Route Optimization in Mobile ATM Networks

Gopal Dommety
Computer and Information Science
Ohio State University
Columbus, OH 43210
dommety@cis.ohio-state.edu

Malathi Veeraraghavan
Bell Laboratories
Lucent Technologies
Holmdel, NJ 07733
mv@bell-labs.com

Mukesh Singhal
Computer and Information Science
Ohio State University
Columbus, OH 43210
singhal@cis.ohio-state.edu

Abstract

Location management schemes that propose setting up the connection to the home location of a mobile and then rerouting the connection to the mobile's current location could result in suboptimal connection paths. In this paper, we propose an *algorithm for optimizing the route of a connection* that becomes suboptimal due to location-based reroutes for mobile ATM (Asynchronous Transfer Mode) networks based on the PNNI (Private Network-to-Network Interface) standard. This algorithm uses the hierarchical route information of the connection and the summarized topology and loading information of the network to determine a "crossover node" such that adjusting the connection from that crossover node results in an optimally routed connection. This route optimization procedure can be executed during mobile connection setup, thus proposing a *one-phase mobile location and connection setup* scheme. We also demonstrate how this route optimization procedure can be executed after a connection is set up to a mobile, thus proposing the *second phase* of *two-phase mobile location and connection setup* schemes. A comparative performance analysis of one- and two-phase connection setup schemes is presented. Measures of comparison are *call setup delay* and the amount of *network resources* allocated to a connection. One-phase scheme incurs higher call setup delays, but results in an optimal route during connection setup. On the other hand, two-phase schemes incur lower call setup delays, but result in connections with suboptimal routes, prior to the execution of the route optimization phase.

1. Introduction

The use of ATM technology in wireless networks is currently under study [1]. In mobile ATM networks, handoff procedures and location management procedures are needed to support user mobility. *Handoff* procedures are needed to reroute connections on which the mobile user is communicating while moving. *Location management* consists of tracking mobiles and locating them for incoming call deliveries. In both these sets of procedures, paths taken by connections could become "suboptimal[1]." For example, most handoff schemes propose performing a local connection reroute rather than an end-to-end connection reroute to keep handoff latencies low. Such reroute operations could result in making the connection path suboptimal. Similarly, location management schemes that propose setting up the connection to the home location of a mobile and then rerouting the connection to the mobile's current location based on location data provided by the home node could result in suboptimal connection paths. In other words, suboptimality that is introduced during connection setup is primarily due to lack of exact information about the location of the mobile at the call originating switch, and suboptimality occurs after connection setup because of user movements by a communicating mobile. It is important to optimize routes of connections since suboptimal paths imply an inefficient usage of network resources.

1. The path of a connection is classified as being "suboptimal" if it is not the best path between the two endpoints of the connection.

This paper proposes an *algorithm for optimizing the route of a connection* using the hierarchical route information of the connection and the summarized topology and loading information of the network. This solution is proposed for ATM networks based on the PNNI (Private Network-to-Network Interface) standard [2]. Based on this algorithm, we propose two solutions for the mobile location and connection setup problem. In the first solution, the route optimization procedure is integrated in the mobile location and connection setup procedure (one-phase method). In this method a connection to a mobile is set up by first routing the connection to the mobile's home location, and then performing an on-the-fly route optimization by altering the route of the connection after determining the location of the mobile from its home node. In the second solution, the route optimization procedure is executed after the mobile location and connection setup procedure is completed (two-phase method). In this method, the connection setup procedure can be continued from the home node to the current location of the mobile and route optimization procedure can be applied subsequently.

The one-phase method incurs a higher *call setup delay*, but results in an optimal route (i.e., using a minimum amount of network resources). On the other hand, the two-phase method incurs a lower call setup delay, but uses a larger amount of *network resources* (prior to the execution of the route optimization phase). A comparative performance analysis of the one- and two-phase methods for location management is presented in this paper. Measures of comparison are call setup delay and the amount of network resources allocated to a connection.

Since the route optimization algorithm is proposed for mobile ATM networks based on the PNNI standard, we begin by providing background information on the PNNI standard and summarize prior work on location management and route optimization in Section 2. A solution to the route optimization problem for PNNI-based ATM networks is proposed in Section 3. Section 3 also describes how this solution can be integrated with the location management procedures in one- and two-phase schemes. Finally, Section 4 presents results of a comparative performance analysis of one- and two-phase schemes for location management, and Section 5 concludes the paper.

2. Background

In this section, we first present a brief overview of the PNNI standards for ATM networks, motivate the need for route optimization by elaborating on the location management procedures that cause suboptimal routes, and then summarize prior work on route optimization.

2.1 Overview of PNNI standards

PNNI-based ATM networks [2] are arranged in hierarchical peer groups as shown in Fig. 1. The lowest level ($l = 3$) consists of ATM switches connected in arbitrary topologies. Each peer group has an elected *Peer Group Leader* (PGL). Nodes within a peer group exchange topology, loading and reachability information us-

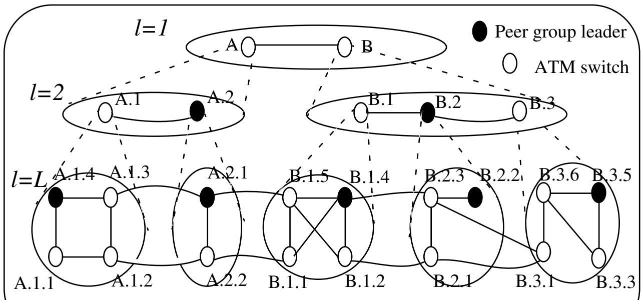

Fig. 1 PNNI-based hierarchical ATM network

ing the *PNNI routing protocol*. The PGL of each peer group represents all nodes within its peer group at the higher-level peer group and sends summarized topology/loading/reachability information about its lower-level peer group to its peers in the higher-level peer group. Each higher-level peer node broadcasts this summarized information to all its nodes in the lower-level peer group. Using this technique, each node has topology/loading/reachability data about its own peer group and all its ancestor peer groups. For example, in Fig. 1, node A.1.1 has the topology information of the entire peer group A.1, as well as the topologies of peer group A and the top-level ($l = 1$) peer group.

The *PNNI signaling protocol standard* defines ATM connection setup and release procedures. In order to set up a connection, the first switch receiving the connection setup request determines the hierarchical source route for that connection. The computed hierarchical source routes are carried as *DTL* (Designated Transit Lists) parameters in the PNNI signaling SETUP messages [2]. A set of DTLs is used to specify the complete path of a connection from the current node to the destination. However, the exact list of nodes is not known due to the hierarchical nature of the source routes. The exact path within each peer group is computed at the *ingress border node* of the peer group, i.e., the call originating switch determines the hierarchical path for the entire connection, and the ingress border node of each peer group determines the connection path within that peer group. On call arrival, the ingress border node stores the previous DTL information, and updates the DTLs in the SETUP message to specify the source route within the peer group. A call being set up according to a specified set of DTLs may be blocked at a node due to a lack of sufficient resources or connectivity. In such situations, the call is cranked back (released) to the border node that created the unusable DTL and an alternate route is attempted. "*Optimality*" of connection paths in PNNI-based networks should be regarded within the context of this hierarchical organization of switches. In other words, because of the hierarchical organization, the "shortest-path" computed by PNNI may not be the true shortest-path. This penalty is paid in return for network scalability.

As an example, a setup request from an endpoint connected to switch A.1.4 to an endpoint connected to switch B.2.2 (see Fig. 1) results in an initial path computation of {A.1.4, A.1.3, A.2, B}. When the SETUP reaches A.2.1, the border node of peer group A.2, it computes the path through peer group A.2 as consisting of only node A.2.1. At the next node, B.1.5 (the border node of peer group B), the path is recomputed as consisting of {B.1.5, B.1.4, B.2}. Similarly, the path through peer group B.2 is computed at B.2.3 as {B.2.3, B.2.2}, and the connection is set up to B.2.2. When the SETUP reaches B.2.2, the hierarchical path information available to B.2.2 from the DTLs is {A, B.1, B.2.3, B.2.2}.

2.2 Motivation

In this section, we briefly describe location management schemes that result in connections with suboptimal routes, thus *motivating the need for route optimization*.

Location management consists of tracking mobiles and locating them in order to deliver incoming calls. Mobile tracking procedures update the home node of a mobile with information about the location of the mobile. Mobile location procedures use the information acquired during the mobile tracking procedures to deliver incoming calls to mobiles. In [3], we proposed a mobile location management scheme called the *mobile PNNI scheme*. In this scheme, reachability updates are propagated with limited scope as mobiles move allowing a small "neighborhood" of switches to know the switch through which the mobile can be reached. In other words, switches within the neighborhood of the mobile have the correct reachability information about the mobile, and the switches outside the neighborhood of the mobile will have default reachability information indicating that the mobile is at its home switch. For example, in Fig. 1, neighborhood of a mobile currently under the switch A.1.1 with scope set to 2 consists of all the switches in the peer group A. Calls originating within the neighborhood of the mobile will be set up optimally on the shortest paths, since all the switches on the paths have correct reachability information.

