Routing and Capacity Assignment in a Network with Different Classes of Messages

Irina Neuman

Information Systems Area
Graduate School of Business Administration
New York University, New York, NY

Abstract

A mathematical model is presented for the problem of jointly assigning routes to the communicating pairs of nodes and capacities to the links in a packet switched network. It is assumed that several classes of flow are using the network, different service requirements and message characteristics being associated with each class. An algorithm that generates good feasible solutions to the model, together with tight lower bounds on the value of the objective function, is presented. Results of numerical experiments using several network topologies are reported.

1 Introduction

A starting point for most of the existing research in the area of backbone network design is the implicit assumption that all messages in the network have similar characteristics and requirements. As a result, a uniform treatment is adopted for all messages, with no distinction being made among different types of applications, each with their own specific characteristics, nor between different user requirements. Such an approach greatly reduces the complexity of the analysis, but in most cases the assumption does not correspond to the real world environment. Explicitly taking into account the characteristics and requirements of different classes of messages not only leads to a solution that is preferable from a global perspective (e.g. depending on the performance criterion, the average delay in the network may be reduced, or an overall less costly design may be achieved), but also the solution is better tailored to the individual user needs.

The practical relevance of the issue is suggested by the fact that routing strategies that differentiate among messages in accordance with their various characteristics and requirements are commonly implemented by many operational networks, SNA [2], DATAPAC [17], and SITA [4] being just some of the examples.

This paper addresses the following problem: how to simultaneously select the link capacities and the routes to be used by the communicating nodes in a network that supports several classes of messages with different priority levels. The model is a generalization of earlier models introduced in [16] and [6].

The capacity and flow assignment aspects of network design are usually dealt with separately in the literature. In most cases, such an approach is not appropriate. The close interaction that exists between the capacity value of a link, and the delay incurred by a given flow on that same link, makes it difficult to claim that a truly good solution has been found for either of the two problems when considered independently.

A growing body of research literature deals with the performance analysis of computer networks. Significant efforts have been made to tailor the general models for networks of priority queues to the specific characteristics of computer communication systems [3,11,15,17,18,19]. Even when simplified networks are considered, the complexity of the underlying phenomena is such that finding optimal or near optimal solutions to these models is a difficult task.

The comparative results in the area of performance evaluation of computer communication systems supporting several classes of service strongly suggest that the overall performance is significantly improved when messages are prioritized. These theoretical indications, together with the experience gained from the operational networks that chose to implement similar methods, are powerful arguments in favor of such schemes. Nevertheless, the literature dealing with the related design issues is very limited. To our knowledge, the only authors who incorporated this important aspect into their design methodology, are K. Maruyama and D. T. Tang. A sequence of their papers deals with increasingly complex aspects of the problem. In [14] only discrete link capacity assignment is considered. Messages are classified according to their processing and delay characteristics, and known priority levels are associated with each message class. The heuristic procedure suggested for the solution of the model attempts to minimize the total link cost, while satisfying the delay requirement constraints specific to each class. An interesting refinement is introduced in [13]. This time the priority levels are no longer assumed to be known in advance, i.e. they are no longer user-assigned, and are instead determined by the system. Thus, an additional reduction in the cost of the capacity assignment can be achieved, by determining the best mapping of message classes into different priority levels. The heuristic is a composite procedure, that alternates between two separate algorithms, for capacity and priority assignment respectively, until a local minimum is found. Finally, in [12] the scope of the proce-
duration is further broadened, by also including an algorithm that handles static flow assignment. The global algorithm starts by determining the flow on each link, based on the maximum available capacities. The initial flow assignment satisfies the throughput requirement, but ignores the capacity constraints. Next, the procedure iterates between the capacity and priority assignment, and the flow assignment algorithms, until no further improvement is possible.

The complexity of the issues involved in network design renders an attempt to find optimal solutions an illusory goal for all but the most trivial cases (e.g. very small networks and/or models based on highly unrealistic assumptions). As a result, the majority of the solution methods suggested in the literature are of a heuristic nature. They do not provide for a way to evaluate the quality of the feasible solution generated, a fact which may significantly hamper their usefulness for real life applications.

