10

Routing Complexity Reduction by Multilayer Decomposition

Åke Arvidsson
University of Karlskrona/Ronneby
Dept. of Telecommun. and Maths., Univ. of Karlskrona/Ronneby, S-371 79 Karlskrona, Sweden. Email: akear@itm.hk-r.se
Tel: +46 455 78053. Fax: +46 455 78057.

Abstract

Routing problems are frequently 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 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. Our work presents the method in formal terms and compares it to other, existing methods of network decomposition. To investigate the performance of our method with respect to complexity reduction and optimality of routing results, we outline a possible application to a real problem and study parts it numerically. It is found that considerable time savings can be made at a limited cost in terms of non-optimality of the final solution. We also discuss the fact 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 connection oriented approach, or the next node for a packet in the connectionless 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 telecommunication 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.* (Arvidsson 1994), and a few times per minute for VC networks, *e.g.* (Szybicki *et al.* 1979). 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 of a network.

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 computational 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 is a large and complex routing problem split into a number of smaller and simpler routing problems. This is achieved by decomposing the network into a number of suitably sized intersecting subnetworks, after which an abstract upper layer is created where the intersecting subnetworks are represented as a simplified network. Each node in the simplified network corresponds to a subnetwork in the original network; and each link in the simplified network corresponds to an intersection between two subnetworks in the original network. The abstraction process may then be repeated on the simplified network in a recursive manner as required to obtain acceptable network sizes. Besides reducing complexity and thereby computation times, MDR also simplifies parallel computation and distributed decision making. Another advantage is that rerouting due to locally confined changes or light over all fluctuations of traffic demands or transmission capacities may be handled within the concerned subnetworks or on a local level rather than by time consuming and costly global updates.

The remainder of this paper is organised as follows: First we give a brief description of the MDR-method in Section 2 after which we discuss how it is related to other, similar works in Section 3. The details of our method are then given in a mathematical language in Section 4. A possible practical application is outlined in Section 5 which is followed by some numerical results in Section 6. Finally, in Section 7, we give some conclusions and point at further research.

2 THE MULTILAYER APPROACH

2.1 Decomposition

A graph model of a complex telecommunication network contains a number of nodes N and links M, both of which may be very large. The computational complexity of a routing problem in such a graph generally increases with increasing N and M. Thus, to reduce the complexity of a routing problem 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, such that solutions to routing problems computed from a reduced model are relevant and applicable to the real 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 (which in itself is a complete graph with nodes and links) is represented by a logical node and the internal structure of the overlap between two subgraphs (which again in itself is a complete graph with nodes and links) is represented by a logical link.

Figure 1 illustrates the concept. To abstract a network, it is partitioned into a number of intersecting subnetworks. The set of intersecting 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 intersecting 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 $l+1$.

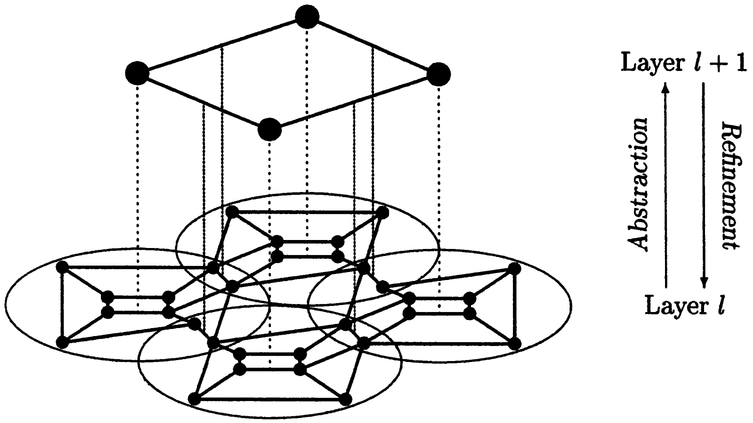

Figure 1 Example of multilayer representation by abstraction/refinement.

Routing algorithm complexity	$N \log N$	N^2	$N^2 \log N$	N^3
Without decomposition	48	1024	1541	32768
With decomposition	42	416	410	4064
Resulting gain (%)	12	60	75	88

Table 1 Computational efforts for various routing algorithms with and without multilayer decomposition for the example in Figure 1.

2.2 Routing

Routing decisions in a network represented as a multilayer decomposition are made sequentially for each layer starting from the highest level of abstraction. The routing problems on each layer are solved by an arbitrary routing optimisation algorithm and the results are used as preconditions to the routing problems on the next lower layer.

