Resource Allocation Algorithms for Multimedia Multiparty Connections

C. Edward Chow

EAS-CS-92-1

University of Colorado at Colorado Springs
College of Engineering and Applied Sciences
Department of Computer Science
Colorado Springs, CO 80933-7150

January 1992

Resource Allocation Algorithms for Multimedia Multiparty Connections

C. Edward Chow[1]
Department of Computer Science
University of Colorado at Colorado Springs
Colorado Springs, CO 80933-7150
Phone: (719) 593-3110
Email: chow@zeppo.uccs.edu

Abstract

This paper addresses the problem of allocating resources to set up multimedia multipoint connections, while efficiently using network resources, in a network that includes signal converters to resolve the heterogeneity of customer/network equipments and information mixers for conference services. Specifically, given a connection request and a physical network, we consider the general resource allocation problem where the allocation costs of special resources and links are comparable so that neither can be neglected. The problem is shown to be NP-hard and the performance result of an optimal algorithm, several heuristic algorithms, and a distributed algorithm is presented.

Keywords: Resource Allocation, Multimedia, Multiparty Communication, Broadband Networks, Optimal Algorithm, Heuristic Algorithm, Distributed Algorithm.

1. Introduction

Rapid advances in switch and transmission technologies have made it possible to provide multimedia multiparty communication services to customers with different physical interfaces. High speed switch fabrics with tremendous bandwidth capacity and multicast capability are being designed. Signal converters that resolve the physical interface differences are being designed and implemented. Conference circuits that merge several video/audio streams are available, although much work remains to be done on digital bridging for both audio and video. Networks offering audio/visual connection services will contain these communication equipments [Addeo87, Alban91, Arango92, Bussey89, Gelman90, Giaco90, Hardt91, Lee88, Root88, Sakata90, Turner86].

Unlike the traditional resource allocation problem which deals only with links or trunk groups allocation [Akse85, Weinrib91], to set up a multimedia multiparty connection among customers with different terminal equipments in a given network, it may be required to reserve special resources from the network, such as the signal converters or the conference bridging circuits, and to reserve bandwidth in the links of the network which provide the transport for the information streams transmitted among special network resources and customer premise equipments. In this paper, links, trunk groups, converters, and conference circuits are all considered network resources. The converters and conference circuits are referred as **special** resources. This work minimizes costs over **both** types of resources.

There may be many special resources in the network that can satisfy the requirement of a complex connection. The allocation cost will be one of the important factors for selecting a particular special resource rather than an equivalent one in a different location. The other factor for selecting a special resource is the cost of links among the selected special resources and out to the user access interface. In this paper we consider the general case of the Resource Allocation problem for multimedia Multiparty Connections, abbreviated as the **RAMC problem**, where the allocation cost of special

1. Some of this research was done while the author was with Bell Communications Research

resources and links are close and can not be neglected. The problem is difficult because the selection of one special resource for the connection will affect the cost of the links that connect this special resource to other required special resources, and therefore the selection decision for other special resources in the same connection. How to efficiently allocate special resources and links for a multimedia multiparty connection becomes a challenging research issue.

To set up or modify connections in this environment, the requests expressed in terms of the signalling protocol [Minzer 91] will first be analyzed by the call processing elements in the networks to determine resource requirements, and then required resources will be allocated according to an efficient resource allocation algorithm. To implement the salient feature of the new signalling system, which hides the heterogeneity of the networks and physical interfaces of the users, special resources such as signal converters or conference circuits may be added to the resource requirements during the resource analysis phase. The available special resources and the connecting links will then be located and reserved, and connections will be set up between the access interfaces of the involved parties and the reserved resources.

We propose a notation called **logical connection** to capture the resource requirements of a complex multimedia multiparty connection. It is based on an augmented directed graph. It serves as an intermediate language for the call processing and is used to encode the result of the resource analysis phase for a user-network signalling request. The result of the resource analysis phase contains additional information such as the location of the special resources and trunk groups used to support the connectivity. The notion of a **multicast edge** is introduced to enable the utilization of the multicast capability in broadband (or other) networks.

Let us consider the RAMC problem in three different circumstances. First is the case where all the resources and parties of a given logical connection are located in the same network. Although this case is the simplest one, the efficient allocation of resources could still be very difficult to achieve, since there may be many ways to allocate available special resources and the allocation sequence may affect the available bandwidth on the connecting links within the network. In the case where bandwidth is heavily utilized, some allocation sequence may not be able to find enough bandwidth to satisfy the connectivity requirement of a given logical connection. It will be shown that the problem is NP-hard and no general polynomial solution can be found.

The second case is that for which all the parties are located in the same network, but the network may run out of the specific types of special resources required for a logical connection and will have to reserve resources in other networks. This is called the **resource hunting problem** and can be solved partially by directory services [Oppe83] [ISO88] [Schwar91] which provides information about available resources in each network. However, in this case the problem of choosing a resource such that the sum of the allocation cost of the resource and the connection cost of the needed links between the requesting network and the requested network is minimal becomes involved.

The most general case is where the parties are located in different networks and available special resources may not exist in these networks. Figure 1 illustrates this RAMC problem.

A heuristic algorithm for solving the RMAC problem in the above case is proposed based on the following observations. It is observed that the selection of a signal converter whose location shortens the link connected to its high-bandwidth side will reduce the total node-to-node cost. It is also a good heuristic to select an information mixer (e.g. an audio bridge) such that the sum of its allocation cost and the cost of all its surrounding links is smallest. To compare the performance of the heuristic algorithm, an optimal algorithm is designed and implemented. Both algorithms are designed based on previous work on path finding algorithms [Chow91a].

A connection management protocol based on the above resource allocation algorithms was reported in [Chow91b] and was briefly described in Section 5. It integrates the management of special network resources with that of links and allocate a multipoint path is set up in a hierarchical fashion. Related works on the connection management protocol for setting up multipoint paths on high speed network with multicast switch fabrics have been reported [Buben91][Ong91]. They did not consider the protocol for managing special network resources and treat it as a separate layer. Their connection management protocols provide path management functionality similar to that of our path finding algorithms, and a resource allocation protocol for hierarchical networks. Ong in [Ong91] briefly touched the issue of handling incremental changes.

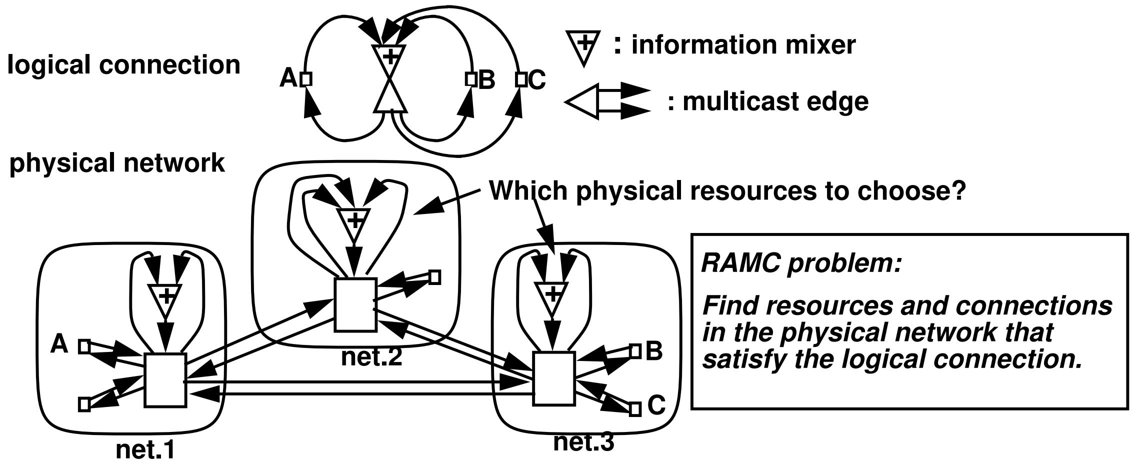

Figure 1. The resource allocation problem for setting up multiparty connections.

This paper addresses the issue of finding resources yielding low total cost for a logical connection. Optimization of the choice of network resources for a series of logical connection requests is an extension to this research. Previous works on trunk group selection [Akse85, Weinrib91] only dealt with circuit-switched networks and point-to-point connection requests, and has to be extended to handle the new dimensions of the RAMC problem, i.e., multicast connections, special network resources, and different granularity of bandwidth.