The remainder of the paper is organized as follows: in section 2, the problem is defined, and a mathematical model is developed; section 3 describes the solution procedure, while methods for obtaining good upper and lower bounds on the optimal value are outlined in section 4; finally, section 5 contains the results and the analysis of the computational experiments conducted with the model, as well as some concluding remarks.

2 Problem Formulation

The limited capacity of network components gives rise to queuing phenomena. These are modeled by associating a server with each link, whose service rate is determined by the link capacity and by the message length. Messages are viewed as customers competing for the link server. Unlimited buffering space and no processing delays at the network is assumed for ease of exposition. Propagation delays, which are negligible for terrestrial links, are also ignored. Messages from each class arrive on the network boundaries according to Poisson processes with known average interarrival times. Message lengths are exponentially distributed for each class. The independence assumption, first introduced in [9], is also used. The resulting model is that of a Jacksonian network of queues, in which average delay measures are easily computable.

Each message class is associated with a known priority level. A head-of-the-line non-preemptive discipline is imposed on the messages waiting at each link. Static routing is assumed in the model. Such routing mechanisms are used in many operational networks (e.g. [2], [4],[20]), and are known to perform well, mostly due to their simplicity and stability.

Two distinct types of costs, which reflect the unified way in which the model deals with the flow and capacity assignment issues, are considered:

1. capacity costs, comprised of a fixed setup cost, and a variable cost, which is a function of the traffic on the line; and

2. queuing costs, associated with the delay incurred by messages in the network.

The model requires the following notation:

- \( L \) = set of links in the network
- \( J \) = total number of priority classes
- \( 1/\mu_j \) = average message length for class \( j \in J \)
- \( I_l \) = set of line types available for link \( l, l \in L \)
- \( Q_{lk} \) = the capacity of line type \( k, k \in I_I \)
- \( S_{lk} \) = the fixed cost of line type \( k, k \in I_I \)
- \( C_{lk} \) = the variable cost of line type \( k, k \in I_I \)
- \( D_l \) = unit cost of delay for messages in class \( j \in J \)
- \( R \) = set of candidate routes
- \( \Pi \) = set of communicating nodes in the network
- \( S_{p,j} \) = set of candidate routes for class \( j \) messages associated with origin-destination pair \( p \in \Pi \). \( S_{p,r}, p \in \Pi \) is defined as \( \cup_{j \in S_{p,j}} \). The sets of candidate routes for different classes of messages are not necessarily disjoint, i.e. \( \cap_{j \in S_{p,j}} \neq \emptyset \).

- \( \lambda_{ij} \) = the class \( j \) message arrival rate for the unique origin-destination pair associated with route \( r \in R \). Also, \( \lambda_{rj} = \lambda_{ij}, \forall r \in S_{p,j} \).
- \( \phi_{ij} \) = the class \( j \) message rate on link \( l \)
- \( F_{l,j} = \phi_{ij}/\mu_j \) = the class \( j \) bit rate on link \( l \)
- \( T_{ij} \) = the average delay incurred on link \( l \) by a class \( j \) message.
- \( \delta_{il} \) = an indicator function, taking the value one if link \( l \) is used in route \( r \), and zero otherwise
- \( \pi_{jr} \) = a decision variable, taking the value one if route \( r \) is chosen to carry the class \( j \) flow of its associated origin-destination pair, and zero otherwise.
- \( y_{lk} \) = a decision variable, which is one if line type \( k \) is assigned to link \( l \), and zero otherwise
- \( T_{ij} \), the average delay on link \( l \) for class \( j \) messages, can be computed as (see [10]):

\[
T_{ij} = \frac{\delta_{ij}(1 - \sigma_j) + \sum_{h \in S_{lj}} \phi_{ij}(y_{lh})^2}{(1 - \sigma_j)(1 - \sigma_{j+1})}
\] (1)

where \( \delta_{ij} = 1/\mu_j \sum_{h \in S_{lj}} Q_{lh}y_{lh} \) is the average 'service time' for class \( j \) messages on link \( l \), and \( \sigma_j = \sum_{h \in S_{lj}} \phi_{ij}y_{lh} \). \( T_{ij} \) includes both the queuing delay incurred by a message while waiting in the buffers of a network switch before transmission, as well as the transmission time.

