ROUTING IN MULTI-DOMAIN NETWORKS

Dragomir D. Dimitrijevic¹, Basil Maglaris², Robert R. Boorstyn²

1. Contel Technology Center
15000 Conference Center Drive
Chantilly, VA 22021-3808

2. Polytechnic University
333 Jay Street
Brooklyn, NY 11201

ABSTRACT¹

We investigate the problem of management and control in a large and, for simplicity, homogeneous packet-switched network. In particular, routing is considered as an important function of network management. The network consists of several individually controlled domains. Domains are interconnected via gateway links. Each domain is controlled by its own Network Control Center, while the overall network performance is managed by the Integrated Network Control Center. Each center has only a portion of the information required for global routing. We investigate the impact of the reduced information available at each center on network performance (average delay in our case). A general approach to designing a hierarchical algorithm for routing in multi-domain networks is presented. A heuristic procedure suitable for packet-switched networks will be proposed. An example will be shown as an illustration of the proposed algorithm. Its performance is compared against a lower bound to the network performance which is governed by the constraint that nodes, due to the lack of information, do not differentiate destinations in other domains. Therefore, each domain is perceived as a single node in a simplified model of the network.

1. INTRODUCTION

1.1 Multi-Domain Networks

In this paper we investigate the problem of management and control in a large and, for simplicity, homogeneous packet-switched network. In particular, routing is considered as an important issue in network management. We shall use the average end-to-end delay over the entire network as a measure of the network performance. This delay is assumed to be a separable function of the link delays. We model all links as independent M/M/1 queues, as commonly assumed in packet-switched networks [Fratta73]. The primary concern of early researchers was to design an efficient, static or dynamic, centralized or distributed, routing algorithm. However, management of today’s computer communication networks involves different kinds of problems shown in the following discussion.

Network heterogeneity: Computer communication vendors and users are faced with growing communication systems that must handle voice, data, and video on an integrated basis, eventually through an Integrated Services Digital Network (ISDN). A modern computer communication network is actually a complex interconnection of many constituent, loosely coupled subnetworks or domains. Generally, there can be different types of subnetworks such as packet-switched, System Network Architecture (SNA), T1 or voice networks. Those subnetworks usually are provided by different vendors. Each of the subnetworks must interface with other networks such as long-haul services from long distance carriers.

The management and control of these networks of interconnected heterogeneous domains is an increasingly important area. In such a heterogeneous environment, it is difficult to provide all the information needed without placing a burden on communication bandwidth and introducing communication, processing, and storage overhead. On the other hand, vendors and users must somehow monitor and control the operation of this entire system to guarantee an acceptable grade of service (the average delay in our case).

For that reason, each domain is controlled by its own Network Control Center (NCC). The management and control of such a system involves cooperation among different NCCs. The distributed approach involves peer-to-peer coordination of the various NCCs involved, while the hierarchical approach introduces an Integrated Network Control Center (INCC), called a “Manager of Managers” by NYNEX, that supervises NCCs and their interaction. AT&T’s Unified Network Management Architecture (UNMA) is related to the evolving International Organization for Standardization - Open Systems Interconnection (ISO-OSI) protocols for passing management information between a network element and an NCC and extends to the NCC-INCC interface.

Size of the networks: Network routing algorithms generally attempt to provide communication between two nodes by sending data messages along the best or shortest paths between them. Unfortunately, in large communication networks it is difficult to maintain knowledge of such paths due to the cost in storage, computation, and communication bandwidth.

A multi-domain network, Figure 1, consists of several individually controlled domains. Domains are interconnected via (possibly multiple) gateway links (heavy lines) and controlled by their own NCCs. We assume coordination among different NCCs. For that reason, we allow a limited amount of information exchange between different NCCs. Due to the network size, we wish to reduce the amount of information exchange while still preserving reasonable network performance. Limitation on the available information raises the problem of myopic decision

¹ This work is supported in part by NYNEX Corporation, applied Research Division, in part by the New York State Science and Technology Foundation under its program for Centers for Advanced Technology, and in part by the Foldes Fellowship Foundation at Polytechnic University.