The remainder of this paper is organized as follows. Section 2 defines a logical connection. Section 3 describes the resource allocation algorithm for the case where all the parties and required resource are available in the same network. Section 4 presents the performance result of three resource allocations algorithms. Section 5 presents a distributed resource allocation algorithm based on hierarchical approach. In Section 6 we discuss the extension of these algorithms and related research issues, and Section 7 is the conclusion.

2. Specification of a Multiparty Connection

The resource requirement of a multiparty connection can be considered as an augmented directed graph, called **a logical connection**, where there are two basic types of vertices: the **end vertex** and the **resource vertex**, and two types of edges: the **point-to-point edge** that connects a source vertex to a destination vertex, and the **multicast edge** that connects one source vertex to multiple destination vertices. A vertex may contain one or more **port**s. There are two types of ports: **inputport** and **outputport**. Each of these ports connects its vertex to an edge. An inputport is a port where a vertex receives the information stream from the connecting edge. An outputport sends information stream out to the connecting edge. An end vertex represents a service access point for the user's equipment. A resource vertex represents the characteristics of a special resource. The edges capture the connectivity and bandwidth requirements among vertices.

The notion of a multicast edge is introduced to enable the utilization of the multicast capability, since switch fabrics may be capable of setting up multicast connections among its inputports and outputports.

In this paper, we assume that there are two types of resource vertices: information mixers and signal converters. Later, in Section 5, we discuss how to extend our results to include more special resource types. Information mixers can be further classified according to the number of information streams which they can merge and their medium type. A signal converter is a device that converts the signal entering from its input port to a different coding format and sends the resulting signal out through its output port. It can be further classified according to the two coding standards of information being converted. Each edge is associated with a bandwidth value.

We use the following notation: Vertex(p) is a function that returns the name of the vertex which contains port p. InputPorts(v) is a function that returns the set of inputports of vertex v. OutputPorts(v) is a function that returns the set of outputports of vertex *v*. A point-to-point edge can be identified as (*sp, dp*) where *sp* is an outputport of a source vertex and *dp* is an inputport of a destination vertex. A multicast edge can be abbreviated as (*sp, D*) where *sp* is an outputport of a vertex and *D* is the set of inputports that are connected to the multicast edge. In some cases, it can also be identified as (*s, DV*) where s is the source vertex and *DV* is the set of destination vertices.

B(*e*) is a function that returns the required bandwidth value of edge e. Type(*x*) is a function that returns the type of vertex *x*. Cost(*x*) is a function that returns the allocation cost of element *x* in a logical connection. A path can be represented as a list of edges, {*e1, e2,..., en*}. The cost of a path, *p*, is the sum of all the allocation costs of edges in *p*. A logical connection captures the connectivity requirement of a multiparty connection.

Figure 2 specifies in graphical notation a logical connection of a 3-way video conference among users with different terminal types: HDTV for user1, NTSC for users 2 and 3. RV1 is a resource vertex of signal converter type

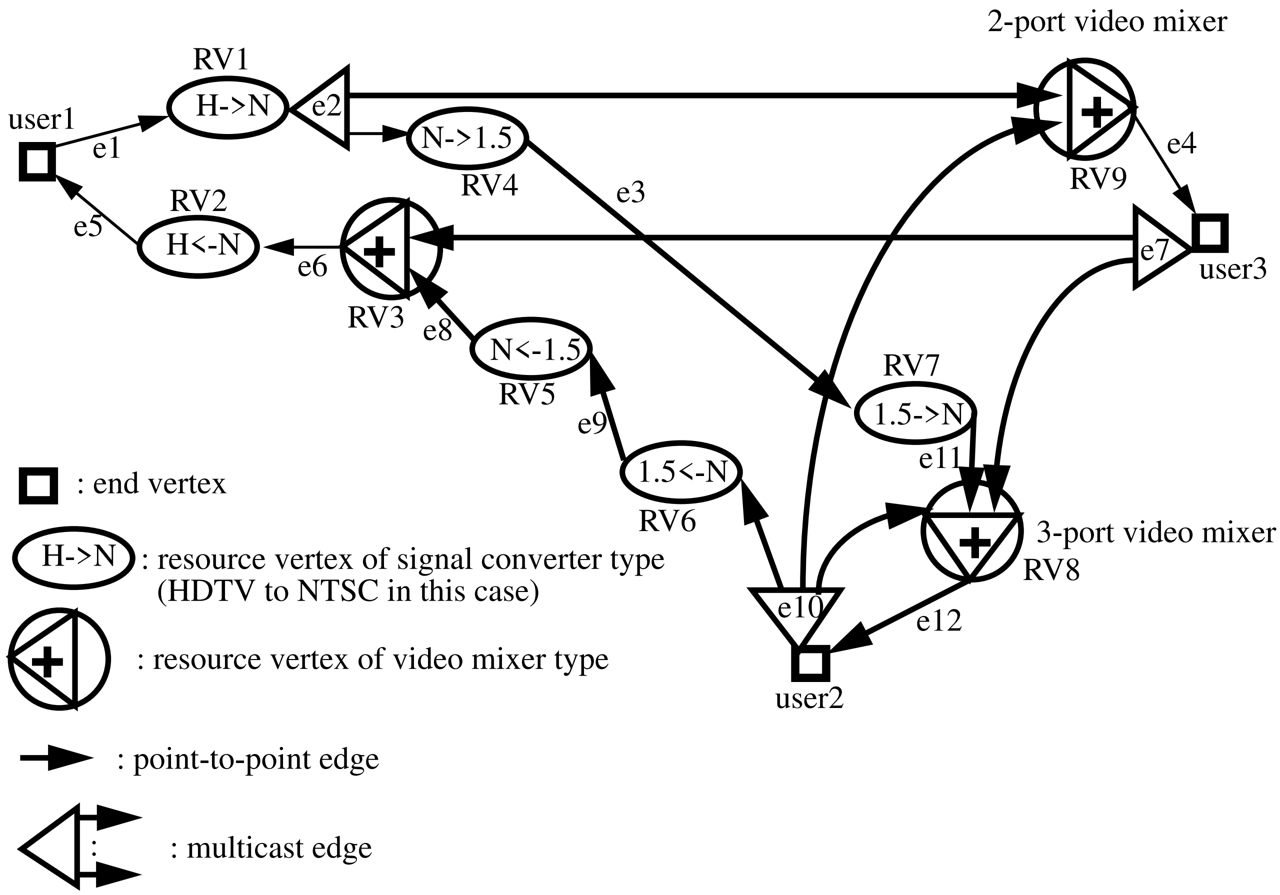

Figure 2. A logical connection of a 3-way video conference.

which converts the signal format of HDTV to that of NTSC. RV4 is a resource vertex of signal converter type which compresses the required bandwidth of NTSC signal to 1.5 Mb/s. RV7 is used to convert the compressed signal back to NTSC normal format. RV9 is a resource vertex of 2-port video mixer type, while RV8 is a 3-port video mixer. The source end of edge *e1* connects to the outputport of user1 and the destination end of e1 connects to the inputport of RV1. The source end of multicast edge *e2* is connected with the outputport of RV1. One of the destination ends of *e2* is connected with the inputport of RV4 and the other destination end of e2 is connected with one of the inputports of RV9. B(*e1*) = 120, B(*e2*)=44.5, and B(*e3*)=1.5 specify (from the presumed definitions of standard codecs) the required bandwidth values of edges *e1*, *e2* and *e3*.

Appendix 1 shows the message of a resource allocation request for setting up the logical connection in Figure 2.

3. Resource Allocation for Multiparty Connections

In this section we describe the optimal and heuristic algorithms that allocate resources in a physical network to satisfy the resource requirement specified in a given logical connection. This allocation can be thought of as a mapping from a logical to a physical connection graph. Assume that a network maintains up-to-date information about (1) its network topology in term of vertices and connecting edges, (2) the bandwidth allocation costs, (3) the available physical resources of the network and their associated allocation costs, and (4) the mapping function that maps the logical name of an end vertex in a logical connection to the corresponding physical vertex in the network.