As a result, the average end-to-end delay in the network for class \( j \) messages can be expressed as:
where $T_j^i = \frac{1}{\gamma_j} \sum_{I \in L} \phi_j T_{ij}^i$

where $\gamma_j = \sum_{I \in L} \lambda_j$ is the total external arrival rate for class $j$.

The above expression becomes untractably complex as the number of message classes increases. We will therefore concentrate in the following on the case of a network supporting just two classes of messages and, without loss of generality, assume that the higher priority is associated with the second class. It is an important case, as indicated by the sharp distinction, both in terms of their requirements as well as of their processing characteristics, between traffic generated by interactive computation, with its tight delay requirement, on one hand, and such applications as file transfer and remote job entry, for which response time is less of a critical factor, and which as a result may be associated with a lower priority, on the other.

From (1), the following expressions are obtained for the average delay on link $l$ for class 1 and class 2 messages, respectively:

$$T_{1l} = \frac{(Q_1 - F_{1l})}{\mu_1} + \frac{F_{2l}}{\mu_2}$$

where the average class $j$ bit flow on link $1$ can be expressed in terms of the decision variables $x_j$ as:

$$F_{ij} = \sum_{r \in R} \lambda_j \beta_i x_j / \mu_j, j = 1, 2$$

The problem of optimally assigning primary routes and link capacities in a network supporting two classes of messages is then equivalent to finding the binary variables $x_j$, and $y_{lk}$ values that satisfy:

**Problem $P_1$**

$$Z_P = \min \left\{ D_1 \sum_{l \in L} \frac{F_{1l}(Q_1 - F_{1l})}{(Q_1 - F_{1l})(Q_1 - F_{12})} + D_2 \sum_{l \in L} \frac{F_{2l}}{(Q_1 - F_{12})} \right\}$$

subject to:

$$F_{1l} + F_{2l} \leq \sum_{k \in L_k} Q_{lk} y_{lk} \forall l \in L$$

$$\sum_{r \in R} x_{ij} = 1 \forall p \in P, j = 1, 2$$

$$\sum_{k \in L_k} y_{lk} = 1 \forall l \in L$$

$$x_j = 0, 1, 1 \forall r \in R, j = 1, 2$$

$$y_{lk} = 0, 1, 1 \forall l \in L, k \in L_k$$

where $F_{ij}, j = 1, 2$ are defined by (4), and $a = \mu_1 / \mu_2$.

The first two objective function terms capture the total cost of delay for the lower priority and higher priority message classes, respectively. The third term corresponds to the total fixed capacity cost, while the fourth represents the total variable cost. The constraints in (6) ensure that the chosen capacity is feasible in terms of the flow assigned to the link. They are equivalent to the constraint set of the NP-complete multiconstrained knapsack problem. The problem studied here is therefore of at least the same complexity. Constraints in (7) and (8) guarantee that only one route is chosen for each origin-destination pair, and only one line type for each link, respectively. Notice that, since $S_{kl}$ and $S_{kl}$ are not necessarily disjoint, the formulation allows for the two types of flow to be directed either along the same or along different routes.

The nature of the problem imposes certain restrictions upon the characteristics of the higher priority messages. On an average, they must be shorter than the low priority messages, and they have to pay for the increase in performance they require. As a result, the following relations must hold among the problem parameters:

1. $a \leq 1$, i.e. the average length of class 2 messages cannot exceed that of class 1 messages, and
2. $D_2 \geq D_1$, i.e. the unit cost of delay is at least as high for class 2 messages as for class 1 messages.

The unit costs of delay are estimates based on user requirements. The model implicitly taking account the different delay requirements of the two message classes. Priority messages, with their tighter response time requirement, are associated with a higher cost of delay, which lowers the average delay they incur in the final solution. As a result, it is no longer necessary to introduce delay bounds specific to each message class (such was, for instance, the approach used in [12] and [14]), and the structure of the constraint set is significantly simplified.