Calls originating outside the neighborhood of the mobile, since the reachability information in the switches outside the neighborhood indicate that the mobile is at its home (default reachability information), proceed toward the home of the mobile. In this case, the following two scenarios are possible: *i)* called mobile is within the neighborhood of its home, or *ii)* mobile is outside the neighborhood of its home. If the called mobile is within the neighborhood of its home, the default reachability information (indicating that the mobile is in its home neighborhood) in switches outside the neighborhood of the mobile is the correct reachability information. Since the switches within the neighborhood as well as outside the neighborhood have correct reachability information, calls get routed via optimal paths. Therefore, if the call originates within the neighborhood of the mobile, as shown in Fig. 2 for the called mobile located in position I, or if the mobile is at its home location, the connection is routed on the shortest path.

To handle calls originating at switches outside this neighborhood when the mobile is not in its home neighborhood, a forwarding pointer is set at the home switch of the mobile indicating the current location of the mobile. On call arrival, originating switches set up connections as per the data indicated in their reachability information. Since the reachability information in switches outside

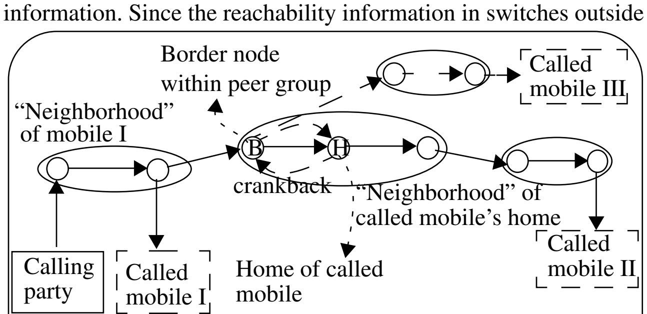

Fig. 2 Combined crankback and call forwarding scheme

the neighborhood of the mobile indicate that the mobile is reachable through its home, the call will be routed toward the home of the called mobile, at which point, it needs to be somehow rerouted to the current location of the mobile. In [3], we proposed *forwarding the call* from the home switch to the current location. This is illustrated in Fig. 2 for the case when the called mobile is in position II. Other options for handling this situation are proposed in [4, 5]. Acharya et al. [4] proposed a *complete release* of the connection from the called mobile's home switch to the calling party's switch followed by a setup from the calling party's switch to the current location of the mobile. Ayyagari et al. [5] proposed using the *crankback* feature of PNNI signaling (as shown in Fig. 2 for the case when the called mobile is located at position III), which allows for the call setup procedure to be cranked back to the last border node (B in Fig. 2) and then rerouted. In [5], the authors note that typically crankback will be executed at a local level, which means that the overall end-to-end path taken by the connection could be "suboptimal."

These three schemes can be classified into *one-phase* or *two-phase* schemes as shown in Fig. 3. In one-phase schemes, such as the complete release scheme of [4], an optimally routed connection is constructed during the call setup procedure, thereby eliminating the need for route optimization. In two-phase schemes, such as [3, 5], the call is delivered to the mobile possibly on a suboptimal route (first phase) and a route optimization (second phase) is performed subsequently to achieve an optimal route.

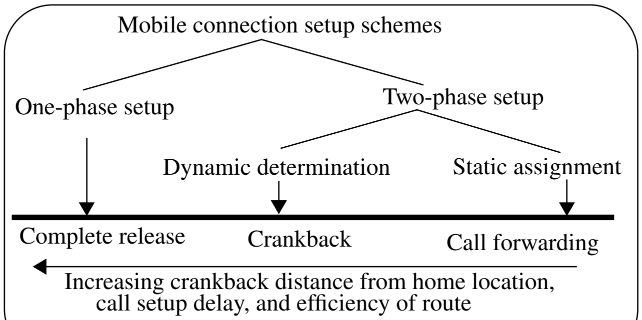

Fig. 3 Spectrum of mobile connection setup schemes

The complete release method incurs a high connection *setup delay*, while the call forwarding scheme (in which the crankback node is statically assigned to be the home node) uses the most amount of *network resources* (albeit for a short period of time until the route optimization phase is complete). The crankback case, in which the node to which the connection is "cranked back" is determined dynamically, is between these two extremes as shown in Fig. 3.

In this paper, we propose a solution to the route optimization problem, for the second phase of the two phase methods. We also demonstrate that the principles of finding the optimal route can be applied to realize a one-phase connection setup scheme without a complete release of the connection from the mobile's home to the called party's switch. In other words, an optimal route can also be achieved using the crankback approach in one phase by cranking back to the "right" node. This is described in Section 3.2

2.3 Prior work

In Section 2.2, prior work on location management was described, thus motivating the need for route optimization. In this section we briefly review prior work on route optimization.

Mobile IP [6, 7] is an extension to the Internet Protocol (IP), which enables hosts to change their point of attachment to the internet with out changing their IP address. In this protocol, a packet for a mobile host is routed to the home network of the mobile as identified by its permanent IP address. The home network tracks the current location of the mobile and tunnels the packet to the current network of the mobile. In order to prevent this "triangle" routing, route optimization extensions have been proposed [8]. These extensions provide a means for communicating-nodes to maintain a binding between the mobile and its current location, and use this binding to tunnel datagrams directly to the mobile. Extensions are also provided to allow datagrams in flight when a mobile node moves, and datagrams sent based on an outdated binding information, to be forwarded directly to the mobile. The primary difference between the route optimization extensions proposed in [8] and the route optimization procedure being proposed in this paper is that, the former extensions have been proposed for the connectionless Internet Protocol, while the latter is being proposed for connection-oriented networks (PNNI-based ATM networks). In connection-oriented networks, in order to reroute a connection, a new connection segment needs to be set up between a "crossover" node and one of the ends of the connection, and the user data needs to be switched from the old path to the new path without loss of ATM cell sequence.

3. Route optimization for location management

In this section, the solution to the *route optimization* problem motivated in Section 2.2 is presented. Before presenting the solution, variations of the problem of route optimization for location management are stated first. In location management there are two variations of the problem of route optimization. In the first variation, a connection is being set up from node c to node h (where c is used to denote the calling party's switch and h is used to denote the home switch of the mobile), and it is recognized that the connection should be set up to the current (visiting) switch of the mobile, v, instead of to h (see Fig. 4). The problem is to redirect the connection, such that an optimal connection is formed between c and v, instead of a connection between c and h. *In the second variation,* a connection already exists between c and v and is suboptimally routed through h (see Fig. 5). The problem is to adjust the connection such that the resulting connection path between c and v is optimal.

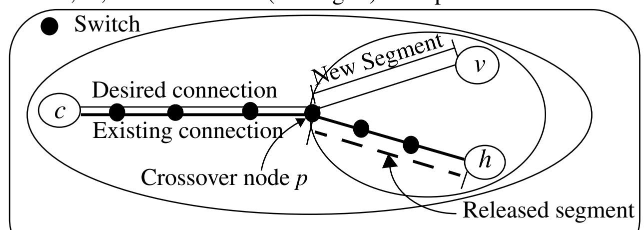

Fig. 4 Route optimization during mobile connection setup

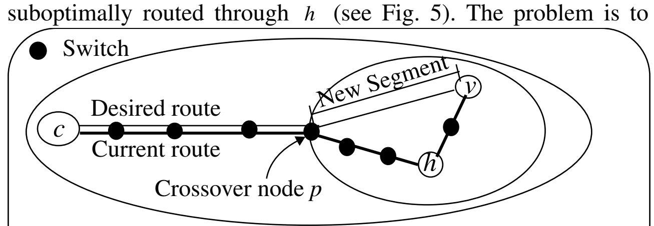

Fig. 5 Route optimization after the connection is established

In the first variation, an optimal connection segment is formed between *c* and *h* but needs to be rerouted to *v*. The desired connection may be achieved by first identifying a "crossover" node on the current path and then rerouting the connection from that node to the current switch of the mobile (see Fig. 4). In the second variation, a connection exists between *c* and *v* and this connection consists of two optimal connection segments, one between *c* and *h* and the other between *h* and *v*. In this variation an optimal path between *c* and *v* can be obtained either by selecting the connection segment between *c* and *h* and rerouting this connection segment to *v* (target node), or by selecting the connection segment between *v* and *h* and rerouting this connection segment to *c* (target node). In this case the rerouting of the connection segment is done by first identifying a crossover node on the connection segment and then setting up a new segment between the crossover node and the target node (see Fig. 5). A sub-problem common to both the variations of the route optimization problem is that of "adjusting" an existing connection between two nodes, *f* and *o* (where *f* is used to denote the "far-end" and *o* is used to denote the "old node"), to form an optimally routed connection between *f* and a third node *t* (target node). For example, in the first variation, the connection that is being formed between *c* and *h* is to be adjusted to form a connection between *c* and *v*. This sub-problem is stated as follows.

Consider an existing optimally routed connection between two endpoints *f* and *o*. The problem is to set up a connection from the far-end (f) to a target node (t). The new connection path should

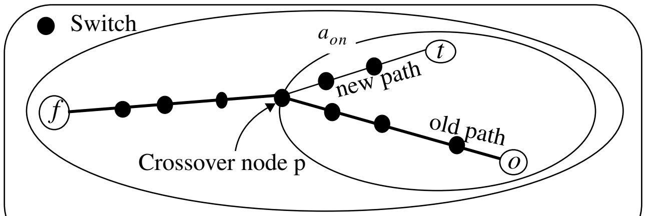

Fig. 6 Rerouting of connection *f-o* to form a connection *f-t*

be optimal and the procedure to achieve the new connection should be efficient. Ideally, rerouting of the connection should be achieved without requiring a complete release of the existing connection between *f* and *o*, and by minimizing the length of the new segment to be set up. The rerouting of the connection can be performed by first identifying a crossover node on the existing connection and then setting up a new connection segment between the crossover node and the target node (see Fig. 6). The crossover node should be selected such that the resulting connection is an optimally routed connection (such a crossover node is referred to as an optimal crossover node). Next, methods to determine an "optimal" crossover node are described. Later, we demonstrate how these methods can be integrated in mobile location and connection setup procedures to achieve optimally routed connections.

