Optimal Distribution of a Hierarchy of Network Management Agents

Kiran R. Bhutani* Bilal Khan†
Center for Computational Science
Naval Research Laboratory, Washington D.C.
{bilal}@cmf.nrl.navy.mil
http://www.nrl.navy.mil/ccs/project/public/DC/web/

September 13, 2003

Abstract

In this paper, we consider several mathematical and algorithmic problems which arise naturally in the optimal deployment of modern network management systems. Specifically, we will consider the problem of minimizing the total communication costs within an architecture consisting of a *distributed hierarchy* of cooperating intelligent agents. We consider several communication cost models, and describe provably optimal schemes for distributing agents among machines in each of these models.

Key Words: Distributed Hierarchy, Network Management, Multi-agent System.

1 Introduction

A network management system (NMS) must be able to coordinate thousands of network devices, and *scale* to perform well as the number of devices increases. To achieve this goal, the designers of the Optiprism [9] NMS drew upon the compelling analogies between scalable distributed computer systems and distributed human organizations, as presented by Fox [5] and others. Indeed, the architecture of Optiprism has its roots in the theory of human organizational hierarchies, as developed in the works of J. R. Galbraith [6], Mount and Reiter [11], Radner [13], Patrick and Dewatripont [12].

Traditionally, network management software has been based on centralized paradigms using on SNMPv1 or SNMPv2c, or weakly distributed hierarchical paradigms using SNMPv2, RMON, CMIP, or CMIP derivatives such as TMN [10, p. 5]. These approaches are feasible in small networks since their communication costs grow prohibitively with

* Department of Mathematics, The Catholic University of America, Washington DC 20064
† Advanced Engineering Sciences ITT at the Center for Computational Science, Naval Research Laboratory, Washington DC 20375

the number of devices [14, p. 4]. In contrast, Optiprism's design draws upon hierarchical network control schemes [2] such as PNNI [1].

The Optiprism NMS is a hierarchy of (software) *managers*, each of which may act as a *supervisor* of several *subordinate* managers. State information about the network is aggregated and flows recursively upwards at each node of the hierarchy, facilitating decisions and actions which are then executed by recursive delegation of subtasks to subordinates [3, pp. 12]. In this way, decentralized information processing facilitates decentralized decision-making [15, pp. 1].

The processing of management requests—such as route selection, fault correlation, and the collection of performance statistics—present computational burdens that prohibit placing all managers on a machine. To address this issue, Optiprism managers behave like cooperating *mobile agents* [10, pp. 9] in a hierarchy that can be distributed across the set of machines that form the computational environment [7, pp. 1].

1.1 Browsers

In most NMS' the human network administrator interacts with the management system using a *browser* application. For scalability, however, such a browser should not communicate with *all* of the NMS' managers at any point in time. In Optiprism, the browser application is itself a leaf node in the management hierarchy. The browser acts as "consultant", changing its position within the hierarchy, and—following Galbraith's mechanistic model [6]—establishing *lateral relationships* with a small subset of the managers based on its present position in the tree. The set of *visible* managers for a given browser consists of: the browser's peers, together with the browser's supervisor's peers, its supervisor's supervisor's peers, and so on. Optiprism thus provides *scalable connectivity* between each browser and the managers that are logically nearby. It also provides a graceful degradation of resolution so that the browser (and hence the human network administrator) has high-resolution access to aspects of the network near the task at hand while maintaining a reasonable—albeit at lower-resolution—perspective on the bigger picture.

Each browsers logical position within the hierarchy can be changed by the administrator using it. Specifically, the browser can be either (i) *promoted*, causing it to become a peer of its current supervisor, or (ii) *demoted*, causing it to become subordinate to a manager that is presently its peer. This *logical movement* of the browser, causes the set of visible managers to change in a manner corresponding to (i) zooming out from and (ii) zooming into particular regions of the network. While managers can be moved between machines, a browsers physical location within the computational environment cannot be changed since it needs to display information on the administrator's machine, and so must reside there.

Adopting a mobile agent-based approach permits Optiprism to move the services (that are provided by managers) closer to their clients (the browsers). If this distribution is done well, one can reduce the system's latency due to network communication [4] [8], thereby yielding better *responsiveness and performance* in the NMS. In this paper, we

consider the problem of how best to distribute the managers across a set of machines so to optimize the responsiveness of browsers and the multi-agent NMS as a whole.

