Network Congestion Measurement and Control

Wasim El-Hajj Dionysios Kountanis Anupama Raju
email: {welhajj, kountan, araju}@cs.wmich.edu
Department of Computer Science
Western Michigan University
Kalamazoo, MI 49008, U. S. A.

Abstract

Given a network $G = (V, E)$ with $|V| = n$, a set of projected communication loads l_{ij} between any two sites $v_i, v_j \in V$ and a static routing strategy R, then the expected value of a buffer on a given site is defined as:

$$v(e_{kl}) = \sum_{i=1}^{n} \sum_{j=1}^{n} l_{ij} \times B_{kl}^{ij} \quad (i \neq j) \tag{1}$$

where $B_{kl}^{ij} = 1$ if $e_{kl} \in E$ is in the path P_{ij} defined by R and $B_{kl}^{ij} = 0$ otherwise. Let B indicate the buffer size at each site. The congestion of the network is defined as:

$$c(G) = \sum_{e_{kl} \in E} (2^{\frac{v(e_{kl})}{B}} - 1) \tag{2}$$

The objective of the problem is to find a routing R which minimizes the function $c(G)$. The corresponding decision problem is defined by $c(G) \leq b$, where b is a certain threshold that should not be exceeded in order to minimize congestion. The problem is proven to be NP-Hard. A number of subproblems and their complexities are investigated. Polynomial approximation algorithms are developed in order to design a routing strategy R which reasonably controls the congestion of the network G.

Keywords: Network, congestion, NP-Complete, routing, algorithm.

1 Introduction

With ever increasing communication traffic across the networks, congestion has become a real problem. Congestion in a network depends not only on the communication patterns but also on the design of appropriate routing strategies. In order to deal with the congestion effectively a deviation index has been introduced in [3] as a congestion metric. In [5] an attempt has been made to use this metric to design routings which control congestion, but the objective function used does not seem to be quite appropriate.

There has been an elaborate survey of routings in [1]. Efficiency of routings are investigated in [7, 4] and optimal solutions in [6]. In [8] a classification of routing strategies and their characteristics are described. In [2] a definition of congestion is centered on the user point of view relatively to the usefuleness of the network.

This paper assumes that there is an a priori knowledge of projected communication patterns among various sites of the network. The predictions are computed across time by using an exponential average function which takes in account previously made predictions and errors. The projected loads at the start are assumed to be the products of sizes of local area networks that e_{ij} connects. The congestion metric used is the exponential function $c(e_{kl}) = 2^{\frac{v(e_{kl})}{B}} - 1$, where $v(e_{kl})$ is the number of messages waiting to be transmitted at a site. This measure of congestion was chosen to reflect human impatience which increases exponentially to an overloaded system. The subtraction of 1 is to keep 1 as the center of the measurement. Note that when $v(e_{kl}) = 0$ then $c(e_{kl}) = 0$, when $v(e_{kl}) = B$ then $c(e_{kl}) = 1$, and when $v(e_{kl}) > 1$ then $c(e_{kl}) > 1$. The above implications satisfy the folowing expectations: no congestion when there is no load in the buffer, completely congested when the buffer is full and overcongested (incoming loads are rejected) when new loads are comming in when the buffer is already full.

The objective function is to obtain $min(c(G))$, i.e. to minimize the congestion of the network. The effect of the minimization is to obtain balance among the various $v(e_{kl})$'s without overloading the network.

The problem is to design a routing table (static) relatively to the projected communication loads such that the objective function is achieved. This problem is proven to be NP-Hard as there are a number of its subproblems, except in the case where all communication loads are equal and the routing is based on the shortest path. Polynomial approximation algorithms are designed and their performances are evaluated and compared.

2 Preliminaries

This section includes the basic concepts and definitions used to describe the problems dealt with in this paper.

A **backbone network** connects a set of Local Area Networks (LAN). Each LAN has a size s defined by the number of local sites it includes. A graph $G = (V, E)$, called **interconnection network** (ICN), connects those LANs. The set E represents a wiring between the set V of sites.

During an interval of time any pair of sites $v_i, v_j \in V$ exchanges information called the **communication load** l_{ij} between these sites. The communication load l_{ij} could be different at different time intervals. An exponential averaging formula is used to calculate the expected loads for the next time interval from this communication history. Thus, the load l_{ij}^t at time interval t between sites $v_i, v_j \in V$ is calculated by:

$$l_{ij}^t = \alpha l_{ij}^{t-1} + (1 - \alpha) L_{ij}^{t-1} \tag{3}$$

where l_{ij}^{t-1} indicates the expected load calculated for $(t-1)$, L_{ij}^{t-1} the actual load that occured at $(t-1)$, and α and $(1-\alpha)$ are relative weights given for the expected loads versus the actual loads. Note that for $\alpha = 1$ the next expected load depends solely on the previously calculated expected load etc.