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 MDR simplifies the routing decision problem by decreasing the dimensionality of the search space. Table 1 gives an idea of the resulting reduction of computational effort for routing algorithms of different complexities. The reduction is achieved at the cost of information loss, which may result in deterioration of the quality of the obtained solution. The extent of this reduction depends on the specific choice

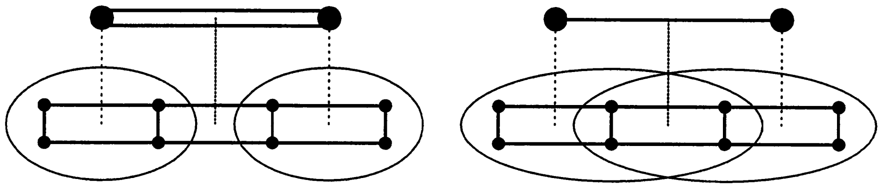

Figure 2 Traditional decomposition by non intersecting subnetworks (left) and our approach by intersecting subnetworks (right).

of subnetworks, the topology of the networks, the traffic interest between various node pairs, *etc.*

3 RELATED METHODS

3.1 Non Intersecting Subnets

The idea of graph decomposition in order to simplify routing problems is not new as such. The difference between our method and traditional approaches lies in the use of intersections. In earlier works are intersections not used, but networks are described as non intersecting subnetworks, each of which is abstracted to a node in the adjacent higher layer (Antonio *et al.* 1991, Dimitrijević *et al.* 1994, Garcia-Luna-Aceves 1987, Hagouel 1983, Kamimura 1991, Kar *et al.* 1988, Kleinrock *et al.* 1977, Kleinrock *et al.* 1980, Saksena *et al.* 1989). One of the important consequences of the non intersecting 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 decomposition method introduced here with intersecting subnetworks, provides a true link abstraction procedure. By including common switching and transmitting capabilities related to resource control *etc.*, we can achieve a meaningful definition of the logical link as an abstraction of interconnected subnetworks.

The removal of the 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 interconnecting 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 of routing results from upper layers to lower ones is not uniquely defined. A possible solution to this problem is shown in the following section.

4 ABSTRACTION AND REFINEMENT

4.1 Preliminaries

We consider a network described by a graph G, with N nodes and M links the transmission capacities of which given by a transmission capacity matrix C

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

where $c_{o,d}$ is the unidirectional transmission capacity (*e.g.* in bits per second) from a node o to another node d. If no direct physical link exists from o to d we set $c_{o,d} = 0$. Thus, since there are exactly M links, only M of the elements in C are greater than zero and the others equal to zero.

We also introduce a traffic demand matrix Γ

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

where $\gamma_{o,d}$ is the traffic demand (*e.g.* bits per second) from node o to node d. For variable and/or stochastic traffics demands, Γ may refer the *e.g.* expected values at a certain point in time. All elements in Γ are greater than or equal to zero.

A path between an origin o and a destination d is defined as an ordered set of nodes interconnected by direct physical links. Our routing problem consists of for every origin-destination pair (o, d) finding the optimal set of paths to be used, and the optimal proportions of the traffic $\gamma_{o,d}$ that should be sent over each of these. (Optimal may *e.g.* refer to maximum utilisation, minimum delay, or a combination of these.)

4.2 Multilayer Decomposition and Routing

Let the original network (G, C, Γ) be referred to as layer $l = 0$, set $(G, C, \Gamma) \equiv (G^{(0)}, C^{(0)}, \Gamma^{(0)})$. The firsts step in MDR is to form a new graph $G^{(0)\prime}$ by completely covering $G^{(0)}$ with overlapping subgraphs (or, equivalently, completely covering the network by intersecting subnetworks).

A simplified network at a higher level of abstraction, layer $l = 1$, is then formed by taking the subnetworks of $l = 0$ as logical nodes at $l = 1$, and the intersections of $l = 0$ as logical links on $l = 1$. The simplified network is described by $(G^{(1)}, C^{(1)}, \Gamma^{(1)})$ which all are determined from $(G^{(0)\prime}, C^{(0)}, \Gamma^{(0)})$.

A second, further simplified, layer $l = 2$ of higher abstraction may then be formed by repeating the same procedure on layer $l = 1$, and so on up to any layer $l = L$. Each new layer results in a higher degree of abstraction and thus in a network with less nodes and links than on the previous layer. The fact that the size of the network is reduced with each layer means that the computational effort of the associated routing problem is successively decreased through the abstraction process.

