Proceedings of *Conference on High Performance Switching and Routing (ICATM'2000)*,
pp. 491-499, June 2000, Heidelberg, Germany.

Rerouting for Handover in Mobile Networks with Connection-Oriented Backbones: An Experimental Testbed

Andreas Festag, Theodoros Assimakopoulos, Lars Westerhoff, Adam Wolisz
Technical University Berlin, Germany

Abstract

The rerouting of connections for handover in a broadband mobile cellular network is investigated. We address networks with a connection-oriented backbone, which supports quality-of-service (QoS). Moreover it is assumed an *IP-style multicast* on top of the connection-oriented network. We advocate to utilize the IP-style multicast in order to reroute connections for handover. Three rerouting schemes are proposed, which are based on the IP-style multicast. One of the schemes realizes a predictive approach, where a call is pre-established to potential new base stations by setting up a multicast tree with neighboring base stations as leaf nodes. This approach reduces the handover latency caused by rerouting to an absolute minimum and improves communication with strict time constraints and frequent handover.
For experimental investigations of the rerouting schemes a testbed is set up, which is based on an open, non-proprietary, experimental networking environment. Its signaling software offers a direct many-to-many communication, rather than requiring one-to-many overlays and the switching hardware supports scalable and efficient multicast by cell-recycling. The testbed is used to proof the feasibility of the approach and for performance evaluation of the rerouting scheme.

1. Introduction[1]

Handover is an important feature in mobile cellular networks. It describes the mechanism that transfers the association of a mobile end system from one base station - which is presently active - to a new base station. Handover is applied when a user moves through the coverage area of a cellular network and crosses cell boundaries. A handover involves not only the mobile end system and the base stations, but also the backbone, which interconnects the base stations. For standard wired networks, usually fixed end systems are assumed, which means that they do not change their association to a network access point. Therefore changes in standard protocols are necessary to allow mobile end systems. Handover poses many challenges, one of them is the dynamic rerouting of ongoing connections as the mobile hands over from one base station to another. This work is focused on rerouting and we advocate to utilize IP-style multicast in order to reroute connections for handover.
A typical mobile cellular network is shown in **Fehler! Verweisquelle konnte nicht gefunden werden.**. A cell is defined by the radio coverage of a single base station. The base stations are part of a wired, high-speed, connection-oriented backbone. Links are provided to their neighboring base stations, to fixed hosts and gateways via a single or several network nodes. A mobile end system may move freely within the coverage of a wireless cell. When it leaves the cell the connection has to be rerouted. A common approach is to tear down one part of the connection and set up a new part - this is well known as *partial reestablishment* [1] of a connection. Rerouting is performed in that network node, where the old and the new route diverge. In large topologies, this network node might be distant from the base station and the backbone is strongly effected from handover signaling. To restrict the impact of rerouting on the backbone network a *handover domain* is defined. This is the area, where the rerouting is carried out via partial reestablishment; for handover between wireless cells belonging to different handover domains a *full reestablishment* [1] of the connections is necessary.

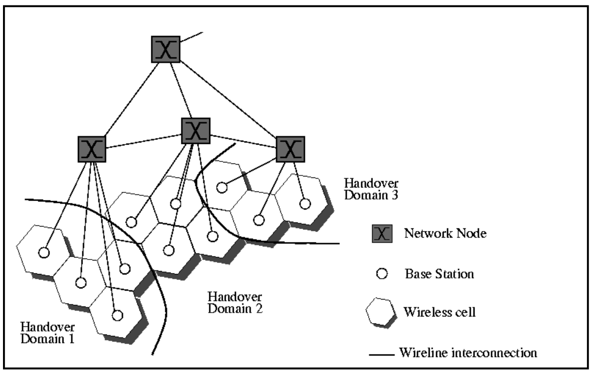

Fig.1 Topology of a mobile cellular network

[1] This work has been partially supported by a grant from the BMBF (German Ministry for Science and Technology) within the priority program ATMmobil.

In the following paper a backbone network is assumed, which is connection-oriented and supports Quality of Service (QoS) by making resource reservations. The backbone network is used to support communication between base stations in IP based communication systems. It is assumed that an *IP-style multicasting* [13], on top of the connection-oriented backbone is available, which is used for group ware applications, such as real-time video conferencing tools and multipoint data distribution services. The IP-style multicasting enables a dynamic many-to-many communication with flexible signaling for joining/leaving the multicast-group, as it is common in IP networks.

We define a mobile-multipoint call *MCALL* - that is a abstraction for a session between multiple (more than two) end points, which combines several parallel connections logically. End points can be added or dropped to/from the *MCALL* dynamically. True IP multicasting supports multipoint-to-multipoint communication inherently and due to the connectionless transport of IP packets, a sending IP host does not need to be explicitly aware of joining a multicast group. Contrary in connection oriented networks the communication between a source and destination requires a connection between the sender and the receivers.