2 Mathematical Model

The computational environment in which the NMS resides is modeled as a set of machines M, connected by a set of links $L \subseteq M \times M$ on a network $G = (M, L)$. In this paper we assume that G is a complete graph.

We model the state of the NMS at any instant in time as a rooted tree $T = (V, E)$, with root node denoted by $r^* \in V$. Denote the leaves of T by $\ell(T) \subset V$. A fixed subset $U \subseteq \ell(T)$ of these leaves are identified as browser agents; the remaining nodes of the hierarchy $V \backslash U$ are manager agents. Each of the browser agents $U = \{u_1, u_2, \ldots, u_k\}$ is operated by a network administrator. We define $m_i \in M$ to be the machine at which the administrator operating browser u_i resides. The initial assignment of browsers to machines is then defined in terms of a map

$$\begin{aligned} g &: U \to M \text{ where} \\ g &: u_i \mapsto m_i \end{aligned}$$

We will assume for simplicity that g is onto, which is to say that there is at least one browser running on each machine. The computational environment G, the browser agents U, the set of manager agents $V \backslash U$, and the initial assignment of browsers g, will all be assumed to remain unchanged over time, In contrast, however, the structure of the tree T may change over time, because each browser may be logically moved by the administrator using it.

Our research begins with the observation, that while g specifies the location of the browsers $U \subseteq V$, it leaves open the question of where the managers $V \backslash U$ should be placed. More formally, we would like to extend g to a function:

$$f : V \to M$$

so that $f|U \equiv g$, and f is optimal in the sense that it minimizes total network communications.

2.1 Properties of Optimal Mappings

Let us begin by considering the behavior of f on just two NMS nodes $u \in U$, $v \in V$. If f determines that u should be assigned to machine $f(u) \equiv g(u)$ and v should be assigned to machine $f(v)$, then whenever u and v communicate, they must do so over the physical network G connecting machines $f(u)$ and $f(v)$. Each act of communication between u and v incurs cost proportional to the distance from $f(u)$ to $f(v)$ in G, which we denote $d_G(f(u), f(v))$.

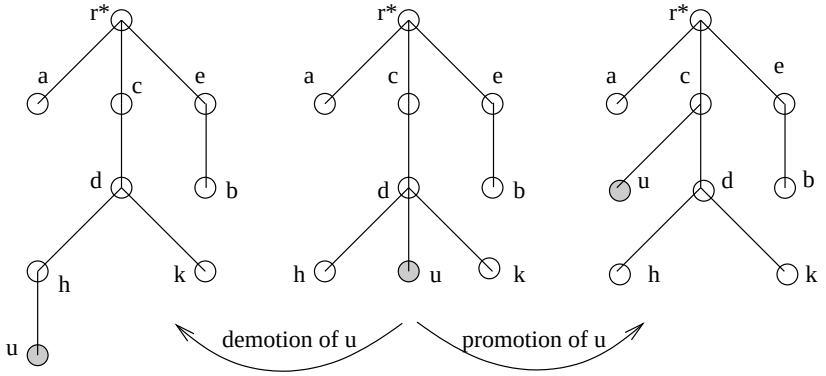
The question remains as to whether browser u and manager v may communicate at all. As noted earlier, for reasons of scalability, each browser $u \in U$ is only permitted to communicate with a restricted subset of so-called *visible* managers, defined as

$$Vis(u) \equiv \bigcup_{k=1}^{h(u)} Vis^k(u)$$

where for each $k \in \mathbb{N}$, $Vis^k(u)$—the k^{th} visibility of u— is the set of $v \in V \backslash U$ that are peers of the $(k-1)^{\text{th}}$ ancestor* of u, and $h(u)$ is the distance from u to the root of the tree.

Because information is aggregated as it flows upwards in the tree, the nodes in the kth visibility of u may be viewed as encapsulating state and control information that is $k-1$ resolution quanta coarser than the resolution of information held by u.

The table and figure below are an example of how the visibility structure of a browser u changes when a browser is promoted or demoted.



