MULTIHOMING IN COMPUTER NETWORKS:
A TOPOLOGY-DESIGN APPROACH

Ariel Orda
Raphael Rom

Department of Electrical Engineering
Technion - Israel Institute of Technology
Haifa, Israel 32000

Revised June 1989

ABSTRACT

Multihoming in networks, i.e., attaching a subscriber to more than a single access point in the network, is a mechanism used to increase several performance criteria.

In this paper we take the topological design view and address the problem of finding optimal multihoming configurations for several topological design criteria. We analyze the problem and demonstrate that except for dual homing, multihoming is algorithmically complex. Optimal algorithms based on maximum matching in graphs and 0-1 integer programming are given for all cases.

I. INTRODUCTION

Multihoming is a situation in which a network subscriber is attached to more than a single node of the network. It is a useful means of improving a number of performance characteristics. For example, it increases subscribers' availability by protecting both against the crash of his node and sometimes against network partitioning. Another aspect of performance improvement due to multihoming is that messages can be forwarded to a subscriber through several paths corresponding to its different points of attachments[1] thereby decreasing end-to-end delivery delay.

More recently, multihoming became the side-effect of internetworking[2,3,4]. Consider a subscriber that is connected to two independent networks; when these networks are interconnected, and when viewed at the internetwork level, our subscriber becomes multihomed. Another case arises when a network is interconnected through more than a single gateway; a subscriber of such a network may be seen from another network as multihomed.

On the other hand, multihoming must be carefully planned because the number of ports in every node is limited and because some configurations are preferred over others. For example, connecting a subscriber to two nodes far apart from one another is costlier than connecting it to two adjacent nodes. This trade-off gives rise to several optimization problems.

In this paper we address the problem of finding an optimal multihoming configuration considering the order of multihoming (i.e., the number of nodes to which a subscriber is attached) and several topological design criteria. As we shall see, some of these problems are tractable; for these efficient algorithms are presented based on maximum matching in graphs. The rest of the problems are shown to be NP-hard, ruling out any optimal polynomial algorithm; for these we formulate the problem as a special case of integer programming, using known techniques for their solution.

In section II we define a general model in whose framework all subsequent problems are defined. Section III deals with double-homing and presents optimal algorithms with respect to a cost function that minimizes the distances between subscribers and nodes. In section IV we demonstrate that multihoming is an NP-hard problem for other than dual-homing and in section V we consider other dual-homing problems that turn out to be complex.

II. MODEL AND PROBLEM FORMULATION

We model our network as an undirected graph $G(V,E,C)$ in which the vertices V represent the network's nodes, E its edges, and C an arbitrary cost function $C:E \rightarrow R^+$. To this network we would like to attach a set of n subscribers $S = \{s_1, s_2, \ldots, s_n\}$ in a multihomed fashion so as to achieve some optimum (to be subsequently defined). Each node of the network is also characterized by a *port index* indicating the number of different subscribers that can be connected to that node. It is clear that not every distribution of ports leads to a solution. For example, if the total number of ports is less than $2n$ no solution exists. We restrict ourselves in the following to cases where a solution does exist.

We define a *multihomed attachment* (MHA) as an unordered pair of nodes* $[v_1, v_2]$ to which a subscriber can potentially be attached in a multihomed fashion (we say that v_1 and v_2 *appear* in the attachment). The multihomed distance *MHd* between a node $w \in V$ and an attachment $MHA = [v_1, v_2]$ is the smallest distance between w and any of v_1, v_2 i.e., $MHd = \min\{d(v_1, w), d(v_2, w)\}$ where the distance between nodes is the usual minimum-cost path in G according to the cost function C. We define a configuration as the set $A = \{A_1, A_2, \ldots, A_n\}$ in which subscriber s_i is attached to the network in the multihomed attachment A_i.

Several problems can now be formulated in which we are seeking a configuration $A = \{A_1, A_2, \ldots, A_n\}$ attaching all n users in a multihomed fashion under the constraint that each node v appears in at most $p(v)$ attachments, where $p(v)$ is the port index of node v. We address some of these in the following subsections.