3D.1.1.
CH2979-3/91/0000-0257 $1.00 © 1991 IEEE 0257
The consequences of the reduced information exchange are:

1. Reduced communication overhead that leads to better network performance
2. Reduced processing and memory requirements in each NCC
3. Suboptimal network performance

1.2. Previous Work

Various network routing algorithms have been suggested in the literature as a means of reducing computational and storage complexity and communication bandwidth at the expense of network performance. Kamoun proposed in his doctoral thesis [Kamoun76, Kleinrock77] an approach of clustering and a suboptimal shortest path (non-bifurcated) routing algorithm. Even a large homogeneous network may be divided into several loosely coupled domains. The analysis is generalized to k levels of clustering [Hagouel83a,b]. Another algorithm, suitable for virtual circuits, was proposed by Baratz and Jaffe [Baratz84]. It gives optimal path lengths by introducing a limited broadcast of the call-request packets during the establishment of a session.

None of the aforementioned papers have analyzed overall network performance when the information about the topology and the traffic requirements is incomplete. A lower bound to the average delay in a network consisting of two individually controlled domains is analyzed in [Dimitrijevic89]. It was shown that despite the additional constraints, the entire requirement matrix is needed to achieve the lower bound of network performance while keeping link flows consistent. An alternative, heuristic approach to this problem was proposed in [Chao89]. It required additional measurements in order to preserve consistency of the link flows. By “inconsistency” we mean discrepancy between expected and actual link flows enforced by actual network operation. Inconsistency is caused by the lack of information about the network. Unlike [Dimitrijevic89] and [Chao89], in the procedure we shall develop, NCCs will configure their routes based on inconsistent estimates for the internet flows. A more detailed study of the problem considered in this paper can be found in [Dimitrijevic90].

1.3. Paper Outline

Section 1 of this paper contains the introduction. Section 2 contains a description of operations performed by the INCC. A procedure for routing in multi-domain networks will be outlined. It involves the High Level Routing (HLR) performed by the INCC as well as the Low-Level Routing (LLR) performed by the NCCs in each domain individually. Section 3 concentrates on the LLR and local performance optimization subject to the constraints imposed by the INCC and HLR and with incomplete information available. A way to simplify the model of a domain through Equivalent Capacities associated with each gateway link will be given. The Equivalent Capacities are a way to convey information about local congestions in a domain without giving too many internal details, and thus introducing an excessive overhead. It will be shown how a domain can improve its local performance after acquiring more information in the learning period, which we call A Posteriori Routing. Section 4 contains an example as an illustration of the proposed algorithm. Section 5 contains the conclusion.

2. OPERATIONS PERFORMED BY THE INCC

2.1. Properties of Routing in Multi-Domain Networks

In the following section we shall describe a hierarchical routing algorithm and determine the amount of information available from a practical point of view to each of the network parts. In order to design an applicable routing algorithm, we impose some a priori constraints on available information, as well as some properties of the algorithm.

Node independent internet information: Any information on traffic across domains is independent of specific nodes. This means that the amount of information about the internet is independent of the size of individual domains. Each domain does not distinguish nodes in other domains, i.e., all messages destined to the same domain are considered as a single commodity. This property implies destination independent routing and more compact routing tables in each node.

Loop-free routing: The resulting routing must be loop-free. To prevent loops, we impose an additional constraint, since the detailed internal structure of a domain is not globally known. We allow that each domain on a route can be visited only once. This means that the routing viewed on a domain-to-domain basis is also loop-free, although a loop on a domain-to-domain basis does not necessarily mean a loop in the actual path. Since each domain knows all the details about its internal structure, a loop-free route within the domain can be provided.

Destination dependent statistics: Each entity in the network keeps statistics necessary for the routing algorithm and reports to a higher level. For example: nodes keep statistics about generated traffic from each internal node to an external domain and reports it to the NCC. NCCs report the condensed information about the internet requirements to the INCC.

Limited internet information exchange: We allow only a small amount of internet information exchange. To achieve this goal, we allow a small number of iterations that involve changes in the overall routing. Excessive amount of global information exchange may not be manageable from a practical point of view.