The initial expected load, to start the calculations of l_{ij}^t, is considered to be the product of the site sizes i.e. $l_{ij}^0 = s_i.s_j$.

The use of exponential averaging to calculate l_{ij}^t is based on the belief that "history repeats itself" or the display of locality by the communication exchanges.

There are various other factors which influence the communication in a backbone network. For the **bandwidth** (BW) of the wiring it is assumed that each line ($e \in E$) can independently transmit in either direction (full duplex). Thus, there are two loads associated with an $e = (v_i, v_j) \in E$, the load l_{ij} from v_i to v_j and the load l_{ji} from v_j to v_i.

A load l_{ij} from a site v_i to reach a site v_j has to follow a path P_{ij} through the network. The path P_{ij} consists of a sequence of $v \in E$ that connects v_i to v_j i.e. $P_{ij} = v_{i_1}, v_{i_2} \cdots v_{i_r}$, where $v_i = v_{i_1}$, $v_j = v_{i_r}$, and $(v_{i_x}, v_{i_{x+1}}) \in E \quad \forall \quad 1 \le x \le r - 1$.

A routing R defines the path P_{ij} for all pairs $v_i, v_j \in V$. R can be represented as a two dimentional table with P_{ij}s as its entries (static). A routing R can impose restrictions on the paths. One such restriction is $P_{ij} = \overline{P_{ji}}$ i.e. the path from v_j to v_i is the traverse path from v_i to v_j. Paths can be restricted to preserve heads and tails. Consider a path P_{ij} from v_i to v_j and v_k in the path P_{ij}, then P_{ik} is a subpath (head) of P_{ij}. Similarly, P_{kj} is a subpath (tail) of P_{ij}.

Every $e_{ij} = (v_i, v_j) \in E$ is associated with two buffers, one at each end, in order to hold the frames (equal size loads) to be transmitted. Buffer B_{ij} holds frames waiting to be transmitted from v_i to v_j and buffer B_{ji} holds frames to be transmitted in the opposite direction. The loads waiting to be transmitted will be indicated sometimes as v_{ij} or v_{ji} instead of $v(e_{ij})$ or $v(e_{ji})$ in order to simplify the notation. These accumulated loads, v_{ij}, depend on two factors; the communication loads between sites and the routing R. For given communication loads the routing R can be adjusted in order to reduce the overall number of frames waiting for transmission.

It is assumed that each buffer has fixed size B. The larger the number of frames waiting in a buffer the larger the congestion will be for the associated transmission. If a buffer is full then incoming frames are rejected and they have to be retransmitted. This event makes the congestion to go from bad to worst. Let congestion $c(e_{ij})$ for the $e_{ij} \in E$ be

$$c(e_{ij}) = 2^{\frac{v_{ij}}{B}} - 1 \tag{4}$$

As it was mentioned in the introduction, $c(e_{ij})$ captures the above described concepts. For the whole network G, the congestion is defined to be the summation of congestions of all its sites. i.e.
$c(G) = \sum_{(v_i,v_j)\in E}(2^{\frac{v_{ij}}{B}} - 1), \;\; 1 \leq i, \; j \leq n$

Note that there is a time interval associated with the congestion at each site and by extention on G. Choosing a routing that minimizes congestion is an NP-Hard problem.

3 Congestion Problems

The general congestion problem and a number of its subprolems are defined below.
$\mathbf{\Pi_1}$**.** Given a network $G = (V, E)$ with $|V| = n$, a set of communication loads l_{ij} between any two sites $v_i, v_j \in V$ and a buffer size B for every endpoint $e = (v_i, v_j) \in E$, find a routing R on G such that the congestion of G is minimum i.e. $min\, c(G) = min \sum_{e_{kl}\in E}(2^{\frac{v(e_{kl})}{B}} - 1)$, where $v(e_{kl}) = \sum_{i=1}^{n}\sum_{j=1}^{n} l_{ij} \times B_{kl}^{ij} \quad (i \neq j)$ with $B_{kl}^{ij} = 1$ if $e_{kl} \in E$ is in the

path P_{ij} defined by R and $B_{kl}^{ij} = 0$ otherwise.

Figure 1 illustrates a network (a ring in this case) with 5 sites. Each site has two buffers associated with it, one for each edge emanating from it. It also illustartes a routing R defined with a table of 0's and 1's. The 0,1 indicate that the load is moved clockwise or counterclockwise respectively. The second table contains the loads l_{ij} sent from site i to site j. The integer sums on the sites are the $v(e_{kl})$ for each end of $e_{kl} \in E$ relatively to R.

Figure 1: Given: A ring network with 5 sites, loads l, and a routing R. The figure represents the buffer loads when every site sends a number of frames to every other site. In the routing table, 1 means right and 0 means left.

