A Model for Enhancing Connection Rerouting Using Active Networks

Kwan Wu Chin, Mohan Kumar
Department of Computer Science,
Curtin University of Technology,
GPO Box U 1987, Perth 6845,
Western Australia
(chinkw,kumar)@cs.curtin.edu.au

Craig Farrell
NDG Software
Suite 200, 12680 High Bluff Drive
San Diego CA 92130 USA
craig@ndgsoftware.com

Abstract

Active networks (ANs) are a new paradigm in computer networking. In ANs, programs are injected into routers and switches to extend the functionalities of the network. In this paper we propose a new model based on active connection rerouting (ACR). In the ACR model, programs are dynamically injected into active switches (ASs) in wireless ATMs to faciliate efficient connection rerouting during mobile host (MH) migration. ACR has all the benefits of traditional path extension schemes such as cell order preservation and fast handoff without the side effects of looping and increased end-to-end delay. Moreover it is efficient, scalable, generic and requires minimal buffering. We present an analytical model to compare the performance of ACR with other path rerouting schemes. We find that ACR's performance is comparable to the traditional path extension schemes. We show that by taking advantage of the underlying routing protocol, locality of the MH's movement and computational power of switches, an efficient and fast handoff can be achieved. Above all we show why ANs are ideal in solving the problem of connection rerouting and how current end-host(s) based approaches can be enhanced.

1. Introduction

Handoff is an important aspect in wireless networks allowing MHs to access computer networks from anywhere in the world. The integration of asynchronous transfer mode (ATM) and wireless networks offers the promise (to mobile users) of seamless multimedia services through the provision of end-to-end QoS. This integration introduces a new set of problems which include connection rerouting during handoff, and QoS maintenance at wireless links and across handoffs.

Ensuring a low handoff latency, avoiding cell loss and misordering due to handoff are crucial in minimizing disruptions to traffic flow. There have been several works on connection rerouting [2, 10, 4, 20, 4, 3, 1, 9, 7]. The majority of connection rerouting schemes involve locating a crossover switch (CX) which is a node on the original path. During path rerouting new connections are created from the CX to the MH's current location. These new connections are then spliced onto the MH's original connections. Connections from the CX to the previous base station (BS) are then torn down. In this paper we will use BS_1 to denote the previous BS or current BS and BS_2 denotes the BS, the MH is handing off to. We represent the geographic cell in italics (*cell*) and use the default font to represent message cell. The CX discovery algorithm must ensure that the rerouted path is optimal and maintains the original QoS of the connection prior to handoff. Taking Figure 1(a) as an example, the fixed host (FH) has an established connection to the MH. When the MH migrates across *cell* areas, the connection is rerouted at the CX. When the MH migrates to *Cell 2* a new connection is setup from CX_1 to *Cell 2*. Once the cells from the old path, CX_1 to BS_1 are drained the translation table at CX_1 is updated. The percentage of the original path used in the optimised path depends on the location of the CX. The CX discovery algorithms must make a tradeoff between maximum path reuse and the optimal path.

ANs [16] are a new paradigm for solving network problems. This paradigm uses the computational power at intermediate network nodes to facilitate processing of traffic passing through. The main goal of our approach is to exploit ASs along the route from the FH to the MH by programming the ASs with the ACR algorithm. ACR is mobile aware and serves to optimise the path after handoff. The ACR scheme consists of two parts. The first part involves extending the path to the MH's current location during handoff. The main advantages of using path extension are low latency handoff, preservation of cell ordering and little or no cell loss. The second part involves a two stage path optimisation process. Optimisations are only invoked after the handoff has finished, therefore handoff latency is not affected by delays due to path rerouting. The first optimisation stage is invoked by an AS along the extended path if the AS detects an optimimisable path then a partial optimisation is performed. The second stage invokes an optimisation program to determine whether there is any optimal route to the FH. This stage is invoked by creating an *active* connection setup from the new BS back to the FH. When the setup message traverses through each AS, it checks whether the message is traversing through the extended path or an alternative path. If an alternative path exists then the path is further optimised. These stages of optimisation are transparent to the MH. The intra-network processing allows minimal signaling and no processing at end-hosts for path rerouting.

We evaluate the performance of ACR analytically and by simulation. Analytically we compare ACR to current rerouting methods [10, 4, 17, 9, 1] based on the required signaling messages, handoff latency, communication disruption, bandwidth requirements and buffer requirements. In the simulation study we implemented the ACR model as a proof of concept and to validate its performance in mesh, tree and redundant tree topologies. The performance metrics measured are *path reuse* and *optimisation gain* for both partial and full optimisations, and buffer requirements.

2. Background

This section mainly reviews existing CX discovery algorithms and approaches to handoff in wireless ATM. The next section introduces ANs which are the basis of our scheme.

2.1. CX Discovery Algorithms

Work on rerouting in wireless ATMs can be broadly categorised into the following:

- *Full Path Extension* [2]: In this approach, the original connection is extended from the old BS to the new BS. Hence results in low latency handoff and no cell loss. The main assumption is that BSs are connected by pre-established virtual circuits (VCs) which allow fast forwarding. The main limitations are increasing end-to-end delay of the route and looping. Therefore a path optimisation is required to address these limitations.

- *Incremental re-establishment* [10, 20]: This class of methods involve invoking a CX discovery algorithm. Handoff latency is affected by the discovery algorithm, update time of translation table at the CX and connection time between the BS and MH. This class of algorithms require buffering at the CX or BSs to avoid cell loss and cell misordering.

- *Path splicing* [3]: A path is established from the new BS to the FH. If the path coincides with the original path, then the CX will splice the new path onto the existing path. Buffering at the BSs is used to prevent cell misordering. Cells arriving from the CX are buffered until the previous BS finishes forwarding.