Parallel work of NCCs: All NCCs should work independently of each other as much as possible to achieve maximum parallelism.

Processing and memory efficiency: The algorithm must be provably more efficient than the centralized routing scheme. This involves computational efficiency and the size of routing tables in all nodes.

Memoryless routing: We assume that all messages are handled based on their destination, regardless of the previous nodes that are visited. Many systems apply this kind of routing. Other systems (e.g., bridged local area networks) apply source routing implementations [Hammer88] where the source of a packet specifies the entire route from the source to the destination.
2.2. A Procedure for Routing in Multi-Domain Networks

The following section gives a brief description of a hierarchical algorithm for routing in multi-domain networks. We divide the problem of designing the hierarchical routing algorithm into different parts to be studied separately.

1. **Initialization**: The INCC initializes the internet (domain-to-domain) routing.

2. **High-level routing**: Based on a simplified configuration of the network, the INCC determines routes for all internet requirements. After the high-level routing is determined, each of the NCCs is informed about the INCC’s decision concerning the routing in the corresponding domain.

3. **Low-level routing**: Based upon the information about the states of its gateway links received from the INCC, each NCC decides about its local routing. This part of the algorithm will be described in Section 3.

4. **Computation of gateway penalties**: Each of the NCCs computes penalties for each of its gateway links in order to avoid congested areas in the network. Note that no matter how the penalties are obtained, each of the gateway links is characterized by a single number which is sent to the INCC. A heuristic approach to the computation of the penalties will given in Section 3.

5. **Stopping criterion**: Steps 2-4 are repeated until a stopping criterion is satisfied. In our case, the INCC terminates the procedure when the relative difference between two consecutive values of its objective function becomes smaller than 1%. Since the algorithm is a heuristic one and there is no proof that it converges, an upper limit to the number of iterations must be imposed.

6. **A posteriori routing**: The initial routing is readjusted after a learning period when more information about the network is acquired. This part of the algorithm will be discussed in Section 3.

### 2.3. Computational and Memory Complexity

Let \( n \) be the number of nodes in a multi-domain network and \( \alpha \) be a constant between 0 and 1. We shall denote the number of domains in the network as \( O(n^\alpha) \) and the average number of nodes per domain as \( s = O(n^{1-\alpha}) \). Generally, the order of complexity of a routing algorithm (shortest path, Flow Deviation, Extremal Flow [Bertsekas87]) is \( O(n^2) \), and the size of the routing tables is \( O(n) \). The order of complexity of the high-level routing is \( O(s^2) \). The order of complexity of the low-level routing performed in each domain is \( O(s^3) \). Since all NCCs perform their low level routing algorithm simultaneously the overall computational complexity is equal to the sum of the two parts. The minimum complexity, \( O(\sqrt{n^3}) \), is achieved when \( \alpha = 0.5 \). This is significantly better than \( O(n^3) \) for the global optimum. Similarly, the minimum memory complexity, \( O(\sqrt{n}) \), is achieved for \( \alpha = 0.5 \).

### 2.4. High-Level Routing

We assume that the INCC has some topological information about the domains. The information is restricted to existence of the domains and their connectivity via gateway links. This means that the topology of the internet is known. In the graph that describes the internet topology, each node corresponds to a domain in the network, while each directed arc corresponds to a gateway link.

To be more specific, we allow the following information about the network:

1. Each gateway link \( g \) is characterized by its *Equivalent Capacity* (EC) \( C'_g \), obtained from the domains that correspond to gateway link \( g \). The equivalent capacity is a measure of congestion at both sides of the corresponding gateway link. In our case, we reduce the original gateway capacity. The reduction is a function of congestion in the domain. Gateway capacities leading to more congested areas are more reduced, thus forcing the traffic to avoid those areas.

2. We also assume availability of the average data rates for the domain-to-domain requirements. In a packet-switched network, it will simply be \( r_{ij} \), defined as the total amount of traffic that has to be sent from domain \( I \) to domain \( J \). Each \( r_{ij} \) is obtained from the NCC in the source domain \( I \).