For example, $v(e_{45}) = 5 + 6 + 7 + 1 = 19$. The $5, 6, 7$, and 1 are loads traversing e_{45} according to routing R.

The problem Π_1 is an optimization problem. The corresponding decision problem $\Pi_1^{'}$ is defined below.

$\mathbf{\Pi_1^{'}}$. Given $G = (V, E)$ with $|V| = n$, a set of communication loads l_{ij} between any two sites $v_i, v_j \in V$, a buffer size B for every endpoint $e \in E$ and a bound $b \in Z^{+}$. Is there a routing R on G such that $c(G) = \sum_{e_{kl} \in E} (2^{\frac{v(e_{kl})}{B}} - 1)$ is bounded by b i.e. $c(G) \leq b$, where
$v(e_{kl}) = \sum_{i=1}^{n} \sum_{j=1}^{n} l_{ij} \times B_{kl}^{ij} \quad (i \neq j)$ with $B_{kl}^{ij} = 1$ if $e_{kl} \in E$ is in the path P_{ij} defined by R and $B_{kl}^{ij} = 0$ otherwise.

The problem Π_2 defined below is a subproblem of Π_1 if G is restricted to be a ring. The major characteristic of the ring network is that only two routes (left and right) are considered. Right means 1 and left means 0. The two possible routes from v_i to v_j can be defines as:

$$right = v_{i\,i+1}, v_{i+1\,i+2}, \ldots, v_{j-1\,j}$$

$$left = v_{i\,i-1}, v_{i-1\,i-2}, \ldots, v_{j+1\,j}$$

Note that the subscripts are taken to be mod n.

$\mathbf{\Pi_2}$. Π_1 restricted to rings.
Of course there is a corresponding decision problem $\Pi_2^{'}$ which is a subproblem of $\Pi_1^{'}$.

Subproblems $\mathbf{\Pi_3}$, $\mathbf{\Pi_3^{'}}$ of Π_2, $\Pi_2^{'}$ respectively are derived by preserving the paths i.e. $P_{ij} = \overline{P_{ji}}$. This condition is satisfied if a certain path is followed from v_i to v_j, then the opposite path should be followed from v_j to v_i. So, if the right route is chosen to go from site 1 to site 4, then the left route should be chosen to go from site 4 to site 1. The general equation that ensures this property is: $\frac{\sum B_{kl}^{ij}}{\sum B_{kl}^{ji}} = 1$

If, in addition to the above restrictions, the preservation of path tails is also required, subproblems $\mathbf{\Pi_4}$, $\mathbf{\Pi_4^{'}}$ of Π_3, $\Pi_3^{'}$ respectively are defined. Refering back to Figure 1, where the path from 1 to 4 goes right, the paths which preserve the tail are the following:

1,2	2,3	3,4	1,5	5,4
2,3	3,4		5,4	
3,4				

The triangles in the above illustration suggest the following general formula for preserving the tails:

$$X + Y = 1 \qquad where, \tag{5}$$

$$X = \frac{\sum_{k=i}^{j}(B_{kk+1}^{kj} + B_{k+1k+2}^{kj} + \ldots + B_{j-1j}^{kj})}{(|i-j| \times |i-j+1|)/2}$$

$$Y = \frac{\sum_{k=i}^{j}(B_{kk-1}^{kj} + B_{k-1k-2}^{kj} + \ldots + B_{j+1j}^{kj})}{(n-|i-j|) \times (n-|i-j|+1)/2}$$

Term X defines the right path and term Y defines the left path ($i \neq j$). The summation for term X ($k \neq j$) goes from $k = i$ up to j. And the summation for term Y ($k \neq j$) goes from $k = i$ down to j. i.e. if site 1 ($i = 1$) is sending frames to site 4 ($j = 4$) then, the values of k will be

{1,2,3} in term X and {1,5} in term Y. If the path from i to j goes right, then Y will be 0 and Xwill be one. The opposite is true if the path goes left. Assuming that the path from 1 to 4 goes right, then

$$\frac{(B^{14}_{12} + B^{14}_{23} + B^{14}_{34}) + (B^{24}_{23} + B^{24}_{34}) + (B^{34}_{34})}{12/2} + \frac{(B^{14}_{15} + B^{14}_{54}) + (B^{54}_{54})}{6/2} = 1$$

where $B^{14}_{12} = B^{14}_{23} = B^{14}_{34} = 1$ since (1,2), (2,3), (3,4) are subpaths of 1,4. $B^{24}_{23} = B^{24}_{34} = 1$ since 2,3 and 3,4 are subpaths of 2,4. $B^{34}_{34} = 1$ since 3,4 is the same path as 3,4. $B^{14}_{15} = B^{14}_{54} = 0$ since 1,5 and 5,4 are not subpaths of 1,4 (note that the route considered goes right). $B^{54}_{54} = 0$ since 5,4 is not in the path of 5,4 when going right. Hence, the 1's and 0's will be exchanged if the route was going left.