When sufficient simplification has been achieved at some level of abstraction, say at layer $l = L$, is the routing problem for this layer solved (by any suitable algorithm) according to the conditions given by $(C^{(L)}, \Gamma^{(L)})$. Let the solution be denoted by $B^{(L)}$. To get a useful result, $B^{(L)}$ must be now be successively refined into $B^{(l)}$ for $l = L - 1, \ldots, 0$.

The solution $B^{(L-1)}$ to the second highest layer $l = L - 1$ can be decomposed into the solutions $B_n^{(L-1)}$ for all subnetworks $n = 1, \ldots, |\mathcal{O}^{(L-1)}|$ in $G^{(L-1)\prime}$. The solution $B_n^{(L-1)}$ for subnetwork n is determined from the the conditions within n, which we refer to as internal conditions and are given by $(C^{(L-1)}, \Gamma^{(L-1)})$, and the conditions with respect to the subnetworks intersecting n, which we refer to as external conditions and are determined also by $(B^{(L)}, G^{(L-1)\prime})$.

Next, the solution $B^{(L-2)}$ on the next lower layer may be determined in a similar way, *i.e.* by finding $B_n^{(L-2)}$ for all subnetworks $n = 1, \ldots, |\mathcal{O}^{(L-2)}|$ in $G^{(L-2)\prime}$ from $(C^{(L-2)}, \Gamma^{(L-2)})$ and $(B^{(L-1)}, G^{(L-2)\prime})$. The same process of solving for $B_n^{(l)}$ for all subnetworks $n = 1, \ldots, |\mathcal{O}^{(l)}|$ from $(C^{(l)}, \Gamma^{(l)})$ and $(B^{(l+1)}, G^{(l),\prime})$ is then repeated until the final solutions $B_n^{(0)}$ for all subnetworks $n = 1, \ldots, |\mathcal{O}^{(0)}|$ of the real network have been determined.

Summing up, the basic steps may described in algorithmic form as follows

```
Set l=0;
repeat
   Define intersecting subnetworks at layer l;
   Define abstract links in layer l+1;
   Define abstract traffics in layer l+1;
   Set l=l+1;
until simplified;
```

```
Solve the routing problem on layer l;
repeat
   Set l=l-1;
   Define external links to subnetworks at layer l;
   Define external traffics to subnetworks at layer l;
   Solve the routing problems for each subnetwork on layer l;
until l equals 0;
```

(a) Subnetwork Definition

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

With respect to complexity reduction, maximum gain is achieved if all routing problems have the same search space, *i.e.* if all subnetworks have the same number of nodes, and this is the approach we have used in our numerical studies below.

The impact on suboptimality of the final result is more complicated and not considered here. Some remarks on the issue are found in the papers on non intersecting subnetworks quoted above and some general references for clustering are (Everitt 1993, Kaufman *et al.* 1990). We also add the heuristic observation is that suboptimality may be reduced by keeping nodes with a large mutual traffic interest within the same subnetwork, but leave the details for further study.

To formally describe the partitioning into subnetworks of a layer l, we let $\mathcal{M}_n^{(l)}$ denote the set of subnetworks on layer l (the set of nodes on layer $l+1$) of which node n on layer l is a member, $\mathcal{N}_{n'}^{(l)}$ the set of all nodes on layer l which belong to subnetwork n' on layer l (node n' on layer $l+1$), $\mathcal{N}_{n'-n''}^{(l)}$ the set of nodes on layer l which belong to subnetwork n' but not to subnetwork n'' on layer l (to node n' but not to node n'' on layer $l+1$), and $\mathcal{O}^{(l)}$ the set of all subnetworks on layer l (nodes on layer $l+1$).

Nodes in intersections are said to be common nodes with respect to the intersecting subnetworks to which it belongs. Thus, a node n on layer l is common to subnetworks n' and n'' on layer l if and only if $n \in \mathcal{N}_{n'}^{(l)} \cap \mathcal{N}_{n''}^{(l)}$. The opposite is referred to as proprietary and a node n on layer l is proprietary to subnetwork n' with respect to subnetwork n'' on layer l if and only if $n \in \mathcal{N}_{n'-n''}^{(l)}$.