It has to be noted, that the assumed IP-style multicast does not rely on standard ATM [18]. As it is well known multicasting for standard ATM is based on unidirectional point-to-multipoint connections and root-initiated (ATM Forum UNI 3.0) and leaf-initiated (ATM Forum UNI 4.0) join operations, respectively. Also, the assumed IP-style multicasting does not use a multicast server, which employs a mesh of one-to-many connections.

In the paper only the backbone network is considered. The connections, as well as the multicast end in the base station. The communication between mobile end systems and base stations is assumed to be separated and specific wireless protocols are incorporated for the wireless hop. In particular, the data sent via the wireless link have to be mapped to the connections in the backbone. However, this is beyond the scope of the paper.

In this paper, we advocate to utilize the IP-style multicasting in mobile networks with a connection-oriented backbone in order to reroute connections for handover. The IP-style multicasting enables to compose the rerouting mainly of *add-* and *drop-end point* operations. An important benefit is that a network node does not have knowledge about an ongoing handover process. Additionally adding/dropping an end point to/from a *MCALL* makes the rerouting of multiple parallel connections more efficient than rerouting them sequentially, since several operations are combined into a single *transaction*.

Three rerouting schemes are proposed. One of the schemes applies a predictive approach to reroute a connection, where the connection is pre-established to potential new base stations (e.g. neighboring base stations) in advance. It reduces the service interruption to an absolute minimum. The predictive approach provides a useful improvement for communication with strict time constraints in a cellular network with frequent handover [2], [8], [9], [10]. The maintenance of the connection tree to some or all of the neighboring base stations can be efficiently realized by adding/dropping single base stations from the *MCALL*.

Various handover schemes have been proposed, which can be roughly categorized into hard, soft and predictive handover schemes. For rerouting as part of the overall handover procedure, new schemes are defined, since they are based on the ability to join/leave a multicast-group and rely on the order of their execution. A *hard* rerouting performs a drop-/add-end point operation, a *soft* handover makes an add-/drop-end point operation. The *predictive* scheme carries out their operation in advance, preferring a add/drop order. As a matter of course the schemes adopts some of its ideas from [1], [2], [4], [6] and [8].

The rerouting for handover in connection-oriented networks has been extensively investigated, see e.g. [1]-[8] most of them for Wireless/Mobile ATM systems. Nevertheless, there are only a few experimental investigations that compare the various schemes ([5], [6]). So far no experimental investigations with an infrastructure that support IP-style multicast over connection-oriented networks have been performed. Moreover to the best as for the authors knowledge, this testbed is the *first* treatment for experimental investigations of handover, which applies a predictive approach in connection-oriented networks to minimize the handover duration.

For experimental investigations a testbed is set up. The testbed is based on an open, non-proprietary networking environment [13]. Its signaling software offers a direct many-to-many communication, rather than requiring one-to-many overlays and the hardware bases on a switching fabric which supports scalable and efficient multicasting by cell-recycling. The control is separated from the switching fabric, which enables a remote control, a modification of the signaling software for mobility purposes and a flexible environment for development, testing and measurements. The goal of the experimental investigations is to verify the rerouting approach and to compare the different rerouting schemes with regard to QoS parameters like rerouting latency and signaling overhead in the backbone.

This paper is organized as follows. In the next section, the call and connection management architecture is described and three different rerouting schemes are proposed. Then the experimental testbed with its hardware and software modules is described. Finally the handover experiments and the selected scenarios are described and performance issues are discussed.

2. Call and connection management architecture and rerouting schemes

In this section the rerouting schemes are explained. First the assumed call and connection management architecture is described as far as it is relevant for the rerouting schemes and features offered by the IP-style multicast are outlined.

The network architecture - as shown in figure Fig. 1 - consists of network nodes and node controllers, base stations, mobile hosts, location server, correspondent hosts, gateways. A network node offers transmission resources and is controlled by a node controller. A node controller in turn can be assigned to a single or to several network nodes. In the latter case several network nodes are managed by a *Node Management* which abstracts the network nodes so that they are logically presented as a single node. The Node Management manages connections within the node. The node controller is responsible for call management, connection management and the management of local resources.

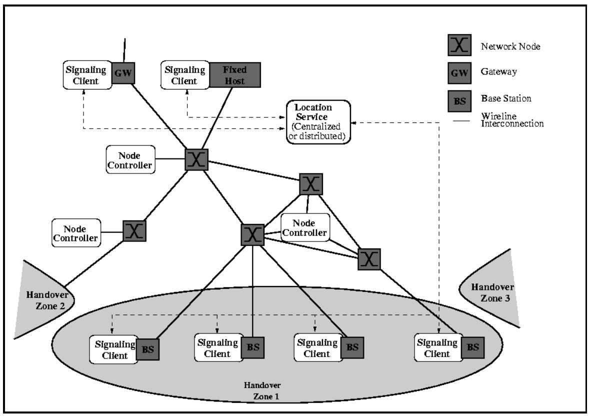

Fig. 1 Call and connection management architecture

It is assumed, that a base station acts as a connection end point and has - as well as the correspondent host and gateway - an associated signaling client. This signaling client interacts with the call management in the node controller. Thus, signaling clients do not have direct access to network resources. Instead the call management provides an interface to the network resources, which can be considered as a User Network Interface (UNI)[2]. In turn the call management uses the facilities provided by the connection management to perform the network operations requested by the signaling client.