Subproblems $\mathbf{\Pi_5}$, $\mathbf{\Pi_5'}$ of Π_4, Π_4' respectively are defined if both the preservation of paths and tails are required. Preserving the path is given by the following equation:

$$\frac{(B^{ij}_{ii+1} + B^{ij}_{i+1i+2} + \ldots + B^{ij}_{j-1j})}{|j-i|} + \frac{(B^{ij}_{ii-1} + B^{ij}_{i-1i-2} + \ldots + B^{ij}_{j+1j})}{|n-j+i|} = 1$$

In order to go from i to j, left or right routes are considered. If the right route is considered then the path from i to $i+1$, $i+1$ to $i+2$, up to $j-1, j$ should also follow the right route. Consider the illustration of Figure 1. Each site has 2 output buffers. One is used if the path goes right and the other if the path goes left. If site 1 sends frames to site 2, the routing path is right and the load sent is 1. So, the amount 1 is stored in the output buffer (horizontal lines) of site 1. Also if site 2 sends frames to site 4, following the right path, then the load size 5 is stored in the output buffers of site 2 and site 3. If site 2 sends data to site 5 and the routing path is left, then the load size 4 is stored in the output buffers (oblique lines) of site 2 and site 1.
To preserve the path form site 1 to site 4 following the right route, the following conditions should be satisfied:

- The path from site 1 to 2 should be right
- The path from site 2 to 3 should be right
- The path from site 3 to 4 should be right

The above three conditions result in

$$\frac{B^{14}_{12} + B^{14}_{23} + B^{14}_{34}}{3} + \frac{B^{14}_{15} + B^{14}_{54}}{2} = 1$$

Notice that, the subscripts of B indicate the subpaths from 1 to 4 (1,2,3,4 or 1,5,4) and the superscripts indicate the source (1) and the destination (4). Assuming that the path from 1 to 4 is to the right, then $B^{14}_{12} = 1$ because the subpath 1,2 is part of the subpath 1,4. Similarly, $B^{14}_{23} = B^{14}_{34} = 1$. $B^{14}_{15} = 0$ because 1,5 is not a sub path of 1,4 since the right path was chosen. Also $B^{14}_{54} = 0$. So, the above equation is true and thus, the path is preserved. In the case where 1 to 2 goes right and 2 to 3 goes left, the above equation is not satisfied.

Preserving the path and the heads defines subproblems $\mathbf{\Pi_6}$, $\mathbf{\Pi_6'}$ of Π_4, Π_4' respectively.

Preserving the heads dictates that the routing from sites 1 to 2 goes right and the routing from 1 to 3 goes also right. If going left from 1 to 4, the heads are preserved if the route from 1 to 5 also goes left. Therefore, the paths produced are:

1,2	2,3	3,4	1,5	5,4
1,2	2,3		1,5	
1,2				

Arguing as previously, the head preservation results in

$$Z + W = 1 \ , \quad where \tag{6}$$

$$Z = \frac{\sum_{k=j}^{i}(B^{ik}_{k-1k} + B^{ik}_{k-2k-1} + \ldots + B^{ik}_{ii+1})}{(|i-j| \times |i-j+1|)/2}$$

$$W = \frac{\sum_{k=j}^{i}(B^{ik}_{k+1k} + B^{ik}_{k+2k+1} + \ldots + B^{ik}_{ii-1})}{(n-|i-j|) \times (n-|i-j|+1)/2}$$

Note that the +/- operations are mod n and $(i \neq j)$. Term Z defines the right path and its corresponding sum goes from $k = j$ down to i $(k \neq i)$. Term W defines the left path and its corresponding sum goes from $k = j$ up to i $(k \neq i)$. i.e. if site 1 $(i = 1)$ is sending frames to site 4 $(j = 4)$ then, the values of k will be {4,3,2} in term Z and {4,5} in term W. For the right path from 1 to 4 the above equation results in

$$\frac{(B^{14}_{34} + B^{14}_{23} + B^{14}_{12}) + (B^{13}_{23} + B^{13}_{12}) + (B^{12}_{12})}{12/2} + \frac{(B^{14}_{54} + B^{14}_{15}) + (B^{15}_{15})}{6/2} = 1$$

Where $B^{14}_{12} = B^{14}_{23} = B^{14}_{34} = 1$ since (1,2), (2,3), (3,4) are subpaths of 1,4. $B^{13}_{23} = B^{13}_{12} = 1$ since 1,2 and 2,3 are subpaths of 1,3. $B^{12}_{12} = 1$ since 1,2 is the same path as 1,2. $B^{14}_{15} = B^{14}_{54} = 0$ since 1,5 and 5,4 are not subpaths of 1,4. $B^{15}_{15} = 0$ since 1,5 is not in the path of 1,5 when going right. The 1's and 0's will be exchanged if the route was chosen to go left.

