gamma''_{o'',d''}$ as

$$\gamma''_{o'',d''} = \gamma_{o,d}$$

i.e. as the original traffics for $o'', d'' \in \mathcal{I}(n')$,

$$\gamma''_{o'',d''} = \begin{cases} 1 \displaystyle\sum_{\substack{\delta \in \mathcal{P}\delta' \\ \delta' \neq n'}} \beta'_{n',\delta'}(n', \hat{d}'')\gamma_{o,\delta} + \frac{1}{2} \sum_{\substack{\delta \in \mathcal{C}\delta' \\ \delta' \neq n'}} \beta'_{n',\delta'}(n', \hat{d}'')\gamma_{o,\delta} & o'' \in \mathcal{P}(n') \\ \displaystyle\frac{1}{2} \sum_{\substack{\delta \in \mathcal{P}\delta' \\ \delta' \neq n'}} \beta'_{n',\delta'}(n', \hat{d}'')\gamma_{o,\delta} + \frac{1}{4} \sum_{\substack{\delta \in \mathcal{C}\delta' \\ \delta' \neq n'}} \beta'_{n',\delta'}(n', \hat{d}'')\gamma_{o,\delta} & o'' \in \mathcal{C}(n') \end{cases}$$

i.e. as the sum of all outbound traffics with respect to $\hat{d}''$ for $o'' \in \mathcal{I}(n'), d'' \in \mathcal{E}(n')$,

$$\gamma''_{o'',d''} = \begin{cases} 1 \displaystyle\sum_{\substack{\omega \in \mathcal{P}\omega' \\ \omega' \neq n'}} \beta'_{\omega',n'}(\hat{o}'', n')\gamma_{\omega,d} + \frac{1}{2} \sum_{\substack{\omega \in \mathcal{C}\omega' \\ \omega' \neq n'}} \beta'_{\omega',n'}(\hat{o}'', n')\gamma_{\omega,d} & d'' \in \mathcal{P}(n') \\ \displaystyle\frac{1}{2} \sum_{\substack{\omega \in \mathcal{P}\omega' \\ \omega' \neq n'}} \beta'_{\omega',n'}(\hat{o}'', n')\gamma_{\omega,d} + \frac{1}{4} \sum_{\substack{\omega \in \mathcal{C}\omega' \\ \omega' \neq n'}} \beta'_{\omega',n'}(\hat{o}'', n')\gamma_{\omega,d} & d'' \in \mathcal{C}(n') \end{cases}$$

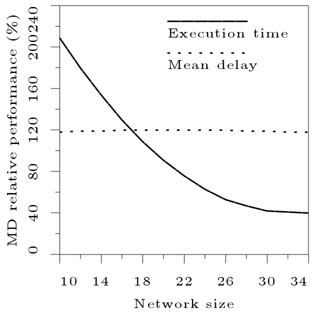

Figure 4: *Relative execution time (solid line) and mean delay (dotted line) for MDR vs. network size.*

i.e. as the sum of all inbound traffics with respect to $\hat{o}''$ for $o'' \in \mathcal{E}(n'), d'' \in \mathcal{I}(n')$, and

$$\gamma''_{o'',d''} = \sum_{\substack{\omega' \neq n' \\ \delta' \neq n'}} \beta'_{\omega',\delta'}(\hat{o}'', n')\beta'_{\omega',\delta'}(n', \hat{d}'')\gamma'_{\omega',\delta'}$$

i.e. as the sum of all transit traffics with respect to $\hat{o}''$ and $\hat{d}''$ for $o'' \in \mathcal{E}(n'), d'' \in \mathcal{I}(n')$.

5 Numerical Results

In this section the computation time and optimality of the result with and without MDR will be compared.

5.1 External Performance

We first examine how the two approaches behave with respect to network size. To prevent the flow deviation algorithm from finding the optimal result directly, which would reduce the value of the comparison, we keep the average load on every link of approximately 50% of its nominal capacity. For the same reason we distribute the traffics unevenly in the network. That means that every node sends all its traffic to one specific node and receives traffic from only one node, *i.e.* all elements but one are zero in every row of , .

Figure 4 shows the relative execution times and mean delays *vs.* network size, where a least square estimation with a second grade polynomial has been used to smooth the curves.