To make the concepts of subnetworks and intersections meaningful, it is assumed that all subnetworks have at least one proprietary node and one common node with respect to every intersecting subnetwork. The former means that for two intersecting subnetworks n' and n'' we require that $\mathcal{N}_{n'-n''}^{(l)} \neq \emptyset \wedge \mathcal{N}_{n''-n'}^{(l)} \neq \emptyset$ and the latter that $\mathcal{N}_{n'}^{(l)} \cap \mathcal{N}_{n''}^{(l)} \neq \emptyset$. Similarly, to make abstraction meaningful, it is assumed that subnetwork interaction is completely described by intersections. Formally stated, this means that for a link on layer l between two nodes o and d we require that $\mathcal{M}_o^{(l)} \subseteq \mathcal{M}_d^{(l)} \vee \mathcal{M}_o^{(l)} \supseteq \mathcal{M}_d^{(l)}$.

(b) Abstracting Links

We define the transmission capacity $c_{o',d'}^{l+1}$ of a link on layer $l+1$ from a node o' to another node d' as the minimum of (i) the total transmission capacity from proprietary nodes of subnetwork o' with respect to d' to nodes of subnetwork d' and (ii) the total transmission capacity from nodes of subnetwork o' to proprietary nodes of subnetwork d' with respect to o',

$$c_{o',d'}^{(l+1)} = \min\left(\sum_{\substack{\forall o \in \mathcal{N}_{o'-d'}^{(l)} \\ \forall d \in \mathcal{N}_{d'}^{(l)}}} \frac{c_{o,d}^{(l)}}{|\mathcal{M}_d^{(l)}| - |\mathcal{M}_o^{(l)}|}, \sum_{\substack{\forall o \in \mathcal{N}_{o'}^{(l)} \\ \forall d \in \mathcal{N}_{d'-o}^{(l)}}} \frac{c_{o,d}^{(l)}}{|\mathcal{M}_o^{(l)}| - |\mathcal{M}_d^{(l)}|} \right) \tag{3}$$

where $|\cdot|$ refers to the cardinal number of its argument. The denominator refers to the number of subnetworks the link in question interconnects, and means that the transmission capacity of a link on layer l which will appear in more than one link on layer $l+1$ is split equal among those links.

By applying equation (3) repeatedly starting from $l = 0$, we may now abstract the transmission capacity matrix up to any layer L.

(c) Abstracting Traffics

We define the traffic demand $\gamma_{o',d'}^{(l+1)}$ on layer $l+1$ from a node o' to another node d' as the sum of all traffics between the members of the two nodes

$$\gamma_{o',d'}^{(l+1)} = \sum_{\substack{\forall o \in \mathcal{N}_{o'}^{(l)} \\ \forall d \in \mathcal{N}_{d'-o'}^{(l)}}} \frac{\gamma_{o,d}^{(l)}}{|\mathcal{M}_o^{(l)}||\mathcal{M}_d^{(l)}|} \tag{4}$$

where the denominator refers to the number subnetwork pairs that the traffic affects, and means that the traffic demand of a traffic on layer l which will appear in more than one traffic on layer $l+1$ is split equal among those traffics.

By applying equation (4) repeatedly starting from $l = 0$, we may now abstract the traffic matrix up to any layer L.

(d) Solving Routing Problems

The conditions of the routing problem on layer L are determined by the transmission capacity matrix $C^{(L)}$ and the traffic demand matrix $\Gamma^{(L)}$, which are computed from (3) and (4) respectively. Given these, the solution $B^{(L)}$ can be computed by any applicable algorithm.

For the subsequent layers $l = L-1, \ldots, 0$, each layer l poses $|\mathcal{O}^{(l)}|$ similar routing problems, one for each subnetwork. The conditions regarding the flows within the subnetworks (the internal flows) are determined by the corresponding transmission capacity matrices $C^{(l)}$ and traffic demand matrices $\Gamma^{(l)}$, which are obtained from (3) and (4). The conditions regarding the flow in to and out from the subnetworks (the external flows) are determined by the flows on the upper level, which are given by $B^{(l+1)}$.

Consider a specific subnetwork n of any layer l below L and let the subnetworks which intersects n be represented by two sets of virtual nodes. The first set of virtual nodes is used to terminate links and traffics which are directed out of n, and the second one to initiate links and traffics which are directed into n. Virtual nodes are indicated by a circumflex, *e.g.* $\hat{n}$. Links (traffics) which are initiated or terminated by virtual nodes are referred to as virtual links (traffics) respectively. The formulae for determining the transmission capacities and traffic demands of these are yet to be defined.

In what follows, let $B^{(l)}$, the result of a routing algorithm on layer l, be expressed in terms of fractions of traffic demands routed over each link such that $\beta^{(l)}_{o,d}(\omega, \delta)$ is the fraction of the traffic from a node o to another node d on layer l (corresponding to a traffic demand $\gamma^{(l)}_{o,d}$) which is routed over the link from a node ω to another node δ on layer l (corresponding to a transmission capacity $c^{(l)}_{\omega,\delta}$).