The resource allocation (RAMC) problem is defined as follows:

> Given a network *N* and a logical connection *L*, for each resource vertex in *L*, allocate physical resource(s) in *N* with the same resource type, and for each edge in *L*, allocate the bandwidth in one or more edges in *N*, preserving the connectivity specified in *L*.

To facilitate the presentation, we use "physical" to refer to elements in *N* and "logical" for those in *L*. The apostrophe symbol is used to represent the allocation relationship. For example, for an edge *e* in *L, e'* represents the allocated physical edge(s) in *N* for *e*. Similarly, for a vertex *v* in *L, v'* represents the allocated physical vertices in *N* for vertex *v*. Note that the bandwidth in several physical edges of *N* may be allocated for a logical edge in *L*. The allocation cost of an edge in *L* is the sum of all the bandwidth allocation costs in the corresponding physical edges in *N*. The location of a physical resource affects the cost of edges connected to that resource. The allocation cost of an edge *e* is decided by the location of the two physical resources connected to edge *e'*. The allocation cost of *L* is the sum of all the allocation costs of edges and vertices in *L*.

Given the set of resource vertices in a logical connection, there are many possible ways to allocate the resource vertices and each may yield a different allocation cost value for *L*. Since the allocation cost of resource vertices and that of edges are interrelated, it is not clear which way will yield the minimum total cost.

Given a network *N* and a logical connection *L*, an **optimal algorithm for RAMC** finds the mapping of the list of resources in *L* to a list of corresponding physical resources in *N* such that the allocation cost of *L* is minimum.

Theorem 1: The general case of a RAMC problem is an NP-hard problem.

Proof: The degenerated case of a RAMC problem is that the logical connection contains no special resource and only consists of point-to-point edges or multicast edges. Since the resource allocation problem for a multicast edge is shown to be NP-hard [Chow91a], the general case of a RAMC problem is a NP-hard problem.

3.1 An Optimal Algorithm for RAMC Problems

A greedy optimal algorithm for a RAMC problem is to generate all the possible resource allocations to the special resources in the logical connection and then find the resource allocation which has minimum overall cost, i.e., cost of all special resources plus cost of the associated allocated links. The top level of a version of the greedy optimal algorithm for a RAMC problem called the ORAOP algorithm is presented as follows:

ORAOP algorithm:

```
VLoop* vl = new VLoop(lc, pn);
float minCost = 0.0;
float cost = 0.0;
Seq* in;
while (!vl->EndLoop()) {
```

```
        in = vl->GetIndex();
        cost = ComputeCost(in, lc, pn);
        if (cost < minCost) lc->ReplaceMin(cost);
        vl->AdvanceIndex();
    }
```

The algorithm is called ORAOP since it finds the optimal resource allocation for the special resources and uses an optimal multicast path finding algorithm to allocate multicast edges. The algorithm is written in object-oriented fashion, specifically in C++ notations. The lc is a pointer to the logical connection and pn is a pointer to the physical network. The VLoop object implements the nested loops with variable number of levels. For a logical connection of k special resource types, VLoop realizes a k-nested loops. The return value of GetIndex method is a sequence of numbers. Each number is the index of the particular loop. The sequence variable *in* is in fact pointing to the current resource allocation. The hardest part of the algorithm implementation is to generate the allocation patterns implied by the index number.

Theorem 2: The ORAOP algorithm is a non-polynomial algorithm.

Proof: Let N be the number of special resource types in the logical connection, let Li be the number of special resources of type i in the logical connection, Pi be the number of special resources of type i in the physical network. For special resource type i, there are $C(Pi, Li)$ ways to allocate resources. The number of possible resource allocations for the RAMC is $T = \prod_{i=1}^{N} C(Pi, Li)$, where $C(Pi, Li)$ denotes the number of Li-combinations of an Pi-set. Since each possible resource allocation needs to be examined for its cost, the loop that computes the cost and compares it with the current minimum cost will be executed T times. As the network size increases, the Pi increases and the $C(Pi, Li)$ grows exponentially.

3.2 Heuristic RAMC algorithms

The heuristic RAMC algorithms differ in two aspects: how each individual resource is allocated and what is the allocation order among the required resources in a logical connection. In this section the allocation rules for different types of resources are discussed. Section 3.3 discusses the resource allocation sequence.

Resource allocation rule for an end vertex

This is a simple name mapping function. The corresponding ID of an end vertex in a physical network can be obtained by looking over the network database.

Resource allocation rule for a vertex of signal converter type

For a signal converter in a logical connection, the simplest allocation rule, SCR1, is to allocate a signal converter of the compatible type with the lowest utilization cost in the physical network. A more complex rule, SCR2, is to allocate a signal converter of the compatible type and whose utilization cost together with the cost of the path to the connected port(s) of the high bandwidth end is minimum. Rule SCR1 can be executed very fast while Rule SCR2 involves the overhead of finding the cost of the shortest path for each possible candidate in the physical network.

Notice that the high bandwidth end of the signal converter may be connected to a multicast edge or a point-to-point edge. The resource allocation rule for signal converters also assumes that a back-to-back direct connection of the high bandwidth ports of two signal converters will not be generated by an intelligent resource analysis procedure.

Assume that we are to allocate special resource in network N for the signal converter SC of a logical connection L shown in Figure 3. The highbandwidth port of SC is vp and vp is connected to a multicast edge e. Assume that vertices A, B, C are allocated vertices and ap, bp, and cp are destination ports in e. By applying the above resource allocation rule, first three signal converters, *SC1', SC2', SC3'* are considered as candidates, since they are of the same resource type as

SC. We then find the efficient multicast path *e'* from port *vp'* of each candidate signal converter to ports *ap', bp'* and *cp'*. If the sum of the cost of *SC2'* and the cost of the corresponding multicast path *e'* is the lowest compared with those of *SC1'* and *SC3'*, *SC2'* is allocated for *SC*. Note that in Figure 3, we only show links connected to *SC2'* and omit other links in network *N*.

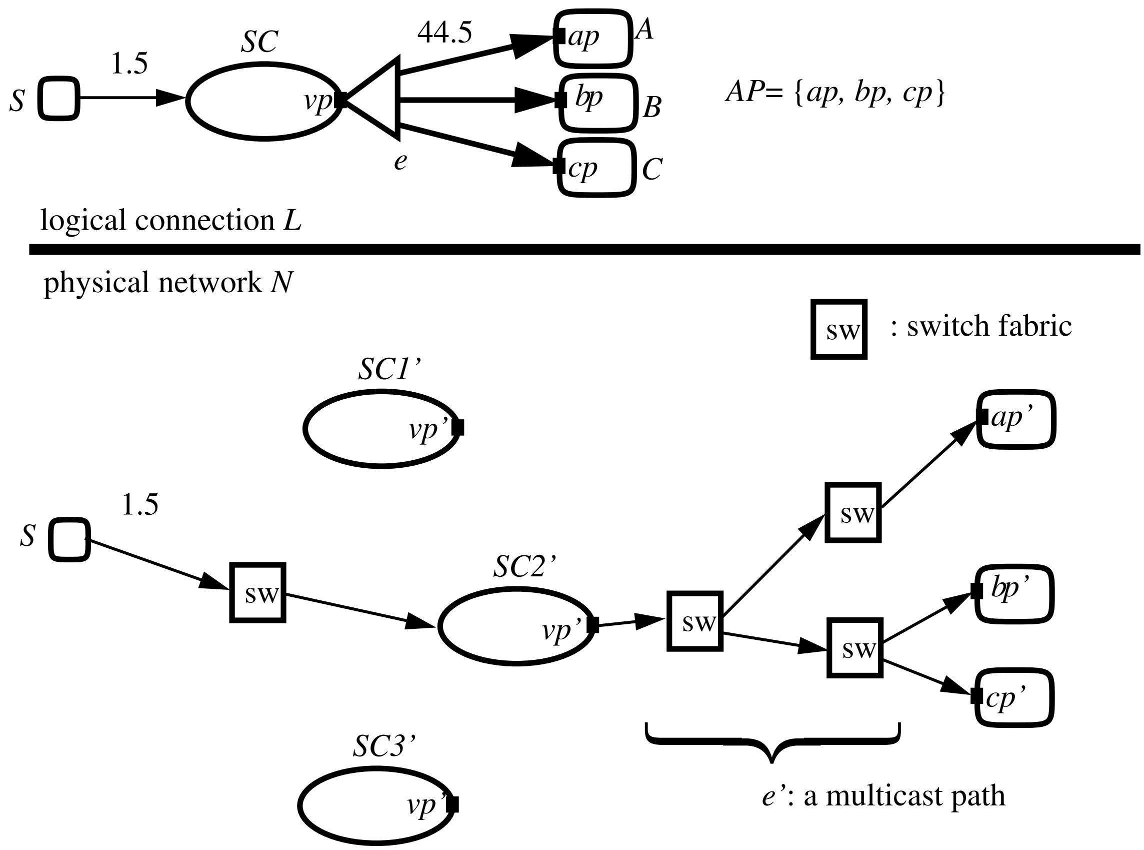

Figure 3. Resource allocation for a signal converter.