- *Tree Routing or Multicast Based* [1]: This class of algorithms aims to reduce the handoff rate through clustering of cells onto a root node. Each cell has a preestablished connection to the root node. During cell forwarding the root node forwards the cell onto the correct path. This scheme prevents cell misordering and cell loss through multicasting but suffers from cell duplications. The main limitations are: existence of cell duplicates, excessive buffering and pre-reservation of bandwidth [18].

- *Anchor Rerouting* [7]: In this scheme a designated switch at the edge of the network acts as the CX. This means no CX discovery is required, and a new connection can be made to the new BS. This results in faster handoff but the resulting path is not optimal.

- *Full Connection Restablishment* [1]: This requires the original connection to be torn down and a new connection made to the MH's current location. The main disadvantages are high handoff latency due to signalling overheads and are not viable when handoff rate is high. On the other hand an optimal path is guaranteed.

A comprehensive study on CX discovery algorithms is presented in [18] and performance studies of various connection rerouting described above can be found in the following reference: [5].

2.2. Active Networks

The AN paradigm allows end-nodes to program network elements so that user specific computation can be performed. ANs allow the injection of customised programs into the network. Programs can extend the capabilities of network element(s) or provide customised computation on a set of packets which results in faster network protocol and flexible network functionality [16]. This means that the deployment of protocols (such as the changes described in this document) are not governed by the slow process of standardisation [14]. Two methods have been proposed for computation at network elements, programmable switches [6] and the capsule approach [19].

In this paper we take the programmable approach where the ACR program is preloaded into switches. The ACR program is then instantiated during connection setup. We assumed a set of functions are available at each switch that is accessible through predefined interfaces which provide access and modification of translation table, traffic customisation, connection management and allocation of memory for maintaining states.

2.3. Why Use Active Networking in Mobile Networks?

The most beneficial aspect of our work is the processing undertaken by ANs. We envisage to obtain better solutions to network problems such as routing to MHs compared to solutions which are deployed from the edge of the network. In the area of mobile networks, intra network processing is advantageous due to the dynamic location of MHs and QoS requirements. To ensure seamless execution of applications running at the MHs a fast response time is required to mask the disruption caused by handoff.

The performance of ANs and security concerns are debatable issues. We argue that despite the computational overheads experienced by AN-based protocol, its benefits out-weigh its overheads. Security issues will be not be covered in this paper. Further information on this aspect of ANs can be found in [14]. In heterogeneous networks, due to the varying QoS inherent in mobile networks, adaptability is an important factor. To cater to a vast variety of applications and networks, MHs can inject programs into BSs on demand. Hence, services at foreign networks can be augmented. Secondly, signaling messages generated for location management and path rerouting are dependent on handoff rate. To reduce the number of signaling messsages a common practice is to utilise a hierarchical network topology. ANs provide a generic solution that is independent of any topology and also enable the filtering of unnecessary signaling messages.

We have seen various approaches to handoff in wireless ATMs, but none has considered the use of ANs. CX discovery algorithms incur high latency if they are initiated from the edge of the network and this latency plays a vital role if dynamic routing is involved. Therefore the use of intra-network processing capabilities enable the design of an efficient path optimisation strategy. As we will see in Section 3.3 an active solution is more cost-effective than traditional methods of rerouting due to intra-network services.

3. The ACR Model

ASs are "active" because they allow programmers to inject customised program(s) into the network for a given flow. In our approach, ASs detect and reroute virtual connection (VCs) or virtual path (VPs) pertaining to the FH and MH. Apart from that ASs keep track of MH location which enable ASs to inform FH(s) of a MH's current location during connection setup.

3.1. Programming ASs

In ACR we assume the integrated approach for location management proposed by Acharya et al. [3]. The scheme proposed by Acharya et al. [3] is similar to location management in Mobile IP [12]. The FH issues a SETUP[1] message to a host without knowing whether the end-host is a MH. When the home agent (HA) of the MH intercepts this message, either a CONNECT or RELEASE is returned. A CONNECT response is generated when the host is static or when the MH is in its home network. A RELEASE is generated if the MH is in a foreign network. The RELEASE contains the MH's current location allowing the FH to setup an optimal path to the MH. The CONNECT and RELEASE messages contain an element which identifies the connection to be a mobile connection. This allows the FH to determine if it is communicating with a MH in order to determine whether switches need to be programmed.

ACR returns an active CONNECT or RELEASE message to the HA instead of the ordinary CONNECT and RELEASE messages. These *active* messages contain the required security association and an embedded program to be used for loading the ACR program into ASs. The advantages of having an HA deploy handoff programs include, loading of the predefined copy of the active program and distribution of security association from the HA. Upon receipt of a CONNECTION message, a FH proceeds to establish a connection to the MH's current location.

This sequence of events are described in Figure 1(b). Refering to Figure 1(b) an active SETUP message is initiated by the FH and sent to the MH via the MH's home network. The HA intercepts this message and replies with a RELEASE message, assuming that the MH has migrated to a foreign network. Embedded within this RELEASE message are two elements, the first is used to load the ACR scheme at ASs and the second is used to indicate to the FH that the corresponding host is a MH. Using the MH's current address embeded within the RELEASE message, the FH then makes a direct connection to the MH. ASs along the path that intercept the setup signal sent by the FH establish the required security association and load the ACR scheme. Once the program is loaded it has access to information pertaining to the call. Other accesses such as translation table or new call creation are through predefined interfaces. In addition once an AS is programmed, if other connection request(s) directed to the MH are intercepted by an AS on the path, it returns a RELEASE message notifying the FH of the MH's care-of-address. Hence a faster connection setup time is experienced by the FH.