(e) Refining Links

We consider a subnetwork n on level l and define the transmission capacity of a virtual link $c^{(l)}_{o,d}$ on layer l from a node o to another node d (one of which is a part of n, and one of which is a virtual node to n) as the sum of all transmission capacities between the two nodes, *i.e.*

$$c^{(l)}_{o,\hat{d}} = \sum_{\forall d \in \mathcal{N}^{(l)}_{\hat{d}-n}} \frac{c^{(l)}_{o,d}}{|\mathcal{M}^{(l)}_{o}| - |\mathcal{M}^{(l)}_{d}|} \tag{5}$$

for outbound virtual links from nodes in n to terminating virtual nodes $\hat{d}$, and

$$c^{(l)}_{\hat{o},d} = \sum_{\forall o \in \mathcal{N}^{(l)}_{\hat{o}-n}} \frac{c^{(l)}_{o,d}}{|\mathcal{M}^{(l)}_{d}| - |\mathcal{M}^{(l)}_{o}|} \tag{6}$$

for inbound virtual links to nodes in n from initiating virtual nodes $\hat{o}$. The denominators are the same as for link abstraction, Equation (3).

By applying equations (3) and (5)–(6) repeatedly starting from $l = L-1$, we may now form the refined transmission capacity matrices for all subnetworks on all layers down to 0.

(f) Refining Traffics

We consider a subnetwork n on level l and define a virtual traffic demand $\gamma_{o,d}^{(l)}$ on layer l from a node o to another node d (one of which is a part n or a virtual node to n, and one of which is a virtual node to n) as the sum of all traffic demands between the two nodes, *i.e.*

$$\gamma_{o,\hat{d}}^{(l)} = \sum_{\forall d' \neq n} \sum_{\forall d \in \mathcal{N}_{d'-n}^{(l)}} \frac{\beta_{n,d'}^{l+1}(n,\hat{d})\gamma_{o,d}^{(l)}}{|\mathcal{M}_o^{(l)}||\mathcal{M}_d^{(l)}|} \tag{7}$$

for outbound virtual traffics from nodes in n to terminating virtual nodes $\hat{d}$, where the first sum refers to all terminating subnetworks d' but n and the second sum to all terminating nodes d in d',

$$\gamma_{\hat{o},d}^{(l)} = \sum_{\forall o' \neq n} \sum_{\forall o \in \mathcal{N}_{o'-n}^{(l)}} \frac{\beta_{o',n}^{l+1}(\hat{o},n))\gamma_{o,d}^{(l)}}{|\mathcal{M}_o^{(l)}||\mathcal{M}_d^{(l)}|} \tag{8}$$

for inbound virtual traffics to nodes in n from initiating virtual nodes $\hat{o}$, where the first sum refers to all initiating subnetworks o' but n and the second sum to all initiating nodes o in o', and

$$\gamma_{\hat{o},\hat{d}}^{(l)} = \sum_{\substack{\forall o' \neq n \\ \forall d' \neq n}} \sum_{\substack{\forall o \in \mathcal{N}_{o'-n}^{(l)} \\ \forall d \in \mathcal{N}_{d'-n}^{(l)}}} \frac{\beta_{o',d'}^{l+1}(\hat{o},n)\beta_{o',d'}^{l+1}(n,\hat{d})\gamma_{o,d}^{(l)}}{|\mathcal{M}_o^{(l)}||\mathcal{M}_d^{(l)}|} \tag{9}$$

for virtual traffics which pass through n from initiating virtual nodes $\hat{o}$ of n to terminating virtual nodes $\hat{d}$ of n, where the first sum refers to all initiating subnetworks o' but n and all terminating subnetworks d' but n, and the second sum to all initiating nodes o in o' and all terminating nodes d in d'.

The denominators are the same as for traffic abstraction, Equation (4). By applying equations (4) and (7)–(9) repeatedly starting from $l = L-1$, we may now form the refined traffic matrices for all subnetworks on all layers down to 0.