In the backbone network an IP-style multicast is supported. It offers flexible one-to many, many-to-one and many-to-many communication. We identify the following features of multicast which can be utilized for rerouting [Table 1]:

Feature	*Description*
Open/Close	Establishment, modification and release of channels between senders and transmitters of a multicast group
Add/Drop	Add and Drop hosts to/from a multicast group. This includes root and leave initiated join/leave and the maintenance of the multicast tree.
Trace	Tracing membership of hosts in a multicast group. This might include geographical knowledge about neighboring base stations.
Third party signaling	Perform signaling on behalf of another host.
Resource reservation	Reserving resources in network nodes. Possibly pre-reservation of resources without using the resources.

Table 1 Features of IP-style multicast

As a first step an existing implementation of an IP-style multicast is investigated, which offers all of the mentioned features from the above table (except the geographical knowledge about neighboring base stations). It is based on CMAP, version 3.0 [16], [17]. We have extended the implementation by a tracing functionality, which is already included in the specification.

The core of the IP-style multicast scheme is the MCALL – as already defined a *mobile multipoint call*. *MCALLs* can be created, modified and deleted. A *MCALL* is set up between a signaling client and the call management of a node controller. It contains one or several parallel connections. Several signaling clients can participate in a *MCALL*, which might be connected to different node controllers. The signaling clients can add, modify and drop end points dynamically to/from a MCALL. Moreover, once a MCALL has been opened, connections can be added, modified and dropped to/from the MCALL. This facilitates a dynamic n-way multipoint-to-multipoint communication. If only two clients participate in a *MCALL*, this case is considered as a specific case of a *MCALL*. To establish a *MCALL* in this architecture, a signaling client requests the associated node

2 Used in a common sense, not ATM Forum UNI.

controller to open a *MCALL* and the signaling client becomes the *root* node of the *MCALL*. The request contains the address of the correspondent signaling client(s) and the desired connection(s) with QoS parameters. The call management in the switch controller requests the connection management to set up the connection(s) to the network. Then the *MCALL* request is routed to the correspondent signaling client, while resources are reserved on each link and a multicast tree within the network is established. The request returns successfully when the connections with the desired QoS can be accepted.

The assumed IP-style multicasting enables that joining/leaving a *MCALL* can be initiated by the root or by signaling clients, which already participate in the *MCALL*. Moreover a signaling client can request to join/leave a *MCALL* on behalf of another signaling client. This is referred to *third party* signaling.

One important feature in this architecture is that the route information is distributed in the network and no centralized call and connection management exists. The distributed architecture impacts the design of the rerouting schemes. In some handover schemes [6], [7] [12] the anchor point can change: It is determined at connection setup and remains fixed for the lifetime of the connection or the anchor point is dynamically selected for each handover. This has some benefits, but requires, that all switches along the route are capable for rerouting and increases complexity and signaling overhead for selection of a optimal anchor point. However, the above described architecture relies on multipoint communication, where end points can dynamically added and dropped to/from a multicast tree. Maintaining a multicast tree is strongly dependent of the multicast routing protocol. In our case selecting a proper network node to act as an anchor point is inherently given by the multicast routing protocol and can be fixed or dynamic.

Rerouting consists mainly of two main sequential operations:

- Adding a new signaling client to a *MCALL*
- Dropping the old signaling client from the *MCALL*.

This can be accomplished in three ways [Fig. 2]:

- *Hard rerouting:* The old base station is dropped from the *MCALL* before the new base station is added.
- *Soft rerouting:* The new base station is added to the *MCALL* before the old base station is dropped.
- *Predictive rerouting:* The *MCALL* to potential new base stations is pre-established. Therefore potential new base stations (which form a set) are added to the *MCALL* at call setup. The set of potential new base stations has to be updated after handover: some base stations have to be added, some dropped.

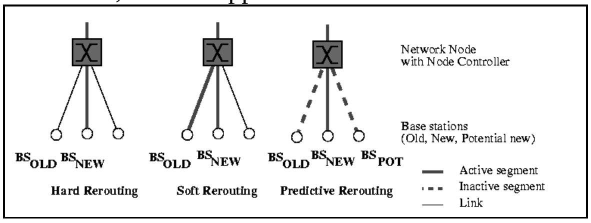

Fig. 2 Rerouting schemes

So far the rerouting schemes are described for forward handover, where signaling is sent via the link to the new base station. When the schemes are applied for backward handover, the signaling is sent via the old link. The main difference is the following: For forward handover the new base station carries out the add- and drop-end point operations, whereas for backward handover the old base station performs the operations. Therefore it is not necessary to design a new rerouting scheme for backward handover.