Resource allocation rule for a vertex of information mixer type

Similar to the resource allocation rules to those of signal converters, IMR1 is the rule that allocates the information mixer with the compatible type and with the lowest cost in the physical network. Rule IMR2 examines both the cost of information mixers and the cost of edges to the connected vertices and picks the one where the sum of the information mixer utilization cost and the cost of the paths to the connected vertices is minimum. Note that the connected paths can be either point-to-point path or multicast path. Even the inputport of a information mixer can be connected to one of the segments of a multicast edge. A detailed and rigorous specification of the above two rules can be found in [Chow91b].

Resource allocation rule for an edge

Allocate the shortest path in the physical network for this edge. Rule OP1 uses optimal path finding algorithm while HP1 uses heuristic path finding algorithm to find the corresponding path. Note that for a multicast edge, the execution time of the OP1 will grow exponentially when the network size and the number of the destination vertices of the multicast edge increase.

3.3 Resource Allocation Sequence

The allocation cost of a logical connection on a network depends not only on the selection criteria in the above resource allocation rules for resource vertices and edges, but also on the sequence imposed by the resource allocation algorithm on how elements in the logical connection are selected for the resource allocation. Here is one of the possible resource allocation sequences:

Step 1. Allocate all end vertices based on the mapping functions.

Step 2. Allocate all signal converters whose high bandwidth ends are directly connected to allocated vertices.

Step 3. Let *I* be the set of unallocated information mixers whose directly connected vertices are all allocated vertices.
If ($I \neq \emptyset$) then allocate each information mixer in I;
Let U be the set of unallocated information mixers
If ($U \neq \emptyset$) then {
allocate one of the remaining unallocated information mixers;
go to Step 2.}

Step 4. Allocate remaining unallocated edges.

If there is no resource available during the allocation of an element of the logical connection, the algorithm stops and fails. Intuitively, the algorithm divides the resource vertices in the logical connection into layers of resource vertices depend on how many edges it takes to connect to the closest end vertices. The resource allocation algorithm allocates physical resources layer by layer until all the resource vertices are allocated. For example, a possible allocation sequence for the resource vertices in the logical connection in Figure 2 is {RV1, RV2, RV6, RV9, RV3, RV4, RV5, RV8, RV7}. If RV1 is selected and allocated before RV4 is considered for allocation, then the allocation sequence will be {RV1, RV4, RV2, RV6, RV9, RV3, RV5, RV8, RV7}.

The above resource allocation sequence can be incorporated with different resource allocation rules and forms different resource allocation algorithms.

4. Performance of the Resource Allocation Algorithms

The resource allocation algorithms can be also be classified according to the type of algorithm used in the allocation of special resources and in the allocation of link bandwidth. The ORAOP class of RAMC algorithms finds the optimal special resource allocation and uses optimal path finding algorithm. The HRAOP class of RAMC algorithms applies heuristic rules such as those mentioned in Section 3.2 to allocate the special resources and uses an optimal path finding algorithm to find and allocate paths. The HRAHP class of RAMC algorithms applies heuristic rules/algorithms to find all the resources.

To evaluate the performance of the heuristic RAMC algorithms, we have designed and implemented several ORAOP, HRAOP, and HRAHP algorithms, and experiments were carried out to test versions of their implementations on several logical connections and physical network configurations. The computation time of these algorithms includes the reservation of resources on the physical network and the creation of data structures that records the allocation results. The algorithms are implemented in Gnu C++ and the test results are generated on SUN sparc2 workstations.

Figure 4 shows the computation time of the three algorithms for a logical connections on eight networks. Networks 3 to 9 contain 3 to 9 clusters of resources respectively. Each cluster consists of a multicast switch fabric and a list of special resources. The clusters are connected in two counter-rotating rings. Network 10 is a hierarchical network where three networks, each contains three clusters of resources, are fully connected. The computation time shows exponential increase for both the ORAOP and HRAOP algorithms over the network size.

To compare the effectiveness of the heuristic resource allocation rules, we have implemented four HRAHP algorithms. Algorithm X encodes Rules SCR1 and IMR1. Algorithm B uses Rules SCR1 and IMR2. Algorithm M applies

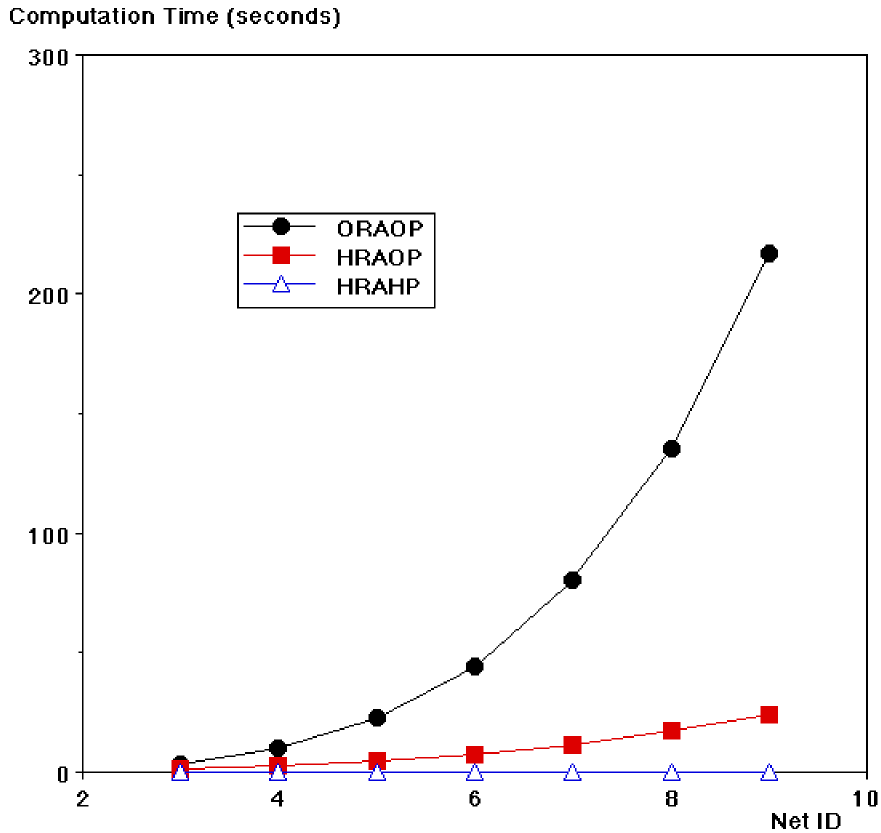

Figure 4. Peformance of three resource allocation algorithms.

Rules SCR2 and IMR1. Algorithm H utilizes Rules SCR2 and IMR2. Figure 5 shows the connection cost of the resource allocation result performed by these four algorithms for the logical connection, lc4, in Figure 2 on the aforementioned eight networks. In general we see consistent cost reduction by applying rules which trying to use additional information. However there are also rare exception cases, such as that of Algorithm M on Network 4.

Figure 6 shows the execution time of the resource allocation performed by the four algorithms on the same case. Here we did not see any exception case.

Figure 7 shows the computation time of the Algorithm X for four logical connections. Lc1 has one special resource, lc2 has three special resources, lc3 has two special resources, and lc4 has nine special resources. Figure 8 shows the graphical representations of logical connections lc1, lc2, and lc3. Lc4 is the same as the logical connection shown in Figure 2. Algorithms B and H which implement rule IMR2 was able to generate the optimal results for lc1 and lc2. Algorithm H was able to generate the optimal results for LC3. For lc4, the cost differences between those computed by the ORAOP algorithm and those by the HRAHP algorithms are between 8% to 60%. There seems to be enough room for improving the heuristic algorithms to get closer to the optimal performance.