III. OPTIMAL SUBSCRIBER-TO-NETWORK DUAL-HOMING ATTACHMENTS

Consider a multihomed configuration $A = \{A_1, A_2, \ldots, A_n\}$ and consider an arbitrary subscriber s_i whose multihomed attachment is A_i. We define the subscriber-to-network attachment cost of this subscriber as the sum of the distances between its attachment and all other nodes in the network, that is $CSTN(A_i) \stackrel{\Delta}{=} \sum_{v \in V} MHd(v, A_i)$. With this definition the subscriber-to-network attachment minimization problem can be formulated as follows: find a configuration A such that its total cost $CSTN(A) \stackrel{\Delta}{=} \sum_{A_i \in A} CSTN(A_i)$ is minimized.

---

*This definition is formally correct only for the dual-homing case. Defining similar quantities for the general multihoming case is a trivial extension but requires more cumbersome notation.

In order to demonstrate the importance of using an optimal algorithm for solving such a problem, consider the 8 node network of Figure 1 in which the numbers designate edge costs. Suppose that all but two subscribers have already been attached to the network leaving nodes 1, 2, 3, and 4 with one available port each and nodes 5, 6, 7, and 8 with no available ports. In our terminology this is stated by specifying the port indices as follows:

$$p(1)=p(2)=p(3)=p(4)=1 \qquad p(5)=p(6)=p(7)=p(8)=0$$

Our task is to connect the last two subscribers. Consider first a single-homed attachment. A simple calculation shows that the *CSTN*'s (i.e., the total distance from network nodes) of the four available attachments (1), (2), (3), (4) are 140, 80, 140, and 60 respectively. Thus we attach one subscriber to node 2 and the other to node 4, resulting in a total *CSTN* of 140. Consider now double-homed attachments. The multihomed distances of the possible attachments are as follows:

MHD(1,2)=70	MHD(1,3)=120	MHD(1,4)=40
	MHD(2,3)=70	MHD(2,4)=30
		MHD(3,4)=40

Suppose we use a "greedy" algorithm, i.e., we attach the first subscriber to the best attachment, the second subscriber to the best among the remaining available attachments, etc. In our case this results in attaching one subscriber to (2,4) and the other to (1,3) to yield a *CSTN* of 150, which is worse than in the single-homing case! However, using an optimal algorithm we attach the subscribers to attachments (1,2) and (3,4) (or symmetrically to (1,4) and (2,3)) resulting in a total *CSTN* of 110.

To solve the problem we first break it into two subproblems--single and multiple port cases--which we solve separately. Then, we address the problem of mixed subscribers and finally we show simple extensions of the problem to related cases.

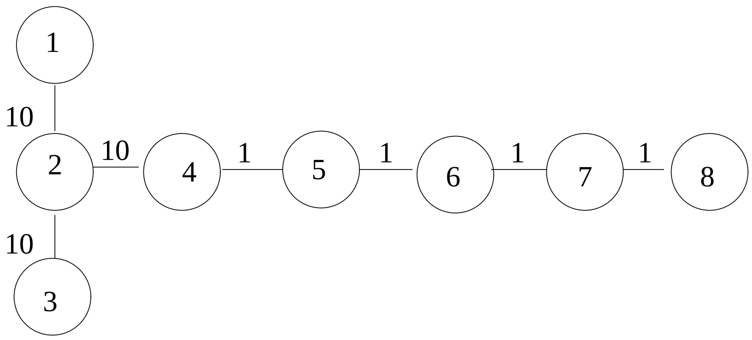

Figure 1: Example Network

A. The Single-Port Case

Here we consider a simplified case of the problem--the case in which every node has a single port, that is, $p(v)=1$ for all $v \in V$. Our approach is based on reducing the problem through a sequence of transformations to a standard matching problem, making use of known solutions to this latter problem.