It has to be noted that we assume a location management: In principal a mobile end system registers with the base station and the current location of the mobile can be determined by a location service. For investigations of rerouting, location management is not our concern and it is not relevant whether the location service is centralized or distributed.

2.1 Hard rerouting

Hard rerouting is the easiest scheme. There is only a single connection end point per mobile at the same time. Since no data are duplicated, the data overhead is minimized. Two drawbacks can be identified: a) The duration of service interruption is relatively long, because the service is interrupted until the old base station is dropped and the new base station added and b) there is a gap between the drop- and the add-end point-operation and data might get lost.

The steps involved in the hard rerouting scheme (only forward handover considered) are (see Fig. 3):

- The mobile host (MH) sends a handover request message (`HO_REQ`) to the actual base station. In the considered case, this is the new base station (BS_NEW). The handover request contains the *MCALL* identification and the address of the old base station.
- The new base station traces the *MCALL* with a `TRACE_CALL_REQ` and `TRACE_CALL_RESP` message. The response message contains the end points and connections with its QoS attributes of the *MCALL*. The base station checks if resources are available for the connections.

- The new base station requests to drop the old base station from the *MCALL* (DROP_EP_REQ). The network node, which acts as an anchor point (anchor node) informs the old base station (BS_OLD) about dropping the end point, which is responded by the base station (INVITE_DROP_EP_REQ, INVITE_DROP_EP_RESP).

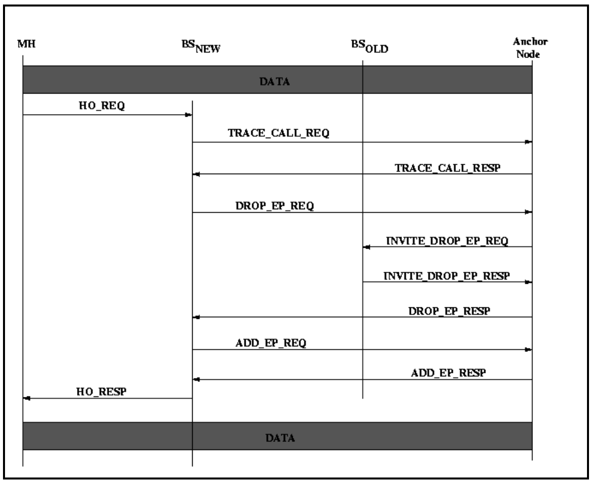

Fig. 3 Basic scheme for hard rerouting

- The anchor node responses to the new base station, that the old base station was dropped (DROP_EP_RESP).
- The new base station requests to add the own end point to the call (ADD_EP_REQ) and the anchor node responses with an ADD_EP_RESP message.
- The new base station sends and HO_RESP message to the mobile hosts and indicates that the handover is finished.

2.2 Soft rerouting

The soft rerouting scheme changes the sequence of drop-end point and add-end point-operation from hard rerouting. Therefore for a limited time at least more than two base stations belong to the *MCALL*. Considering backward handover a basic assumption for soft rerouting is, that the mobile host has connectivity at least to the old and to the new base stations simultaneously and the radio cells of the base stations overlap.

The main benefit of the scheme is that the handover response message (HO_RESP) can be sent to the mobile when the new end point (new base station) is added to the *MCALL*. After then the end point of old base station is dropped from the *MCALL*, but this operation does not contribute to the duration of the service interruption. Therefore the handover duration is minimized. The data overhead is higher than in the hard rerouting scheme since data are duplicated and sent to the new and old base station for a short time. Moreover data can be duplicated or mis-ordered in the mobile host.

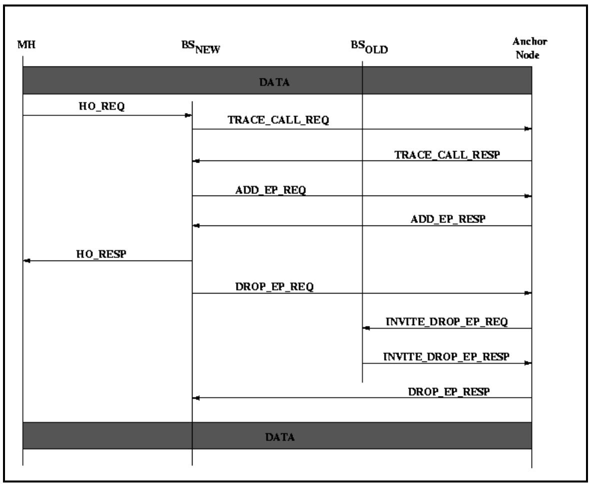

Fig. 4 Basic scheme for soft rerouting

The steps of the soft rerouting scheme (Fig. 4) are very similar to the hard rerouting scheme, except that adding the new end point (ADD_EP_REQ, ADD_EP_RESP) and dropping the old end point (DROP_EP_REQ, DROP_EP_RESP) are exchanged and the handover response message is sent after the add-end point operation.