The ASs only need to be programmed once. If there are multiple connections, the ASs only need to record the con-

[1] Additional Information Elements are added in signalling message. This include SETUP, CONNECT and RELEASE

nection information. This improves scalability since only one program is loaded for all connections going to the MH. Furthermore the data maintained at each AS requires minimal storage. Table 1 shows the data maintained by the AN-handoff program at each AS.

3.2. Handoff

In the ACR scheme handoff is achieved firstly by extending the current path to the new BS. After that the path is lazily optimised by ASs. The following paragraphs describe the handoff algorithm and path optimisation in detail.

In Figure 1(c) when the MH detects that it is in the vicinity of *Cell 2*, it sends a HO_REQUEST to BS_1 and this message is forwarded to BS_2. If BS_2 is able to support the required QoS, then BS_2 sends an active SETUP message to BS_1. In addition to a new path being formed, all ASs between BS_1 and BS_2 are programmed. This completes the path extension process [2]. Once the MH has migrated, BS_2 sends a HO_DATA_FORWARD message to BS_1. BS_1 then updates its translation table to merge the two paths. Once this is done, a cell is forwarded onto the new path. Note that using preestablished connection between BSs will further improve handoff latency. Finally a HO_REPLY is sent from the BS_2 to the MH through BS_1. If the path extension fails a HO_REJECT is returned.

In the above we assume the MH is able to detect signal strength from neighbouring *cell* areas and determine when it is approaching the next BS. This enables the MH to preallocate the extended connection before handoff occurs. In the case of *hard handoff* (no prediction of handoff), after the MH establishes a radio link with the new BS (BS_2), the active SETUP is sent from BS_2 to BS_1. BS_1 automatically updates its translation table without waiting for the HO_DATA_FORWARD message. Once the translation table is updated at BS_1 the handoff process is completed. Note that all uplink connections from the MH are routed through BS_1 after handoff. These links are then optimised in a manner similar to the downlink connections.

The ACR scheme maintains cell ordering and prevents cell loss, thus no disruptions to traffic flow are encountered and results in low handoff latency. Note that handoff latency is a factor of the signaling latency of the path extension thus it is not affected by the path optimisation performed by the AS.

3.3. Partial Route Optimisation

Once the path has been extended and all the ASs along the path have been programmed, the ASs determine whether optimisations are needed. The AS in which the extended and original paths intersect is designated as the CX. In other words the AS that has reachability to both BS_1 and BS_2 is designated as the CX. The CX then updates its translation table to splice the extended path onto the original path. This process is explained in Section 3.5.

Refering to Figure 2(a), BS_1 and BS_2 share a common AS. After path extension we see that cells have to traverse through BS_1. This can be avoided by updating the translation table at the AS. Moreover this scheme does not require any further signaling.

Figure 2(b) shows another example but with multiple ASs involved. This figure shows why a connection setup needs to come from BS_2. In Figure 2(b) AS_1 to AS_3 are ASs on the original path. When a connection is made from BS_2, the extended path intersects the old path at AS_2. In Figure 2(b) we can see that a loop is formed between AS_2 and BS_1. To resolve this loop AS_2 is designated as the CX. Once AS_2 updates its translation table and cells forwarded onto BS_1 are drained, the path AS_2-AS_3-BS_1-AS_3-AS_2 is torn down. If BS_1 was to extend the path, then AS_3 will be chosen as the CX resulting in a loop between AS_2 and AS_3.

During cases where the MH moves backward and forward (or zig-zag) between BS_1 and BS_2 the partial optimisation process will result in BS_1 being chosen as the CX. Hence the extended path from BS_1 to BS_2 is torn down.

3.4. Full Route Optimisation

This step is performed after the partial optimisation. It serves to check whether there is an alternative path to the FH. If an alternative path is found optimisation is performed. To optimise the extended path, BS_2 makes an *active* connection setup to the FH. An AS on the original path that intercepts this setup request is designated as the CX. This CX then splices the new path onto the old path given that the connection setup has traversed through an alternative path instead of the extended path. Refering to Figure 2(c) we see that there is an alternative path leading to the FH. As the setup message traverses through AS_5 and AS_4 it detects whether these are programmed during path extension. At this point the program still has not determined whether full path optimisation is needed. Upon arrival at AS_6 the program detects that AS_6 has not been programmed and hence the setup message has been routed through an alternative path. Therefore full path optimisation is needed. When the setup program arrives at AS_1 it determines AS_1 is on the original path. At this point the program stops traversing and updates the translation table to splice the new path onto the old path. The setup message stops at AS_1. If AS_6 is on the original path and if no alternative route has been taken then the program is terminated and the allocated resources are freed. If the CX

[2] We assumed that BS_2 has sufficient bandwidth for maintaining the MH's QoS. Work on satisfying QoS is not within the scope of this paper. Interested readers are referred to [15]

chosen does not have the appropriate resources to maintain the required QoS, the setup message is forwarded along the route towards the source. The message continues to be forwarded until a suitable CX is found.

The above two stage approach takes advantage of the locality of MH's movements and fast handoff provided by path extension. The full path optimisation is mainly invoked when the MH has moved across a domain. Schemes such as Nearest Common Node Rerouting [4] work well in hierarchical topologies. ACR provides a generic solution which provides an optimal path regardless of the underlying topology. This will be shown in Section 5.1.2.

3.5. Connection Rerouting

This process is needed to splice the new path onto the old path. This involves updating the translation table at the CX and draining cells from the old path. When the CX detects a setup request it forwards the request to BS_1. After the connection has been setup we need to drain the cells in the original path. Firstly the CX sets up a connection to BS_2 given the traffic contract of the original path. When a path has been succcessfully setup the translation table is updated. A capsule is sent along the new path to inform BS_2 to buffer all incoming cells from the new path until a marker cell from the old path has been received. The CX then sends a marker cell along the old path to mark the last of the cells forwarded. When BS_2 receives the marker cell it forwards the buffered cells. The paths from BS_2 to BS_1 and from the CX to BS_1 are torn down.