The INCC determines routes for all domain-to-domain requirements. To simplify the problem, we wish to make use of some of the well-known routing algorithms using the \( C'_g \) and \( r_{ij} \) as input data. In this work, we apply the Flow Deviation Algorithm [Pratta73] using the equivalent capacities of the gateway links instead of their real capacities. We used the actual gateway capacities as an initial value for the equivalent capacities.

The INCC’s objective is to minimize “the average delay” using the equivalent capacities instead of real ones. The result of this algorithm is a set of \( f_{gj} \) defined as the flow of messages in gateway link that are destined for domain \( J \). This number is reported to the domains adjacent to gateway link \( g \) and is used as a constraint in the low-level routing optimization. Note that the produced routing using this objective function is loop-free i.e., none of the domains is visited more than once by the route for any source-destination pair.

### 3. OPERATIONS PERFORMED BY NCCs

#### 3.1. Formulation of the Problem

In this section we formulate the problem of low level routing. The low level routing algorithm optimizes a local objective function subject to the constraints imposed by the INCC. Based on the information received from the INCC, each NCC iteratively determines routes to the best of its knowledge. The routes are determined for each of the following four types of traffic:

**Internal traffic**: This type of traffic is generated at and destined for a node within a domain, say domain \( I \). The information about the internal traffic is local and known to NCC \( I \).

**Outgoing traffic**: This type of traffic is generated in the domain and destined for a node outside of the domain. The flows in each gateway link of the domain destined for other domains are determined by the INCC.

**Transit traffic**: This type of traffic is generated and destined for outside of a domain and passes through the domain. The information about the total traffic in each gateway link and destined for each domain is received from the INCC. Based on this type of traffic can be handled as outgoing traffic generated in the incoming gateway node. Therefore, we can assume that there is
no transit traffic. The outgoing requirements for the gateway nodes are modified to incorporate the amount of traffic coming from their incoming gateway links.

**Incoming traffic:** This type of traffic is generated in a source node outside of a domain and destined for a node within the domain. It is the only type of traffic for which insufficient information is received from the INCC. The INCC provides to each domain only the total amount of traffic destined to it on each of the gateway links adjacent to the domain. This is not enough for the local routing since the incoming traffic must be routed to its actual destination node. Therefore, it is necessary to assume some breakdown per destination node and do a locally optimal routing for the assumed incoming traffic pattern. We shall present three different estimation policies about the incoming traffic in Section 3.3.

In the proposed algorithm, flows in the gateway links are controlled by the INCC and cannot be changed by NCCs. Flows in the gateway links are used as constraints for the low-level routing. For that reason we assume that the domain being considered, domain $I$, consists of its nodes and internal links only. The topology of such a graph and the capacities of the links are known to NCC $I$. Furthermore, NCC $I$ knows the following:

- the internal node-to-node requirements, $r_{ij}$
- the outgoing node-to-domain requirements, $r_{ij} I \neq J$
- the total incoming traffic destined to each node, $r_{ii}$
- the flows of messages, $f_{gi}$, in all gateway links $g$ entering domain $I$ and destined domain $J$. By $g \rightarrow i$ and $g \leftarrow i$ we denote gateway links $g$ in and out of gateway node $i$, respectively.

The low-level routing problem can be formulated as follows. Minimize the local average delay, or more precisely, the average number of packets, $D_I$, defined as

$$D_I = \sum_{i \in I} \frac{f_i}{C_i} - f_i$$

such that the total link flows $f_i$ belong to the set of feasible flows $\Omega$, and $I \subseteq \Omega$ denotes the set of internal, i.e., non-gateway links $I$ in domain $I$. We shall assume that the link flows always satisfy the capacity constraints ($f_i < C_i$). The capacity constraints can be relaxed by extending each term in the sum (1) beyond a certain link utilization (99% in our case) using its second order Taylor series approximation.