2.3 Predictive rerouting

The predictive rerouting scheme adopts some ideas from [2], [8], [9]. In this scheme base stations, which are potential candidates for handover of mobile hosts, form a set of base stations. When a *MCALL* is set up to a base station, the end points of the base stations, belonging to the set, will also be added to the *MCALL*. The actual base station is called *active*, the other base stations *inactive*. When a handover occurs, the new base station becomes active and serves the mobile host. The set of base stations is updated by the new base station. Therefore the base station compares the actual set with the own set of the base stations. Some end points are added to or dropped from the call. The former active base station becomes inactive or its end point is dropped. The main advantage of the predictive rerouting scheme is, that the new handover segment is already established when the handover occurs. Thus, the handover latency is minimized. There are two alternatives for using the multicast tree: Whether the anchor switch sends data to the active base stations only, deactivates the other links and does not send data to inactive base stations *or* the anchor node sends data to the whole set of base stations. In the second case the base stations which are not active buffer the data and drop them when they are outdated. Both cases seem feasible. In the first case a modification in the anchor node is required to activate/deactivate connections (which is a low level functionality, since the VC/VP table have to be modified). In the second case the point of decision is moved to the base station, but the overhead is much higher. An alternative is given, if the IP-style multicast implementation offers, that an end point can be added to the multicast group without sending data to the end point.

The steps for the predictive rerouting scheme are mainly an extension of the soft handover scheme. The difference is, that the handover response (HO_RESP) can be sent immediately after the base station has become active. Then the multicast tree is updated. As an example in Fig. 5 the end points of a potential base station (BS_POT) is added to the *MCALL* and the end point of the old base station (BS_OLD) is dropped.

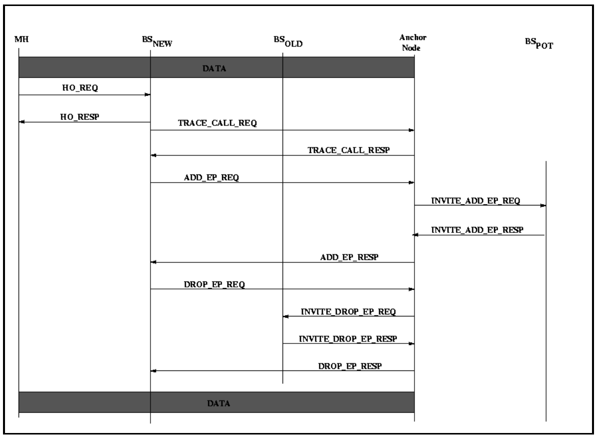

Fig. 5 Basic scheme for predictive rerouting

3. Related work

The proposed rerouting schemes are different from previous proposed schemes. Therefore it was necessary to introduce new schemes.

In [2] a virtual connection tree is established to potential new base stations. The wireless cells of the virtual tree cover a relatively large *neighboring mobile access region* and it is assumed that most mobile users remain within this area for the lifetime of a connection. This avoids the modification of the virtual tree and the incorporation of the network call processors for every handover attempt. A virtual tree consists of a set of connections (in each direction), associated with a path from a root node (ATM switch) to one of the leaf nodes (base stations). One of the branches is activated, when the mobile enters the new cell and the switch receives cells on that specific branch. This is interpreted as handover.

In [4] a mesh of ATM virtual paths is pre-established between all base stations. ATM virtual circuits are used to distinguish between the different data streams. The benefit is that the connections are pre-established and resources pre-allocated.

Both approaches in [2] and [4] have similarities with our predictive rerouting in its general idea to shorten the handover latency by pre-establishing ATM connections, but in both proposals the pre-established connection tree is not modified.

In [6] a *break-make* and a *make-break* handover approach (among others) were introduced. This refers to the sequence of operation for modification of the switching table in the switch. Nevertheless these schemes do not utilize multicast in the network. Instead the make-break and break-make scheme refers to the multicasting facilities within the ATM switch. A multicast tree in the backbone is *not* established. Instead in our scheme a multicast tree is set up and possibly modified after each handover event.

In [1] a *Multicast-based Reestablishment* scheme is proposed, which uses multicast operations to execute handover. The main idea is very similar to our hard and soft rerouting schemes, whereas some differences can be recognized: In [1] the old base station is triggered to tear down its old connection segment. In our scheme *third party* signaling is assumed, where a signaling client is allowed to perform add- and drop-end point operations on behalf of another signaling client. Moreover in our scheme not only the old and new base station is included, but also potential new base stations. The third party signaling makes the maintenance of such a multicast-tree more easy, since all operations can be carried out by the actual base station. However the authors of [1] describe their scheme in general, without considering a specific multicast implementation.