Preserving the heads, tails, and paths defines problems $\mathbf{\Pi_7}$, $\mathbf{\Pi_7'}$ respectively.

Finally, if $l_{ij} = x$ for every communication load then subproblems $\mathbf{\Pi_8}$, $\mathbf{\Pi_8'}$ are defined.

There can be more subproblems defined by introducing new restrictions, but the ones presented above are the ones commonly considered in the design of network routings. In the next section, the complexities of the above subproblems will be studied.

4 Problem Complexities

In this section the complexities of the problems defined above are investigated. The general problem and all its subproblems except the last one are proven to be NP-Hard. The problem Π_8 is proven to be polynomial by using the shortest path algorithm.

Lemma 1. *Let $G = (V, E)$ be a ring network with fixed loads $l_{ij} = x$ for every pair of sites. Then, the shortest balanced routing produces the following buffer loads:*

$$\frac{n^2-1}{8}x \text{ } if \text{ } n \text{ } is \text{ } odd$$

$$\frac{n^2}{8}x \text{ } if \text{ } n \text{ } is \text{ } divisible \text{ } by \text{ } 4$$

Alternately $(\frac{n^2+4}{8}x)$ *and* $(\frac{n^2+4}{8}-1)x$ *if n is divisible by 2 and not by 4*

proof.

- case1: n is odd. Let a site $i = \frac{n-1}{2} - 1$ as in Figure 2. The shortest paths that contributes to the right buffer of i are presented in Table 1. Therefore, the total load for the buffer is $x + 2x + \ldots + \frac{n-1}{2}x = \frac{n^2-1}{8}x$. Because all sites are symmetrical, the same load goes to each buffer.

- case2: n is multiple of 4. Let a site $i = \frac{n}{2} - 1$ as in Figure 3. Notice that there are $n/2$ diameters in this ring. For each site, at an end of a diameter, there are 2 shortest paths to the opposite site. Therefore, there is a choice to be made. The choices are made alternatively to the right and the next to the left direction because there is an even number of diameters. So, the choices are balanced. The shortest

Figure 2: n sites where n is odd.

Figure 3: n is divisible by 4

Figure 4: n is divisible by 2 and not by 4

paths that contribute to the right buffer of i are presented in Table 2. Therefore, the total load for the buffer is $2(1+3+\ldots+(\frac{n}{2}-1))x = \frac{n^2}{8}x$

- case3: As in Figure 4, n is multiple of 2 but not of 4. In this case the number of diameters is odd and thus, no balanced choices can be achieved. Thus, the right or left choices will be one more than the other. Arguing similarly as before, the loads will alternatively be between $(\frac{n^2+4}{8})x$ and $(\frac{n^2+4}{8} - 1)x$.

Theorem 1. *Let $G = (V, E)$ be a ring with $|V| = n$ and let the expected communication costs to be constant ($l_{ij} = x$) for every pair of sites(v_i, v_j), then the balanced shortest path routing gives the smallest network congestion $c(G)$.*

Table 1: loads for a network with odd number of sites

From Site	To Site	Contributed Load
1	$\frac{n-1}{2}$	x
2	$\frac{n-1}{2}$, $\frac{n-1}{2}+1$	$2x$
$\vdots$	$\vdots$	$\vdots$
$\frac{n-1}{2}-1$	$\frac{n-1}{2}$, $\frac{n-1}{2}+1, \ldots, (n-1)$	$\frac{n-1}{2}x$

Table 2: loads for a network where $n \bmod 4 = 0$

From Site	To Site	Contributed Load
1	$\frac{n}{2}$	x
2	$\frac{n}{2}$	x
3	$\frac{n}{2}$, $\frac{n}{2}+1$, $\frac{n}{2}+2$	$3x$
4	$\frac{n}{2}$, $\frac{n}{2}+1$, $\frac{n}{2}+2$	$3x$
$\vdots$	$\vdots$	$\vdots$
$\frac{n}{2}-1$	$\frac{n}{2}$, $\frac{n}{2}+1, \ldots, (n-1)$	$(\frac{n}{2}-1)x$

Proof. The proof will be carried out by cases.

- n is odd.
  If n is odd, then the shortest path routing will give according to Lemma 1 congestion $2^{\frac{v}{B}}-1$ for each site, where $v = \frac{n^2-1}{8}x$. Therefore, the shortest path congestion is
  $c_1 = 2n(2^{\frac{v}{B}} - 1) = 2n2^{\frac{v}{B}} - 2n$
  Assume a right path from site i to site j according to the shortest path routing. If a communication load x folows the longest path (left) from site i to site j, then the loads will be:

  - For n sites v will not be affected by the change.
  - For $(j-i)$ sites from i to j going right, the load will be $v-x$.
  - For $(n-(j-i))$ sites from i to j going left, the load will be $v+x$.