$Vis^1(u)$	$\emptyset$
$Vis^2(u)$	k
$Vis^3(u)$	$\emptyset$
$Vis^4(u)$	a, e

$\leftarrow$

$Vis^1(u)$	h, k
$Vis^2(u)$	$\emptyset$
$Vis^3(u)$	a, e
$Vis^4(u)$	$\emptyset$

$\rightarrow$

$Vis^1(u)$	d
$Vis^2(u)$	a, e
$Vis^3(u)$	$\emptyset$
$Vis^4(u)$	$\emptyset$

For simplicity, we exclude the possibility of multicast communication between browsers and managers, and consider only point-to-point communication. We may then define a *local cost function* $C_{v,u} : M \rightarrow \mathbb{R}^+$, for each $v \in V$ and $u \in U$, whose value $C_{v,u}(m)$ is

*By $(k)^{\text{th}}$ ancestor of u we mean the $(k+1)^{\text{th}}$ node on the path from u to r^*. For example, $Vis^1(u)$ consists simply of the peers of u, since the 0^{th} ancestor of u is u itself.

interpreted as the cost that would be incurred during pairwise communication between the nodes u and v if v were placed on machine vertex m in the network G, while u was placed at machine $f(u) \equiv g(u)$. Later we will propose candidate expressions for $C_{v,u}$ that are tailored to address particular concerns.

The family of local cost functions

$$\mathcal{C} = \{C_{v,u} \mid v \in V \backslash U, u \in U\}$$

can then be used to define the *global cost function* C_*, whose minimization would in turn be used as the criterion for finding a globally optimal f.

$$C_*(f, T, U, g) = \sum_{v \in V \backslash U} \sum_{u \in U} C_{v,u}(f(v))$$

2.2 The Problem

Given: a network management system consisting of a hierarchy of nodes $T = (V, E)$ distributed over a physical network of machines $G = (M, L)$, where the NMS contains both browsers agents $U \subseteq V$ whose physical location is given by the map $g : U \to M$ and manager agents $V \backslash U$ whose physical location is unknown.

NMS Communication Minimization Problem. *Determine the optimal physical locations for the all nodes of the NMS, including manager agents. In other words extend g to a function $f : V \to M$ in a manner that minimizes $C_*(f, T, U, g)$.*

3 Mathematical Preliminaries

We now define some functions which will be used to specify the local cost functions, and for describing the algorithm for constructing the optimal agent assignment function f.

For each $v \in V \backslash U$, let $q_v : U \to \{0, 1\}$ be defined as

$$q_v(u) \quad = \quad \begin{cases} 1 & \text{if } v \in Vis(u) \\ 0 & \text{if } v \notin Vis(u) \end{cases}$$

and let $p_v : U \to \mathbb{N}$ be given by

$$p_v(u) \quad = \quad \begin{cases} k & \text{if } v \in Vis^k(u) \\ \infty & \text{if } v \notin Vis^k(u) \text{ for any } k. \end{cases}$$

p_v and q_v are closely related to the visibility relation.

For each $v \in V$ the set of all browsers which can see v is called $q_v^* = \{w \in U \mid q_v(w) = 1\}$; of these the subset which see v with smallest visibility are denoted

$$p_v^* = \{w \in U \mid p_v(w) = min_{x \in U}(p_v(x)) \text{ and } p_v(w) \neq \infty\}$$

For each $u \in q_v^*$ we denote the set of all browsers that can see v *and* are mapped to the same machine as u by the initial mapping g as $q_v^*(g,u) = \{w \in q_v^* |\ g(w) = g(u)\} \subseteq q_v^*$. Analogously, for $u \in p_v^*$ the set of browsers that see v with smallest visibility *and* are mapped by g to the same machine as u, is denoted

$$p_v^*(g,u) = \{w \in p_v^* |\ g(w) = g(u)\} \subseteq p_v^*$$

Finally $q_v^{\oplus} = \{g(w) \in M |\ w \in q_v^* \text{ and all } g(w) \text{ distinct}\}$ is the set of all machines that have been assigned at least one browser from q_v^*, and the set of all machines that have been assigned at least one browser from p_v^* is

$$p_v^{\oplus} = \{g(w) \in M |\ w \in p_v^* \text{ and all } g(w) \text{ distinct}\}$$