4. Experimental testbed

The testbed is composed of hardware and software components and tools to monitor the rerouting operation. The hardware consists of a network node, a node controller, at least two base stations and a correspondent host (Fig. 6a).

The network node is an open, non-proprietary cell switch from the Washington University Gigabit Switch (WUGS) program [13] intended for experimental research. It is an 8 port cell switch supporting up to 2.4Mbps per port. The switch is equipped with two dual OC-3 line cards and six G-Link line cards (1.2Gbps). The switch supports multicast efficiently through cell recycling. This is a technique, where cells arriving on an input port are sent to an output port and optionally recycled back to the input port, which can be repeated. The cell recycling adds a very small delay, but yields a scaling gain [14].

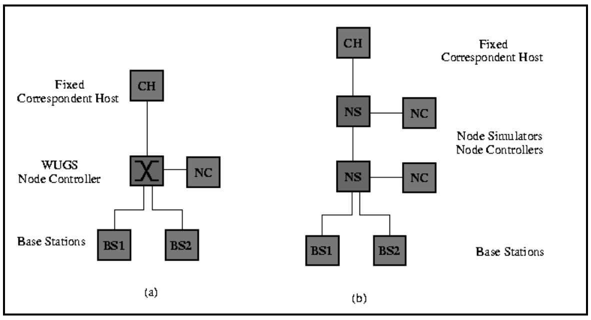

Fig. 6 Experimental testbed

The switching and control functionality are separated. The control functionality resides on a Node Controller (NC), which communicates with the node through an switching port. This facilitates remote control, the modification of the signaling software for mobility purposes and a flexible environment for development and testing. For development and tests the Node Controller can communicate via TCP with a Node Simulator, which emulates the control behavior of the switch. Thus, the signaling software can be tested in networks, which consists of several (simulated) network nodes (Fig. 6b). The Node Controller and the base stations are Intel PCs Pentium II, 266MHz with 64MB of main memory. They are equipped with off-the-shelf ATM cards ENI 155Mbps and open, proprietary network cards built around a custom IC called the ATM Port Interconnect Chip (APIC), respectively [15]. The Node Controller is running Linux 2.2.10, the base stations run NetBSD 1.4.1. Additionally the Node Controller and the base stations are equipped with an Ethernet card. The Ethernet cards are only used, when the switching hardware is replaced by a node simulator.

The signaling protocols (Fig. 7) are also part of the WUGS-environment. It consists of (bottom up):

- *Node Controller (NC):* The node controller (Node/Switch Management) realizes the same switch control API as the switch controller, but allows several switches to be treated as a single abstract switch. The node controller hides the details of how the switches within such a group (called a node) are linked together and translates requests relating to the node's external ports into commands for individual Switch Controllers.
- *Connection Management (Conn_M):* The connection management implements a distributed control for a network with multiple nodes). Connection managers communicate with one another using a general signaling protocol supporting a very flexible multicast call model.
- *Call Management (Call_M):* The call management realizes a session abstraction (*MCALL*) used by signaling clients to request and manage connections in an ATM network. It allows general, dynamic, multi-connection, multicast sessions. These sessions are supported through the UNI signaling protocol CMAP (Connection Management Access Protocol) [16].

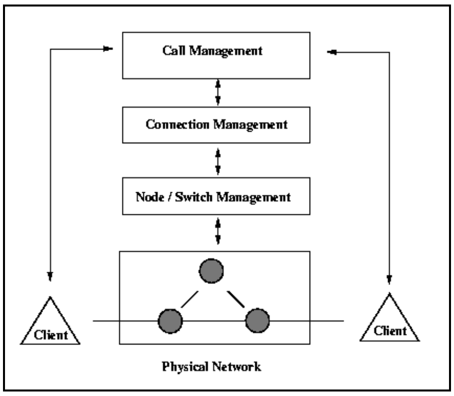

Fig. 7 Signaling protocols in the WUGS environment

The implementation of the Call Management offers a very flexible communication: When a *MCALL* is opened by a signaling client, the connections (possibly several) are set up. The client, which manages the *MCALL* is designated the *owner* of the *MCALL*. This is not necessarily the *root* node of the *MCALL* - this is the node, which is the *MCALL* origin. Additional clients can be added to a *MCALL* by invitation of the owner, by request of a client, which adds itself and by request from a client (which is not necessarily in the *MCALL*) on behalf of another client. Once a *MCALL* has been opened, connections can be added, modified and dropped to/from the *MCALL*. *MCALL* features (about access, modification, tracing, monitoring, etc.) can be set according to the needs. The signaling clients have been implemented to perform rerouting of connections when handover occurs.

4.1 Experimental Investigations

The following scenario has been investigated:

The network consists of a network node with node controller, a correspondent host, 4 base stations and a mobile host (see Fig. 8).