3.1 Optimal crossover node determination

The *optimal crossover node* between two connection paths $f-o$ and $f-t$ is a switch p on the old path $f-o$, such that a concatenation of path $f-p$ and $p-t$ results in an "optimal" (shortest) path between nodes f and t (see Fig. 6). It should be noted that, for a connection path $f-o$ that needs to be rerouted to form the connection $f-t$, there could be several optimal crossover nodes. Among these nodes, the node that minimizes the cost (e.g., length) of the new segment is referred to as the *minimal crossover node*.

Ideally route optimization should be accomplished by selecting the minimal crossover node as the crossover node and then rerouting the connection to *t*. In certain scenarios, it may not be possible to either compute the minimal crossover node or to reroute the existing connection at the minimal crossover node. For example, hierarchical path information of the connection is required to determine the minimal crossover node. In certain cases this information is not available, as a consequence it may not be possible to determine the minimal crossover node. Another important consideration is that, in PNNI based networks, the standards stipulate that only the node that created a hierarchical DTL can change that DTL. This restriction enforces the principle of hierarchical source routing, thereby preventing possible loops in the resulting connection. Rerouting from the minimal crossover node may not confirm to the restriction, but results in a loop free connection. In cases when it is not possible to compute the minimal crossover node or rerouting of the connection from any node on the path is not permitted, rerouting of the connection is performed from an optimal crossover node that may not be the minimal crossover node.

Fig. 7 shows the flow chart for the determination of the crossover node. In this procedure, first an optimal crossover node is

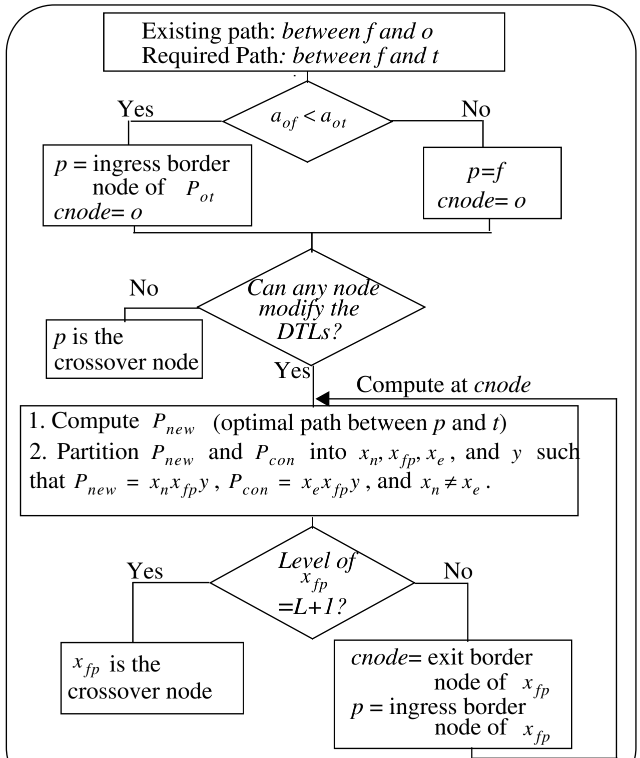

Fig. 7 Determination of crossover node

identified using the hierarchical organization of the network and then, the minimal crossover node is determined using the current path information and the summarized topology and loading information. In this section, the procedure to determine an optimal crossover node is described first. This procedure can be used to reroute connections when it is not possible to compute the minimal crossover node or rerouting of the connection from any node on the path is not permitted. Later, the procedure to determine the minimal crossover node is described. The notation used in this description is defined below.

Definition: Ancestors-are-siblings peer group, denoted by P_{ij}, of two nodes i and j is the lowest peer group in the hierarchy at which ancestors of both nodes belong to the same peer group. For example, in Fig. 1, ancestors-are-siblings peer group of nodes A.1.1 and A.2.1 is A.

Definition: The *ancestors-are-siblings level* of two nodes i and j, denoted by a_{ij}, is the level[2] at which the ancestors of the two nodes i and j belong to the same peer group, i.e., the level of their ancestors-are-siblings peer group. For example, in Fig. 1, the ancestors-are-siblings level of A.1.1 and A.2.1 is 2, since the level of peer group A is 2.

Next, the procedure to determine an optimal crossover node is described. Two aspects of the PNNI based networks that are fundamental to this procedure are: *i)* every node has only the summarized information regarding the topology of the network (nodes outside of a peer group do not have information regarding internal structure of that peer group), and *ii)* connections are routed using hierarchical source routing. The basis of this procedure is that, if a connection is routed using source routing and the nodes outside a peer group do not know the internal details of the peer group, then the connection to any node within the peer group will follow the

2. The numbering of levels in the hierarchy is such that lower level peer groups have higher level values assigned to them, i.e., the level numbers increase from top to bottom in the hierarchy.

same route until the first node of that peer group. The location of the optimal crossover node depends on the exact relation between the ancestors-are-siblings levels of the nodes *f*, *o*, and *t*.

If $a_{of} < a_{ot}$ *(Case I of Fig. 8),* the scenario is one in which the target node (*t*) is closer[3] to the old node (*o*) than the far-end (*f*). In this case, the optimal crossover node is the ingress border node of the *ancestors-are-siblings peer group* of the old and the target nodes (P_{no}) through which the connection setup procedure for the $f-o$ connection entered peer group P_{to}. Since nodes contain only the summarized topology and reachability information, nodes outside a peer group cannot distinguish between exact locations of the different nodes reachable through that peer group. Therefore, a connection from the far-end to either the target or the old node will follow the same route until it reaches the ingress node (p) of the *ancestors-are-siblings peer group* of the old and the target nodes (P_{to}). Once the connection setup arrives at the ingress node *p*, it computes the source route to the exact location. Therefore, at *p*, the connection to the target node (*t*) would have a different source route than a connection to the old node (*o*). Hence, an optimal path can be obtain by adjusting the existing connection from the crossover switch *p*. This scenario ($a_{of} < a_{ot}$) is expected to occur frequently during route optimization during/after call setup, since majority of the mobiles are situated in and around their home area.

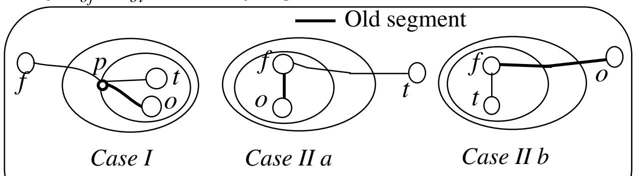

Fig. 8 Relative positions of nodes *f*, *o* and *n*

Example: Consider the scenario in which there is an existing connection between A.1.1 (far-end) and B.3.1 (old node), and a new connection between A.1.1 and B.2.1 is desired (see Fig. 9). In this case a_{of}, a_{ot} and P_{to} are 1, 2, and B respectively. Since $a_{of} < a_{ot}$, the crossover node is the ingress border node of peer group B, which is B.1.1. Therefore, the optimal path between A.1.1 and B.1.2 is achieved by setting up a new segment from B.1.1 to B.2.1, as shown in Fig. 9.

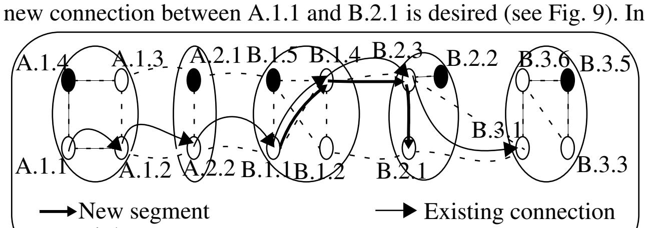

Fig. 9 Rerouting connections

If $a_{of} > a_{ot}$ *(Case II a of Fig. 8)*, the scenario is one in which the old node (*o*) is closer to the far-end (*f*) than to the target node (*t*). In this situation there may not be any segment common between the existing connection and the new desired connection. Therefore, the optimal crossover node is the far-end (*f*). The new path is optimal, since the entire path between the far-end and the target node is being set up.

If $a_{of} = a_{ot}$ *(Case II b of Fig. 8)*, the scenario is one in which the target node is closer ($a_{ft} > a_{fo}$) to the far-end than to the old node or equidistant ($a_{ft} = a_{fo}$) to the far-end and the old node. In this situation too there may not be any segment common between the existing connection and the new desired connection. The optimal crossover node is the far-end (*f*).

3. Closeness is defined with respect to the node relationships defined by the peer groups. Note that closeness according to this criterion is not necessarily close with respect to number of hops or geographical distance.

The crossover node determined by the procedure described above is hence forth referred to as *the optimal crossover "point"* (denoted by *p*). The significance of the optimal crossover point is three fold. *First*, if a new connection is established from this node to the target node, then the resulting connection path between the far-end and the target node will be optimal. *Second*, the optimal crossover point is determined without the use of previous path information. *Third*, the current PNNI standards stipulate that only the node that created a hierarchical DTL can change that DTL. This optimal crossover node is the node that created the DTL that is to be changed. Therefore, rerouting of the connection at this optimal crossover node conforms to the stipulation.