4 Local Communication Cost Functions

We turn now to defining appropriate local cost functions (see subsection 2.1)—a necessary pre-requisite before an optimal f can be found. If the network G is a complete graph, then we might consider the BINARY local cost function below

$$C_{v,u}(m = f(v)) = \begin{cases} 0 & \text{if } m = g(u) \text{ or if } v \notin Vis(u) \\ 1 & \text{if } m \neq g(u) \text{ and } v \in Vis(u) \end{cases} \tag{1}$$

To see how such a cost function would affect C_* fix $v \in V \backslash U$. Clearly, if for all $u \in U$, $v \notin Vis(u)$ then f can map v to any machine $m \in M$ without affecting $C_*(f,T,U,g)$. If there exists a unique $u \in U$ for which $v \in Vis(u)$, then f must map v to the machine $f(u) \equiv g(u)$ in order to optimize $C_*(f,T,U,g)$. The remaining, interesting case occurs when the set of browsers observing v has size greater than 1, that is, $|q_v^*| > 1$.

In this case, the cumulative contribution to C_* by observers $u \in q_v^*$ can grow to be as large as the size of $|q_v^*|$. This biases the importance of $f(v)$ in C_*, so to counteract this bias, we consider the following UNIFORMLY BOUNDED local cost function

$$C_{v,u}(m) = \begin{cases} 0 & \text{if } m = g(u) \text{ or if } v \notin Vis(u) \\ \frac{1}{|q_v^*|} & \text{if } m \neq g(u) \text{ and } v \in Vis(u) \end{cases} \tag{2}$$

This new cost function ensures that the total contribution to C_* by browsers that can see v remains bounded by 1.

Since nodes in the kth visibility of a browser u are, in a sense, encapsulating state and control information that is $k-1$ resolution quanta coarser than the resolution of information held by peers of u, it might be desirable to emphasize the relative importance of mapping agents that are within smaller visiblity of u to $g(u)$. While the previous cost function does not do this, it could be modified to take visibility into account.
One such VISIBILITY-GRADED local cost function is

$$C_{v,u}(m) = \begin{cases} 0 \text{ if } m = f(v) = g(u) \text{ or if } v \notin Vis(u) & \\ \frac{1}{d^{((k(k+1)/2)-1}} & \text{if } p_v(u) = k \end{cases} \tag{3}$$

where d is the maximum degree in the tree T.

Observe that if $w \in p_v^*$ and $p_v(w) = k$, then there are at most

$$d(d-1) + d^2(d-1) + d^3(d-1) + ... + d^{k-1}(d-1) + d^k(d-2) = d(d^{k-1}-1) + d^k(d-2)$$

browsers which can see v with visibility $k+1$. These are the browsers at depth $d_T(w, r^*)+1$ in the tree, that is one level below the level of browser w. This VISIBILITY-GRADED local cost function has the property that the cost a manager pays to a single browser w in p_v^* where $p_v(w) = k$ is more than the total cost it pays to all possible browsers which can see v with visibility $k+1$.

This fact follows from the next lemma.

`Lemma 4.1.` *Let d be the maximum downwards degree of a vertex in the tree, and let $w \in p_v^*$. If $p_v(w) = k$, then $\frac{1}{d^{((k(k+1)/2)-1)}} > (d(d^{(k-1)}-1) + d^k(d-2))\frac{1}{d^{(((k+1)(k+2)/2)-1)}}$*

Proof. True since $d^{(((k+1)(k+2)/2)-1)-(k(k+1)/2-1)} \geqslant (d(d^{(k-1)}-1) + d^k(d-2))$

The left hand side simplifies to $d^{(k+1)}$ and the right hand side to $d^{k+1} - d^k - d$. Hence the result. □

Indeed the above lemma also proves that the cost an agent pays to a single browser w such that $p_v(w) = k$ is more than the total cost it pays to all browsers which can see v with visibility $k+i$ for $i \geq 1$ and $(k+i) \leq t$ where t is the depth of the tree.

5 Optimal Mappings

We now answer the problem of determining an assignment f of NMS nodes to machines that minimizes the global communication cost C_* for each of the local cost functions described in the previous section.