The following points outline the sequence of events during connection rerouting at the CX:

1. Inform BS_2 to buffer incoming cells from the new path.
2. Update translation table. Therefore new cells from the FH will be forwarded onto the new path here after.
3. Inject a marker cell onto the old path. When this marker cell appears at BS_2 the buffered cells can be forwarded to the MH.
4. The CX then tears down the original path upon receiving notification from BS_2.

4. Analytical Model for ACR

This section provides an analytical study of the ACR algorithm and compares its performance to other connection rerouting schemes in the literature [10, 4, 17, 9, 1]. The analytical derivations outlined are based on the work by [5]. A brief summary of Bui et al.'s work will be presented in the following section.

4.1. Basis of study

Bui et al.'s paper compares the performance of various rerouting schemes [5]. These include *Connection Re-establishment* [10], *Path Extension*[4], *Anchor Rerouting*[7], *Dynamic Re-routing* [17], *Multicast Based Rerouting* [9] and *Static Virtual Connection Tree based Rerouting* [1]. These rerouting schemes are compared based on their complexity (number of signaling messages required), handoff latency, communication disruption, bandwidth requirements and buffer requirements. Algebraic equations are derived for each of the above performance parameters. These equations are derived based on the processing and traversing time of signaling messages generated during handoff. The equations derived for the ACR model are shown in Tables 5, 6 and 7. A detailed treatment on deriving these equations and the parameters used is presented in [8].

4.2. Handoff Delay

In the ACR scheme handoff latency is dependent on the time it takes to establish a radio connection with BS_2 and the time to setup (including programming ASs) path(s) to BS_1. In schemes such as dynamic rerouting, handoff latency depends on fast CX discovery. Handoff delay needs to be at a minimum to ensure seamless operation of real-time applications during handoff. Here we compare the handoff delay of the ACR scheme to other schemes [5].

The expression for handoff delay for the ACR scheme can be seen in Table 5. Also shown is the expression for handoff latency if BSs have direct connection to each other and soft handoff scenario. Other rerouting schemes, namely connection re-establishment, path extension and dynamic (CX is chosen during handoff) are also compared. The expressions for other rerouting schemes mentioned above can be found in [5]. In ACR, the main delays are due to programming/setup of ASs between BS_2 and BS_1 and transmission of messages over the radio link. In connection re-establishment the latency is dependent on the number of switches between the FH and BS_2 (denoted by $N_{old-org}$).

Figure 3(c) shows a plot of handover delay against N (the number of switches between BSs or the number of switches above the BSC where the CX resides). We can see from Figure 3(c), connection re-establishment has a constant handoff latency and has the second highest handoff latency. The handoff latency recorded for the ACR scheme is almost the same as the path extension scheme. Although there is a 20% overhead with our active solution the ACR scheme still outperforms the non-active path extension scheme. This is due to the reduction in signalling messages required compared to the path extension scheme outlined in [5]. Soft handoff has the fastest handoff latency followed by the ACR scheme with 1-hop configuration. Multicast has the lowest handoff latency due to its higher bandwidth cost.

4.3. Disruption Time

The disruption time determines the interval in which no cells are forwarded to the MH. Figure 3(a) shows a comparison of disruption times for various schemes. The behaviour of disruption time in Figure 3(a) is quite similar to that of handoff latency (Figure 3(c)). This is mainly due to the direct relation between handoff latency and disruption time. The ACR has the benefit of minimising disruption time since it is only dependent on the time required to setup the extended path and the time taken for the first cell to be forwarded from BS_1. In the dynamic scheme, delay is incurred mainly in the CX discovery phase.

4.4. Overall Buffer Requirements

As mentioned the ACR schemes require buffering at the MH and BS to prevent cell loss and misordering. From Table 7 we see that for ACR, buffer requirement at the MH (upstream data) is dependent on the disruption time. On the other hand the buffer requirement at BS_2 is dependent on the propogation delay of the marker cell from the CX, on the path $CX - BS_1 - BS_2$.

The total buffer requirements for each scheme are shown in Figure 3(b). From Figure 3(b) we see that the overall buffer requirements for the ACR scheme are in the range 12.5 - 800 Kbs. Comparing between ACR-soft and ACR scheme we find that there is only a slight reduction in buffer requirements. If a topology with BSs connected to each other is used, buffer requirements are significantly reduced.

5. Simulation Study of ACR

Our simulation studies were done using Opnet [11]. In these simulations we studied buffer requirements during path optimisation and performance of ACR in different network topologies. We looked at three network topologies; mesh, tree and redundant tree.

A persistent source is set to generate frames at 30 frames/sec with frame size of 65K. The source uses CBR service over links of 155Mb/s. During connection setup the ACR algorithm is loaded by the AS management module and is only allowed to access call information pertaining to the call. The call information contains a traffic contract and interfaces to the translation table and management module.

In the simulator, ten BSs are randomly connected to the ASs. Each BS has an ATM interface which allows it to create connections to another BS or CX. Furthermore our handoff program is loaded during connection setup enabling rerouting and path extension during handoff. The BSs broadcast advertisement messages every n seconds. The advertisements are needed to enable MHs to detect that they are in the cell area of the given BS. We have tested three values of n and analysed its affect on handoff latency. In each simulation the data recorded are shown in Table 2.