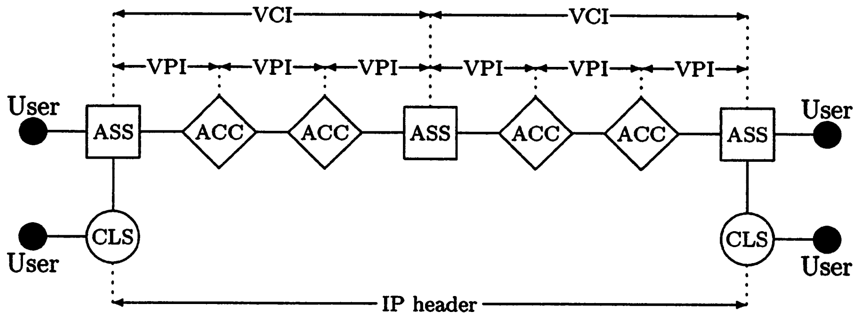

Figure 3 Addressing on a link between two CLSs for connectionless traffic in ATM networks.

5 APPLICATION TO DATA TRAFFIC IN ATM

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

We consider a physical network of links and nodes. The links are typically optical transmission systems and the nodes include ATM cross connects (ACCs), ATM switching systems (ASSs), and connectionless servers (CLSs) (Heijenk *et al.* 1995, Vickers *et al.* 1994), Figure 3. The CLSs are attached to ASSs in a way similar to ordinary subscribers, and the ASSs are connected to one or more ACCs, which in turn are directly interconnected.

The physical network is enhanced in two steps by two logical overlay networks, *viz.* a network of virtual paths (VPs) built upon the physical network which interconnect ASSs through a number of intermediate ACCs, and a network of virtual channels (VCs) built upon the virtual path network which interconnect CLSs through a number of intermediate ASSs.

Users of packet switched services such as connectionless TCP/IP deliver packets to their local CLS either directly using *e.g.* a Local Area Network (LAN) interface, or via their local ASS using the normal ATM interface. Either of the two interfaces may be switched or permanent. The CLSs in turn transport the packets 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 routing.

The routing functions in the CLSs can control the packet flows over VCs to other CLSs as required using the features offered by the protocols of TCP/IP suite. Other parts of the CLSs can arrange their logically direct VC links to other CLSs as required by establishing, modifying, and removing VCs by interacting with their local ASSs. Such actions would typically be carried out by signaling between the two entities using a control plane protocol. The ASSs in turn can arrange their logically direct VP links to other ASSs as required by establishing, modifying, and removing VPs by interacting with their local

ACCs. Such actions would typically be carried out by interaction between the two entities over a management plane protocol.

It is well known that most kinds of data traffic exhibits burstiness and intensity variations in many time scales, *e.g.* (Gagnaire *et al.* 1995, Leland *et al.* 1994, Paxson *et al.* 1994) and it is clear that no single means, *e.g.* buffers, can handle these variations. On the contrary, 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.

Having three different means of the handling traffic flows, *viz.* packets, VCs, and VPs, it is suggested (Arvidsson *et al.* 1996) to treat traffic variations as occurring in three distinct time scales and to devote one means of adjustment to each of the time scales. We thus foresee the use of VPs to handle variations in the long term time scale, VCs for medium term time scale variations, and packets for short term time scale variations.

Such a control system would consist of thee logical levels which carry out traffic demand measurements and collect transmission capacity information. The first level collects on-line measurements of slow traffic variations, *e.g.* by passing load measurements through a low pass filter to form a series of traffic demand matrices $\Gamma^{(0)}_{\mathrm{LP}}(t)$ for various instants t in time. A corresponding series of transmission capacity matrices $C^{(0)}_{\mathrm{PHY}}(t)$ describes the status of the physical network at these instants. The two matrices define a routing problem of VPs where MDR may be applied to speed up the computation of the VP network. The second level deploys a band pass filter for producing another series of traffic demand matrices $\Gamma^{(0)}_{\mathrm{BP}}(t)$ on the medium time scale and takes the current status of the VP network as the transmission capacity matrices $C^{(0)}_{\mathrm{VPN}}(t)$. The VC routing problem determined by the two matrices can use MDR to quickly engineer the VC network. Similarly, MDR can be applied to speed up the routing of packets on the third level from traffic demands matrices $\Gamma^{(0)}_{\mathrm{HP}}(t)$ obtained by high pass filtering load measurements, and transmission capacities $C^{(0)}_{\mathrm{VCN}}(t)$ taken from the VC network.

We are thus 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.

Clearly, the faster the time scale of the routing problem, the more important is the speed of the routing algorithm employed and hence the more attractive is MDR.