5.1 The BINARY and UNIFORMLY BOUNDED Case

We assign the NMS nodes to machines via $f : V \to M$ defined as follows:

$$f(v) = \begin{cases} g(v) & \text{if } v \in U \\ g(w) & \text{if } |q_v^*| = 1 \text{ or if } |q_v^{\oplus}| = 1, w \in q_v^* \\ g(w) & \text{if } |q_v^{\oplus}| > 1 \text{ and } |q_v^*(g,w)| = \max_{u \in q_v^*} |q_v^*(g,u)| \\ m_1 & \text{if } q_v^* = \emptyset \end{cases}$$

where m_1 is a fixed machine. If there is more than one vertex w that satisfies the conditions, then we simply use the leftmost vertex among all possibilities—which w is used does not effect the optimality of the assignment function.

The assignment f described above is an optimal solution to the NMS Communication Minimization Problem. To see this we prove the following:

`Lemma 5.1.` *Given any $t : V \to M$ such that $t|U = g$ and any $v \in V \backslash U$, define a function $t' : V \to M$ as:*

$$t'(w) \quad = \quad \begin{cases} t(w) & \text{if } w \neq v \\ f(v) & \text{otherwise} \end{cases}$$

Then $C_*(t', V, U, g) \leqslant C_*(t, V, U, g)$

Proof. The change in total cost $C_*(t, V, U, g)$ due to the new function t' occurs only when $v = w$. Thus the only term in the total cost that can change with the new mapping t' is the term $\sum_{u \in U} C_{v,u}(t(v))$. It is enough to show that

$$\sum_{u \in U} C_{v,u}(t'(v)) \leqslant \sum_{u \in U} C_{v,u}(t(v)) \tag{4}$$

There are two cases to consider,

(1) If $t(v) = f(v)$ or if $q_v^* = \emptyset$, then the (4) becomes an equality, so the total cost does not get worse with this change, that is, $C_*(t', V, U, g) \leqslant C_*(t, V, U, g)$

(2) Let $t(v) \neq f(v)$, and since in this case $q_v^* \neq \emptyset$, it follows either $|q_v^{\oplus}| = 1$ or $|q_v^{\oplus}| > 1$.

  (2.1) If $|q_v^{\oplus}| = 1$ and if $t(v)$ is a different machine that $q_v^{\oplus}$, then v pays a non-zero cost to all the browsers u which are in q_v^* and which are mapped to the machine $q_v^{\oplus}$, that is $C_{v,u}(t(v))$ is non-zero for all $u \in q_v^*$. Under the mapping t' an agent v pays no cost to browser u since v is assigned to the machine that already has all browsers u which can view v. Hence $C_*(t', V, U, g) < C_*(t, V, U, g)$.

  (2.2) If $|q_v^{\oplus}| > 1$ then there is more than one machine that has a browser which is in q_v^*. Since $t(v)$ is not the same machine as $f(v)$, it follows v pays non-zero cost to a larger subsets of $u \in q_v^*$ since $f(v)$ is the machine that has maximum number of browsers that can see v. On the other hand since $t'(v)$ is the same machine as $f(v)$ it follows that v pays non-zero cost to a smaller subset of q_v^*. Thus, all this shows that $\sum_{u \in U} C_{v,u}(t'(v)) < \sum_{u \in U} C_{v,u}(t(v))$

Hence the function f as define above indeed minimizes C_* for the case of BINARY and UNIFORMLY BOUNDED local cost functions. □

5.2 The VISIBILITY-GRADED Case

To describe the function f which minimizes C_* in the case of the VISIBILITY-GRADED local cost function, we need to introduce the *choice* function $* : V \backslash U \to U$. Intuitively, the choice function assigns to each manager v a corresponding browser $v^* \in U$ which most significantly influences the physical location of v.

5.2.1 The Choice Function

For each manager $v \in V \backslash U$, we will now (constructively) specify a browser $v^* \in U$, with the property defining $f(v)$ to be $g(v^*)$ is often likely to be a good choice: we call v^* the *choice* of v. Recall that p_v^* consists of browsers which can see manager v with smallest visibility, say k. Put $\beta_k^v \equiv |p_v^*|$ and $\alpha_{(v,j)}^w$ to be the number of browsers on the machine $g(w)$ which can see v with visibility j, for $k \leqslant j \leqslant h$. Clearly $\alpha_{(v,k)}^w \leqslant \beta_k^v$ for all $w \in p_v^*$. Then, for $w \in p_v^*$, the number of browsers on machine $g(w)$ which see v is

$$\alpha_{(v,k)}^w + \alpha_{(v,k+1)}^w + \alpha_{(v,k+2)}^w + ... + \alpha_{(v,h)}^w$$

where h is the depth of the tree. Put

$$|p_v^*(g,w)| = \stackrel{max}{u \in p_v^*} |p_v^*(g,u)| \tag{5}$$

There are two cases to consider:

(1) $p_v^{\oplus} = 1$. Then for all $w \in p_v^*$ we have $|p_v^*(g,w)| = \stackrel{max}{u \in p_v^*} |p_v^*(g,u)| = |p_v^*|$. We pick any $v^* \in p_v^*$ as the *choice* of v.

(2) $p_v^{\oplus} > 1$. That is, there are at least two distinct machines that have browsers from p_v^*. At this point again there are two cases:

(2.1) Either there is exactly one browser $v^* \in U$ such that $|p_v^*(g,v^*)| = \stackrel{max}{u \in p_v^*} |p_v^*(g,u)|$ in which case we call such a browser the *choice* of v.

(2.2) There are at least two browsers w_1 and w_2 in p_v^* such that $g(w_1) \neq g(w_2)$, but w_1 and w_2 satisfy:

$$|p_v^*(g,w_1)| = |p_v^*(g,w_2)| = \stackrel{max}{u \in p_v^*} |p_v^*(g,u)| \tag{6}$$

For such w_1 and $w_2 \in p_v^*$ we compare the two sequences of numbers:

$$\alpha_{(v,k)}^{w_1}, \alpha_{(v,k+1)}^{w_1}, \ldots, \alpha_{(v,h)}^{w_1}$$
$$\alpha_{(v,k)}^{w_2}, \alpha_{(v,k+1)}^{w_2}, \ldots, \alpha_{(v,h)}^{w_2}$$

Clearly the first term of each sequence is same, and so we compare second term onwards one by one until for the first j, $k \leqslant j \leqslant h$, $\alpha_{(v,j)}^{w_1} \neq \alpha_{(v,j)}^{w_2}$. WLOG we may assume $\alpha_{(v,j)}^{w_1} > \alpha_{(v,j)}^{w_2}$. At this point we call w_1 the *choice* of v. This argument can be generalized to the case where there are n browsers in p_v^* which satisfy condition (2) stated above. If the sequences of such numbers are not the same, the first w_i with higher value for $\alpha_{(v,j)}^{w_i}$ is declared the *choice* of v; otherwise the leftmost vertex is taken as the *choice*.

Having defined the choice function $*$, we can now define an assignment f of NMS nodes to machines for the case of a VISIBILITY-GRADED local cost function as follows:

$$h(v) = \begin{cases} g(v) & \text{if } v \in U \\ m_1 & \text{if } p_v^* = \emptyset \\ g(v^*) & \text{otherwise,} \end{cases}$$

where v^* is the *choice* of v as described in section 5.2.1, and m_1 is a fixed machine.

The following lemma shows that this extension function is optimal in the sense that it minimizes the total communication cost C_*.

`Lemma 5.2.` *Given any $t : V \to M$ such that $t|U = g$ and any $v \in V \backslash U$, define a function $t' : V \to M$ as:*

$$t'(w) \quad = \quad \begin{cases} t(w) & \text{if } w \neq v \\ h(v) & \text{otherwise} \end{cases}$$

Then $C_*(t', V, U, g) \leqslant C_*(t, V, U, g)$

Proof. The change in total cost $C_*(t, V, U, g)$ due to the new function t' occurs only when $v = w$. Thus the only term in the total cost that can change with the new mapping is the term $\sum_{u \in U} C_{v,u}(t(v))$. It is enough to show $\sum_{u \in U} C_{v,u}(t'(v)) \leqslant \sum_{u \in U} C_{v,u}(t(v))$

There are two cases to consider,

(1) If $t(v) = f(v)$ or if $p_v^* = \emptyset$, then the two sides of the above inequality are equal and so the total cost does not get worse with this change.

(2) Let $t(v) \neq f(v)$, and since in this case $p_v^* \neq \emptyset$, it follows either $|p_v^{\oplus}| = 1$ or $|p_v^{\oplus}| > 1$.