  The new congestion will be:
  $c_2 = (n2^{\frac{v}{B}} - n) + (j-i)2^{\frac{v-x}{B}} - (j-i) + (n-j+i)2^{\frac{v+x}{B}} - (n-j+i)$
  $\Rightarrow c_2 = n2^{\frac{v}{B}} + (j-i)2^{\frac{v-x}{B}} + (n-j+i)2^{\frac{v+x}{B}} - 2n$

  Since the shortest path from i to j was chosen, $j-i < n-(j-i)$.

  $c_1 <^? c_2$
  $\Rightarrow 2n2^{\frac{v}{B}} - 2n <^? n2^{\frac{v}{B}} + (j-i)2^{\frac{v-x}{B}} + (n-j+i)2^{\frac{v+x}{B}} - 2n$

$\Rightarrow n2^v <^? (j-i)2^{v-x} + (n-j+i)2^{v+x}$
$\Rightarrow n <^? (j-i)2^{-x} + (n-j+i)2^x$
$\Rightarrow n <^? (j-i)2^{-x} + n2^x - (j-i)2^x$
$\Rightarrow n(1-2^x <^? (j-i)(2^{-x} - 2^x)$
$\Rightarrow TRUE$

- n is divisible by 4.
  According to Lemma 1, the load for each of the $2n$ sites will be the same and equal to $\frac{n^2}{8}x$.
  Assume the same changes in the routing and similar arguments as in the previous case, then $c_1 < c_2$.
- n is divisible by 2 but not by 4.
  According to Lemma 1, the loads for n alternating sites will be $v_1 = \frac{n^2+4}{8}x$ and for the remaining n sites $v_2 = (\frac{n^2+4}{8} - 1)x$. Thus, congestion c1 is:
  $c_1 = n2^{\frac{v_1}{B}} + n2^{\frac{v_2}{B}} - 2n$
  If the shortest path from i to j changes to the longest path, then
  - $n/2$ sites will have v_1 loads.
  - $n/2$ sites will have v_2 loads.
  - $(j-i)/2$ sites will have $v_2 - x$ loads.
  - $n - (j-i)/2$ sites will have $v_1 + x$ loads.
  - the remaining sites will have the same loads as before

  Figure 5 illustrates the loads for $n = 6$.
  Therefore,
  $c_2 = (\frac{n}{2}2^{\frac{v_1}{B}} - \frac{n}{2}) + (\frac{n}{2}2^{\frac{v_2}{B}} - \frac{n}{2}) + (\lfloor\frac{j-i}{2}\rfloor 2^{\frac{v_2-x}{B}} - \lfloor\frac{j-i}{2}\rfloor) + (\lfloor\frac{n-j+i}{2}\rfloor 2^{\frac{v_1+x}{B}} - \lfloor\frac{n-j+i}{2}\rfloor) + (\frac{n}{4}2^{\frac{v_1}{B}} - \frac{n}{4}) + (\frac{n}{4}2^{\frac{v_2}{B}} - \frac{n}{4})$

  $c_1 <^? c_2$
  $\Rightarrow c_1 = n2^{\frac{v_1}{B}} + n2^{\frac{v_2}{B}} - 2n <^?$
  $(\frac{3n}{4}2^{\frac{v_1}{B}} - \frac{3n}{4}) + (\frac{3n}{4}2^{\frac{v_2}{B}} - \frac{3n}{4}) + (\lfloor\frac{j-i}{2}\rfloor 2^{\frac{v_2-x}{B}} - \lfloor\frac{j-i}{2}\rfloor) + (\lfloor\frac{n-j+i}{2}\rfloor 2^{\frac{v_1+x}{B}} - \lfloor\frac{n-j+i}{2}\rfloor)$
  $\Rightarrow n2^{v_1} + n2^{v_2} <^? \frac{3n}{4}2^{v_1} + \frac{3n}{4}2^{v_2} + \lfloor\frac{j-i}{2}\rfloor 2^{v_2-x} + \lfloor\frac{n-j+i}{2}\rfloor 2^{v_1+x}$
  $\Rightarrow \frac{n}{4}2^{v_1} + \frac{n}{4}2^{v_2} <^? \lfloor\frac{j-i}{2}\rfloor 2^{v_2-x} + \lfloor\frac{n-j+i}{2}\rfloor 2^{v_1+x}$

  Substituting $v_1 = v_2 + x$, then from Lemma 1,

  $\Rightarrow \frac{n}{4}2^{v_2+x} + \frac{n}{4}2^{v_2} <^? \lfloor\frac{j-i}{2}\rfloor 2^{v_2-x} + \lfloor\frac{n-j+i}{2}\rfloor 2^{v_1+2x}$

Figure 5: A ring network with 6 sites, all loads between sites are equal to x, and communicate with the shortest path routing. Notice, half of the buffers have loads = 4 and the other half have alternately loads = 5.