The solid line shows the execution time with MDR normalised by the result without MDR, so that 100 % corresponds to equal execution times. It is noted that the slope is negative and decreasing, *i.e.* MDR performs better the larger the network with a larger difference for smaller networks. For a network with 30 nodes is our method about 60% faster.

Similarly, the dotted line shows resulting mean delay with MDR normalised by the result without MDR, so that 100% corresponds to equal mean delays. The figure shows that the relative mean delay is almost independent of network size, and that the loss of information resulting from decomposition only has a minor impact on the quality of the result, or about 20%.

It is concluded that there is an overhead associated with MDR which determines its applicability. For small networks, here about 18 nodes or less, MDR as nothing to offer, but for larger networks can considerable savings in execution time be obtained at a restricted cost in steady state performance. It is emphasised that this nominal performance degradations might in fact be reversed when applied to a dynamic environment, where a formally more accurate solution might very well be less accurate in practice. This happens if the computation takes so long time that the offered traffics might have changed in the mean time.

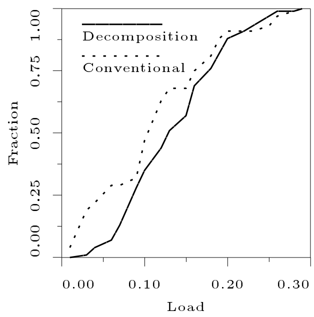

Figure 5: *The distribution of traffic loads with MDR (solid line) and without (dotted line).*

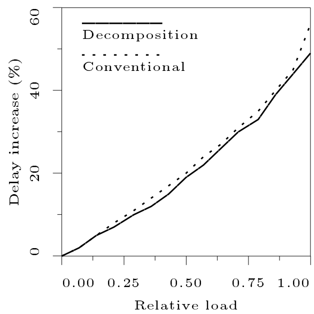

Figure 6: *Mean delay vs. load with MDR (solid line) and without (dotted line).*

5.2 Internal Performance

We now compare the distributions of flows with MDR and without. For a network of twenty nodes and a traffic matrix the elements of which are generated at random according to a normal distribution we obtain the results shown in Figre 5.

The figure shows, for each load value, the fraction of links which has a load less than this value. As could be expected from the suboptimality demonstrated above, MDR results in an increase in the number of highly loaded links.

5.3 Load Sensitivity

We also examine the influence of the traffic load on both algorithms. For the same network and traffic matrix as above, we find a maximum load scaling factor, *i.e.* the smallest scaling factor for which the algorithm no longer can find a solution, and study the mean delay in whole range of loads.

The result is shown in Figure 6 as relative mean delay *vs.* relative traffic load. The continuous line represents the result with MDR and the dotted the the result without. The two curves follow each other closely, and it is concluded that the difference between the two methods is independent of the load.

5.4 Traffic Sensitivity

Next we examine the consequences of small traffic changes. For the same network as before were 15 independent traffic matrixes generated at random as above. The mean was set high enough to prevent the flow deviation algorithm to find the optimal solution directly. It is reasonable to assume that the mean delay over these traffic matrices also will be normally distributed.

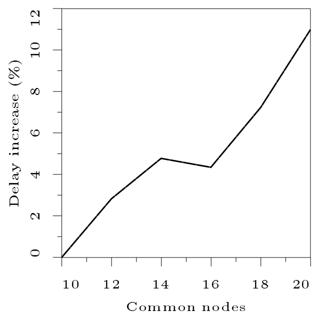

Figure 7: *Mean delay vs. number of common nodes.*

Computing a confidence interval with significance level 0.999 for the mean delay over the traffic matrices and normalising with respect to the mean, we obtain a (0.99,1.01). It is concluded that small changes in traffic give very small changes in mean delay. The algorithm is consequentially not sensitive to minor traffic changes.

5.5 Subnetwork Sensitivity

Finally we present some experimental results on how sensitive our method is to subnetwork partitioning. In particular, we use the same network of twenty nodes as before and study the performance for varying number of common nodes while the number subnetworks is fixed to five.