6 NUMERICAL RESULTS

6.1 Background

Although the concepts of MDR are applicable to any network and any routing algorithm, *e.g.* VP networks and VP routing, VC networks and VC routing, or packet networks and packet routing, we have to be specific when it comes to numerical examples. We have chosen to study the packet level and to deploy the method of flow deviation (Kleinrock 1970) for this purpose. Flow deviation determines the routing of given traffic demands in a network of given transmission capacities under the presumption that the network can be modelled as an open queuing network of M/M/1 queuing systems. The result obtained is optimal with respect to mean delay for an arbitrary packet.

The reasons for chosing this method is that it is well known, simple to implement, and allows us display the *pros* and *cons* of MDR in a clear way. The purpose of the numerical examples is only to compare computation times and results with and without MDR and not to present significant results for data traffic routing in ATM networks. It is emphasised that our specific choices therefore neither intend to indicate any restrictions to the usage of MDR, nor to suggest that this specific method is particularly suitable for routing data traffic in ATM networks.

The current work represents a significant generalisation and provides a tidier and more stringent formalisation of an earlier work (Arvidsson 1995). New, numerical examples over a wider range of parameters and that completely cover the current extensions were still under production at submission deadline. Rather than excluding numerical results altogether, we have chosen to include the numerical examples from the earlier work. These are not as extensive as would be desirable and obey more stringent restrictions than assumed here. The most important restriction in the earlier work is that no n node on any level l may be a member of more than two subnetworks, $|\mathcal{M}_n^{(l)}| \leq 2 \; \forall n, l$.

6.2 External Performance

We first examine how MDR behaves 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 at approximately 50% of its nominal capacity. For the same reason we distribute the traffics unevenly in the network. That means that every node

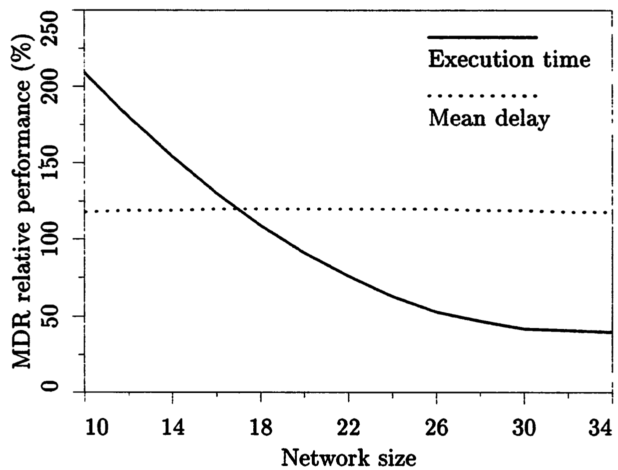

Figure 4 Relative execution time (solid line) and mean delay (dotted line) for MDR *vs.* network size.

directs its entire traffic demand to one specific node and receives traffic from only one node, *i.e.* all elements but one are zero in every row of Γ.

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.

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 (for a network with 30 nodes is our method about 2.5 times faster) at a restricted cost in 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 have changed in the mean time.

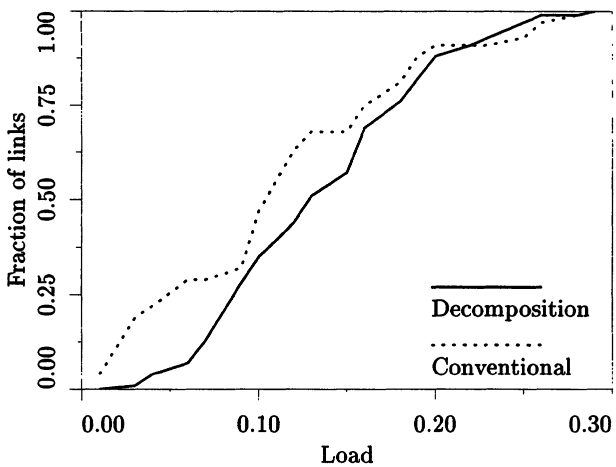

Figure 5 The distribution of traffic loads with MDR (solid line) and without (dotted line).

6.3 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 Figure 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.

6.4 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 the 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 one 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.