5. A Distributed Resource Allocation Algorithm for RAMC problems

In this section we present and discuss the performance of a distributed resource allocation algorithm based on hierarchical approach for solving the RAMC problem in large networks. It can be applied to cases where parties involved in a logical connection are located at different networks. In this approach, the network elements are organized as a

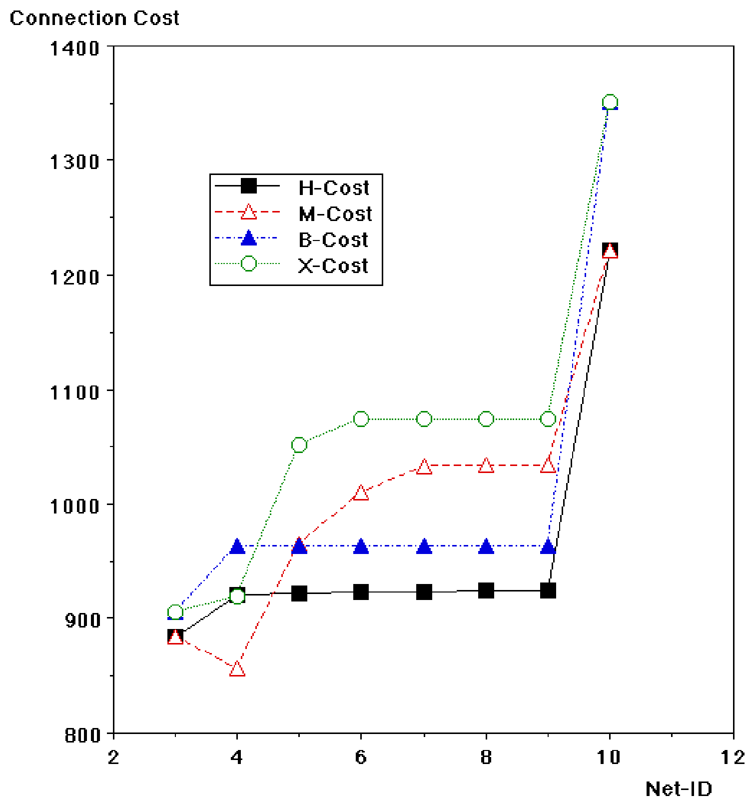

Figure 5. Connection cost computed by algorithms using different heuristics.

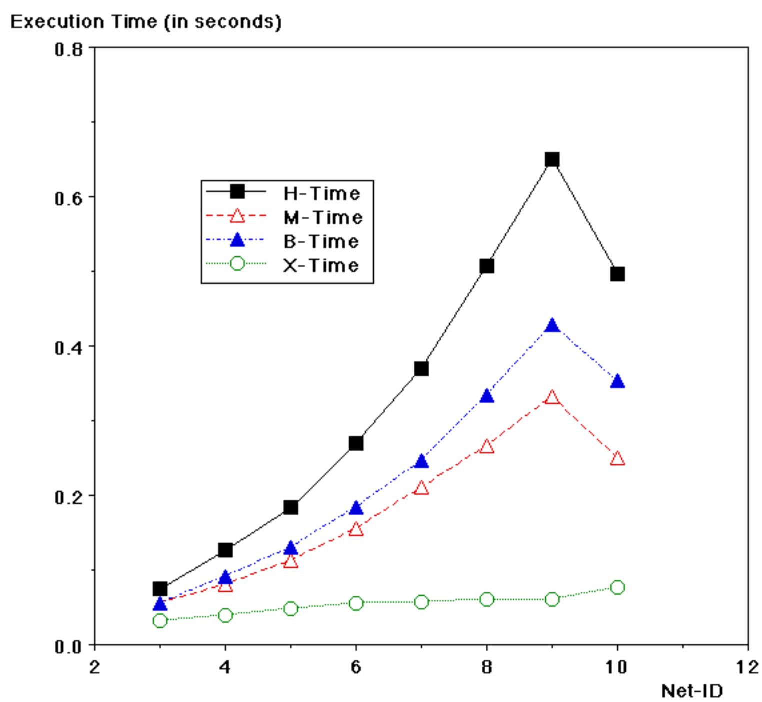

Figure 6. Execution time of the four heuristic algorithms.

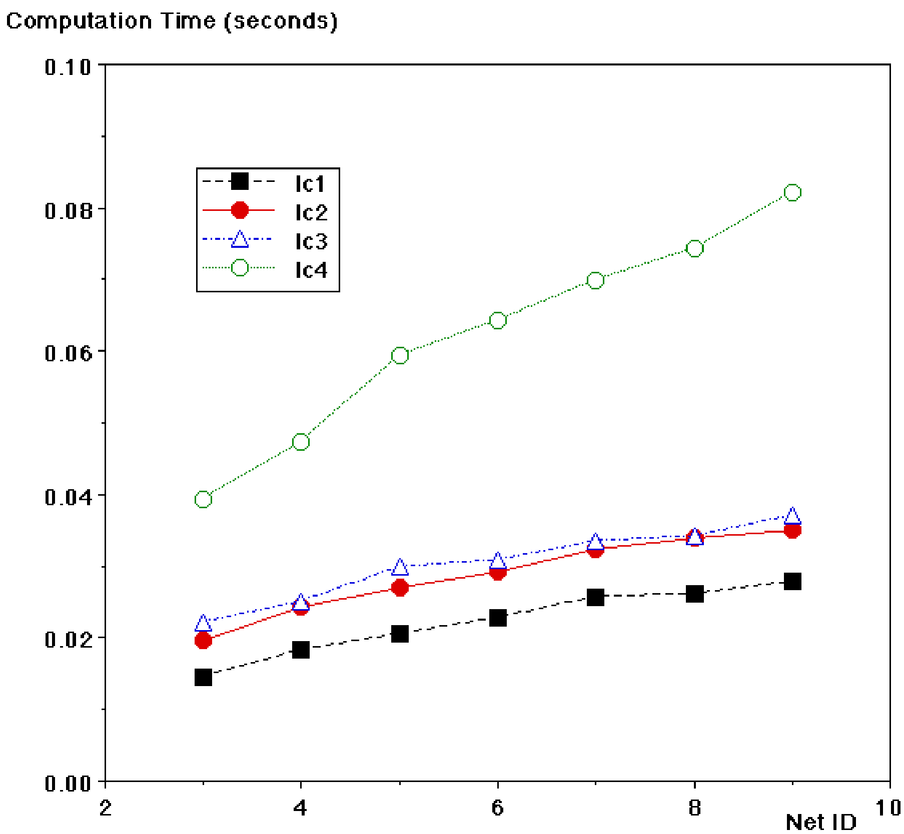

Figure 7. Performance of Algorithm X on 4 logical connections.

hierarchical network. A node in a level corresponds to a subnetwork in the level below. An edge in the higher level corresponds to edges that interconnect two subnetworks representing the corresponding two nodes connected by the edge. Each subnetwork in the hierarchical network is managed by a resource manager which allocates resources and bandwidth in the subnetwork.

After the initiation of each subnetwork, its resource manager sends a NetworkConfiguration message to its "parent" resource manager in the hierarchical level above. The message contains the list of special resources in the subnetwork and the available outgoing/incoming trunks to other subnetworks. On receiving a NetworkConfiguration message the resource manager updates the corresponding node's resource list and the inter-node trunk information, records the address information of the sending resource manager for future communication, and then relays the information upward to the hierarchy.

When a resource manager receives a local connection request, it first examines if all the end vertices are in the subnetwork. If all the end vertices are in the subnetwork, the resource manager checks whether there are enough resources to satisfy the request by executing one of the resource allocation algorithms mentioned in Section 3. If there are end vertices not in this subnetwork or the special resources do not exist, the request will be sent it to its parent resource manager. The resource allocation algorithms executed by the resource managers not in the bottom level are a little bit different than those mentioned in Section 3. Here the resource manager is trying to allocate special resources in the resource lists of its nodes and in fact to partition the logical connection into smaller logical connections so that the overall cost is minimum. Note that only those edges in the logical connection that are allocated with inter-node trunks will contribute to the connection cost. Therefore an estimated cost is used to allocate the resources and it is not an optimal algorithm. Again if the resource requirement can not be satisfied, the request will be relayed upward.