It is possible for a part of the existing segment from the optimal crossover point to the old node (P_{con}) to be in common with the new segment (P_{new}) from the crossover node to the target node (see Fig. 10). Next, procedure to determine the *minimal crossover node* is presented, thereby minimizing the length of the new segment. It should be noted that, *rerouting from the minimal crossover node can be performed only if any node on the path can change the hierarchical source route (DTLs) of the connection.*

Minimal crossover node determination: As explained earlier, it is possible for a part of the existing segment from the optimal crossover point (*p*) to the old node (P_{con}) to be in common with the new segment (P_{new}) from the crossover node to the target node (see Fig. 10). In order to reduce the connection setup delay and signaling load on the network, it is desirable to minimize the length of the new segment setup. Thus, motivating the need for determination of the "minimal crossover node".

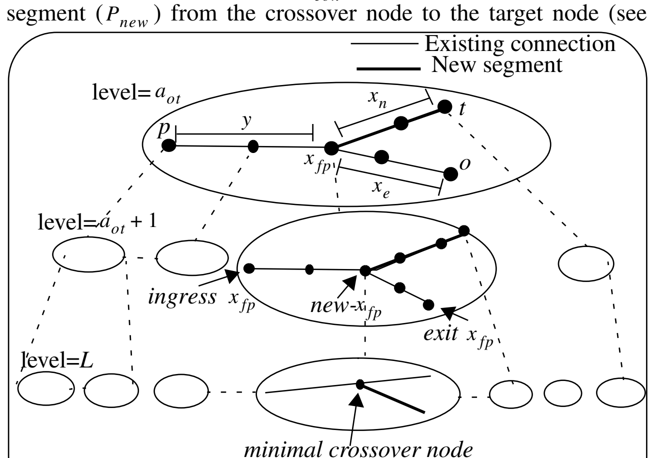

Fig. 10 Minimal crossover node determination

Two aspects of the PNNI based networks that are fundamental to this procedure are: *i)* every node has summarized information regarding the topology and the loading conditions of the network. Therefore, a node can compute the local view of the hierarchical route between any two nodes in the network. For example in Fig. 9, B.1.2 can compute the shortest path between A.1.1 and B.3.2 to be {A, B.1.1, B.1.4, B.2, B.3}. *ii)* It is possible to construct the local view of the existing connection, P_{con}, using the hierarchical source route information stored in the ingress border nodes on the existing path. Hence, at any node it is possible to compare the local view of the existing connection and the local view of the desired connection, and identify the minimal crossover node or the peer group containing the minimal crossover node.

Fig. 7 shows the flow chart for the determination of the minimal crossover node (if any node can modify the DTLs). First the optimal crossover point is determined using the procedures described earlier. Next, this crossover node is refined by identifying a new-crossover node that is closer to the old node (o). This process is repeated until the minimal crossover node is determined.

In order to identify a new-crossover node that is closer to the old node, the path from the optimal crossover point to the target node, P_{new}, is computed[4] (precomputed path is used if one is available) and compared with the local view of the existing path[5]. The purpose of the comparison is to partition the new and the old segments into x_n, x_{fp}, x_e, and y such that $P_{new} = x_n x_{fp} y$ (concatenation of the three paths), $P_{con} = x_e x_{fp} y$, and x_n and x_e are different (note that only the hierarchical path information is known about both the segments), as shown in Fig. 10. In other words, a connection from the optimal crossover point to the target node will follow the routes y, x_{fp} and x_n, and the existing connection follows the routes y, x_{fp} and x_e. Therefore, the minimal crossover node is a *node in the peer group* x_{fp} and the ingress border node of x_{fp} in the new optimal crossover node.

If the peer group x_{fp} corresponds to a switch (peer group of level $L+1$), x_{fp} is the minimal crossover node. Otherwise, the procedure to refine the crossover node is repeated at the *exit border node* of x_{fp}, i.e., P_{con} (ingress border node of x_{fp} to current node) and P_{new} (ingress border node of x_{fp} to the target node) are compared and a *new-*x_{fp} is determined (see Fig. 10). Note that, by this procedure the minimal crossover node is determined in less than $L + 1 - Max\{a_{on}, a_{fo}, a_{fn}\}$ iterations. By this refined selection of the crossover node, the length of the new segment to be set up can be minimized, thereby reducing the associated setup delays and the signaling load. However, there is processing overhead associated with this procedure. Heuristics can be used to terminate the search process, i.e., at a particular level instead of repeating the procedure to compute the minimal crossover node at the exit border node of x_{fp}, the ingress border node of x_{fp} can be selected as the crossover node.

Example: Consider the scenario in which an existing connection between A.1.1 (f) and the B.3.1 (o) is to be rerouted to form a connection between the switch A.1.1. and the switch B.2.1 (n). In the previous example the optimal crossover node is computed to be B.1.1. In this scenario, the old segment, P_{con}, is {B.1, B.2, B.3.1} and the new segment, P_{new}, is {B.1, B.2}. Applying further refinement procedure, y, x_e, x_n and x_{fp} are computed to be {B.1}, {B.3.1}, {} and {B.2}, respectively. Therefore, the new crossover node is the ingress border node of peer group B.2, B.2.3.

Example: Consider the scenario in which an existing connection between A.1.1 (f) and the B.3.1 is to be rerouted to form a connection between the switch A.1.1. and the switch A.1.3 (see Fig. 11). In this case a_{of}, a_{on} and P_{no} are 1, 1, and A, respectively. Since $a_{of} = a_{on}$, the optimal crossover node is the calling switch (A.1.1). Therefore, the old segment (P_{con}) is {A, B.1, B.2, B.3.1} and the new segment is {A}. Applying further refinement procedure, y, x_e, x_n and x_{fp} are computed to be {}, {}, {B.1, B.2, B.3.1} and {A}, respectively. Since A is not a peer group of level *L+1*, the procedure to compute the minimal crossover node is repeated at the exit border node of peer group A (x_{fp}), A.2.2. The old segment (P_{con}) and the new segment are computed to be {A.1, A.2.2} and {A.1}, respectively. Applying further refinement procedure, y, x_n, x_e and x_{fp} are computed to be {}, {}, {A.2.2} and {A.1}, respectively. Repeating this process at the exit border node of peer group A.1 (*new-*x_{fp}), A.1.2, the minimal crossover node is computed to be A.1.2.

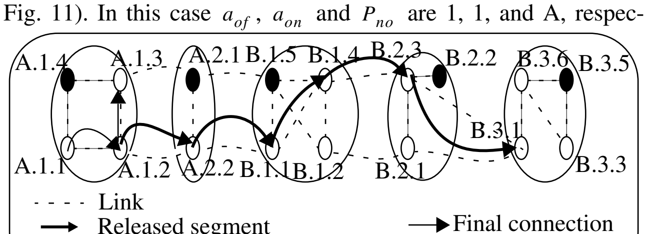

Fig. 11 Rerouting connections

4. Each node has summarized information regarding the topology and the loading conditions of the network.
5. If multiple optimal paths exist between the crossover node and the target node, the path that maximizes y is selected

Next, we integrate the crossover node determination algorithm in the location management schemes described in Section 2.2. For location management, the algorithm can be applied while setting up a connection (*route optimization during call setup*) or after a connection is completely setup (*route optimization after call setup*). Similarly, for handoffs, this algorithm can be applied to determine the optimal crossover node during or after a handoff [9].

3.2 Route optimization during mobile connection setup

In this section, we present a *one-phase mobile connection setup scheme* by which an optimal connection is achieved while setting up the connection to a mobile. In this scheme, route optimization procedure is integrated with the moble connection setup procedure, i.e., a connection path suboptimality is detected during connection setup and on-the-fly route optimization is performed.

In this scheme, like in the mobile-PNNI scheme[3], reachability information regarding the location of the mobile is propagated within the neighborhood, and on call arrival to a mobile, the call is setup with each switch “believing” in its reachability information. Calls originating inside the neighborhood of the mobile or if the mobile is located in its home neighborhood, the call is optimally routed to the mobile. For calls originating outside the neighborhood of a mobile that is not currently located in its home neighborhood, the call originating switch sets up the connection to the home of the called mobile. *However*, when the call setup arrives at the home switch of the mobile, instead of performing a fast local rerouting of the connection to the mobile, rerouting of the connection is performed such that the resulting connection is optimally routed.

On arrival of the call setup message at the home switch, an optimal connection is formed from the calling party’s switch to the home switch of the mobile and it needs to be rerouted to the current location of the mobile (home switch is informed about the mobile’s current switch location through the process of registration). This corresponds to the variation of the problem in which a connection is being set up from node c to node h, and it is recognized that the connection should be set up to v instead of to h (see Fig. 4).