  (2.1) If $|p_v^{\oplus}| = 1$ and if $t(v)$ is a different machine than $p_v^{\oplus}$, then v pays non-zero cost to all browsers u which are in p_v^* and which are mapped to the machine $p_v^{\oplus}$, that is $C_{v,u}(t(v))$ is non-zero for all $u \in p_v^*$. Under the mapping t' the agent v pays no cost to any browser $u \in p_v^*$ since v is assigned to the same machine that already has all browsers that see v with smallest visibility. Hence in this case clearly $C_*(t', V, U, g) \leqslant C_*(t, V, U, g)$.

  (2.2) If $|p_v^{\oplus}| > 1$ then there is more than one machine that has a browser which is in p_v^*. Since $t(v)$ is not the same machine as $h(v) = g(v^*)$, it follows for some j, $k \leqslant j \leqslant h$, $\alpha_{(v,j)}^{g(w)^*} > \alpha_{(v,j)}^{t(v)}$. Thus v pays higher cost to browsers in $g(v^*)$ which can see v, since $g(v^*)$ has more browsers than $t(v)$. On the other hand since $t'(v)$ is the same machine as $h(v)$, it follows that v pays lower cost browsers on machine $t(v)$ (see Lemma 4.1) . We have therefore shown that $\sum_{u \in U} C_{v,u}(t'(v)) \leqslant \sum_{u \in U} C_{v,u}(t(v))$. Hence the function f indeed optimizes the VISIBILITY-GRADED local cost function.

□

6 Remarks on the Dynamic Case

It is interesting to consider the dynamic version of the problem, where the structure of the tree changes whenever an administrator promotes or demotes his/her browser in the NMS hierarchy. In such a setting, any browser u can see any agent v with 1 visibility by using a finite sequence of number of logical movements. The next lemma formalizes this.

`Proposition 6.1.` *Let $u \in U$ and $v \in V \backslash U$ be any two vertices in V whose distance within the tree is $d_T(u,v) = m$. Then there is a sequence of $\ell \leqslant m$ logical movements $\tau_1, \ldots, \tau_\ell$—where τ_i, $(i = 1 \ldots j)$ is either promotion or demotion of u—such that*

$$v \in Vis^1(\tau_\ell \cdots \tau_2 \tau_1 u)$$

Proof. By induction on m. Since $u \neq v$, $m \geqslant 1$.

Base case ($m = 1$). Since u is a browser, it is a leaf—hence u must be the subordinate of v. Then, taking τ_1 to be promotion of u, we get that $v \in Vis^1(\tau_1 u)$.

Inductive step. Suppose the lemma holds for all browsers u and managers v for which $d_T(u,v) \leqslant m-1$. Now let us consider $u \in U$ and $v \in V \backslash U$ for which $d_T(u,v) = m$. Let $u = v_0, v_1, v_2, \ldots, v_{m-1}, v_m = v$ be the sequence of vertices along the path from u to v. By inductive hypothesis, there is a sequence $\tau_1, \ldots, \tau_{\ell-1}$ (where $\ell - 1 \leqslant m-1$) of logical movements which satisfy: $v_{m-1} \in Vis^1(\tau_{\ell-1} \cdots \tau_2 \tau_1 u)$. Now $v_m = v$ is either the supervisor or the subordinate of v_{m-1}. If it is the supervisor, then we take τ_ℓ to be promotion; otherwise we take τ_ℓ to be demotion. The resulting sequence $\tau_1, \ldots, \tau_\ell$ satisfies $v \in Vis^1(\tau_\ell \cdots \tau_2 \tau_1 u)$. □

Suppose the NMS hierarchy T changes over time due to logical mobility by any of the browsers, and let $T_1, T_2, \ldots T_i, \ldots$ be the sequence of tree structures, and $f_1, f_2, \ldots, f_i, \ldots$ the corresponding sequence of assignments of NMS nodes to machines. Note that whenever $f_i \not\equiv f_{i+1}$, this means that some managers in the NMS hierarchy need to be physically moved at stage $i+1$. Specifically, for each $v \in V \backslash U$ satisfying $f_{i+1}(v) \neq f_i(v)$, manager v must be physically moved from machine $f_i(v)$ to machine $f_{i+1}(v)$.