5.1. Results from Simulation Studies

This section outlines results from our simulation study. Firstly we highlight the buffer requirements during route optimisation given varying delay of the extended path. The significance of these results is the quantification of the number of cells being buffered given varying drainage time. Finally the performance of ACR is evaluated for the three network topologies discussed in Section 5.1. The simulation studies demonstrate that our scheme provides a reasonable tradeoff between optimal path and path reuse for the mesh and tree topologies. Low path reuse or high optimisation gain implies higher latency in the extended path, hence higher drainage time. It is crucial to provide a tradeoff between these two variables to minimise buffer requirements.

5.1.1 Buffer Requirements During Route Optimisation

Figure 3(d) shows that in the event of large end-to-end delay in the extended path, more cells are buffered. In LANs with delays less than 10ms the buffer requirement is less than 10 cells. The buffer requirement can be reduced further by buffering only throughput dependent traffic such as data transfer which are susceptible to loss. We can further improve the overall buffer requirements if custom architectures such as [9] are used to reduce handoff rate. An observation we made with CBR traffic is that the size of the buffers remain stable after handoff due to identical incoming and outcoming rates at the buffer.

5.1.2 Performance of Varying Network Topologies

The results of the experiments on mesh, tree, and redundant tree topologies are shown in Table 4. The results shown in Table 4 are average values of performance parameters from ten simulation runs. The results show path reuse and optimisation gain from each optimisation stage. The objective of these experiments is to investigate the effectiveness of the proposed scheme in achieving optimal path and maximal path reuse. The folowing items explain each performance parameter in Table 4:

- *Path Reuse (Partial Optimisation)* is the percentage ratio of the number of switches common to original and partially optimised path, and the number of switches in the original path. If the value is high then full optimisation is likely to yield significant optimisation gain.

- *Path Reuse (Full Optimisation)* is the percentage ratio of the number of switches common to original and fully optimised paths and the number of switches in the original path. The path includes new ASs (if any) from BS_1 to BS_2. A high value implies a low optimisation gain was achieved.

- *Optimised Gain 1* is the gain when full optimisation is performed on the extended path before partial optimisation is performed. In other words, this gain gives an indication of how much shorter is the optimised path compared to the extended path. This value is calculated as:

$$\frac{(OrigPL + EPL) - OpPL}{(OrigPL + EPL)} \times 100 \qquad (1)$$

- The *Optimised Gain 2* is the gain due to the full optimisation stage after partial optimisation has been performed on the extended path. This value is calculated as:

$$\frac{OpPL}{POPL} \times 100 \qquad (2)$$

- The *Optimised Gain 3* is a measure of the gain achieved after partial optimisation is performed on the extended path. A high value means partial optimisation is sufficient in the given topology. The expression for this value is:

$$\frac{(OrigPL + EPL) - POPL}{(OrigPL + EPL)} \times 100 \qquad (3)$$

- The *Partial Optimisation Runs* is the ratio between the number of instances when the partial optimisation is executed and the total number of simulations run.

We see that the mesh topology has the highest path reuse under partial optimisation. This is because in most cases during handoff the path is extended from the switch local to BS_1 to BS_2. Therefore we see all the ASs on the original path being reused in the extended path. Since the CX is the AS local to BS_1, partial optimisation did not yield any optimisation gain as shown by Optimised Gain 3. In the tree topology the extended path has to traverse up the hierarchy to reach BS_2. After partial optimisation ASs from the CX to BS_1 are omitted, hence we observed less ASs from the original path being reused.

When full optimisation is run we see that the mesh topology has the lowest reuse value. The values for path reuse for the tree topology are similar because no full optimisation was needed. It is observed that partial optimisation is sufficient, and full optimisation did not result in a shorter path.

In Optimised Gain 1 we see that full optimisation results in significant reduction in path length. Furthermore we see that a reasonable tradeoff is observed between path reused and optimized path. This is important because the main problem with any rerouting algorithms is the drainage latency on the old path. As we have seen in Figure 3(d) high drainage latency would lead to high buffer requirements. To keep buffer requirements at a minimum we need to make a tradeoff between path reuse and optimal path. Maximisation of path reuse would entail more cells to drain. The buffer requirement at the BS is proportional to the propogation delay of the extended path. A high optimisation gain means the extended path length is significantly longer than the optimised path. Therefore an increase in drainage time will be observed if the path is significantly optimised[3]. In the results we see that the AN-handoff protocol provides a reasonable tradeoff between optimisation gain and path reuse.

It can be observed that partial optimisation is sufficient for the tree topology. This is shown by the 0.0% gain recorded for Optimised Gain 2. Both the mesh and redundant tree requires full path optimisation to be performed to yield a further 19.6% and 17.4% gain respectively.

The low Optimised Gain 3 value for the mesh topology indicates that partial optimisation is not sufficient. A slightly higher value is observed for redundant tree because in most cases during handoff the extended path needs to traverse up the hierarchy involving two or three ASs on the original path. As we can see partial optimisation for both mesh and redundant tree is insufficient to yield total gains of 42.0% and 43.0% respectively.

In retrospect, we observe that in the partial optimisation stage approximately 50% of the optimisation gain is achieved. The proposed two-stage scheme has the advantage of limiting the end-to-end of the extended path. If full path optimisation alone is performed, the extended path is significantly longer compared to the extended path after partial optimisation. For example in the mesh topology the extended path is 58% longer than the optimised path. Given that partial optimisation does not require any additional signaling overhead the cost incurred is minimal. Apart from that we can program the ASs to invoke full path optimisation under given constraints. For example, full path optimisation is omitted in tree network. And full path optimisation after the MH has migrated across multiple domains is invoked.

Table 3 shows the rate at which the CX was found. At worst our scheme needed 7 hops before the CX is found. On average we needed 4.4, 3.4 and 4.6 hops for the mesh,

[3]We only consider the number of hops. A shorter path might have longer delays given factors such as bandwidth and traffic on the given path.

tree and redundant tree topologies respectively.