As a first step we build an auxiliary complete graph $G1=(V,E1,C1)$ having the same set of nodes as G and defining the cost of an edge as the subscriber-to-network attachment cost of a subscriber attached to the two nodes, i.e., $C1([v,w]) = CSTN([v,w])$. Selecting an edge from $E1$ is interpreted as attaching a subscriber to the two nodes at the ends of the edge, and incurring an attachment cost equaling the cost of the edge. Our problem can therefore be stated as the following matching problem:

Problem M1: Given $G1(V,E1,C1)$, find a matching $M1$, such that: (1) $|M1| = n$ and,
(2) $\sum_{e \in M1} C1(e)$ is minimal.

A necessary and sufficient condition for the existence of a solution is the existence of a matching of cardinality n. In the discussion that follows we assume this condition is fulfilled.

Since most matching algorithms deal with maximum cost, we transform the matching problem M1 into a maximum weight matching problem by constructing a graph with the same topology and a new cost function in the following way. Let C' be a cost slightly higher than the highest edge-cost in $G1$, that is, $C' > \max_{e \in E1} \{ C1(e) \}$ and define a cost function $C2(e)=C'-C1(e)$. On the graph $G2(V,E1,C2)$ we define the following matching problem:

Problem M2: Given $G2(V,E1,C2)$, find a matching $M2$, such that: (1) $|M2| = n$ and,
(2) $\sum_{e \in M2} C2(e)$ is maximal.

It is evident that any solution of Problem M1 is a solution of Problem M2, and vice versa.

Problem M2 is a maximum weight matching problem constrained by the number of edges that must be selected. We now introduce another transformation with the goal of relaxing the constraint. To that end we construct a graph $G3(V3,E3,C3)$ by augmenting $G2$ with a set $\tilde{V}$ of auxiliary nodes and a set $\tilde{E}$ of auxiliary edges. Specifically:

1. $|\tilde{V}|=|V|-2n$. (Since we assumed the existence of a matching of cardinality n, $|V|-2n$ is nonnegative).
2. $\tilde{E}$ contains an edge between every auxiliary node and every original one.
3. Let $C'' \stackrel{\Delta}{=} \max_{E1}[C2(e)]$ and define

$$C3(e) = \begin{cases} C2(e)+nC'' & e \in E1 \\ (n+1)C'' & e \in \tilde{E} \end{cases}$$

Note that the cost of an auxiliary edge is higher than that of any other edge.

We will show that a maximal weight matching of $G3$ will solve our minimization problem. To do so we first prove the following lemma.

Lemma 1: Any maximal weight matching of $G3$ contains exactly n edges of $E1$.

Proof: First, we show that all nodes of $\tilde{V}$ appear in any maximal weight matching of $G3$. Let $M3$ be a maximal weight matching of $G3$, let $(v,w) \in E1$ be an edge in $M3$, and assume $\tilde{v} \in \tilde{V}$ is a node not appearing in $M3$. We construct a set of edges M' containing the edges of $M3$ with (v,w) replaced by $(v,\tilde{v})$. M' is clearly a matching. The cost associated with M' is

$$\sum_{e \in M'} C3(e) = \sum_{e \in M3} C3(e) - C3([v,w]) + C3([v,\tilde{v}]) > \sum_{e \in M3} C3(e)$$

contradicting the assumption that $M3$ is a maximum weight matching. Thus, all nodes of $\tilde{V}$ appear in $M3$. This eliminates $|V|-2n$ nodes of V, leaving $2n$ nodes of V from which at most n edges of $E1$ can be constructed.