To better evidence the underlying structure of the problem, a set of derived decision variables is next introduced. The $f_{ij}, j = 1, 2$ variables are defined as the portion of the utilization of link $l$ attributable to type $j$ flow:

$$f_{ij} = \frac{\sum_{r \in R} \lambda_j \beta_i x_{ij} / \mu_j}{\sum_{k \in L_k} Q_{lk} y_{lk}}$$

In terms of the augmented set of decision variables, the problem becomes that of finding the $f_{ij}, x_{ij}$ and $y_{lk}$ variables that satisfy:

**Problem $P_1$**

$$Z_P = \min \left\{ D_1 \sum_{l \in L} \frac{f_{1l}(1 - f_{1l}) + a f_{1l} f_{2l}}{(1 - f_{1l})(1 - f_{1l} - f_{2l})} + D_2 \frac{f_{2l}}{1 - f_{1l}} \right\}$$

subject to:

$$\sum_{r \in R} \lambda_j \beta_i x_{ij} / \mu_j \leq f_{ij} \sum_{l \in L} Q_{lk} y_{lk} \forall l, j = 1, 2$$

$$f_{1l} + f_{2l} \leq 1 \forall l \in L$$

$$f_{lk} \geq 0 \forall l \in L, k \in L_k$$

where $f_{ij}, j = 1, 2$ are defined by (4), and $a = \mu_1 / \mu_2$. 

782
and: (7)-(10).

3 Solving the Model

A Lagrangian relaxation to Problem P1 is obtained by multiplying the capacity constraints in (12) by a vector of non-positive Lagrange multipliers \( \{a_i, i \in L, j = 1, 2\} \), and adding them to the objective function. With the coupling constraints no longer present, the Lagrangian problem can be decomposed into a problem depending only on the link decision variables \( f_{lj} \) and \( y_{ik} \), and a second problem over the routing variables \( x_{ij} \). Each of these problems, in turn, can be further decomposed over the links in the network, and over the origin-destination pairs and message classes, respectively.

The \([J] \times [II]\) subproblems associated with a given traffic type for each of the communicating pairs, have a simple structure:

**Problem P1(\(\alpha, p, j\))**

\[
L(\alpha, p, j) = \min \left\{ \sum_{r \in S_{ pj}} a_r x_{pj} \right\}
\]

subject to:

\[
\sum_{r \in S_{ pj}} x_{pj} = 1
\]

\[
x_{pj} = 0, 1 \quad r \in S_{pj}
\]

where: \(a_r = \sum_{L} a_i L \alpha_i b_{ri} / \mu_i Q_i\).

The subproblems are readily solved by setting to one that \(x_{pj}\) variable that has the lowest coefficient in the objective function, i.e.

\[
a_{ij} = \min_{r \in S_{pj}} \{a_r\} \Rightarrow x_{pj} = 1
\]

The \([L] \times [II]\) link subproblems resulting from the decomposition are more complex:

**Problem P1(\(\alpha, l\))**

\[
L(\alpha, l) = \min \left\{ D_1 f_1 (1 - f_2) + a_1 f_1 f_3 + D_2 f_2 (1 - f_1 - f_2) + \sum_{k \in L} S_{ik} y_{ik} \right\}
\]

\[
+ f_3 \sum_{k \in L} Q_{ik} y_{ik} (C_{ik} + \alpha_1) + f_3 \sum_{k \in L} Q_{ik} y_{ik} (C_{ik} + \alpha_2)
\]

subject to:

\[
f_1 + f_2 \leq 1
\]

\[
\sum_{k \in L} y_{ik} = 1
\]

\[
f_1, f_2 \geq 0
\]

\[
y_{ik} = 0, 1
\]

The set of candidate capacities is likely to be of small cardinality. It is therefore possible to simplify the above problem by successively fixing the \(y_{ik}\) variables to all the possible values that satisfy the constraints in (16) and (18).

The subproblem becomes:

**Problem P1(\(\alpha, l, k\))**

\[
L(\alpha, l, k) = \min \left\{ D_1 f_1 (1 - f_2) + a_1 f_3 f_2 + D_2 f_2 (1 - f_1 - f_2) \right. \]

\[
+ f_3 y_{ik} (C_{ik} + \alpha_1) + f_3 y_{ik} (C_{ik} + \alpha_2) + S_{ik}
\]

subject to: (15) and (17), where the \(k\) index corresponds to the \(y_{ik}\) variable set to one.

The numerical solution to the subproblem, is based on the following theorem:

**Theorem 1**

The objective function of Problem P1(\(\alpha, l, k\)) is unimodal over \(\Omega = \{f_1, f_2 : f_1 + f_2 \leq 1, f_1, f_2 \geq 0\}\).