The total flow $T_{IJ}$ that a node $i$ in $I$ sends to domain $J$ (including both locally and externally generated traffic) appears like local traffic generated in $i$ and is computed as

$$T_{IJ} = \begin{cases} r_{ij} & \text{internal } i \land J \neq I \\ r_{ij} + \sum_{g \rightarrow i} f_{gi} & \text{gateway } i \land J \neq I \\ \sum_{g \leftarrow i} f_{gi} & \text{gateway } i \land J = I \\ 0 & \text{all other cases} \end{cases}$$

Note that we did not include the local requirements $r_{ij}$ of domain $I$ in the third and fourth line of the right hand side of (2). This separates the traffic with, uncertain destination ($T_{IJ}'$) due to lack of information, and certain destination ($r_{ij}$).

The total flow $R_{ij}'$ that a node $i$ in $I$ sends and relays to domain $J$ (that has not been generated in $I$) appears like local traffic destined to $i$. It is computed as

$$R_{ij}' = \begin{cases} R_{ij}' & \text{any } i \land J = I \\ 0 & \text{internal } i \land J \neq I \\ \sum_{g \leftarrow i} f_{gi} & \text{gateway } i \land J \neq I \end{cases}$$

Let define the following notations:

- $P_{ij}$ is the set of paths between nodes $i$ and $j$
- $\pi_{ij}$ is the fraction of the internet traffic between nodes $i$ and $j$ on path $p$
- $\phi_{ij}^p(J)$ is the fraction of the internet traffic between nodes $i$ and $j$ on path $p$ destined to domain $J$
- $x_{ij}$ is the amount of the internet traffic that node $i$ sends to node $j$ and node $j$ relays to domain $J$

The global flow in each link $l$ can be determined as

$$f_l = \sum_{i \in Z} \sum_{j \in Z} \pi_{ij}^l + \sum_{p \in P_l} \sum_{i \in Z} \phi_{ij}^l$$

The linear constraints are

$$\sum_{i \in I} \pi_{ij}^l = 1 \quad \forall i, j$$
$$\sum_{p \in P_l} \phi_{ij}^l = 1 \quad \forall i, j, l$$
$$\sum_{i \in Z} x_{ij} = T_{ij}' \quad \forall i, j$$
$$\sum_{i \in Z} x_{ij} = R_{ij}' \quad \forall j, l$$

It can be shown [Launberger84] that if (i) the global minimum exists, and (ii) the objective is convex, then the global minimum can be found using a descent direction procedure such as the Flow Deviation Algorithm [Fratta73].

### 3.2. The Low-Level Routing Algorithm

After the formulation of the problem in the previous paragraph, we can outline the low-level routing algorithm as a constrained version of the Flow Deviation Algorithm. A formal proof that the algorithm converges to the global minimum of (1) subject to constraints (2-8) can be found in [Dimitrijevi60].

1. Initialize variables so they satisfy the constraints (2-8).
2. Find the shortest paths and their lengths $d_{ij}$.
3. Establish the extremal flows:
   1. Establish the extremal flows due to the local traffic $r_{ij}$ by sending it along the shortest paths.
   2. Establish the extremal flows due to the transit traffic. We shall denote these flows as $T_{ij}'$. Find the set of $T_{ij}'$, for all $J \neq I$, that minimizes:

   $$\sum_{i \in Z} x_{ij} d_{ij}$$

   subject to (7) and (8) as a linear transportation problem.
Send the determined $x'_{ij}$ along the shortest paths.

3.3. Establish the extremal flows due to the incoming traffic, i.e., find $x'_{ij}$ according to some estimation policy. Send the obtained $x'_{ij}$ along the shortest paths. This issue will be discussed in the next paragraph.

4. Add all flow components above (3.1, 3.2, and 3.3) and obtain overall extremal flows $\nu_i$. Find an optimal $\lambda$, such that the convex combination between old link flows $f_i$ and extremal link flows $\nu_i$ produces new link flows $f_i = (1-\lambda)f_i + \lambda \nu_i$, that minimizes the total number of packets in the local links.