Let us separate $M3$ into $M3' \subset E1$ and $\tilde{M} \subset \tilde{E}$, i.e., $M3 = M3' \cup \tilde{M}$. The above argument indicates that $|M3'| \le n$. Let us assume that $|M3'| < n$ (or $|M3'| \le n-1$). The total weight of the matching $M3$ is

$$\begin{aligned} W(M3) &= \sum_{e \in M3} C3(e) = \sum_{e \in \tilde{M}} C3(e) + \sum_{e \in M3'} C3(e) = \\ &= |\tilde{V}|(n+1)C'' + |M3'|nC'' + \sum_{e \in M3'} C2(e) < \\ &< |\tilde{V}|(n+1)C'' + |M3'|nC'' + |M3'|C'' \le \\ &\le |\tilde{V}|(n+1)C'' + n^2C'' - C'' \end{aligned}$$

We now construct a matching M in the following way. Let M' be any matching from $G1$ of cardinality n (such a matching does exist). Let $\tilde{M}$ be the set of edges connecting each node of V not appearing in M' with some node of $\tilde{V}$. Clearly $M = M' \cup \tilde{M}$ is a matching, $|\tilde{M}| = |\tilde{V}|$ and $|M'| = n$. The weight of M is

$$\begin{aligned} W(M) &= \sum_{e \in M} C3(e) = \sum_{e \in \tilde{M}} C3(e) + \sum_{e \in M'} C3(e) = \\ &= |\tilde{V}|(n+1)C'' + |M'|nC'' + \sum_{e \in M'} C2(e) = \\ &= |\tilde{V}|(n+1)C'' + n^2C'' + \sum_{e \in M'} C2(e) > |\tilde{V}|(n+1)C'' + n^2C'' > \\ &> |\tilde{V}|(n+1)C'' + n^2C'' - C'' > W(M3) \end{aligned}$$

Contradicting the assumption that $M3$ is a maximal matching, and thus $|M3'| = n$ □

(Note that, in the case described here a simpler proof could be constructed using the full connectivity of the nodes in V. However, we also make use of this result in the next section where the full connectivity does not hold.)

Algorithm A1

1. Construct the distance matrix of G [5,6].
2. Construct $G3$ through the construction of $G1$ and $G2$
3. Execute a maximum weight matching algorithm on $G3$ (see for example[7]); let $M3$ be the resultant matching.
4. For every $[v,w] \in M3 \cap E1$, attach a subscriber to nodes v and w creating a dual-homed attachment.

Theorem 1: Algorithm A1 solves the single-port minimum subscriber-to-network attachment problem.

Proof: Follows directly from Lemma 1, along with the construction rules of the graphs.

□

B. The Multiple-Port Case

This case is dealt in much the same way as the previous case. We introduce a set of transformations to result in a graph on which a maximum weight matching will be executed. Because of the similarity to the single-port case, this will be presented in a more condensed form.

Construct a graph $G4(V4,E4,C4)$ as follows:

- Nodes. The set of nodes $V4$ consists of the union of two sets $V*$ and $\tilde{V}$. $V*$ is obtained by splitting every node $v \in V$ into $p(v)$ nodes (the nodes resulting from the split of some node v are called its descendents). The auxiliary set of nodes $\tilde{V}$ contains $|V*| - 2n$ nodes.
- Links. The set of links $E4$ consists of the union of two sets $E*$ and $\tilde{E}$. $E*$ includes links from every descendent of nodes in V to every descendent of every other node (i.e., $E*$ is the set of edges for a complete graph with the edges among descendents of the same nodes removed). The set $\tilde{E}$ contains the edges from every node in $\tilde{V}$ to every node in $V*$.
- Cost Function. Define, as before, for all nodes in V $C1([v,w]) \triangleq CSTN([v,w])$. The cost function $C4$ is obtained from $C1$ in the same way $C3$ is obtained from $C1$ with the set $E1$ replaced by $E*$.

We note, again, that selecting an edge from $E*$ is interpreted as attaching a subscriber to the parents of the nodes at the ends of the edge. We thus offer the following algorithm as the solution to our problem.

Algorithm A2

1. Construct the distance matrix of G.
2. Construct $G4$ according to the above description.
3. Execute a maximum weight matching algorithm on $G4$; let $M4$ be the resultant matching.
4. For every $[v,w] \in M4 \cap E^*$, attach a subscriber to the parents of v and w creating a dual-homed attachment.