6. Discussion

Our results show a number of benefits in using the ANs approach to MH rerouting. As can be seen from the results, our optimisation scheme is not biased towards optimal path or path reuse. Therefore the drainage delay of cells from the extended path is kept at a minimum. This is important since we have shown in section 5.1.1 that buffer requirements are a function of the propogation delay on the extended path. We have also demonstrated that by utilising the routing protocol and customised processing within the network layer a simple and efficient path optimisation scheme is achieved.

Traditionaly the use of path extension has been limited due to its increased end-to-end delay [10, 13, 18] and chances of introducing routing loops. The incorporation of partial path optimisation removes any looping which might occur during path extension. Path extension fits into the nature of handoff in wireless ATMs. Since cells need to be forwarded from BS_1 a connection has to be made to BS_2. By making path extension as the initial stage of our handoff protocol coupled within efficient optimization scheme we were able to overcome problems encountered in other schemes.

7. Conclusions

We have presented a connection rerouting model based on ANs. Analytical studies have shown that the performance of ACR is comparable to traditional path extension and without the side effects of increased end-to-end delay and looping. Furthermore ACR is generic, scalable, efficient and requires minimal buffering. Apart from that the route optimization process does require a CX discovery algorithm to be deployed from end-host(s). Above all we have evaluated the effectiveness of ANs in enhancing traditional methods for connection rerouting in wireless ATM. Furthermore the advantages of performing connection rerouting within the network are shown and limitations of traditional methods are addressed.

References

[1] A. S. Acampora and M. Naghshineh. An architecture and methodology for mobile executed handoff in cellular ATM networks. *IEEE JSAC*, 12(8):1365–1375, Oct. 1994.

[2] A. Acharya, S. Biswas, L. French, J. Li, and D. Raychaudhuri. Handoff and location management in mobile ATM networks. In *3rd International Workshop on Mobile Multimedia Communications (MoMuC-3)*, Princeton, New Jersey, Sept. 1996.

[3] A. Acharya, J. Li, B. Rajagopalan, and D. Raychaudhuri. Mobility management in wireless ATM networks. In *IEEE Infocom 98*, San Francisco, CA. USA, Apr. 1998.

[4] B. A. Akyol and D. C. Cox. Rerouting for handoff in a wireless ATM network. *IEEE Personal Communications*, 3(10):26–33, Oct. 1996.

[5] B. A. J. Banh, G. J. Anido, and E. Dutkiewicz. Handover rerouting schemes for connection oriented services in mobile atm networks. In *INFOCOM'98*, San Francisco, USA, 1998.

[6] S. Bhattacharjee, K. L. Calvert, and E. Zegura. An architecture for active networking. In *High Performance Networking (HPN'97)*, Apr. 1997.

[7] M. Cheng, S. Rajagopalan, L. F. Chang, G. P. Pollini, and M. Barton. PCS mobility support over fixed ATM networks. *IEEE Communications Magazine*, 35(11):82–92, Nov. 1997.

[8] K. W. Chin, M. Kumar, and C. Farrell. A model for enhancing connection rerouting using active networks. Technical Report 2, Curtin University of Technology, Bentley, Western Australia, June 1999. Available at: http://www.cs.curtin.edu.au/čhinkw/Publications/.

[9] R. Ghai and S. Singh. An architecture and communication protocol for picocellular network. *IEEE Personal Communications Magazine*, 1(3):36–46, 1994.

[10] K. Keeton, B. A. Mah, S. Seshan, R. H. Katz, and D. Ferrari. Providing connection-oriented network services to mobile hosts. In *Proceedings of the USENIX Symposium on Mobile and Location-Independent Computing*, Massachusetts, Aug. 1993.

[11] MIL3 Inc. Opnet Modeler 3.5.A. Network Simulation Software, 1997.

[12] C. Perkins. RFC2002: IP mobility support, Oct. 1996.

[13] R. Ramjee, T. L. Porta, J. Kurose, and D. Towsley. Performance evaluation of connection rerouting schemes for ATM-based wireless networks, June 1997.

[14] J. M. Smith, D. J. Farber, C. A. Gunter, and S. M. Nettles. Switchware: Accelerating network evolution. Technical report, University of Pennsylvania, 1996. MS-CIS-96-38.

[15] M. Srivastava and P. P. Mishra. On quality of service in mobile wireless networks. In *Proceedings of the 7th Intl. wkshp on Network and O/S Support for Digital Audio and Video (NOSSDAV)*, St Louis, Missouri, May 1997.

[16] D. L. Tennenhouse and D. J. Wetherall. Towards an active network architecture. *CCR*, 26(2):5–18, Apr. 1996.

[17] C.-K. Toh. The design and implementation of a hybrid handover protocol for multi-media wireless lans. In *ACM MOBICOM'95*, Berkeley, California, 1995.

[18] C.-K. Toh. Performance evaluation of crossover switch discovery algorithms for wireless ATM lans. In *IEEE INFOCOM'96*, San Francisco, Mar. 1996.

[19] D. J. Wetherall, J. Guttag, and D. L. Tennenhouse. ANTS: A toolkit for building and dynamically deploying network protocols. In *IEEE OPENARCH'98*, San Francisco, Apr. 1998.

[20] R. Yuan, S. K. Biswas, L. J. French, J. Li, and D. Raychaudhuri. A signaling and control architecture for mobility support in wireless ATM networks. *ACM/Baltzer Mobile Networks and Applicatioins*, 1(3), Dec. 1996.