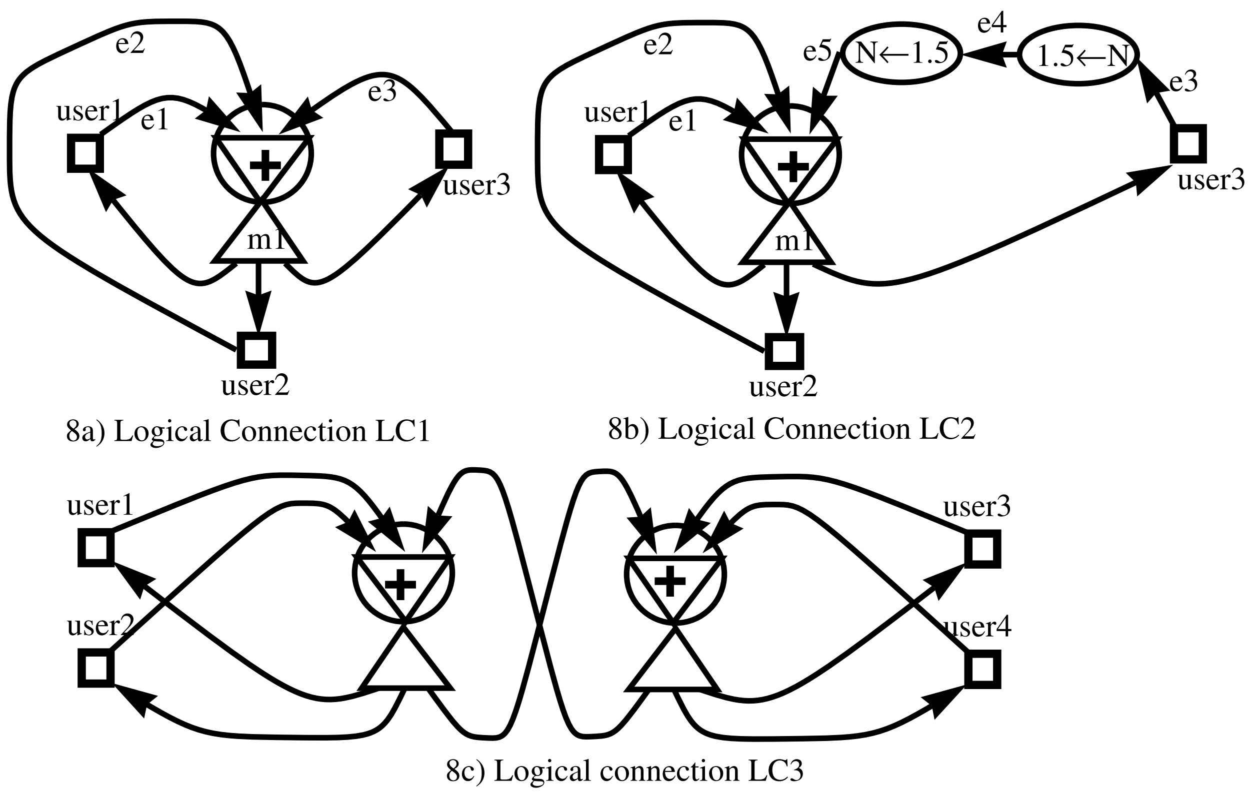

Figure 8. The three logical connections for the resource allocation experiments.

After partitioned the logical connection, the smaller connection requests will be sent to the corresponding resource manager below. A protocol similar to the two-phase locking is used to guarantee that either all the requests are allocated by the low level resource managers or they are all aborted.

On receiving the connection request from the parent resource manager, a resource manager knows that all the special resources are designated and it only has to allocate the bandwidth in the links. If all the bandwidth requirement can be satisfied and allocated, an ACK message is sent back to the parent resource manager. Otherwise a NAK is sent. One receiving all the ACKs for the partitioned logical connection, an ACK is sent back to the originating resource manager.

A distributed resource allocation system was designed and implemented to evaluation the performance this algorithm. It takes 0.228 seconds to allocate the resources for the logical connection lc4 on a two-level hierarchical network where the lowest level has three subnetwork each of the same configuration of network Net 3 mentioned in Section 4. The hierarchical network is the same as Network Net-10. The 0.228 seconds includes the transmission time of all eight messages, the execution time of the resource allocation algorithms in all four resource managers, and the time to print the allocation result. The allocation cost for lc4 is 1368.75 and is about 10% more over the cost, 1262, calculated by Algorithm H on Net-10. The execution time of Algorithm H on Net-10 for lc4 is 0.491 seconds is more than twice of that spent by the distributed hierarchical algorithm. If it is used in the centralized resource allocation approach, we have to include the time for the request message to send to the resource allocation center for the allocation messages to be delivered to the involved node. This confirms the advantage of the hierarchical approach when the network is large. We also use Algorithm X, which simply picks the special resources with the lowest cost, for our distributed resource allocation algorithm, the cost it generated is still the same but the cost generated by Algorithm X on Net-10 is 1391 and is higher. It is quite natural that the particular strategy of Algorithm X performs badly when network is big and a faraway resource will be picked and the connection cost to the remote resource will be pretty big.

Figure 9 shows the partitioned logical connection generated by the distributed algorithm. Appendix 2 shows the messages sent by the higher level resource manager for lc4.

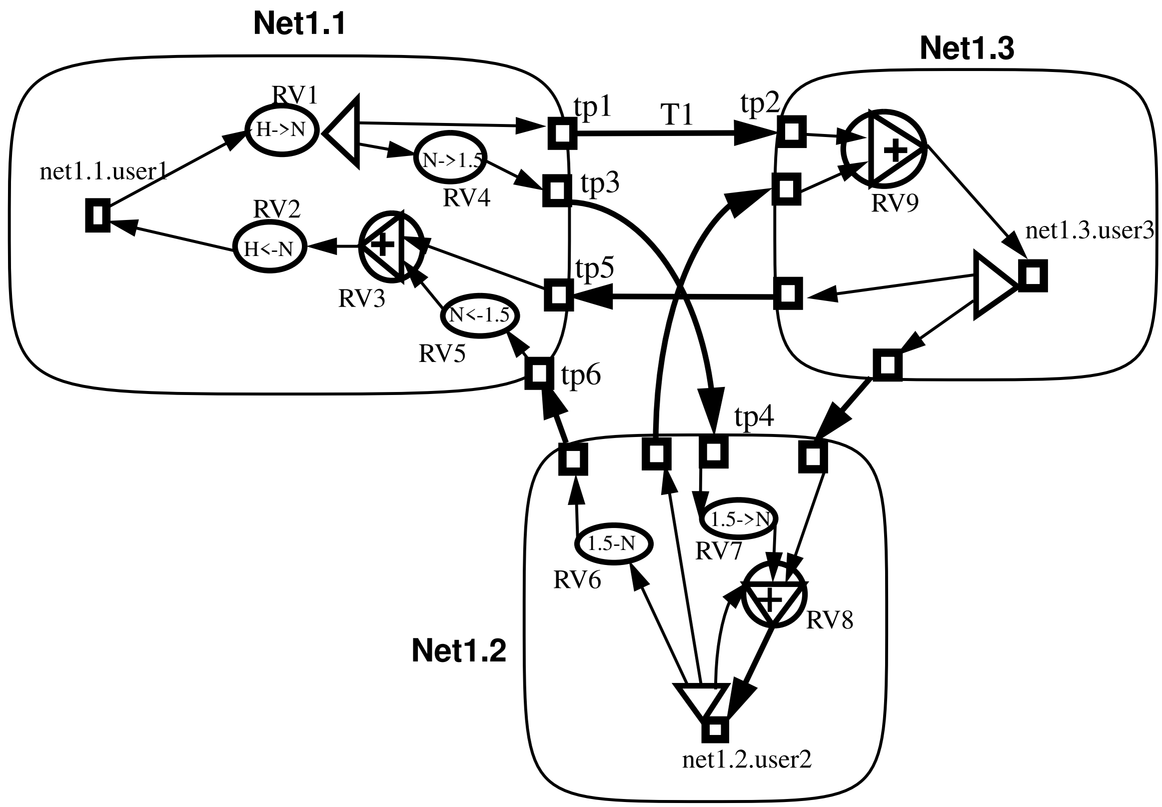

Figure 9. The resource allocation result generated by the distributed algorithm.