Theorem 2: Algorithm A2 solves the multiple-port minimum subscriber-to-network attachment problem.

Proof: A matching, by definition, will select every node at most once; thus, every descendent of a node v will be selected at most once and consequently v will be selected at most $p(v)$ times fulfilling the port index constraint.

It can be shown, following the line of proof of the previous subsection, that Lemma 1 holds for this case as well. Hence, the theorem is proven. □

C. Optimal Attachment for a Mixed Set of Subscribers

Consider a situation in which the set of subscribers to be attached is divided into two--some that need be dual-homed while the rest are regular, single-homed, subscribers and we ask the same question namely, how to best attach this set of subscribers. We show below an algorithm for the single-port case; extension to the multiport case can be done much like in the previous subsection.

Let us consider n dual-homed subscribers and k single homed subscribers. We define $G1$ and $G2$ as before and construct a new graph $G5(V5,E5,C5)$ as follows. The nodes of $G2$ are augmented by two sets of nodes $\tilde{V}$ and $\hat{V}$ so that $|\hat{V}| = k$ and $|\tilde{V}| = |V| - 2n - k$. The edges of $E5$ are those of $E1$ augmented with the set $\hat{E}$ and $\tilde{E}$ containing, respectively, an edge between every node in $\hat{V}$ and $\tilde{V}$ with every node in V. The cost function $C5$ is defined as

$$C5(e) = \begin{cases} C2(e) + nC'' & e \in E1 \\ (n+1)C'' + \sum_{u \in V} d(u,v) & [v,\hat{w}] \in \hat{E} \\ (n+1)C'' & e \in \tilde{E} \end{cases}$$

We now execute a maximum weight matching algorithm on $G5$ and examine the result. With the aid of a lemma similar to Lemma 1 above, one can show that this matching includes exactly n edges of $E1$, k edges of $\hat{E}$, and $|V| - 2n - k$ edges from $\tilde{E}$. An edge from $E1$ in the matching is interpreted as an attachment for a dual homed subscriber while an edge from $\hat{E}$ as an attachment for a single homed one, resulting in the desired minimum cost attachment for our mixed set of subscribers (the edges from $\tilde{E}$ are ignored).

D. Extension to Related Problems

Several slight modifications can be made to the algorithms above to adapt them to different circumstances. One might want to assign an extra cost to specific attachments so that, for example, an attachment to two nodes geographically remote from one another is discouraged. This can be achieved easily by defining the cost as

$$CSTN(A_i) = \sum_{v \in V} MHd(v, A_i) + AC(A_i) \ .$$

Where $AC(A_i)$ is the additional attachment-specific cost.

As an extreme case of the above, one can forbid certain attachments (e.g., between nodes having different security levels). This can be done either by assigning an extremely high value to the corresponding AC or, better yet, exclude these edges from $G1$ altogether.

In many design cases the location of both the network nodes and the subscribers are known; every subscriber is required to be attached to the node nearest it while we are free to choose the second node of the attachment (we refer to this as the "anchored-subscriber" problem). To solve this problem we again modify $G1$ to include only edges between nodes one of which is an anchor point for some node and the other is free. The resulting problem will be of a much smaller order than the unconstrained one.

IV. HIGHER-ORDER MULTIHOMING

In the previous section we presented efficient algorithms for minimal cost subscriber attachment for dual-homed subscribers. Unfortunately, these are the only cases in which polynomial algorithms exist. In this section we show, by standard means of reduction, that higher order multihoming problems (e.g., triple-homing, etc.) are all NP-hard (in the next section we show that even other interesting dual-homing problems are complex).

We address the problem of minimizing the subscriber-to-network attachment cost in a single-port m-multihoming configuration. We define an attachment as an unordered m-tuplet of distinct nodes and extend the definitions of MHd and CSTN in the straightforward way.