The theorem proof contains a lengthy argument that is left out for the sake of brevity. The interested reader is referred to [16] for further details.

The result in theorem 1 implies that any algorithm that numerically searches for the minimum within the domain over which the function is defined is guaranteed to converge to a global optimum. Initial experiments showed that a simple successive substitution method has a good convergence rate. The procedure alternately optimizes the function for fixed \(f_{ik}\) and fixed \(y_{ik}\) until no further improvement is obtained in two subsequent iterations. Theorem 1 ensures that the unique minimum is reached at this point.

The objective function value for Problem P1(\(\alpha, l\)) is computed as:

\[
L(\alpha, l) = \min_{l \in L} L(\alpha, l, k)
\]

Once all the subproblems are solved, the Lagrangian value is given by:

\[
L(\alpha) = \sum_{p \in P} L(\alpha, p) + \sum_{l \in L} L(\alpha, l)
\]

It is a known result in optimization theory [7] that the best lower bound is provided by the vector \(\alpha^*\) that corresponds to:

\[
L(\alpha^* ) = \max_{\alpha \in S^0} L(\alpha) \leq Z_P
\]

The following theorem states the relationship that exists between \(L(\alpha^*)\) and the continuous relaxation of Problem P1.

**Theorem 2**

\[
L(\alpha^*) = \hat{Z}
\]

where: \(\hat{Z}\) is the objective function value of the continuous relaxation of Problem P1.

The proof, based on duality theory, is similar to the one presented in [6] for the no priority case.

4 Subgradient Optimization and Heuristic Procedures

A subgradient procedure, an iterative method successfully applied to a variety of combinatorial problems (e.g. [1], [6],
redundant constraints that will restrict the domain of the
original problem. It is possible to further tighten the lower bound by generating
the one over which the Lagrangian problem is defined. It is used in order to obtain a good estimate of
Lagrangian (though not that of the original
origin-destination pairs whose primary route for class
and destination pair, and each message class, out of which
might render the current capacity assignment no longer
optimal. The algorithm therefore alternates between the
capacity and the route improvement steps, until no
further reduction in the overall cost can be achieved.
Since this search for a local optimum may at times
be quite time consuming, in the current implementa-
tion of the algorithm it is initiated only at the user's
pecific request.

5. Computational Results

The model and algorithm presented in the previous sections are implemented as an interactive system that allows the
network topologies and model parameters to be easily defined and modified by the user. At the end of each major
iteration (defined as a user specified number of subgradient
iterations), a comprehensive output, corresponding to the
current best feasible solution, is produced. In addition to
the current value of the Lagrangian, the overestimate, and
the corresponding average message delays, the output also
shows the candidate line types used

Table 1 shows the candidate line types used
(same for all the links in a network), and the associated

4. Route improvement: It is possible to determine the
best flow assignment for a given capacity assignment
by solving the following problem:

Problem F

\[
Z_F = \min \left\{ \sum_{l \in L} D_l f_l (1 - f_l) + a f_l \sum_{k \in K} Q_k y_{k,l} \right\}
\]

subject to: (8), (9), (12), (14), and (19). The capacity
constraints are once again relaxed, and Problem
F is solved using a procedure similar to the one outlined
earlier. The significant difference is that, with the
capacity variables no longer present, not only the
Lagrangian problem is more efficiently solved, but also
the algorithm converges very fast, solution tolerances
of under 5% being generally obtained in less than 40
subgradient iterations.

5. Local optimum: The route improvement procedure
sometimes significantly alters the flow pattern, which
may render the current capacity assignment no longer
optimal. The algorithm therefore alternates between the
capacity and the route improvement steps, until no
further reduction in the overall cost can be achieved.
Since this search for a local optimum may at times
be quite time consuming, in the current implementa-
tion of the algorithm it is initiated only at the user's
pecific request.