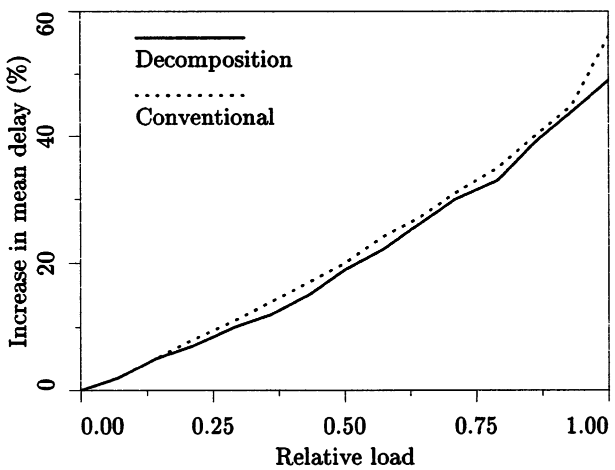

Figure 6 Mean delay *vs.* load with MDR (solid line) and without (dotted line).

6.5 Traffic Sensitivity

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

Computing a confidence interval with significance level 0.999 for the mean delay over the traffic demand matrices and normalising with respect to the centre point, we obtain (0.99, 1.01). It is concluded that small changes in traffic demands give very small changes in mean delay hence MDR does not appear to be sensitive to minor traffic demand changes.

6.6 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 more nodes that are common, the less optimal is the final solution. The reason for this is that the more nodes that appear in more than one subnetwork routing problem, *i.e.* the more nodes

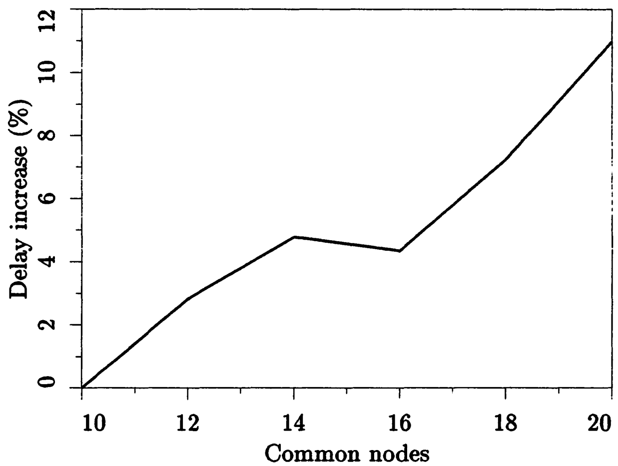

Figure 7 Mean delay *vs.* number of common nodes.

which are classified as common, the less close is the studied problem to the real problem.

Although the values obtained indicate that the increase in delay with the number of common nodes is relatively small, it is concluded that intersections should be kept as small as possible. Taking the relatively modest increase into account, we may also take the results as an indication of that MDR for a given number of subnetworks, which is determined from complexity reduction considerations, is relatively insensitive to the exact partitioning of a network into subnetworks.

7 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.

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 selection of subnetworks and traffic demands. Moreover, for real time applications with changing conditions, speed is as important as mathematical optimality for the applicability of the result when it comes to optimising actual network performance.

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

ACKNOWLEDGEMENT

The current author wishes to acknowledge the efforts of his former co-authors at the Department of Communication Systems at Lund Institute of Technology, Mr. Wlodek Holender, M.Sc., (†1996) who initially proposed the concept of overlapping subgraphs, and Mr. Torgny Karlsson, M.Sc., (now with Ericsson Radio Systems) who implemented the first, draft interpretation of the concept and prepared the numerical examples.

REFERENCES

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.

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

Arvidsson, Å. (1995) Traffic Management in ATM Networks — A Proposal for Imperfect Traffic Information, in *Proc. Second Polish Teletraffic Seminar*, Gdańsk.

Arvidsson, Å., Holender, T., and Karlsson, T. (1996) Reduction of Routing Complexity by a Novel Multilayer Decomposition Method, in *Proc. Fourth IFIP Int. Workshop on Perf. Mod. and Eval. of ATM Netw.*, Ilkley.

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

Everitt, B. (1993) *Cluster Analysis.* Edward Arnold, London.

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, London.

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

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

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, London.

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

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

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

Kaufman, L. and Rousseeuw, P. (1990) *Finding Groups in Data: An Intro-*

duction to Cluster Analysis. Wiley, New York.

Kleinrock, L. (1970) *Queuing Systems, vol. 2: Computer Applications*. Wiley Inter-science, New York.

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

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

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.

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

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

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.

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

Åke Arvidsson received his Ph.D. from the Lund Institute of Technology in Lund, Sweden, in 1990. He is currently acting professor at the Department of Telecommunications and Mathematics, University of Karlskrona/Ronneby, Sweden. Current research interests include bandwidth management and traffic routing in ATM networks, traffic modelling for buffer engineering and call acceptance control in ATM networks, and congestion control mechanisms for intelligent networks. His URL is http://www.itm.hk-r.se/~akear.