If we consider an attachment as a subset of order m of V and the CSTN of that attachment as the cost of that subset, then our minimization problem can be stated as:

Problem MH: Given a set V and a cost function $C: M \rightarrow R^+$ where M is the collection of all subsets of order m of V (i.e., for each $V' \subseteq V$ $|V'| = m$ there is a positive cost $C(V')$). Find a collection A of n mutually disjoint subsets of V, each of order m, such that the sum of their costs is minimal.

To show that this minimization problem is NP-hard we state the following four problems:

Problem MH1: Same as Problem MH except we only ask whether the collection A exists and require that the sum of the costs should be no more than some given upper bound B.

Problem MH2: Same as Problem MH1 except that B is a lower bound.

Problem MH3: Same as Problem MH2 except A is not restricted to n subsets.

Problem MH4: Given a collection S of finite sets, each of size m and a natural number $k \leq |S|$. Does S contain at least k mutually disjoint sets?

Lemma 2: Problem MH is NP-hard.

Proof: Problem MH4 is a version of the set packing problem which is known to be NP-complete For $m \geq 3$[8] from which we show that Problems MH1, MH2, and MH3 are all NP-hard leading to the conclusion that Problem MH is NP-hard as well.

To show that for $m \geq 3$ Problem MH3 is NP-hard transform an instance of Problem MH4 into an instance of Problem MH3 as follows: Let V be the set of all elements that appear in the sets of S and let the cost of each subset of V of order m be 1 if it appears in S and 0 otherwise. Solving MH3 for $B = k$ solves MH4.

If we now let $n = \lfloor |V|/m \rfloor$, i.e., *n* is the maximal number of disjoint subsets of size *m* of *V*, then a solution of Problem MH2 for this instance solves Problem MH3, meaning that MH2 is also NP-hard. Finally, for each $V' \in M$ let $C^*(V') = \max_{V'' \in M} [C(V'')] - C(V')$; solving Problem MH1 for C^* solves Problem MH2 for *C* meaning that MH1 is NP-hard as well, leading to the conclusion that MH is NP-hard. □

The above shows that even the simplest multihoming problem--minimizing subscriber-to-network attachment cost--is NP-hard for configurations other than dual-homing.

Given that the problem is complex we approach its solution by means of special cases of mathematical programming. Let us number the nodes of the network sequentially (and arbitrarily) $1, 2, \ldots, |V|$, and similarly number all possible attachments $1, 2, \ldots, \binom{|V|}{m}$ (where *m* is the degree of multihoming). Define an incidence matrix whose general element a_{ij} equals 1 if node *i* appears in attachment A_j and 0 otherwise. Also, define an attachment indicator x_i so that its value is 1 if a subscriber is attached according to attachment A_i and 0 otherwise. Clearly, the collection of x_i's uniquely determines the configuration. Finally, define a cost c_i of attachment A_i as the *CSTN* of the *m*-tuplet representing the attachment.

Two constraints must be imposed: the total number of attachments chosen must equal *n* and, the port index of every node must not be exceeded. Formally, therefore, we have the following problem:

Minimize: $\sum_i c_i x_i$

Subject to: $\sum_j x_j = n$

$\sum_j a_{ij} x_j \le p(i) \quad i = 1, 2, \cdots$

This is a simple linear 0-1 programming problem for which traditional, fairly efficient, algorithms do exist[9,10].

V. OTHER MULTIHOMING OPTIMIZATION CRITERIA

A. Nonhomogeneous Subscribers

In all previous multihoming problems subscribers were not distinguished from one another. This is not the typical case since it may be desirable to incorporate the identity of the subscriber in the attaching decision. For example, connecting a subscriber to nodes that are remote from it is costlier than connecting it to an attachment whose nodes are close by. In this subsection, we formulate a problem in which the subscriber and the attachment are considered simultaneously.