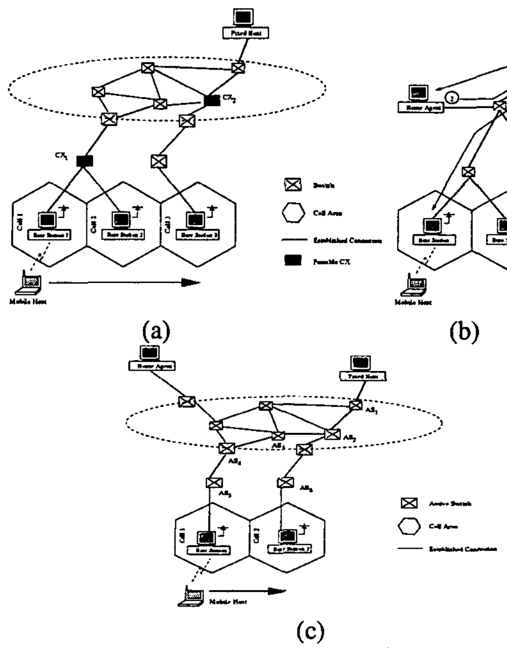

Figure 1. (a) Example of Mobile ATM (b) Location Management in Wireless ATM (c) Handoff in Wireless ATM

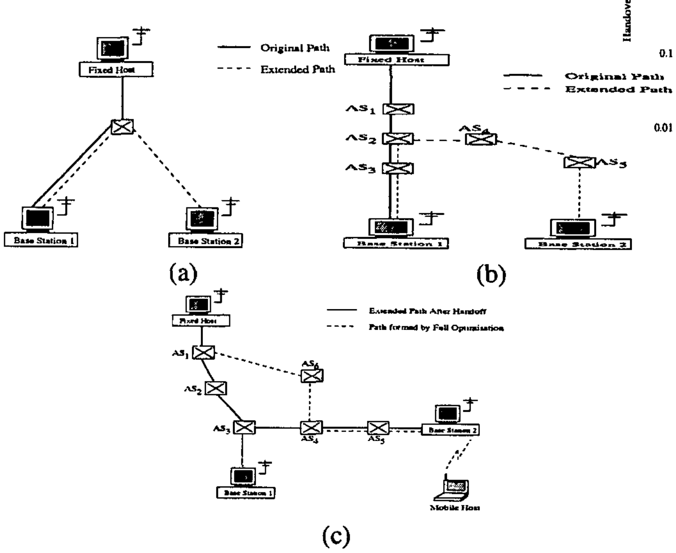

Figure 2. Example topology for: (a) Path Extension (b) Path Extension With Multiple Switches (c) Example of Full Optimisation

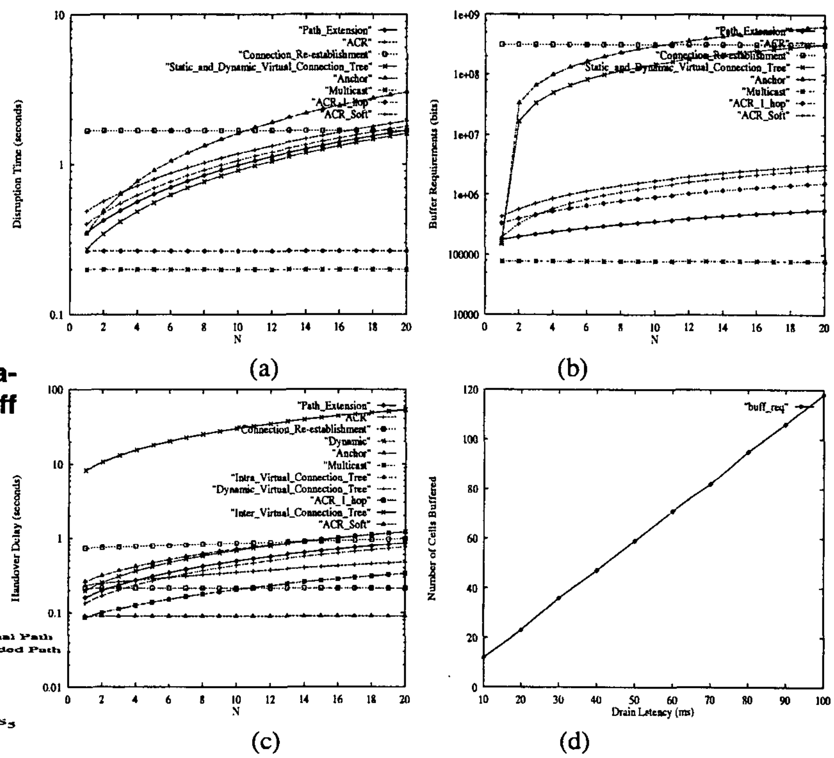

Figure 3. (a) Disruption Time (b) Overall Buffer Requirements (c) Handoff Delay (d) Buffer requirements given varying delay in the original path. (CBR Traffic)

Data	Details
Pointer to the Translation Table	This is needed for path rerouting during optimisation.
MH's care-of-address	The MH's current location.
MH's HA address	The MH's home address.
FH address	The FH which the MH is communicating with (i.e calling party).
Traffic contract	Traffic contract specified (e.g. QoS).
Identification	This is used to create a security association between the FH and AS.
Original or Extended Flag	This flag marks whether the AS is on the original or extended path.

Table 1. Data Structure Maintained by AS

Data	Description
Original Path Length (OrigPL).	This is the number of hops (ASs only) from the FH to the MH (when MH is at BS_1).
Extended Path Length (EPL).	The number of hops from BS_1 to BS_2.
Optimised Path Length (OpPL).	The optimised path length after full optimisation.
Partial Optimised Path length (POPL)	The path length from the FH to BS_2 after partial optimisation.
Convergence Count.	The number of hops needed before a CX is found.
Partial Optimisation Flag.	We also recorded whether partial path optimisation was performed.