5. Computational Results

The model and algorithm presented in the previous sections are implemented as an interactive system that allows the
network topologies and model parameters to be easily defined and modified by the user. At the end of each major
iteration (defined as a user specified number of subgradient
iterations), a comprehensive output, corresponding to the
current best feasible solution, is produced. In addition to
the current value of the Lagrangian, the overestimate, and
the corresponding average message delays, the output also
shows a detailed description of the capacity assignment, specifying for each link its assigned capacity, its message rate
and utilization, the associated fixed, variable, and queuing
costs, and the percentage of the total cost attributable to
it. Thus, the user is presented with a full picture of the
current solution that can be used as a basis for gaining fur-
ther insights into the characteristics of the problem under
consideration.

Three different topologies were used in the experiments
(fig. 1-3). The total average message traffic for all origin-
destination pairs is of four messages for both directions, and
is evenly divided between the two types of flow. The can-
didate routes were generated using the same hybrid procedure
outlined in [5]. Table 1 shows the candidate line types used
(same for all the links in a network), and the associated
capacity costs.

The tolerance measure used for estimating the quality of the results generated by the algorithm is defined as:

\[(Upper \ Bound - Lower \ Bound) / Upper \ Bound\]

The results summarized in the following tables show that the gaps between the lower and the upper bound the algorithm generates are between satisfactory, e.g. 15.5%, and very good, e.g. less than 1%. There does not seem to be an obvious correlation between the parameter values used and the behavior of the algorithm, but the matter requires further investigation.

The experiments were run on a VAX 780 machine. No exact values for the computation times were collected, but they were reasonably small, considering the off-line nature of the problem. The maximum number of subgradient iterations never exceeded 200, and was usually under 100. The amount of time used per iteration was negligible, even for the larger test problem.

From table 1, it can be observed that the average message delay for each message class is only marginally sensitive to the ratio between the two costs of delay, more so for the lower priority messages. On the other hand, as these costs are decreased, so that the objective function is even more dominated by the capacity costs, the average delay experienced by both classes of messages increases, the changes being more significant, once more, for the lower priority traffic.

Table 2 shows the results for fixed costs of delay, and varying average message lengths. As expected, as the message length increases/decreases, the average delay varies accordingly, for both message classes. The variations are rendered more significant by the changes in the capacity assignment.

Further testing of the model is required, before final conclusions can be reached. Nevertheless, the initial results seem to justify the introduction of the priority discipline, and are promising in terms of the quality of the solutions generated by the algorithm.

<table>
<thead>
<tr>
<th>Capacity Set and Base Costs Used in Computational Experiments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable Capacity Setup Distance Variable Cost</td>
</tr>
<tr>
<td>[bps] [dollar/month] [dollar/month/mile] [dollar/month/bps]</td>
</tr>
<tr>
<td>4800 650 0.4 0.360</td>
</tr>
<tr>
<td>9600 750 0.5 0.352</td>
</tr>
<tr>
<td>19200 850 2.1 0.156</td>
</tr>
<tr>
<td>50000 850 4.2 0.030</td>
</tr>
<tr>
<td>100000 2400 4.2 0.024</td>
</tr>
<tr>
<td>200000 1300 21.0 0.020</td>
</tr>
<tr>
<td>400000 1300 60.0 0.017</td>
</tr>
</tbody>
</table>

Table 1: Capacity set and base costs used in computational experiments
Figure 2: Topology and distances for the USA network

Figure 3: Topology and distances for the RING network
### Table 2: Results for different costs of delay