Let C_{ik} denote the cost of attaching subscriber s_k to attachment A_i. This cost may include $CSTN(A_i)$ and the inherent costs of the attachment itself. For example, C_{ik} might be the sum of $CSTN(A_i)$ and the connection cost of the subscriber s_k to both nodes of A_i. Also, if subscriber s_k has a restricted "community of interest" then the $CSTN(A_i)$ portion of C_{ik} will only include the costs to those nodes with which s_k communicates. Let $x_{ik} \in \{0,1\}$ be a variable indicating whether or not subscriber s_k is attached according to attachment A_i, and let the incidence matrix be defined as before. We are thus faced with the following problem:

Minimize: $\sum_{i,k} C_{ik} x_{ik}$

Subject to: $\sum_{i} x_{ik} = 1 \quad 1 \le k \le n$

$$\sum_{k} \sum_{i} a_{ji} x_{ik} \le p(j) \qquad j = 1, 2, \cdots .$$

where the first constraint ensures that every subscriber is attached exactly once, and the second is the port-index constraint.

Lemma 3: The nonhomogeneous subscriber problem is NP-hard.

Proof: By reduction from the three-dimensional matching problem (which is NP-complete). Consider three sets of (not necessarily distinct) elements *U*, *W*, and *Y* each of cardinality *q*, and a set of triplets $T \subseteq U \times W \times Y$. The 3D-matching is the problem of determining the existence of T', a subset of *T* of cardinality *q*, such that no two triplets agree in any of their components.

We transform this problem into an instance of the nonhomogeneous dual-homing one by defining the set *V* as the set of elements that appear in the first or second place in any triplet of *T*, the set *S* as the set of elements appearing in the third place in any triplet of *T*, and $p(v) \equiv 1$. In addition, for an attachment $A_i = [u, w]$ and a subscriber $s_k = y$, C_{ik} is defined as 1 if $(u, w, y) \in T$, and some very large constant *K* otherwise.

With this definition a positive answer to the 3D-matching is equivalent to $|T| \geq q$ and the minimal attachment cost is less than *K*. □

Nevertheless, this is still a linear 0-1 integer programming problem and fairly large problem sizes can be efficiently solved using the appropriate techniques.

B. Minimizing Subscriber-to-Subscriber Attachment Costs

Since end-to-end data flow takes place between subscribers and not network nodes it might be interesting to consider the distances among the subscribers themselves. We therefore define the double multihomed distance (MHD) between two attachments $A_1 = [v_1, v_2]$ and $A_2 = [w_1, w_2]$ as the minimal distance between any of v_1, v_2 and any of w_1, w_2, or $MHD(A_1, A_2) = \min\{MHd(A_1, w_1), MHd(A_1, w_2)\}$ and define the subscriber-to-subscriber attachment cost of subscriber A_i as $CSTS(A_i) \stackrel{\Delta}{=} \sum_j MHD(A_i, A_j)$. The problem of minimizing subscriber-to-subscriber attachment cost is that of finding a configuration *A* such that its total cost $CSTS(A) \stackrel{\Delta}{=} \sum_{A_i \in A} CSTS(A_i)$ is minimized.

Unfortunately, this is also an NP-hard problem. To see this construct a graph $G6(V6, E6, C6)$ in the following way. For each pair of nodes $v_1, v_2 \in V$ there is a single node $\tilde{v} \in V6$ (interpreted as an attachment); we say that v_1 and v_2 correspond to $\tilde{v}$. The set of edges is constructed so that every two nodes $\tilde{u}, \tilde{v} \in V6$ for which all corresponding nodes are distinct, are connected. The cost of an edge is $C6([\tilde{u}, \tilde{v}]) = MHD([u_1, u_2], [v_1, v_2])$.

Minimizing the single-port subscriber-to-subscriber attachment cost is equivalent to finding a clique of order *n* (*n* nodes) in $G6$ such that the sum of its edges is minimal. The multiple port problem can similarly be defined as a minimum-weight clique problem. A formal NP-hardness proof can be easily constructed by reducing the integer programming problem presented next to the general clique problem.