The result in figure 7 shows that the number of common nodes is not critical to optimality in terms of mean delay, which increases only slightly with the number of common nodes. This can be considered as an indication that MDR, for a given number of subnetworks which is determined from complexity reduction considerations, is relatively insensitive to the exact partitioning of the subnetworks.

6 Conclusions and Further Research

The current work has demonstrated that there is a clear potential for accelerating routing algorithms by applying MDR. Comparing to earlier decomposition methods, our method allows for repeated abstraction and therefore offers a higher complexity reduction potential.

Under steady state, the price paid for the complexity reduction is non-optimally of the result obtained. Our numerical examples indicate, however, that this cost is limited and relatively insensitive to problem characteristics such as subnetworks and traffics. Moreover, for real time applications with changing conditions, speed is as important as mathematical optimality for the applicability of the result.

A more detailed numerical study is currently being conducted. This includes more and larger networks plus examples of more than one abstraction layer.

References

[1] Antonio, J., Huang, G., and Tsai, W. (1991) A Fast Distributed Shortest Path Algorithm for a Class of Hierarchically Structured Data Networks, in *Proc. IEEE Globecom '91*, Phoenix, Arizona.

[2] Arvidsson, Å. (1994) Real Time Management of Virtual Paths, in *Proc. IEEE Globecom '94.*, San Francisco, California.

[3] Dimitrijević, D., Maglaris, B., and Boorstyn, R. (1994) Routing in Multidomain Networks. *IEEE/ACM Trans. on Networking*, **2**, 252–62.

[4] Gagnaire, M., Kofman, D. and Korezlioglu, H. (1995) An Analytical Description of the Packet Train Model for LAN Traffic Characterization, in *Perf. Modelling and Eval. of ATM Netw.*, **1**, Chapman & Hall.

[5] Garcia-Luna-Aceves, J. (1987) Regional Node Routing, in *Proc. IEEE Globecom '87*, Tokyo.

[6] Hagouel, J. (1983) Issues in Routing for Large and Dynamic Networks. *Ph.D. dissertation*, Columbia University, New York.

[7] Heijenk, G. and Niemegeers, I. (1995) Modelling the Reassembly Buffer in a Connectionless Server, in *Perf. Modelling and Eval. of ATM Netw.*, **1**, Chapman & Hall.

[8] Kar, G., Madden, B., and Gilbert, R. (1988) Heuristic Layout Algorithms for Network Management Presentation Services. *IEEE Network*, **2**, 29–36.

[9] Kamimura, K. (1991) An Efficient Method for Determining Economical Configuration of Elementary Packet Switched Networks. *IEEE Trans. on Commun.*, **39**, 278–88.

[10] Karlsson, T. (1995) An Evaluation of the Benefits Using Hierarchical Algorithms. *Master thesis*, Lund Institute of Technology, Lund (in Swedish).

[11] Kleinrock, L. (1970) *Queuing Systems, vol. 2: Computer Applications.* Wiley Interscience, New York.

[12] Kleinrock, L. and Kamoun, F. (1977) Hierarchical Routing for Large Networks: Performance Evaluation and Optimization. *Computer Networks*, **1**, 155–74.

[13] Kleinrock, L. and Kamoun, F. (1980) Optimal Clustering Structures for Hierarchical Topological Design of Large Computer Networks. *Computer Networks*, **10**, 221–48.

[14] Leland, W., Taqqu, M., Willinger, W., and Wilson, D. (1994) On the Self-Similar Nature of Ethernet Traffic (Extended Version). *IEEE/ACM Trans. on Networking*, **2**, 1–15.

[15] Paxson, V. and Floyd, S. (1994) Wide-Area Traffic: The Failure of Poisson Modeling, in *Proc. ACM Sigcomm 94*, London.

[16] Saksena, V. (1989) Topological Analysis of Packet Networks. *IEEE J. on Sel. Areas in Commun.*, **7**, 1243–52.

[17] Szybicki, E. and Bean, A. (1979) Advanced Routing in Local Telephone Networks; Performance of Proposed Call Routing Algorithms, in *Proc. 9th Internat. Teletraffic Cong.*, Torremolinos.

[18] Vickers, B. and Suda, T. (1994) Connectionless Service for Public ATM Networks. *IEEE Commun. Mag.*, **32**, 34–42.