In order to reroute the connection, first, the *crossover node is determined* and then the *connection is cranked back* (released) from the home switch to the computed crossover node. Once the connection is cranked back till the optimal crossover point (or minimal crossover node), the *call is routed to the current location* of the mobile. The optimal crossover node is determined using the procedure described in Section 3.1 by considering f as the calling party’s switch, o as the home switch of the called mobile, and t as the current switch of the called mobile. If all nodes are permitted to change the source route (DTLs) of a connection, the connection can be cranked back to the minimal crossover node. In this case, the home node after determining P_{new} and P_{con}, computes x_{fp} using the procedure explained in section 3.1. The connection is then *cranked back till the exit border node of* x_{fp}. If x_{fp} is the minimal crossover node, i.e., the level of x_{fp} is *L+1*, the call setup is directed towards the current location of the mobile. Otherwise, a *new-* x_{fp} is computed and the connection is cranked back further. The process of recomputing x_{fp} and cranking back further is repeated until the connection is cranked back to the minimal crossover node (similar to procedure specified by flow chart in Fig. 7). At the minimal crossover node the connection is rerouted to the current location of the mobile. This rerouting procedure results in an optimal connection, thereby achieving a one-phase mobile connection setup procedure.

Next, details regarding the determination of the "local view" of existing connection (P_{con}) and the desired connection segment (P_{new}) are discussed. During the process of crankback, the source route (specified by the DTLs) is updated (replaced with previously stored DTLs) at the ingress border node of the every peer group. Therefore, at the current node (node executing the crank back procedure) the source route specified by the DTLs corresponds to the local view of the current connection segment (P_{con}). Since every node has summarized information regarding the topology and loading conditions of the network, the desired connection segment between the crossover node and the current location of the mobile, P_{new}, can be computed.

3.3 Route optimization after mobile connection setup

In this section, we present a route optimization procedure that can be used to reroute a connection after a connection is set up to a mobile, thus proposing the *second phase* of *two-phase mobile location and connection setup* schemes. In the first phase, when a call setup arrives at the home switch of the mobile, a fast local rerouting of the connection to the mobile is performed [3, 5]. This results in a connection suboptimally routed through the home switch (or some other local node). In order to optimize the connection route that resulted from the fast rerouting, a route optimization procedure is executed. The need for route optimization is dictated by several factors, such as optimality of the current path, duration of the call, QoS (Quality of Service) requirements of the connection, etc. A simple criterion is the *optimality of the current path*, which can be determined at the home switch by comparing the "local view" of existing and the optimal paths between the calling switch and the current switch of the mobile. Once the need for optimizing the existing route is detected, this route optimization procedure is executed.

In this paper we assume that, when a call arrives at the home switch for a mobile that is not its home neighborhood, the home switch forwards the call to the current location of the mobile. In this scenario, existing suboptimally routed connection between the calling party and the mobile consists of two optimal connection segments, one between the calling switch and the home switch and another between the home switch and the current switch of the mobile. Since there are two connection segments, one of them can be selected and rerouted toward the target node to realize an optimally routed connection. For example, the connection segment between the calling party's switch and the home switch can be selected, and rerouted to the current switch of the mobile (target node) to form an optimally routed connection between the calling party's switch and the current switch of the mobile.

The route optimization procedure after call setup consists of *three* steps. *First*, the target node (the calling party's switch or the mobile's current switch) to which the connection segment between the other two nodes is to be rerouted is determined. *Second*, an optimal crossover node between the old and the new paths is determined, and a new segment is set up between that crossover node and the target node. *Third*, using "Tail" signals and buffering, user data is switched over from the old path to the new path.

Target node determination: Target node identifies the endpoint to which the connection segment between the other two nodes is rerouted. Fig. 12 shows the route of the connection after the call is

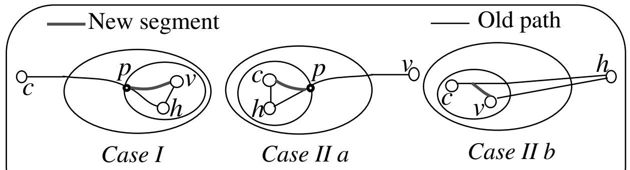

Fig. 12 Connection after the call setup is completed

set up. Prior to the route optimization phase a path already exists between the calling party and the mobile. Therefore, the calling party switch or the current switch of the mobile can be the target node. If the current switch of the mobile is selected as the target node, an optimal routed connection is achieved by setting up a new segment between the current switch of the mobile and the crossover node. In this paper, we assume that the home switch after determining the need for route optimization, selects the target node and informs the target node about the choice, by sending a RT_NODE (Rerouting Target Node) message. This message can be sent directly to the target node via a connectionless transport mechanism. One such transport mechanism is connectionless ATM (CL-ATM) [12].

Considerations such as *minimizing the length of the new segment* to be set up and availability of *support for route optimization* can influence the choice of the target node. It should be noted that the optimality of the resulting connection path is *independent of the choice of the target node*. A heuristic (based on the assumption that the distance between two nodes is inversely proportional to their ancestors-are-siblings level) with the objective of minimizing the length of the new segment is described next. In this heuristic, the target node depends on the relation between the ancestors-are-siblings levels of the nodes c, h, and v, where c, h, and v denote calling party, home switch and the visiting (current) switch of the mobile, respectively. If $a_{hc} < a_{hv}$ *(Case I of Fig. 12),* the home switch is closer to the current switch of the mobile. Therefore, visiting (current) switch can be selected as the target node. If $a_{hc} > a_{hv}$ *(Case IIa of Fig. 12),* calling party's switch can be selected as the target node (the home switch is closer to the calling party's switch). If $a_{hc} = a_{hv}$ *(Case IIb)*, visiting (current) switch of the mobile can be the target node.

Determination of the crossover node and setting up of the new segment: Procedures for the determination of the crossover node and the setting up of the new segment depend on the information about the path of the existing connection. The following three alternatives are considered:

i) No path information of the connection is known: In this case, the optimal crossover point is determined based the choice of the target node, and the relation between the ancestors-are-siblings levels of the nodes. c, h, and v. If target node is the current switch of the mobile, the optimal crossover node is determined using the procedure described in Section 3.1 by considering f as the calling party's switch, o as the home switch, and t as the current switch of the called mobile. If target node is the calling party's switch, the optimal crossover node is determined using the procedure described in Section 3.1 by considering f as the current switch, o as the home switch, and t as the calling party's switch.

In certain scenarios (*cases I* and *IIa of Fig. 12),* the above procedure computes the peer group whose border node is the optimal crossover point. In order to set up a new segment between the target node and the optimal crossover point the identity of the crossover switch is required. Since there is no information regarding the path of the connection, it may not be feasible to determine the exact crossover node (identity of the crossover switch). In this situation, we propose sending a COS_SELECT (Crossover Switch Select) message *along the existing path*. This message contains information of the peer group whose ingress node is the optimal crossover point. The COS_SELECT message is processed by each ingress border node to determine if it is the optimal crossover switch. In other words, the COS_SELECT message is sent along the existing path until in reaches the optimal crossover point. On determination of the optimal crossover switch, the new segment is setup between that switch and the target node.

ii) Hierarchical path of the connection is known: In this alternative, we assume that the hierarchical source route information stored in the ingress border nodes of peer groups during call setup are retained even after the call setup procedure is completed. This information facilitates the determination of minimal crossover node, thereby minimizing the length of the new segment.

The minimal crossover node is determined using the procedure described in section 3.1. In this procedure, the home switch determines x_{fp} based on existing path (P_{con}) and the desired path (P_{new}) (see Fig. 10). Since detailed information regarding the path is not known, it may not be feasible to determine the exact crossover node. In this situation, we propose sending a COS_SELECT message along the existing path with x_{fp} set as a parameter. The COS_SELECT message also carries the local view of the existing path. This local view is updated (replaced with previously stored DTLs) at the appropriate border node of the every peer group (similar to updates during crankback procedures). The COS_SELECT message is processed by the *entry border node* of the peer group x_{fp} (first switch on the path belonging to that peer group). If x_{fp} is the minimal crossover node, i.e., the level of x_{fp} is *L+1*, the new segment setup is initiated by x_{fp}. Otherwise, the current node determines the *new-*x_{fp} by comparing its view of P_{con} and P_{new}, updates the COS_SELECT message, and sends this message further along the existing path. Finally, on determination of the minimal crossover switch, the new segment is setup. Next, we consider the alternative of setting up the new segment, when the detailed path of the connection is available.

iii) Detailed path of the connection is known: Detailed path information facilitates the identification of the exact crossover switch and the setting up of the new segment *directly* by the target node (instead of traversing along the existing path and then setting up the new segment). In this alternative, the new segment can be set up from the target node[6] directly to the optimal crossover node or to the minimal crossover node. The identity of the optimal crossover point (ingress border node of the computed peer group) can be obtained by examining the detailed path. In order to directly set up the new segment to the minimal crossover node, the procedure to identify the minimal crossover node is integrated with the call setup procedure.

Next, details of directly setting up the new segment to the minimal crossover node are discussed. Prior to setting up the new segment, the target node compares the existing path between the home switch and the optimal crossover point (P_{con}) and the new segment between the target node and the optimal crossover point (P_{new}), thereby identifying x_{fp} (as shown in Fig. 10). After x_{fp} is identified, the new segment setup is initiated (towards the minimal crossover switch), by specifying the hierarchical path to the peer group x_{fp} (hierarchical path does not contain the details internal to the peer group x_{fp}). If the peer group x_{fp} corresponds to a switch (peer group of level $L+1$), the new segment setup is completed. Otherwise, the *ingress border node* of x_{fp} determines the *new-*x_{fp} by comparing its view of P_{con} and P_{new}, and then specifies the path towards the crossover switch that is internal to the peer group x_{fp} (the new-x_{fp} is internal to the peer group x_{fp}). This process in repeated until the new segment is setup is completed.