We offer, as before, an integer programming solution to our problem. Let the incidence matrix and the attachment indicator x_i be defined as before, and consider the following minimization problem:

Minimize: $$\sum_i \sum_j x_i x_j MHD(A_i, A_j)$$

Subject to: $$\sum_j x_j = n$$

$$\sum_j a_{ij} x_j \leq p(i) \quad i = 1, 2, \cdots$$

Again, the set of x_i's that solve the above problem determine the optimal subscriber-to-subscriber attachment configuration.

The above is a simple form of a quadratic 0-1 program [10] (in fact, it belongs to the family of quadratic knapsack problems[11]). While in general the problem is nonpolynomial, several efficient algorithms exist for many cases. In particular, because $MHD(A_i, A_j) = MHD(A_j, A_i)$ the problem can be transformed into an equivalent quadratic 0-1 program with continuous relaxation[12] using relaxation techniques for its solution. Note also that some simpler problems such as that of anchored subscribers remain complex (although the size of the problem is dramatically reduced).

VI. CONCLUSION

In this paper we have formally defined the problem of multihoming from a topology design standpoint, and have discussed optimal solutions to various variations of a special case--dual homing--which is the most likely to occur.

We have shown that only a small set of the relevant multihoming problems are polynomial and showed that their solutions are based on solution to matching problems. As is typical in topology design problems, posing a slightly more complicated situation turns the problem from polynomial to NP-hard. Several such situations are described in the paper and their NP-hardness is proved by reduction.

Being a topology design problem multihoming can be naturally defined as an integer (linear or quadratic) programming problem, for which standard solution techniques are in abundance.

The ability to solve the multihoming problem, be it by a matching-based algorithm or by programming methods, is important since multihoming poses challenges beyond those of topological design. For example, the dependence of performance measures on multihoming, the design of efficient routing algorithms for a multihomed environment, controlling the flow (and finding maximal flows) in a multihomed networks to name but a few. In short, the advantages afforded by multihoming are subject for further research.

REFERENCES

1. J.M. McQuillan, ‘‘Enhanced Message Addressing Capabilities for Computer Networks,’’ *Proc. of the IEEE* **66**(11) pp. 1517-1527 (November 1978).

2. V.G. Cerf and P.T. Kirstein, ‘‘Issues in Packet-Network Interconnection,’’ *Proc. of the IEEE* **66**(11) pp. 1386-1408 (November 1978).

3. C.A. Sunshine, ‘‘Interconnection of Computer Networks,’’ *Computer Networks* **1**(3) pp. 175-195 (January 1979).

4. Z.S. Su and J.E. Mathis, ‘‘Internetwork Accommodation of Network Dynamics: Organizational Structure,’’ pp. 428-430 in *Proceeding of IEEE Infocom 85*, Washington, D.C. (March 1985).

5. W.D. Kelton and A.M. Law, ‘‘A Mean-Time Comparison of Algorithms for the All-Pairs Shortest-Path Problem with Arbitrary Arc Lengths,’’ *Networks* **8** pp. 97-106 (1978).

6. G. Yuval, ‘‘An Algorithm for Finding All Shortest Paths Using $N^{2.81}$ Infinite Precision Multiplications,’’ *Information Processing Letters* **4** pp. 155-156 (1976).

7. N. Christofides, *Graph Theory - An Algorithmic Approach,* Academic Press, London (1975).

8. M.R. Garey and D.S. Johnson, *Computers and Intractability,* W.H. Freeman and Company, San Francisco (1979).

9. R.S. Garfinkel and G.L. Nemhauser, *Integer Programming*, John Wiley, New York (1972).

10. H. Taha, *Integer Programming,* Academic Press (1975).

11. G. Gallo, P.L. Hammer, and B. Simeonme, ‘‘Quadratic Knapsack Problems,’’ Research Report 76-43, UNiversity of Waterloo, Waterloo, Canada (1976).

12. P. Hansen, ‘‘Methods of Nonlinear 0-1 Programming,’’ pp. 53-70 in *Annals of Discrete Mathematics*, (1979).