Table 2. Data Recorded During Simulation.

	Mesh	Tree	Redundant Tree
Maximum number of hops	7	4	7
Minimum number of hops	1	1	2
Average numbar of hops	4.4	3.4	4.6

Table 3. Number of hops for Convergence.

	Mesh	Tree	Redundant Tree
Path Reuse (Partial Optimisation)	77.11%	41.4%	58.3%
Path Reuse (Full Optimisation)	39.3%	41.4%	40.73%
Optimised Gain 1	42.0%	47.6%	43.0%
Optimised Gain 2	19.6%	0.0%	17.4%
Optimised Gain 3	18.3%	47.6%	26.5%
Partial Optimisation Runs	50%	100%	20%

Table 4. Performance Parameters for Network topologies.

Rerouting Scheme	Handoff Latency ($T_h andover$)
ACR	$3T_w + 3S_{bts(other)} + 2S_{bts(s-AS)} + S_{bsc(s-AS)} + S_{sw(s-AS)} N_{old-new} + 4T_{bsc(sw)} + S_{sw(other)} N_{old-new} + 2T_{sw} N_{old-new} + S_{bsc(other)} + 2S_{stp} + S_{sw(upd-tbl)}$
ACR-1-hop	$4T_w + 3S_{bts(other)} + 2S_m + 2T_{bsc-bts} + 2S_{bts(s-AS)} + T_{sw} + 4T_{bsc-sw} + 3S_{bsc(other)} + S_{sw(upd-tbl)}$
ACR-Soft $S_{bsc(other)}$	$2(S_m + T_{bsc-bts} + 4T_w + 3S_{bts(other)} +$

Table 5. Expressions for Handoff Latency.

Rerouting Scheme	Disruption Delay ($T_{disrupt}$)
ACR	$T_{HO-latency} + N_{old-new} T_{d(sw)} + T_{d(w)} + T_{d(bsc-sw)} + T_{d(bsc-bts)} + S_m$
ACR-1-hop	$T_{HO-latency-1-hop} + T_{d(bsc-bts)} + T_{d(w)} + S_m$
ACR-Soft	$3T_w + S_m + 2_{bts(other)} + N_{old-new}(T_{sw} + S_{sw(other)} + T_{d(sw)} + T_{d(w)} + T_{d(bsc-bts)} + T_{bsc-bts} + S_{bsc(other)} + 2(T_{bsc-sw} + T_{d(bsc-sw)}$

Table 6. Expressions for Disruption Delay.

Rerouting Scheme	Buffering Nodes	Buffer Requirements
ACR	MH	$T_{HO-latency} \times BW_d$
	BSC_{new}	$((N_{cx-old} + N_{old-new})(T_{d(sw)} +$
		$S_{sw(sw)}) + T_{d(sw)}) \times BW_d$
ACR-1-hop	MH	$T_{HO-latency-1-hop} \times BW_d$
	BSC_{new}	$((N_{cx-old} + N_{old-new})(T_{d(sw)} + S_{sw(sw)}) + T_{d(sw)}) \times BW_d$
ACR-Soft	MH	$T_{HO-latency-soft} \times BW_d$
	BSC_{new}	$((N_{cx-old} + N_{old-new})(T_{d(sw)} + S_{sw(sw)}) + T_{d(sw)}) \times BW_d$

Table 7. Expressions for Buffer Requirements.

Variable	Description	Values
M_{sig}	Signalling message size	50 Bytes
M_d	Data packet size or AAL PDU size	8 Kbytes
BW_{sig}	Signalling channel bandwidth	64 Kbytes
BW_d	Data channel bandwidth	1 Mbps
L_w	Propagation delays for the wireless links	3 μs
$L_{bsc-bts}$	Propagation delays for the BSC-to-BTS Links	50 μs
L_{bsc-sw}	Propagation dealsy for the BSC-to-switch links	50 μs
L_{sw}	Propagation delays for the inter-switch links	50 μs
S_{stp}	Processing time of signalling messages in nodes where only Signal Transfer Point (STP) function is required.	0.3 ms
$S_{sw(setup)}$ $S_{bsc(setup)}$ $S_{bts(setup)}$	Processing time of SETUP message and target BTS determination functions in various types of network nodes.	8 ms 16 ms 32 ms
$S_{sw(disc)}$ $S_{bsc(disc)}$ $S_{bts(disc)}$	Processing time of DISC message in various types of network nodes	4 ms 8 ms 16 ms
$S_{sw(upd-tbl)}$	Time to update translation table	4 ms
$S_{sw(other)}$ $S_{bsc(other)}$ $S_{bts(other)}$	Processing time of signalling messages (other than SETUP, DISC, and fast resource reservation) in various types of network nodes	3 ms 6 ms 12 ms
$S_{sw(sw)}$ $S_{sw(bsc)}$ $S_{sw(bts)}$	Processing time for switching an ATM cell through an ATM switch	10 μs 20 μs 40 μs
$S_{sw(s-AS)}$ $S_{bsc(s-AS)}$ $S_{bts(s-AS)}$	Processing time of SETUP message (for AS) and target BTS determination functions in various type of network nodes.	9.6 ms 19.2 ms 38.4 ms
$N_{old-new}$	Number of switches between BS_1 and BS_2.	N
N_{cx-old}	Number of switches from the CX to BS_1.	N
$N_{old-org}$ $N_{new-org}$	Number of switches between BS_1 or BS_2 and the switch connecting to the FH	20 20
S_m	Signalling message processing time in the MS	5 ms

Table 8. *Parameters Used In Analytical Model.*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
MSWiM 99 Seattle WA USA
Copyright ACM 1999 1-58113-173-9/99/08...$5.00