6.Several connections could be existing through the crossover switch to the target node. In order to uniquely identify the connection that is being route optimized, we assume that during connection setup every connection is assigned a *global identifier*, and this global identifier is sent along in the new segment setup message.

Switching of user data from the old to the new path: "Tail" signals and buffering are used to switch user data from the old path to the new path. Tail signals are special cells sent on the same virtual circuit as the user cells (in-band signals). Tail signals should be easily distinguishable from the user cells. They could for instance be special RM (Resource Management) cells [11]. These Tail signals are sent to signal the switching of user data to the new segment. Tail signals are sent in the upward direction (towards the crossover switch) by the target node and in the downward direction (towards the target node) by the crossover switch (see Fig. 13). After sending the *Tail* signals, the target node configures the switching fabric in the upward direction, and the switchover node configures the switching fabric in the downward direction. Since cells sent on the

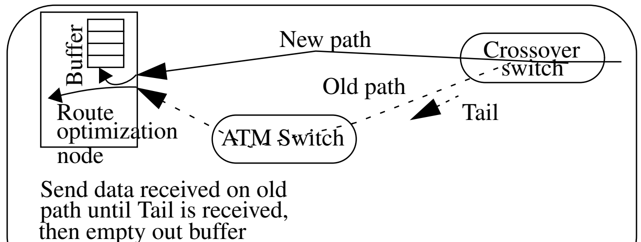

Fig. 13 Switching of user data from the old to the new path

new path may arrive before cells are depleted on the old path (given the old path is longer), the route optimization-initiating node and the switchover node need to buffer cells received on the new segment until they receive the *Tail* signal sent by the other end. After receiving the Tail signals from the other end, the buffered cells are first sent and then, the switching fabric is configured in the other direction. All the transit nodes on the old path can release the connection in the appropriate direction as and when they see the corresponding *Tail* signal. Thus, using this scheme, route optimization is achieved while *maintaining cell sequence.*

4. Analysis and discussion

In this section, a comparative performance analysis of one-phase (route optimization during call setup) and two-phase (route optimization after call setup) methods for location management is performed. As discussed earlier, there is a trade-off between call setup delay and resource utilization in one- and two-phase methods. Using a simple analytic model, we analyze this trade-off between resource utilization and connection setup delays for these methods. Other considerations are signaling load and processing requirements to handle the signaling procedures associated with these schemes. These issues are not considered in this paper. Qualitative considerations, such as the simplicity of signaling procedures, impact on standards, networking with non-mobile-enhanced ATM switches (backwards compatibility), etc., are also discussed.

4.1 Preliminaries

We first define the notation used in this analysis, and then describe a basic property of PNNI standards based hierarchical networks that is repeatedly used in the analysis.

4.1.1 Notation

Table 1 lists the symbols used in this analysis section, along with their definitions.

4.1.2 Property

For three nodes x, y and z, the following relations hold between

Table 1: Notation

Symbol	Meaning
L	Number of peer group levels in the network; level 1 is the *topmost* level and level L+1 represents individual switches in the network (see Fig. 1).
a_{ij}	The *ancestors-are-siblings level* is the level at which the ancestors of the two nodes i and j belong to the same peer group (see Section 3.1).
h, v c, p	Subscripts used to represent the home switch, visiting switch of the mobile, the calling party and the crossover node, respectively.
S	Scope indicates a stopping distance for reachability update propagations in the mobile PNNI scheme; if scope is S, reachability sent by a node i does not propagate to any node j for which $a_{ij} < S$.
N_i	For a given node i, all nodes j such that $a_{ij} \geq S$ is defined as the *neighborhood* of node i.
$m_i + 1$	Number of peer nodes in a peer group of level i for $i = 1, 2, \ldots L$; $m_i = 0$ for $i > L$.
p_i	Average length of the "shortest-path" between nodes of a peer group at level i.
D_{ij}	Distance, in terms of the number of nodes on the route, from node i to node j.
S_d	Delay incurred at a switch during connection setup.
C_d	Delay incurred at a switch during the process of cranking back a call.

their *ancestors-are-sibling levels*:

$$\text{Case I:} \quad a_{xy} < a_{xz} \Rightarrow a_{yz} = a_{xy} \qquad \text{(EQ 1)}$$

$$\text{Case II:} \quad (a_{xy} = a_{xz}) \Rightarrow a_{yz} \geq a_{xz} \qquad \text{(EQ 2)}$$

The proof of this property can be inferred from Fig. 14. If $a_{xy} < a_{xz}$ the arrangement of nodes is shown in *Case I* of Fig. 14, from which it is clear that $a_{yz} = a_{xy}$. A similar argument extends for *Case II.* Also, note that since $a_{ij} = a_{ji}$, we use these terms interchangeably.

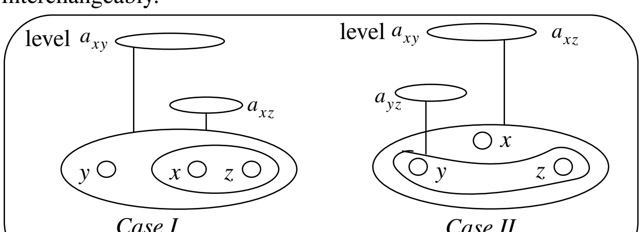

Fig. 14 Relative positions of *x, y,* and *z*

4.2 Location management schemes

In this section, we analyze the performance of one- and two-phase location management schemes. In particular, we analyze the proposed one-phase method (route optimization during call setup), and the two-phase location management scheme in which the first phase consists of a quick forwarding of the connection to the mobile by the home switch [3] and the second phase is the proposed route optimization scheme. Measures of analysis include the *connection setup delays* incurred in these schemes and the amount of *network resources* required. Since the bandwidth requirement of a connection is independent of the scheme used, we use the *number of hops* in the connection to estimate the network resource allocation required in each scheme.

In this two-phase connection setup scheme, the connection is set up with each switch believing in its reachability information. The connection setup delay, $Delay_{two}$, in this method is given in (EQ 3). *The first case*, $a_{cv} \geq S$, implies that the calling party is within the neighborhood (see Table 1) of the called mobile, where S, defined in Table 1, corresponds to the level upto which the reachability information is propagated. Therefore, the connection setup delay can be approximated by $S_d D_{cv}$, where S_d denotes the delay at each switch on the path and D_{cv} is the number of nodes on route from node c (calling party) to node v (visiting location of the mobile).

The second case of (EQ 3) corresponds to the scenario when a mobile is in its home neighborhood. In this situation the entire network has the correct reachability information about the mobile and hence the call is routed directly to the mobile. All the nodes within the neighborhood of the mobile have correct reachability information as a result of localized reachability updates. The summarized reachability information in the nodes outside the neighborhood of the mobile (default information) indicates that the mobile is in its home neighborhood. This information is sufficient to route a call directly to the mobile, since in PNNI networks hierarchical source routing is used to route the connection.

The third case corresponds to the scenario in which the call is first routed to the home of the called mobile (h) and is then forwarded by the home to the current location of the mobile (v). Therefore, the connection setup delay comprises of the delay in setting up the connection from the calling party switch (c) to the home of the called mobile and the setup delay for the connection between the home and the visiting location of the mobile.

$$Delay_{two} = \begin{cases} S_d D_{cv} & a_{cv} \geq S \text{ independent of } a_{hv} \\ S_d D_{cv} & a_{hv} \geq S \text{ independent of } a_{cv} \\ S_d (D_{ch} + D_{hv}) & a_{hv} < S \text{ and } a_{cv} < S \end{cases} \qquad \text{(EQ 3)}$$

Next, we provide a method for estimating D_{ij}, and derive the expression for estimating the average connection setup delay. The distance D_{ij} between nodes i and j is approximated as

$$D_{ij} = \prod_{k = a_{ij}}^{L} p_k \qquad \text{(EQ 4)}$$

where p_k is the average (among all node pairs) length of the "shortest-path" between nodes of a peer group at level k. For the worst case performance, the maximum length (among all node pairs) of the "shortest-path" can be taken. By this definition, the distance D_{ij} between nodes i and j depends on the value of their *ancestors-are-siblings level* a_{ij}. From the property described in (EQ 1)-(EQ 2), we know the relationship between the *ancestors-are-siblings levels* of different nodes. Using (EQ 1) through (EQ 4), an expression to estimate the average connection setup delay, $AvgD_{two}$, is obtained in (EQ 5).

$$AvgD_{two} = \sum_{i=S}^{L+1} P(a_{cv}=i)S_d D_{cv} + \sum_{i=1}^{S-1} P(a_{cv}=i) \quad \text{(EQ 5)}$$

$$\left\{P(a_{hv} \geq S)S_d D_{cv} + \sum_{j=1}^{i} P(a_{hv}=j)(S_d(D_{hv}+D_{hv})) + \right.$$

$$\sum_{j=i+1}^{S} P(a_{hv}=j)S_d(D_{cv}+D_{hv})$$

$$\left. + P(a_{hv}=i)\left\{\sum_{j=i}^{S-1} P\left(\frac{a_{ch}=j}{a_{cv}=a_{hv}=i}\right)S_d(D_{ch}+D_{hv})\right\}\right\}$$

In the two-phase scheme being analyzed, i.e., call forwarding scheme, since the connection is not "backtracked" at any point, the *number of hops* in the connection can be estimated from equations (EQ 3) and (EQ 5) by setting the value of S_d to be 1. The expression to estimate the average number of hops in a connection, $AvgR_{two}$, is given in (EQ 6). We expect that the second phase, i.e., route optimization phase, is performed soon after the connection setup is completed. Hence, the additional network resources needed for extra hops in a connection are only needed for a short period of time.