<table>
<thead>
<tr>
<th>Network</th>
<th>Delay costs</th>
<th>Lower bound</th>
<th>Upper bound</th>
<th>Queuing costs</th>
<th>Fixed cost</th>
<th>Variable cost</th>
<th>Tolerance (%)</th>
<th>Average delays</th>
</tr>
</thead>
<tbody>
<tr>
<td>GTE</td>
<td>d1=2000</td>
<td>131010</td>
<td>138077</td>
<td>18785</td>
<td>82062</td>
<td>28549</td>
<td>5.12</td>
<td>33.3</td>
</tr>
<tr>
<td></td>
<td>d1=6000</td>
<td>8080</td>
<td>8080</td>
<td>8080</td>
<td>8080</td>
<td>8080</td>
<td>13.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>d1=3000</td>
<td>122609</td>
<td>132829</td>
<td>18896</td>
<td>79904</td>
<td>30805</td>
<td>7.69</td>
<td>36.6</td>
</tr>
<tr>
<td></td>
<td>d2=2100</td>
<td>3223</td>
<td>3223</td>
<td>3223</td>
<td>3223</td>
<td>3223</td>
<td>14.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>d1=3000</td>
<td>155802</td>
<td>158340</td>
<td>19232</td>
<td>85680</td>
<td>27255</td>
<td>1.54</td>
<td>27.8</td>
</tr>
<tr>
<td></td>
<td>d2=20000</td>
<td>26172</td>
<td>26172</td>
<td>26172</td>
<td>26172</td>
<td>26172</td>
<td>11.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>c1=50</td>
<td>85750</td>
<td>101861</td>
<td>867</td>
<td>56183</td>
<td>44671</td>
<td>16.29</td>
<td>11.4</td>
</tr>
<tr>
<td></td>
<td>c2=60</td>
<td>220</td>
<td>220</td>
<td>220</td>
<td>220</td>
<td>220</td>
<td>27.8</td>
<td></td>
</tr>
<tr>
<td>USA</td>
<td>d1=2000</td>
<td>1407538</td>
<td>1556373</td>
<td>201303</td>
<td>18897</td>
<td>8.0</td>
<td>9.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>d1=6000</td>
<td>1407538</td>
<td>1556373</td>
<td>201303</td>
<td>18897</td>
<td>8.0</td>
<td>9.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>d1=3000</td>
<td>1414836</td>
<td>1563893</td>
<td>260697</td>
<td>1197921</td>
<td>99039</td>
<td>9.64</td>
<td>100.9</td>
</tr>
<tr>
<td></td>
<td>d1=2100</td>
<td>8236</td>
<td>8236</td>
<td>8236</td>
<td>8236</td>
<td>8236</td>
<td>9.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>d1=3000</td>
<td>1623322</td>
<td>1653890</td>
<td>203370</td>
<td>1200320</td>
<td>98056</td>
<td>7.1</td>
<td>78.1</td>
</tr>
<tr>
<td></td>
<td>d2=30000</td>
<td>76433</td>
<td>76433</td>
<td>76433</td>
<td>76433</td>
<td>76433</td>
<td>9.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>c1=20</td>
<td>1110111</td>
<td>1276015</td>
<td>3558</td>
<td>1166088</td>
<td>112122</td>
<td>13.02</td>
<td>199.2</td>
</tr>
<tr>
<td></td>
<td>c2=60</td>
<td>366</td>
<td>366</td>
<td>366</td>
<td>366</td>
<td>366</td>
<td>12.2</td>
<td></td>
</tr>
<tr>
<td>RING</td>
<td>d1=2000</td>
<td>982060</td>
<td>1083677</td>
<td>215312</td>
<td>726837</td>
<td>108544</td>
<td>9.38</td>
<td>42.3</td>
</tr>
<tr>
<td></td>
<td>d2=6000</td>
<td>32883</td>
<td>32883</td>
<td>32883</td>
<td>32883</td>
<td>32883</td>
<td>5.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>d1=3000</td>
<td>1002900</td>
<td>1107734</td>
<td>379049</td>
<td>709006</td>
<td>409375</td>
<td>9.32</td>
<td>45.2</td>
</tr>
<tr>
<td></td>
<td>d1=2100</td>
<td>12971</td>
<td>12971</td>
<td>12971</td>
<td>12971</td>
<td>12971</td>
<td>5.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>d1=3000</td>
<td>1008378</td>
<td>1125126</td>
<td>222609</td>
<td>705123</td>
<td>108766</td>
<td>10.38</td>
<td>45.9</td>
</tr>
<tr>
<td></td>
<td>d2=30000</td>
<td>87426</td>
<td>87426</td>
<td>87426</td>
<td>87426</td>
<td>87426</td>
<td>4.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>c1=20</td>
<td>669869</td>
<td>703296</td>
<td>4047</td>
<td>650622</td>
<td>138229</td>
<td>15.56</td>
<td>53.5</td>
</tr>
<tr>
<td></td>
<td>c2=60</td>
<td>368</td>
<td>368</td>
<td>368</td>
<td>368</td>
<td>368</td>
<td>5.4</td>
<td></td>
</tr>
</tbody>
</table>

Note: \( l/p_1 = 1000 \) and \( l/p_2 = 400 \) for the GTE and USA networks.