Subtracting 1 from the floor values retains the inequality.

$\Rightarrow \frac{n}{4}2^x + \frac{n}{4} <^? \frac{j-i-1}{2}2^{-x} + \frac{n-j+i-1}{2}2^{2x}$
$\Rightarrow n2^x + n <^? 2(j-i-1)2^{-x} - 2(2^{2x}(j-i-1)) + 2(n-2)2^{2x}$
$\Rightarrow n2^x + n <^? 2(j-i-1)(2^{-x} - 2^{2x}) + 2(n-2)2^{2x}$
$\Rightarrow 2(j-i-1)(2^{2x} - 2^{-x}) <^? 2(n-2)2^{2x} - n2^x$
$\Rightarrow TRUE$

□

Corollary 1. Π_8 *is polynomial.*
proof. *Theorem 1 polynomially constructs a minimum congestion routing on* G*. Therefore,* Π_8 *is polynomial.*

Theorem 2. Π_7' *is NP-C.*

Proof. The unknowns to be computed in this problem are the values of B_{kl}^{ij}'s. These values are either 0 or 1. Thus, Π_7' is a 0-1 integer programming problem; which has been proven [2] to be NP-Complete. □

Theorem 3. Π_7 *is NP-Hard.*

Proof. Let $S(V, E, l_{ij}, B, b)$ be a subroutine which solves the problem $\Pi_7^{'}$. The following program solves Π_7 with a polynomial number of calls to subroutine S i.e. there is a polynomial Turing reduction (α_T) from $\Pi_7^{'}$ to Π_7.
Algorithm 1:

1. Let $b_m = 0$, $b_M = \sum_{i=1}^{n} \sum_{j=1}^{n} l_{ij} \ \ (i \neq j)$.
2. If $b_m = b_M$, then the minimum congestion routing is found. Stop.
3. Set $b = \frac{b_m + b_M}{2}$ (integer).
4. Call $S(V, E, l_{ij}, B, b)$
   if return = yes then $b = b_m$, then go to step 2.
   if return = no then $b = b_M$, then go to step 3.

Obviously, the above algorithm is polynomial with compexity $O(log_2(b_M))$. Thus, $\Pi_7^{'}$ α_T Π_7 and by definition Π_7 is an NP-Hard problem. □

Corollary 2. $\Pi_1^{'}$, $\Pi_2^{'}$, $\Pi_3^{'}$, $\Pi_4^{'}$, $\Pi_5^{'}$, $\Pi_6^{'}$ *are NP-Complete problems.*

Proof. $\Pi_7^{'}$ is a subproblem to each of them and $\Pi_7^{'}$ is NP-Complete by Theorem 2. Thus, all of them are NP-Complete. □

Corollary 3. Π_1, Π_2, Π_3, Π_4, Π_5, Π_6 *are NP-Hard problems.*

Proof. Π_7 is NP-Hard by Theorem 3. Each one of the problems have Π_7 as a subproblem. Thus, all of them are NP-Hard. □

5 Approximation Algorithms

Since the problems of minimizing the objective function are proven to be NP-Hard, three approximation algorithms are suggested to reduce the network congestion. Also a comparison between these algorithms is presented.

- Algorithm 2. Let n be the number of sites in the network, and let l_{ij} be the load from site i to site j. The purpose of this algorithm is to choose a routing that reduces the congestion of the network. This result is achieved by minimizing and balancing the loads locally. A function $K_{ij} = L_{ij}/min(L_{ij})$ is used which measures how many times the chosen load is greater than the minimum load. So, the combination of K_{ij}, the path length, and the minimum load between the different routes decide which route to take. The algorithm is:

for all i, j where $i \neq j$
compute $K_{ij} = L_{ij}/min(L_{ij})$
for all i, j where $i \neq j$

- compute the length of the path from site i to site j ($P_{ij} = |j - i|$) and the path from site j to site i ($P_{ji} = n - |j - i|$)
- compute max loads e_1max and e_2max across path P_{ij} and P_{ji} respectively.
- find min{ $P_{ij} \times K_{ij} + e_1max$, $P_{ji} \times K_{ij} + e_2max$ } and route according to the minimum.