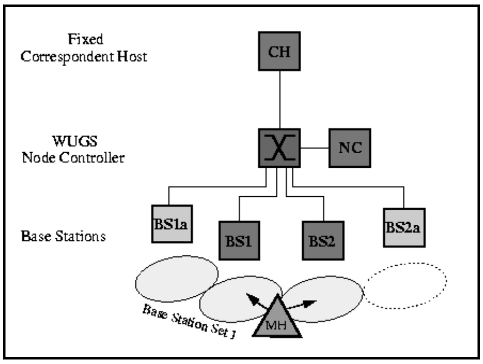

Fig. 8 Selected scenario

For the predictive scheme the base station BS1 has got assigned the set of base stations {BS1, BS2 and BS1a}. The set of base station of BS2 consists of {BS1, BS2, BS2a}. The mobile host has connectivity to the base stations BS1 and BS2, respectively and performs a *ping-pong* handover between base station BS1 and BS2. All base stations are connected to the same network node. The correspondent host opens and closes a *MCALL* and remains fixed.

The software configurations can be seen in Fig. 9.

In the testbed the wireless links between mobile host and base stations have been replaced by Ethernet in order to avoid the impact of an erroneous wireless channel.

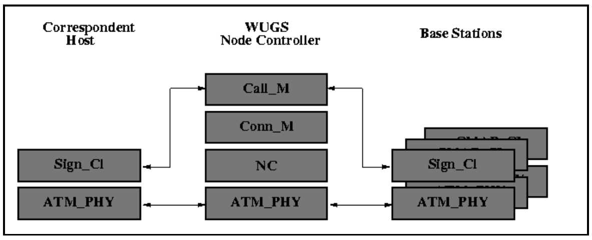

Fig. 9 Software configuration

The purpose of the experiments is twofold. On the one hand it has to be verified that a specific solution of IP-style multicast is suitable for rerouting of connections. Moreover desirable features for handoff rerouting can be identified, which are not part of the multicast in general or of the investigated implementation. On the other hand the rerouting schemes, which have been considered in this paper should be compared. As a starting point the comparison will be performed regarding the following two metrics:

- *Rerouting latency:* Time between handover indication until new end point added
- *Signaling load:* Traffic generated by rerouting operations

At first the scenario with a single switch between the correspondent host and the base stations is considered. It can be predicted that the hard rerouting scheme is slightly worse than the soft rerouting scheme regarding the rerouting latency. This is because in the soft rerouting scheme the connection is already reestablished and data can be exchanged, while the old base station is dropped from the multicast group in parallel. Nearly no difference is expected for the signaling load.

Now, a more complex scenario is considered: Suppose several network nodes on the path between the correspondent host and the base stations (see Fig. 10). If the hard rerouting scheme is applied, then the anchor node is in node A, since the connections between the old base station and node A are released after the drop-end point operation. For the soft rerouting scheme the anchor node is in node C. The choice of the anchor node has a strong impact on rerouting latency and signaling load. For hard rerouting the rerouting latency is increased. While the amount of signaling data equals for the hard and soft rerouting scheme, the impact on the overall network is different.

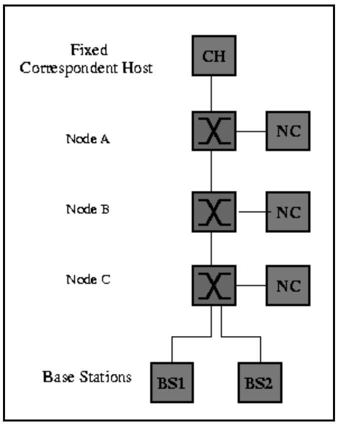

Fig. 10 Scenario with several network nodes

Assuming the rerouting latency for soft rerouting consists of a processing time in the network nodes and base stations and the time required to transmit the signaling messages:

$$T_{RR\_Latency,Soft} = T_{Proc} + T_{Transm}$$

Then

$$T_{RR\_Latency,Hard} \approx n \cdot T_{RR\_Latency,Soft}$$

where n is the number of intermediate network nodes between the correspondent host and the base stations. If the *aggregated* signaling load is defined as

$$L_{Sign,Aggr} = \frac{L_{Sign}}{n}$$

where n is again the number of intermediate nodes and L_{Sign} is the signaling load due to a rerouting operation. Then the signaling load induced by the hard handover scheme is increased by the signaling load of every network node between the correspondent host and base stations:

$$L_{Sign,Aggr,Hard} \approx n \cdot L_{Sign,Aggr,Soft}$$

It can be seen, that the hard rerouting scheme generates much more overhead in large networks. Assuming that the overhead of soft rerouting which occurs due to two simultaneously active links is relative small, the soft rerouting is a better choice for handover than the hard rerouting scheme. It has to be remarked that for forward handover the soft rerouting scheme requires overlapping radio cells and at least two simultaneously received base stations.