\( l/p_1 = 700 \) and \( l/p_2 = 300 \) for the RING network.

### Table 3: Results for different average message lengths

<table>
<thead>
<tr>
<th>Network</th>
<th>Message length</th>
<th>Lower bound</th>
<th>Upper bound</th>
<th>Queuing costs</th>
<th>Fixed cost</th>
<th>Variable cost</th>
<th>Tolerance (%)</th>
<th>Average delays</th>
</tr>
</thead>
<tbody>
<tr>
<td>GTE</td>
<td>d1=1000</td>
<td>131010</td>
<td>138077</td>
<td>18785</td>
<td>82062</td>
<td>28549</td>
<td>5.12</td>
<td>33.3</td>
</tr>
<tr>
<td></td>
<td>d2=400</td>
<td>8080</td>
<td>8080</td>
<td>8080</td>
<td>8080</td>
<td>8080</td>
<td>13.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>d1=1000</td>
<td>147636</td>
<td>160179</td>
<td>26223</td>
<td>95189</td>
<td>30154</td>
<td>7.83</td>
<td>52.0</td>
</tr>
<tr>
<td></td>
<td>d2=400</td>
<td>8413</td>
<td>8413</td>
<td>8413</td>
<td>8413</td>
<td>8413</td>
<td>11.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>d1=1000</td>
<td>115000</td>
<td>127724</td>
<td>181496</td>
<td>74337</td>
<td>30499</td>
<td>9.96</td>
<td>33.5</td>
</tr>
<tr>
<td></td>
<td>d2=100</td>
<td>4732</td>
<td>4732</td>
<td>4732</td>
<td>4732</td>
<td>4732</td>
<td>7.4</td>
<td></td>
</tr>
<tr>
<td>USA</td>
<td>d1=1000</td>
<td>1495088</td>
<td>1556373</td>
<td>261303</td>
<td>1230818</td>
<td>105354</td>
<td>3.94</td>
<td>92.3</td>
</tr>
<tr>
<td></td>
<td>d2=400</td>
<td>18897</td>
<td>18897</td>
<td>18897</td>
<td>18897</td>
<td>18897</td>
<td>8.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>d1=2000</td>
<td>1813968</td>
<td>1950308</td>
<td>234121</td>
<td>1535081</td>
<td>119980</td>
<td>7.00</td>
<td>86.2</td>
</tr>
<tr>
<td></td>
<td>d2=400</td>
<td>72291</td>
<td>72291</td>
<td>72291</td>
<td>72291</td>
<td>72291</td>
<td>9.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>d1=1000</td>
<td>1233601</td>
<td>1420852</td>
<td>198750</td>
<td>1108450</td>
<td>102401</td>
<td>12.76</td>
<td>78.8</td>
</tr>
<tr>
<td></td>
<td>d2=400</td>
<td>10755</td>
<td>10755</td>
<td>10755</td>
<td>10755</td>
<td>10755</td>
<td>4.1</td>
<td></td>
</tr>
<tr>
<td>RING</td>
<td>d1=700</td>
<td>983960</td>
<td>1083677</td>
<td>2153112</td>
<td>726937</td>
<td>108544</td>
<td>9.38</td>
<td>42.3</td>
</tr>
<tr>
<td></td>
<td>d2=300</td>
<td>32883</td>
<td>32883</td>
<td>32883</td>
<td>32883</td>
<td>32883</td>
<td>5.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>d1=1000</td>
<td>1820029</td>
<td>1918065</td>
<td>1464213</td>
<td>1444921</td>
<td>150672</td>
<td>5.04</td>
<td>77.4</td>
</tr>
<tr>
<td></td>
<td>d2=300</td>
<td>150138</td>
<td>150138</td>
<td>150138</td>
<td>150138</td>
<td>150138</td>
<td>4.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>d1=700</td>
<td>964094</td>
<td>977741</td>
<td>217547</td>
<td>641289</td>
<td>102303</td>
<td>1.49</td>
<td>27.4</td>
</tr>
<tr>
<td></td>
<td>d2=150</td>
<td>16000</td>
<td>16000</td>
<td>16000</td>
<td>16000</td>
<td>16000</td>
<td>2.2</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Results for different average message lengths

(D1=2000, D2=6000)
References