- Algorithm 3. This algorithm follows the footsteps of the previous one. It differs from the previous algorithm by the way K_{ij} is calculated. K_{ij} is calculated according to $mean(L_{ij})$ and not to $min(L_{ij})$ and it measures how many times the chosen load is greater than the mean load. The algorithm is:

for all i, j where $i \neq j$
compute $K_{ij} = L_{ij}/mean(L_{ij})$

for all i, j where $i \neq j$

- compute the length of the path from site i to site j ($P_{ij} = |j - i|$) and the path from site j to site i ($P_{ji} = n - |j - i|$)
- compute max loads e_1max and e_2max across path P_{ij} and P_{ji} respectively.
- find min{ $P_{ij} \times K_{ij} + e_1max$, $P_{ji} \times K_{ij} + e_2max$ } and route according to the minimum.

- Algorithm 4. This algorithm uses a different approach to choose an appropriate routing. It calculates the summation of the average and the standard deviation of the loads across different routes. The routing is chosen according to the minimum value. A tie is broken by adding the respective path lengths. The algorithm is:

for all i, j where $i \neq j$

- compute the length of the path from site i to site j ($P_{ij} = |j - i|$) and the path from site j to site i ($P_{ji} = n - |j - i|$)

Table 3: Approximation Algorithms Comparison for loads ≤ 12

Sites	4	5	6	7	8	9	10
App. Alg. 1	5.17	5.67	6.60	7.51	8.41	9.33	10.31
> 1	1.13	0.25	0.075	0.01	0.00	0.00	0.00
App. Alg. 2	5.41	6.07	7.13	8.19	9.18	10.19	11.40
> 1	1.27	0.55	0.12	0.02	0.01	0.00	0.00
App. Alg. 3	4.92	5.37	6.28	7.10	7.85	8.65	9.49
> 1	0.8	0.145	0.08	0.01	0.00	0.00	0.00
Optimal	4.82	5.27					

Table 4: Approximation Algorithms Comparison for loads ≤ 1000

Sites	4	5	6	7	8	9	10
App. Alg. 1	5.55	6.13	7.02	8.05	9.07	10.02	11.13
> 1	1.465	0.63	0.225	0.04	0.02	0.01	0.00
App. Alg. 2	5.57	6.20	7.09	8.16	9.23	10.31	11.44
> 1	1.47	0.645	0.26	0.045	0.03	0.015	0.00
App. Alg. 3	5.16	5.82	6.51	7.36	8.17	8.97	9.77
> 1	1.105	0.43	0.105	0.04	0.01	0.00	0.00
Optimal	4.89	5.29					

- compute the standard deviation (σ) on the loads and the average load (L_{avg}) for each path.
- find min$\{ P_{ij}+\sigma_1+L_{avg1} \ , \ P_{ji}+\sigma_2+L_{avg2} \}$ and route according to the minimum.

The above algorithms are tested with randomly generated loads ranging between 1 - 12 and 1 - 1000. The algorithms ran 200 times for networks of $4, 5, \ldots, 10$ sites. The results are displayed in Tables 3,4 and pictorially in Figure 6. The rows in Table 3,4 which correspond to > 1 indicate the average number of overcongested buffers. Note that the congestion of the optimal routing is calculated only for netwroks of sites 4 and 5, because of the exponential growth of the problem.

The conclusion from Figure 6 is that the best approximation algorithm is Algorithm 4 and the worst is Algorithm 2.

6 Conclusion

As it was shown, the congestion problems on the networks are hard to solve. The only exception is the equal load-shortest path case. The approximation

Figure 6: Histograms indicating the performance of the approximation algorithms

algorithms perform fairly well with algorithm 4 displaying the best performance. The reason is that algorithm 4 attempts to equally distribute the communication loads over all sites. To avoid high load values, the average loads and the path lengths are used as controlling variables.

References

[1] A. Chien. A cost and speed model for k-ary n-cube wormhole routers. In *Proceedings of hot interconnect*, 1993.

[2] Kenneth Steiglitz Christos H. Papadimitriou. *Combinatorial Optimization Algorithms and Complexity.* Library of Congress Cataloging in Publication Data, 1939.

[3] G. Quadros E. Monteiro, F. Boarida and V. Freitas. Specification quantification and provision of quality of service and congestion control for new communication services. *AFCEA European symposium and IEEE COMSOC*, 1997.

[4] P. Fraigniand and C. Garoille. Memory requirement for the universal routing schemes. *Fourteen annual symposium on principles of distributed computing*, 1995.

[5] K. Kokkinos. *Load balancing and congestion avoidance routing.* PhD thesis, Western Michigan University, Computer Science, 2002.

[6] S. Carlson M. Degermark, A. Brodnick and S. Pink. Small forward tables for fast routing lookups. In *Proceedings of the SIGCOMM 97 conference on application technologies*, 1997.

[7] J. Vanleaeuwen M. Flammini and W. Marchetti-Spaccamela. The complexity of interval routing in random graphs. *Twentieth international symposium on mathematical foundations of computer science, lect. Notes computer science (Springer)*, 1995.

[8] P. Mohapatra. Wormhole routing techniques for directly connected multicomputer systems. *ACM computing survey*, 1998.