The predictive scheme reduces the rerouting latency to an absolute minimum. Nevertheless there is a remarkable overhead, since

$$L_{Sign,Aggr,Predictive} \approx (s-1) \cdot L_{Sign,Aggr,Soft}$$

where s is the size of the base station set and it is assumed that all base station belonging to the set have to be added and no base station have to be dropped. Moreover buffers can used in the base stations to buffer user data, which increases the overhead. If buffering is avoided, then the overhead is increased, since the reserved resources have to be enabled. Nevertheless as smaller the set of base stations, the smaller the overhead. In an extreme case, where the set of base stations equals 2, the overhead is as small as for the soft rerouting scheme. Thus, the predictive rerouting scheme offers the potential to reduce the overhead (see e.g. [8]) and improves communication with strict timing constraints and frequent handover.

First preliminary results show that the rerouting latency of the hard rerouting is about 400ms, for the soft rerouting about 200ms. As expected the predictive rerouting scheme minimizes the rerouting latency to the duration of the mobile's re-registration. Precise measurement data will be published separately.

5. Conclusions

In the paper it is advocated to use IP-style multicast in order to reroute connections for handover in a mobile cellular network. The main benefit of the approach is that the network nodes in the backbone have no knowledge about handover and no mobility specific signaling is required, since the rerouting operations can be composed of basic multicast operations (add/drop-end point-to/from multicast-group). Three rerouting schemes are described: Hard, soft and predictive rerouting.

For experimental investigations a testbed is set up, which bases on an open, non-proprietary, networking environment. Its signaling software offers a direct many-to-many communication and the switching hardware supports scalable and efficient multicast by cell-recycling. The testbed proofs the feasibility of the approach for an existing implementation of an IP-style multicast over connection-oriented networks. In networks with many network nodes hard rerouting generates a remarkable overhead. The usage of soft rerouting might be limited by the wireless technology since it requires overlapping wireless cells and at least two simultaneously received base stations. The predictive rerouting scheme limits the rerouting latency to an absolute minimum It has some drawbacks regarding its overhead, but it holds the potential to reduce the overhead.

6. Literature

[1] K. Keeton; et al.: Providing Connection-Oriented Network Services to Mobile Hosts. Proc. of USENIX Symposium on Mobile and Location-Independent Computing, Cambridge, USA, 1993

[2] Acampora, et al.: An Architecture and Methodology for Mobile-Executed Handoff in Cellular ATM Networks, IEEE Journal on Selected Areas in Communication, Oct. 1994

[3] K. Eng, et al.: BAHAMA: A Broadband Ad-Hoc Wireless ATM Local Areas Network. ACM/ Baltzer, J. on Wireless Networks, V. 1, No.2, 1995

[4] I. Katzela, et al.: Virtual Trees Routing Protocol for a Wireless ATM LAN, Proc. of ICUPC '96, Sept. 1996

[5] C. K. Toh, et al.: The Design and Implementation of a hybrid handover protocol for Multi-media Wireless LANs, Proc. Of ACM Mobicom, 1995

[6] R. Ramjee, et al.: Performance Evaluation of Connection Re-Routing Schemes for ATM-based Wireless Networks, IEEE/ACM Transactions on Networking, Vol. 6, No. 3, June 1998

[7] C. K. Toh, et al.: A Unifying Methodology for Handovers of Heterogeneous Connections in Wireless ATM Networks. Computer Communication Review, Vol. 27, No. 1, 1997

[8] R. Ghai, et al.: "An Architecture and Communication Protocol for Pico-cellular Networks, USENIX Symposium on Mobile and Location-Independent Computing, 1993

[9] M. Stemm, et al.: Vertical Handoffs in Wireless Overlay Networks. ACM Mobile Networking (MONET), Special Issue on Mobile Networking in the Internet, Winter 1998

[10] A. Festag, et al.: Interconnection of Wireless Cells - a Multicast-based Approach, Proc. of ICUPC '97, Oct. 1997

[11] C. Perkins: IPv4 Mobility Support. IETF RFC 2002

[12] Rauhala, et al.: WATM Specification 1.0. URL: http://www.atmforum.org

[13] WUGS - Washington University Gigabit Switch Environment. Wash.U, St. Louis, USA, URL: http://boushi.arl.wustl.edu/gigabitkits/kits.html

[14] T. Chaney, et al. Design of a Gigabit ATM Switch. Proc. INFOCOM 1997

[15] Z. Dittia; et al.: The APIC Approach to High Performance Network Interface Design. URL: http://boushi.arl.wustl.edu/gigabitkits/kits.html

[16] J. de Hardt: Connection Management Software (CMSS) Architecture. URL: http://boushi.arl.wustl.edu/gigabitkits/kits.html

[17] K. Cox, et al.: Connection Management Access Protocol (CMAP) Specification. URL: http://boushi.arl.wustl.edu/gigabitkits/kits.html

[18] ATM Forum: ATM User-Network Interface (UNI) Signaling Specification Version 4.0., 1996