5. Update $x_{ij}$ as $x_{ij} = (1-\lambda)x_{ij} + \lambda x'_{ij}$ where $x_{ij} = \sum_j x'_{ij}$.

6. Go to step 2 unless a stopping criterion is satisfied.

3.3. Estimation Policies

The following section explains how to determine $x'_{ij}$ and establish the extremal flows for the incoming traffic in step 3.3 in the LLR. This step depends on the estimation policy. We consider the best case, proportional case, and worst case.

**Best case:** We assume that the breakdown of the total incoming traffic will be the best possible for the domain’s local performance. We wish to find $x'_{ij}$ such that

$$\min \{x'_{ij} \mid \min \{x'_{ij} \mid J \neq lD_{ij}^{op}\} \} = \min \{x'_{ij} \mid D_{ij}^{op} \}$$

(10)

where $D_{ij}^{op}$ denotes the minimum number of packets (minimum delay) in domain $I$ for a particular set of $x'_{ij}$. The symbol $\min\{a(x)\}$ denotes the minimum of $a$ over the set of variables $x$.

In the best case estimation policy, steps 3.2 and 3.3 of the LLR are equivalent, i.e., we have to minimize (9) subject to (7) and (8) for $J = I$ in step 3.3.

**Proportional case:** In this case, we assume that the incoming traffic from each of the gateway nodes $i$ that is destined for a particular destination node $j$ is proportional to the total incoming traffic from that gateway node

$$x'_{ij} = \frac{T_i^j}{S} \quad i,j \in I$$

(11)

where $S$ is the total incoming traffic

$$S = \sum_i T_i^j = \sum_j R_j^i$$

(12)

Therefore, the $x'_{ij}$ and $x_{ij}$ are kept fixed during the execution of the LLR. The maximum entropy principle [Papoulis84] can be used as a justification [Dimitrijevich90] for the proportional assumption. The experiments showed that the proportional estimate generally gives better network performance since the best and worst case are not likely to occur.

**Worst case:** In this case, we assume that the breakdown of the total incoming traffic from each of the gateway nodes will be the worst possible for the domain’s local performance. We wish to find $x'_{ij}$ such that

$$\max \{x'_{ij} \mid \min \{x'_{ij} \mid J \neq lD_{ij}^{op}\} \}$$

(13)

In this case, $x'_{ij}$ must be determined in a different way than the other two cases. The problem can be formulated as a nonlinear transportation problem that can be solved using the following algorithm.

**Algorithm W:**

To find an optimal routing using the worst case assumption, we use the computational procedure for solving a transportation problem ([Gass84] pp 339-340). In each iteration, the cost coefficients $d_{pq}$ are linearized using the LLR and are assumed fixed.

1. Initialize $x'_{ij}$ by finding a basic solution $B$ to the constraints (7) and (8) for $J = I$. If there are $n$ nodes in the domain, there will be at most $2n-1$ non-zero variables $x'_{ij}$ since the number of constraints (7) and (8) is $2n$, one of them being linearly dependent of the other $2n-1$ constraints. We say that they belong to the basis $B$.

2. Perform the LLR while keeping $x'_{ij}$ fixed and $x'_{ij} = x'_{ij}$.

3. Compute the shortest path lengths $d_{ij}$ between all pairs of nodes $i$ and $j$ in $I$.

4. Solve a system of $2n-1$ linear equations with $2n$ variables, $a_i b_j (i = 1, 2, ..., n)$.

$$a_i + b_j = -d_{ij} \quad \forall (i,j) \in B$$

5. Find $p$ and $q$ that will give

$$\max \{(p, q) \in B \mid a_p b_q = -d_{pq}\}$$

where $B$ denotes the set of variables $x'_{ij}$ that are not in basis $B$.

6. If $a_p + b_q + d_{pq} \leq 0$, stop the algorithm. Otherwise go to the next step.

7. Introduce $x'_{pq}$ into the basis by giving it the maximum value so that the non-negativity of the $x'_{ij}$ is still preserved. A basic variable that becomes equal to zero is eliminated from the basis.

8. Go to step 2.

At the end of Algorithm W, the routing will be optimal assuming that the actual breakdown of the incoming traffic is the worst possible for the domain’s local performance.