$$AvgR_{two} = \sum_{i=S}^{L+1} P(a_{cv}=i)D_{cv} + \sum_{i=1}^{S-1} P(a_{cv}=i)\left\{P(a_{hv} \geq S)D_{cv}\right. \quad \text{(EQ 6)}$$

$$+ \sum_{j=1}^{i} P(a_{hv}=j)(D_{hv}+D_{hv}) +$$

$$\sum_{j=i+1}^{S} P(a_{hv}=j)(D_{cv}+D_{hv})$$

$$\left. + P(a_{hv}=i)\left\{\sum_{j=i}^{S-1} P\left(\frac{a_{ch}=j}{a_{cv}=a_{hv}=i}\right)(D_{ch}+D_{hv})\right\}\right\}$$

In the one-phase connection setup scheme, like the two-phase scheme, the connection is set up with each switch "believing" in its reachability information. But on connection arrival to the home switch of the mobile, instead of forwarding the connection, the connection is cranked back to an optimal crossover point and rerouted to the mobile. The connection setup delay, $Delay_{one}$, in this method is given in (EQ 7). The first two cases of this equation are the same as those of (EQ 3). The third case corresponds to the scenario in which the connection proceeds to the home of the called mobile ($S_d D_{ch}$), then cranked back from the home to the crossover node ($C_d D_{hp}$, where C_d denotes the delay at each node to perform crankback), and then rerouted from the crossover node (p) to the current location of the mobile ($S_d D_{pv}$).

$$Delay_{one} = \begin{cases} S_d D_{cv} & a_{hv} \geq S \quad \text{independent of } a_{cv} \\ S_d D_{cv} & a_{cv} \geq S \quad \text{independent of } a_{hv} \\ S_d(D_{ch}+D_{pv}) + C_d(D_{hp}) & a_{hv} < S \text{ and } a_{cv} < S \end{cases} \quad \text{(EQ 7)}$$

Using the approximation given in (EQ 4) (the distance D_{ij} between nodes i and j depends on the value of their *ancestors-are-siblings level* a_{ij}), the expression to estimate the average call setup delay (given in (EQ 8)) can be obtained from (EQ 7).

$$AvgD_{one} = \sum_{i=S}^{L+1} P(a_{cv}=i)S_d D_{cv} + \sum_{i=1}^{S-1} P(a_{cv}=i)\left\{P(a_{hv} \geq S)S_d D_{cv}\right. \quad \text{(EQ 8)}$$

$$+ \sum_{j=1}^{i} P(a_{hv}=j)(S_d(D_{ch}+D_{pv}) + C_d D_{hp}) \quad +$$

$$\sum_{j=i+1}^{S} P(a_{hv}=j)(S_d(D_{ch}+D_{pv}) + C_d D_{hp})$$

$$\left. + P(a_{hv}=i)\left\{\sum_{j=i}^{S-1} P\left(\frac{a_{ch}=j}{a_{cv}=a_{hv}=i}\right)(S_d(D_{ch}+D_{pv}) + C_d D_{hp})\right\}\right\}$$

Applying the properties described in (EQ 1)-(EQ 2) (which give the relationship between the *ancestors-are-siblings level* of different nodes), (EQ 8), can be rewritten as below:

$$AvgD_{one} = \sum_{i=S}^{L+1} P(a_{cv}=i)S_d D_{cv} + \sum_{i=1}^{S-1} P(a_{cv}=i) \quad \text{(EQ 9)}$$

$$\left\{P(a_{hv} \geq S)S_d D_{cv} + \sum_{j=1}^{i} P(a_{hv}=j)(S_d(D_{hv}+D_{cv}) + C_d D_{hv}) \quad + \right.$$

$$\sum_{j=i+1}^{S} P(a_{hv}=j)(S_d(D_{cv}+D_{hv}) + C_d D_{hv})$$

$$\left. + P(a_{hv}=i)\left\{\sum_{j=i}^{S-1} P\left(\frac{a_{ch}=j}{a_{cv}=a_{hv}=i}\right)(S_d(D_{ch}+D_{cv}) + C_d D_{ch})\right\}\right\}$$

In the one-phase crankback scheme, the connection is rerouted during call setup such that the resulting connection traverses an optimal path. Hence, the number of hops in the connection depend only on the optimal path between calling party and the current location of the mobile (D_{cv}). The expression to estimate the average number of hops in a connection, $AvgR_{one}$, is given in (EQ 10).

$$AvgR_{one} = \sum_{i=1}^{L+1} P(a_{cv}=i)D_{cv} \quad \text{(EQ 10)}$$

4.3 Numerical results

In this section, we *quantitatively compare* the one-phase (route optimization during call setup) and the two-phase (route optimization after call setup) schemes. The measures of comparison include the average and maximum call setup delays and the amount of network resources required (number of hops in the connection).

Input Data: Values of input parameters assumed for this numerical computation are shown in Table 2 (see Table 1 for definitions of these parameters). Input parameters S_d and C_d are estimated from measured data [10]. The probability distributions used for calling patterns and user location are presented in the Appendix. We observe that the exact numerical results are dependent on the exact values chosen for these parameters. However, the trends observed are more or less independent of these values. Sensitivity of the comparative results to these input parameters has been studied but not included in this paper due to space considerations.

Table 2: Input parameters

Parameter	Value
m_i, $1 \leq i \leq L$	10
p_i, $1 \leq i \leq L$	3
L	5
S_d	2.6 ms
C_s	2.3 ms

The variation of connection setup delays (average and maximum delays) and the resource requirement (average and maximum number of hops) with the variation of S (which defines the extent of the "neighborhood") for the one- and the two-phase methods is shown in Fig. 15. From the plot showing the variation in *connection setup delay*, we observe that the one-phase scheme (route optimization during call setup) incurs a higher call setup delay than that of the two-phase scheme. It should be noted that the call setup delay depends on the locations of the calling party, called mobile and the home of the called mobile, and the topology and loading conditions of the network. The estimates of the average delays for both the methods are comparable, but there is significant difference in the maximum delay incurred in both these methods. In the one-phase scheme, the procedure to crankback to the minimal crossover node will help in reducing the average call setup delay. The average delay in the one-phase scheme is approximately 10% higher than of the forwarding scheme ($S = 6$), but the maximum delay in the one-phase scheme is approximately 30% higher than that of the forwarding scheme. This worst case scenario occurs when the calling party, the home of the called mobile and the current location of the mobile are far-away from each other. In this scenario, the call is routed to the home of the called mobile, cranked back all the way to the calling party switch and then rerouted to the current location of the mobile in the one-phase scheme leading to large delays. At lower values of S (neighborhood to which reachability data is propagated is larger), the call setup delays are lower, since most of the calls are routed directly to the mobile, instead of routing to the home and then rerouting to the mobile. When $S = 1$, since all the calls are routed directly to the mobile in both the schemes, the call setup delays are minimum. The disadvantage of having a large neighborhood (i.e., low value of S) is that, cost of updating reachability information regarding the location of a mobile, when a mobile moves from one location to another, is very high [3].

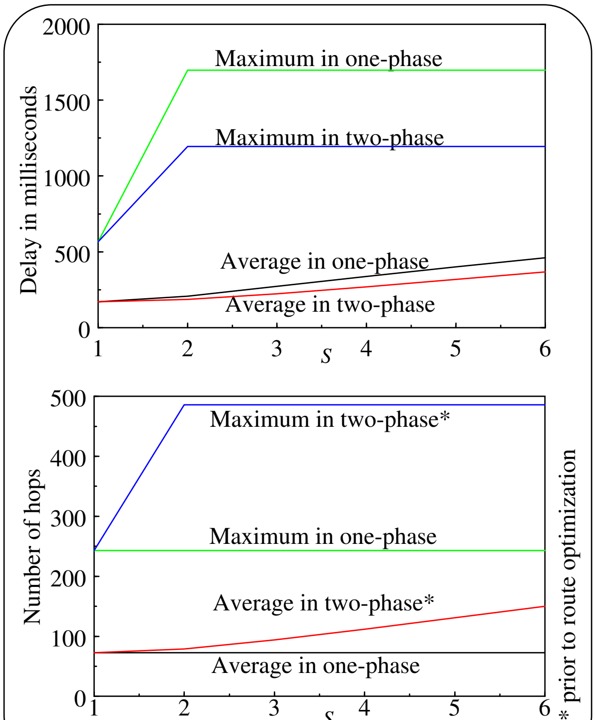

Fig. 15 Delay and resource comparisons

The variation of the *number of hops in a connection* for the two methods shows similar but opposite trends than the corresponding variation in delays. The number of hops in connections resulting from the two-phase scheme are higher than that in the one-phase scheme. This is expected trend since the one-phase scheme results in optimal connection paths. The plot of average number of hops in a connection show that significant improvements in resource utilization can be achieved by the one-phase scheme. In the worst case (the plot for maximum number of hops), the two-phase forwarding scheme (prior to route optimization) requires twice the amount of communication resources as the one-phase scheme (route optimization during call setup). In the two-phase scheme, the number of hops in connection after the route optimization phase is equal to the number of hops in a connection set up by the one-phase scheme. Hence, in the two-phase scheme, route optimization procedure helps in reducing the number of hops in the original connection significantly. At low values of S (i.e., neighborhood to which reachability data is propagated is large), the number of hops in a connection are lower in the two-phase method, since most of the calls are routed directly to the mobile, instead of routing to the home and then rerouting to the mobile. Since the one-phase method results in a optimally routed connection, the number of hops in a connection does not vary with the variation in S.