6. Discussion

The role of the logical connection can be expanded to serve as a low level user to network signaling interface and to allow users to have more control over the resource allocation policy. For example, the user can specify the preferred locations of some resources in a connection request or indicate resources that have already been reserved. The resource allocation can be easily extended to take this into consideration.

When setting up multimedia multiparty connections across multiple domains such as New Jersey Bell, AT&T, and Pacific Bell, interesting situation may arise where each domain may reduce its allocation cost of some resources to allure customers. Since the proposed distributed algorithm only uses the allocation cost of special resources and the allocation cost of inter-domain trunks as a basis for resource allocation, it may unwisely decide to allocate resources on a domain whose internal link cost is fairly expensive. This problem can be solved by requiring each domain to report the internal link cost associated with the use of its resources from outside of the domain and by modifying the algorithm to take this additional information into consideration.

The proposed algorithms assume that the allocation costs of vertices and edges are close and resources of common resource types may be available in other subnetworks of a large network. Here we consider some of the extreme cases and see how the algorithms can be modified to take advantage of these additional information.

First is the case where the allocation cost of resources dominates and the edge cost can be neglected. Here the problem reduces to that of finding the available cheapest connected resource in the network. Simple queries to the directory service together with a simplified version of the path finding algorithm which checks the connectivity and bandwidth can solve the problem.

In case of homogeneous networks with the same media, the same user terminal type, and no signal compression, the signal converters are no longer needed. The problem reduces to the allocation of information mixers and bandwidth of edges in the network.

The resource allocation algorithms are presented in a framework which allows easy additions of new resource types. Such additions involve creating the resource allocation rule for that particular resource type, and modifying the resource allocation sequence.

The paper did not address the issues of incremental changes to an existing connection. How to provide mechanism to guarantee the atomic operation of each connection request is an important research issue.

7. Conclusions

Multimedia multiparty communication services such as Cruiser [Root88] and desktop conference services [Sakata90] require the network to allocate information mixers or signal converters and to provide efficient connections among these special network resources and the customers. How to represent the complex connections in these services and how to design algorithms that efficiently allocate required resources and the link bandwidth to support the connections become an important research issue.

We have offered a heuristic approach to allocate resources that minimizes message exchange and sets a benchmark for other heuristic resource allocation proposals. It is hoped this work will be supplanted by future research that generates algorithms that allocate more nearly optimal physical connection graphs in “shorter” processing time.

A comprehensive notation, called logical connection, is presented to capture the resource and connectivity requirements of complex multimedia multiparty connections. The notation of the multicast edge and its resource allocation rules enable the use of the multicast capability provided by switch fabrics in broadband networks. The logical connection notation can be used as part of a low level user to network signaling interface to allow users to have more control on the resource allocation policy by specifying the preferred locations of resources or previous reserved resources in a connection.

The resource allocation algorithms are designed to map the logical resources requirements into the physical available resources in a multiple network environment. These algorithms are presented in a framework of resource allocation rules for each resource type and a resource allocation sequence. The framework allows easy addition of new resource types.

By integrating the management of special network resources and that of the network bandwidth, the hierarchical resource allocation algorithm is able to allocate network resources more efficiently. For a complex multimedia multiparty connection with users at different networks, the algorithm partitions the corresponding logical connection into smaller logical connections according to the network boundary. The resource allocation protocol then sends the aggregated resource request to each network and guarantees the atomic operation of the request, i.e., either all are accepted or none of them are accepted. By requesting the network resources in each network in a lump sum manner instead of piecemeal, our approach reduces the number of messages exchanged among the involved networks and thus improves the performance.

Acknowledgment

The author would like to thank Rong Chang, Jason Lin, Shoshana Hardt-Kornacki, Minfa Huang, and Steve Minzer for their valuable comments. The author thank Howard Bussey for his insightful comments on the limitations of the resource allocation algorithms. This work will not be as complete without the encouragement from Steve Weinstein.

References

[Addeo87] Addeo, E.J., Gelman, A., Massa, V.F., "A Multi-Media Multi-Point Communication Services Capability for Broadband Networks," *Proc. ISS'87*, paper C5.5, March 19, 1987.

[Akse85] Akselrod, B. and Langholz, G., "A Simulation Study of Advanced Routing Methods in a Multipriority Telephone Network," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-15, No. 6, Nov./Dec. 1985.

[Alban91] Albanese, A., Bussey. H., Weinstein, S., and Wolf, R., "A Multi-Network Research Testbed For Multimedia Communications Services," *Proc. ICC'91*, Denver, June, 1991.

[Arango 92] M. Arango, P. Bates, G. Gopal, N. Griffeth, G. Herman, T. Hickey, W. Leland, V. Mak, L. Ruston, M. Segal, M. Vecchi, A. Weinrib, S.-Y. Wuu, "Touring Machine: A Software Infrastructure to Support Multimedia Communications," submitted to *Multimedia '92, 4th IEEE COMPSOC International Workshop on Multimedia Communications*, Monterey, April 1992.

[Buben91] Bubenik, R., DeHart, J., and Gaddis, M., "Multipoint Connection Management in High Speed Networks," *Proc. of InforCom'91,* pp. 59-68, Miami, Florida, April 7-11, 1991.

[Bussey89] Bussey, H.E., Porter, F.D., and Raitz, H.E., "A Second Generation BISDN Prototype," *International Journal of Digital and Analog Cabled Systems,* Vol. 1, No. 4, January 1989.

[Chow91a] Chow, C.-H., "On Multicast Path Finding Algorithms," *Proc. of InfoCom'91,* pp. 1274-1283, Miami, Florida, April 7-11, 1991.

[Chow91b] Chow, C.-H., "Resource Allocation for Multimedia Multiparty Connections in Broadband Networks," University of Colorado at Colorado Springs, *Tech Report EAS-CS-91-6*, October 1991.

[Gelman90] Gelman, A.D., and Halfin, S., "Performance Analysis of Information Providing Services," *Proc. of GlobeCom'90*, San Diego, 1990.

[Giaco90] Giacopelli, J.N., Littlewood, M., and Sincoskie, W.D., "Sunshine: A High Performance Self-Routing Broadband Packet Switch Architecture," *Proc. International Switching Symposium (ISS'90),* Vol. 3, paper P21, May 27-June 1, Stockholm, Sweden, pp. 123-129.

[Hardt91b] Hardt-Kornacki, S.L. and Ness, L.A., "Optimization Model for the Delivery of Interactive Multimedia Documents," to appear in *Proc. of GlobeCom'9*1.

[ISO88] ISO, "Information processing systems: open systems interconnection - the directory - overview of concepts, models, and service," Draft International Standard ISO 9594-1: 1988.

[Lee88] Lee, T.T., "Nonblocking Copy Networks for Multicast Packet Switching," *IEEE Journal on Selected Areas In Communications,* Vol. 6, No. 9, pp. 1455-1467, Dec. 1988.

[Minzer91] Minzer, S., "A Signalling Prototype for Complex Multimedia Services," to appear in *IEEE Journal on Selected Areas In Communications,* Dec. 1991.

[Ong91] Ong, L and Schwartz, M., "Distributed Control of Multiparty, Multimedia Broadband Services," to appear in *Proc. of ICC'91.*

[Oppe83] Oppen, D. and Dala, Y., "The Clearinghouse: a decentralized agent for locating named objects in a distributed environment," *ACM Transactions on Office Information Systems*, Vol. 1, No. 3, July 1983.

[Root88] Root, R., "Design of a Multimedia Vehicle for Social Browsing," *Proc. of Conference on Computer Supported Cooperative Work*, Portland, OR, Sept. 26-28, 1988.

[Sakata90] Sakata, S., "Development and Evaluation of an In-House Multimedia Desktop Conference System," *IEEE Journal on Selected Areas In Communications,* Vol. 8, No. 3, pp. 340-347, April 1990.

[Schwar91] Schwartz, M.F., "Resource Discovery and Related Research at the University of Colorado," *Technical Report CU-CS-508-91,* Department of Computer Science, University of Colorado, Boulder, Colorado, 1991.