As we showed in section 5, it is possible to compute an optimal assignment f_i given the tree T_i. In practice, recomputing f_i at each time step may not be feasible for two reasons (1) computing f is relatively expensive, taking $O(|V|^2)$ time, and (2) the costs of physically moving managers between machines when $f_{i+1} \not\equiv f_i$ may outweight the benefits of making the change.

We propose that the assignment function f should be recomputed only when the cost function increases by a predefined factor κ. Specifically, we recompute a new optimal f when

$$\frac{C_*(f_i, T_{(i+1)}, U, g)}{C_*(f_i, T_i, U, g)} \geqslant \kappa$$

Setting $\kappa = 1$ results in recomputing an optimal assignment at each step, while setting κ to larger values results in less frequent computation of new assignment functions. Thus

κ acts as a parameter which permits tunable tradeoff between the costs of computing good assignments over time, and the communication costs inherent in using sub-optimal assignments.

7 Conclusions and Future Work

In this paper, we have developed algorithms for distributing a hierarchy of network management agents in a manner which minimizes the network communication required between browsers and managers. We showed that for three different natural local cost functions, it is possible to compute a globally optimal assignment.

In this work, we considered the NMS agents to be distributed across a network of machines modelled as a complete graph, and charged unit cost for communication between any two distinct machines. In the future, we hope to consider the more general problem where the computational environment is modelled as an arbitrary graph whose edges are not uniformly weighted.

8 Acknowledgements

The first author thanks the Center for Computational Sciences at the Naval research Lab Washington D.C. for providing funding for this work during the summer of 2000. We would like to thank Dardo Kleiner and David Talmage for many spirited discussions about hierarchical network management and the Optiprism NMS.

References

[1] ATM Forum, Private Network-Network Interface Specification, version 1.0, 1996.

[2] A. Battou, K. Bhutani, and B. Khan. "Two approaches for aggregation of peer group topology in hierarchical PNNI networks". *International Journal of Intelligent Automation and Soft Computing*, 6(2), 2000.

[3] M. M. Cheikhrouhou, P. Conti, and J. Labetoulle. "Intelligent agents in network management, a state-of-the-art". *Networking and Information Systems*, 1(1):9–38, 1998.

[4] D. Chess, B. Grosof, C. Harrison, D. Levine, C. Parris, and G. Tsudik. "Itinerant Agents for Mobile Computing". *IEEE Personal Communications*, 2(5):34–49, 1995.

[5] M. S. Fox. "An organizational view of distributed systems". *IEEE Transactions on Systems, Man, and Cybernetics*, 11(1):70–80, 1981.

[6] J. R. Galbraith. *Organization Design*. Addison-Wesley Publishing Company, 1977.

[7] C. Ghezzi and G. Vigna. "Mobile code paradigms and technologies: A case study". In *Proceedings of the First International Workshop on Mobile Agents*, Berlin, Germany, 1997.

[8] L. Hurst, P. Cunningham, and F. Sommers. "Mobile agents — smart messages". In *Proceedings of the 1st International Workshop on Mobile Agents*, Berlin, Germany, 1997.

[9] B. Khan, D. D. Kleiner, and D. Talmage. "Towards an agent-based distributed hierarchical network management system for all-optical networks". In *Proceedings of 3rd International Workshop on Mobile Agents in Telecomunications Applications*, Montreal, Canada, August 15-18, 2001.

[10] J.-P. Martin-Flatin and S. Znaty. "A simple typology of distributed network management paradigms". In *8th IFIP/IEEE Int. Workshop on Distributed Systems: Operations & Management (DSOM'97)*, 1997.

[11] K. Mount and S. Reiter. "A model of computing with human agents". Technical Report 890, The Center for Mathematical Studies in Economics and Management Science, Northwestern University, Evanston, IL, 1990.

[12] B. Patrick and M. Dewatripont. "The firm as a communication network". *Quarterly Journal of Economics*, 109(4):809–839, 1994.

[13] R. Radner. "The organization of decentralized information processing". *Econometrica*, 61:5, 1993.

[14] M. G. Rubinstein and O. C. M. B. Duarte. "Evaluating tradeoffs of mobile agents in network management". *Networking and Information Systems*, 2(2):237–252, 1999.

[15] T. V. Zandt. "Real-time hierarchical resource allocation". Technical report, Kellogg Graduate School of Management, Northwestern University, 1997.