Besides the extra amount of resources required prior to route optimization phase, the two-phase call setup method has an overhead of route optimization signaling and buffering of cells. In order to minimize the amount of resources required for a connection and to avoid the additional signaling overhead of the second phase in the two-phase scheme, it is preferable to execute route optimization during call setup. However, the high worst case call setup delays associated with the one-phase scheme may not be acceptable for certain applications. Therefore, in such situations a fast local rerouting of the connection is required, and in order to minimize the number of hops in a connection, route optimization should be executed subsequently. Since the performance (connection setup delay and the amount of communication resources) of these schemes depend on the topology of the network, and the exact locations of the calling party, called mobile and home of the called mobile. We propose that, on call arrival at the home of the called mobile, the home switch decides which scheme to execute, the one-phase scheme or the two-phase scheme. Some of the factors on which this decision can be based include: duration of the call, QoS (Quality of Service) requirements of the connection, location of the calling party and the called mobile, topology and loading conditions of the network, the associated call setup delays, number of hops in the connection, and mobility support offered at the switches (discussed in Section 4.4).

4.4 Mobility support

In this section, we discuss the issues of mobility support needed at switches, backward compatibility with existing and planned infrastructure, and the buffering requirements.

In order to facilitate the easy integration of wireless ATM into the existing and planned fixed ATM networks, an approach of incremental deployment of *mobile-enhanced switches* needs to be taken. Therefore, it is important to consider isolating mobility-related features to mobile-enhanced switches. Examples of mobility-related features are support of crankback for rerouting calls to mobiles, support for the transport of mobility related signaling messages along the existing path, support for buffering during route optimization, etc. Call forwarding during connection setup is easily implementable. In the call forwarding method, only the home switch of the mobile needs to be mobile-enhanced. In the one-phase scheme, additionally, the crossover node has to be a mobile-enhanced switch and the intermediate nodes need to support mechanisms for the transport of mobility related messages. In order to determine the minimal crossover node (see Section 3.1), the intermediate nodes (i.e., exit node of x_{fp}) that compute the new crossover node have to be mobile-enhanced switches.

Next, we address the question of how best to support route optimization with limited deployment of mobile-enhanced switches. In a network with limited deployment of mobile-enhanced switches, if the optimal crossover point is not a mobile-enhanced switch, a mobile-enhanced switch that occurs between the optimal crossover node and the far-end is selected. If no such mobile-enhanced node is available, shorter but a suboptimal path can be obtained by choosing a node between the optimal crossover point and the target node as the crossover node. Another issue to be addressed is, how does the home switch (or other switches) know about the availability of mobile-enhanced switches on the path (before executing the route optimization procedure). A possible approach is to maintain a list of mobile-enhanced switches on the existing path. This list can be obtained during call setup, by having a list (DTL) in the SETUP message to which the mobile-enhanced switches on the path add their address.

Buffering is required to prevent out-of-sequence cells and cell loss. For some types of traffic that are loss insensitive, such as real-time audio and video, buffering may not be required. Other types of traffic, such as data traffic, are typically loss sensitive. Though loss-sensitive traffic may be transported over a reliable transport protocol, such as TCP (Transmission Control Protocol), it is desirable to prevent cell loss at the lower layer. During the route optimization procedure, buffering is required to switch the flow of data from the old segment to the new segment. The amount of buffer space required is equal to number of cells received at the crossover node during the time required to flush the old segment.

5. Conclusion

In this paper, we proposed an *algorithm for optimizing the route of a connection* that becomes suboptimal due to location-based reroutes, for mobile ATM networks based on the PNNI standard. This algorithm uses the hierarchical route information of the connection and the summarized topology and loading information of the network, to determine a “crossover node” such that adjusting the connection from that crossover node results in an optimally routed connection. We demonstrated how this route optimization procedure can be executed during call setup, thus proposing a *one-phase connection setup* scheme. We also demonstrated how this route optimization procedure can be executed after a connection is setup, thus proposing the *second phase* of the *two-phase connection setup* schemes. This route optimization procedure can also be used to perform rerouting of connections during/after mobile handoffs. A comparative performance analysis of one- and two-phase connection setup schemes is presented. One-phase scheme incurs higher *call setup delays*, but results in an optimal route during connection setup. On the other hand, two-phase schemes incur lower call setup delays, but result in connections with suboptimal routes, prior to the execution of the route optimization phase. The route optimization phase in the two-phase methods shows a significant reduction in the amount of network resources required for a connection.

References

[1] D. Raychaudhuri and N. D. Wilson, “ATM-Based Transport Architecture for Multiservices Wireless Personal Communication Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 12, pp. 1401-1414, Oct. 1994.

[2] The ATM Forum Technical Committee, “Private Network-Network Specification Interface v1.0 (PNNI 1.0),” March 1996, af-pnni-0055.000.

[3] M. Veeraraghavan and G. Dommety, “Mobile Location Management in ATM Networks,” *IEEE Journal on Selected Areas in Communications*, under review, August 1996. (Short form appeared in Proc. of ICC '97, June 8-12, Montreal, Canada).

[4] A. Acharya, S. Biswas, L. French, J. Li, D. Raychaudhuri, “Handoff and Location Management in Mobile ATM Networks,” *Proc. of the 3rd International. Workshop on Mobile Multimedia Communications*, Princeton, New Jersey, September 1996.

[5] A. Ayyagari, J. Harrang, S. Ray, “Call Establishment/Termination in Wireless PNNI,” ATM_Forum 96-1410, October 1996.

[6] Charles Perkins, “IP Mobility Support,” RFC 2002, October 1996.

[7] David B. Johnson and Charles Perkins, “Mobility support in IPv6,” Internet Draft, draft-ietf-mobileip-ipv6-02.txt, November 1996. Work in progress.

[8] David B. Johnson and Charles Perkins, “Route optimization in Mobile IP,” Internet-Draft, draft-ietf-mobileip-optim-04.txt, February 1996. Work in progress.

[9] G. Dommety, M. Veeraraghavan, and M. Singhal, “Route Optimization in Mobile ATM Networks,” Technical Report, Computer and Information Science, Ohio State University.

[10] M. Veeraraghavan, G. L. Choudhury, and M. Kshirsagar, “Implementation and Analysis of PCC (Parallel Connection Control)”, *IEEE INFOCOM,* Kobe, Japan, April 1997.

[11] The ATM Forum Technical Committee, “ATM User-Network Interface (UNI) Signaling Specification Version 4.0,” January 1996, ATM Forum/95-1434R9.

[12] M. Veeraraghavan, P. Pancha and G. Dommety, “Connectionless ATM using an ATM Switch Router,” *Proc. of European Conference on Multimedia Applications, Services and Techniques (ECMAST) 1997*, May 21-23, Milan, Italy.

Appendix

In this section, we describe the probability distributions, $P(a_{cv})$, $P(a_{ch})$, $P(a_{hv})$, and the conditional probabilities used to compute the average call setup delay and the number of hops in a connection, in (EQ 5), (EQ 6), (EQ 9), and (EQ 10).

We assume that the distribution of calls to a mobile follows a uniform distribution, i.e., it is equally probable for a mobile to receive a call originating at any switch. So the probability of a call originating from a switch for which $a_{cv} = x$ depends on, the number of nodes, for which $a_{cv} = x$ and the total number of nodes in the network. At level x, we exclude the peer group of level $x+1$ to which the called mobile belongs, and determine the number of nodes at level L that belong to other peer groups of level $x+1$. Thus, the probability $P(a_{cv} = x)$ is given by:

$$P(a_{cv} = x) = \frac{\left(\prod_{i = x+1}^{L} (m_i + 1)\right) m_x}{\prod_{i=1}^{L} (m_i + 1)} \qquad \text{(EQ 11)}$$

Since a call can originate from anywhere, the distribution of a_{ch} is the same as the distribution of a_{cv}. The conditional probability $P\left(\frac{a_{ch} = j}{a_{cv} = a_{hv} = k}\right)$ is given by (EQ 12).

$$P\left(\frac{a_{ch} = j}{a_{cv} = a_{hv} = k}\right) = \frac{\left(\prod_{i=j+1}^{L} (m_i + 1)\right) m_j}{\prod_{i=k}^{L} (m_i + 1)} \qquad \text{(EQ 12)}$$

Next, we consider the distribution of a_{hv}. The choice of this distribution is based on the premise that a majority of the mobiles roam in and around their respective homes. We choose a distribution such that the probability of being located close to home is high. The distribution is such that the probability of being located at the home switch is f, in the L^{th} level peer group of the home switch as f^2, in the $(L-1)^{th}$ peer group as f^3, etc. When $L \geq 1$, then $f < 1$ and the probability of being located further and further away from the home switch decreases. Given that a mobile is located somewhere in the network,

$$f + f^2 + f^3 + \ldots + f^L = 1 \qquad \text{or} \qquad \frac{1 - f^{L+1}}{1 - f} = 2 \qquad \text{(EQ 13)}$$

Under this assumption, the distribution of a_{hv} is given by (EQ 14).

$$P(a_{hv} = x) = f^{L - x + 2} \qquad \text{(EQ 14)}$$