[Turner86] Turner, J.S., "New directions in communications (or which way to the information age?)," *IEEE Commun. Mag.*, Vol. 24, No.10, pp. 8-15, Oct. 1986.

[Weinrib91] Weinrib, A. and Gopal, G., "Limited Waiting: An Adaptive Overload-Control Strategy for Circuit-Switched Networks," *IEEE Journal on Selected Areas In Communications*, Vol. 9, No. 2, pp. 157-164, Feb. 1991.

Appendix 1: Logical connection request lc4 in a message.

```
[user1 m2 reserve lc4 ((EndVertex, user1, IPS=1, OPS=1, ADDR=steve)
    (EndVertex, user2, IPS=1, OPS=1, ADDR=chow)
    (EndVertex, user3, IPS=1, OPS=1, ADDR=mike)
    (H_2_N, rv1, IPS=1, OPS=1)
    (N_2_H, rv2, IPS=1 OPS=1)
    (PIP, rv3, IPS=2, OPS=1)
    (N_2_T1, rv4, IPS=1, OPS=1)
    (T1_2_N, rv5, IPS=1, OPS=1)
    (N_2_T1, rv6, IPS=1, OPS=1)
    (T1_2_N, rv7, IPS=1, OPS=1)
    (PIP, rv8, IPS=3, OPS=1)
    (PIP, rv9, IPS=2, OPS=1)
    (Edge, e1, SP=user1.OP.0, DP=rv1.IP.0, B=120)
    (Edge, e2, SP=rv1.OP.0, DPL=(rv9.IP.0, rv4.IP.0) B=44.5)
    (Edge, e3, SP=rv4.OP.0, DP=rv7.IP.0, B=1.5)
    (Edge, e4, SP=rv9.OP.0, DP=user3.IP.0, B=44.5)
    (Edge, e5, SP=rv2.OP.0, DP=user1.IP.0, B=120)
    (Edge, e6, SP=rv3.OP.0, DP=rv2.IP.0, B=44.5)
    (Edge, e7, SP=user3.OP.0, DPL=(rv3.IP.0, rv8.IP.2) B=44.5)
    (Edge, e8, SP=rv5.OP.0, DP=rv3.IP.1, B=44.5)
    (Edge, e9, SP=rv6.OP.0, DP=rv5.IP.0, B=1.5)
    (Edge, e10, SP=user2.OP.0, DPL=(rv6.IP.0, rv9.IP.1, rv8.IP.0) B=44.5)
    (Edge, e11, SP=rv7.OP.0, DP=rv8.IP.1, B=44.5)
    (Edge, e12, SP=rv8.OP.0, DP=user2.IP.0, B=44.5))]
```

Appendix 2: The messages sent by the network resource manager to the resource manager s in the subnetworks.

```
Request is sent to net1.2 msg=
[net1 net1.2_1 reserve lc4.net1.2
((EndVertex net1.2.user2 IPS=1 OPS=1)
(N_2_T1 net1.2.sw3_N_2_T1_3 IPS=1 OPS=1 C=4)
(T1_2_N net1.2.sw3_T1_2_N_4 IPS=1 OPS=1 C=4)
(PIP net1.2.sw3_pip_3 IPS=3 OPS=1 C=74)
(Edge e6_net1.2 SP=sw3_pip_3.OP.0 DP=net1.1.sw1.IP.29 B=44.5)
(Edge e7_net1.3 SP=net1.3.sw1.OP.31 DPL=(sw3_pip_3.IP.0
net1.1.sw1.IP.29) B=44.5)
(Edge e8 SP=sw3_T1_2_N_4.OP.0 DP=sw3_pip_3.IP.1 B=44.5)
(Edge e9 SP=sw3_N_2_T1_3.OP.0 DP=sw3_T1_2_N_4.IP.0 B=1.5)
(Edge e10_net1.2 SP=net1.2.user2.OP.0 DPL=(sw3_N_2_T1_3.IP.0
net1.3.sw1.IP.31 net1.1.sw1.IP.29) B=44.5)
(Edge e12_net1.1 SP=net1.1.sw1.OP.29 DP=net1.2.user2.IP.0 B=44.5))]

Request is sent to net1.1 msg=
[net1 net1.1_1 reserve lc4.net1.1
((EndVertex net1.1.user1 IPS=1 OPS=1)
(H_2_N net1.1.sw1_H_2_N_1 IPS=1 OPS=1 C=10)
(N_2_H net1.1.sw1_N_2_H_2 IPS=1 OPS=1 C=10)
(N_2_T1 net1.1.sw3_N_2_T1_3 IPS=1 OPS=1 C=4)
(T1_2_N net1.1.sw3_T1_2_N_4 IPS=1 OPS=1 C=4)
(PIP net1.1.sw3_pip_3 IPS=3 OPS=1 C=74)
(Edge e1 SP=net1.1.user1.OP.0 DP=sw1_H_2_N_1.IP.0 B=120)
(Edge e2_net1.1 SP=sw1_H_2_N_1.OP.0 DPL=(sw3_N_2_T1_3.IP.0
net1.3.sw1.IP.30) B=44.5)
(Edge e3 SP=sw3_N_2_T1_3.OP.0 DP=sw3_T1_2_N_4.IP.0 B=1.5)
(Edge e5 SP=sw1_N_2_H_2.OP.0 DP=net1.1.user1.IP.0 B=120)
(Edge e6_net1.2 SP=net1.2.sw1.OP.29 DP=sw1_N_2_H_2.IP.0 B=44.5)
(Edge e7_net1.3 SP=net1.2.sw1.OP.29 DP=sw3_pip_3.IP.2 B=44.5)
(Edge e10_net1.2 SP=net1.2.sw1.OP.29 DP=sw3_pip_3.IP.0 B=44.5)
(Edge e11 SP=sw3_T1_2_N_4.OP.0 DP=sw3_pip_3.IP.1 B=44.5)
(Edge e12_net1.1 SP=sw3_pip_3.OP.0 DP=net1.2.sw1.IP.29 B=44.5))]

Request is sent to net1.3 msg=
[net1 net1.3_1 reserve lc4.net1.3
((EndVertex net1.3.user3 IPS=1 OPS=1)
(PIP net1.3.sw3_pip_3 IPS=3 OPS=1 C=74)
(Edge e2_net1.1 SP=net1.1.sw1.OP.30 DP=sw3_pip_3.IP.0 B=44.5)
(Edge e4 SP=sw3_pip_3.OP.0 DP=net1.3.user3.IP.0 B=44.5)
(Edge e7_net1.3 SP=net1.3.user3.OP.0 DP=net1.2.sw1.IP.31 B=44.5)
(Edge e10_net1.2 SP=net1.2.sw1.OP.31 DP=sw3_pip_3.IP.1 B=44.5))]
```

BIBLOGRAPHIC DATA SHEET	1. Report No. **EAS-CS-92-1**	2.	3. Recipient's Accession Number

4. Title and Subtitle

Resource Allocation Algorithms for Multimedia Multiparty Connections

5. Report Date

January 1992

6.

7. Author(s)

C. Edward Chow

8. Performing Organization Rept. No.

EAC-CS-92-1

9. Performing Organization Name and Address

Department of Computer Science
University of Colorado at Colorado Springs
Colorado Springs, CO 80933-7150

10. Project/Task/Work Unit No.

11. Contract/Grant No.

12. Sponsoring Organization Name and Address

13. Type of Report & Period Covered

14.

15. Supplementary Notes

16. Abstracts

This paper addresses the problem of allocating resources to set up multimedia multipoint connections, while efficiently using network resources, in a network that includes signal converters to resolve the heterogeneity of customer/network equipments and information mixers for conference services. Specifically, given a connection request and a physical network, we consider the general resource allocation problem where the allocation costs of special resources and links are comparable so that neither can be neglected. The problem is shown to be NP-hard and the performance result of an optimal algorithm, several heuristic algorithms, and a distributed algorithm is presented.

17. Key Words and Document Analysis. 17a. Descriptors

Resource Allocation
Multimedia Communication
Multiparty Communication
Broadband Networks
Optimal Algorithm
Heuristic Algorithm
Distributed Algorithm

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages 16
	20. Security Class (This Page)	22. Price