3.4. Computation of Equivalent Capacities

The purpose of equivalent capacities is to penalize usage of paths leading to congested areas in the network. Let $g$ be a gateway link connected to gateway node $i$. Let $C_g$ be the capacity of $g$. In our implementation, we compute the equivalent capacities $C'_g$ from the form

$$\frac{C'_g}{(C'_g - f_g)^2} = \frac{C_g}{(C_g - f_g)^2} + \sum_j \alpha_{ij} d_{ij}$$

(14)

as a heuristic function of condition estimated traffic in the domain. A coefficient $\alpha_{ij}$ in (14) is the fraction of the total internet traffic from gateway node $i$ to any other node $j$. Due to lack of information, these coefficients can be estimated only in function of the estimated policy.

3.5. A Posteriori Routing

The initial routing is produced based on an estimate of the missing information about the states of the gateway links. However, the estimated parameters are not exact and therefore the local performance of the produced routing is not optimal. The determined routing can be used for a period of time until steady state
flows are established and each domain learns about the missing information. After the learning period, each domain can adjust its routing. This a posteriori routing can be done in two different ways:

**Unchanged use of the gateway links:** Each domain keeps the previously determined use of the gateway links unchanged and readjusts the use of its internal paths. A new, locally optimal routing is computed with these requirements. This approach requires only one measurement of the actual incoming requirements since the local changes will not affect other domains.

**Changed use of the gateway links:** In this version of the a posteriori routing, each domain is allowed to change its use of the gateway links by transit and outgoing traffic subject to the constraints imposed by the INCC in the last iteration of the routing. However, this adjustment can affect the rest of the network and therefore more than one iteration i.e., learning period, may be necessary. Note that there is no guarantee that this process converges. In all examples that we tried, three to five iterations were necessary for convergence.

In the a posteriori phase of the routing algorithm the INCC, is not involved and the total flows in the gateway links remain unchanged. Therefore, all domains can measure and adjust their routing asynchronously.

**4. A GRADUALLY LOADED NETWORK AS AN EXAMPLE**

We have carried a number of examples in order to evaluate the proposed heuristic algorithm. In this section we present just one of them as an illustration. All results are obtained analytically with all links modeled as independent M/M/1 queues. The performances of the algorithm is compared against the global optimal solution (e.g. obtained from the Flow Deviation for the global network) and the lower bound generalized from [Dimitrijević89] The lower bound is determined using global knowledge and additional constraints, that is: (i) each node routes messages destined to another domain as a single commodity regardless of the particular destination node, and (ii) a message never enters a particular domain more than once.

Figure 2 depicts the topology of a four-domain network. Capacities of all links are equal to 50 units. The gateway links are depicted with heavy lines. The initial end-to-end requirements are generated as exponentially distributed random numbers with the average value equal to 1. The network is gradually loaded by multiplying the initial requirements by the relative load factor ranging from 1 to 2.6. Figure 3 shows that the average delay per packet in the network is close to the lower bound even when the maximum link utilization exceeds 90%. The maximum link utilization is also close for all three algorithms as shown in Fig. 4.

**5. CONCLUSION**

A general approach to designing a hierarchical algorithm for routing in multi-domain networks is presented and a heuristic routing algorithm is developed in this paper. A simplified model of a packet-switched network is used. The overall network performance is managed through the high-level routing that is performed by an Integrated Network Control Center. The computational complexity of the proposed algorithm grows as \( O(n^2) \) for an appropriately partitioned network and the size of the routing tables is \( O(\sqrt{n}) \). An example is presented as an illustration of the proposed heuristic algorithm. Generally, the algorithm provides network performance that is no more than 10% above the lower bound although some special examples can be made up where the algorithm performs much worse. In many cases, the network performance was less than 5% above the lower bound.

**REFERENCES**


**3D.1.6.**
Figure 1. A multi-domain network

Figure 2. A four-domain network topology

3D.1.7.
Figure 3. Average delay for the gradually loaded network in Fig. 2.

Figure 4. Maximum link utilization for the gradually loaded network in Fig. 2.