Providing Seamless Mobility with Competition based Soft Handover Management

Johan Kristiansson and Peter Parnes

Department of Computer Science & Electrical Engineering, Media Technology
Luleå University of Technology,
971 87 Luleå, Sweden
{Johan.Kristiansson, Peter.Parnes}@csee.ltu.se

Abstract. As host mobility and radio interference in wireless networks cause packet losses and delays, it is difficult to develop useful mobile real-time media applications. This paper describes a new handover strategy for end-to-end mobility called Competition based Soft Handover Management (CSHM). During a handover, redundant packet streams are sent through multiple connections which are later merged into one stream when received by the other end-point. As each network connection competes with other connections in contributing to the merged packet stream, the handover process can be viewed as a competition.
As a proof of concept, CSHM has been implemented in Resilient Mobile Socket, RMS, an application-layer mobility scheme and used together with Marratech Pro, which is a commercially available e-meeting application. By using this prototype, the paper shows that it is possible to minimize redundant packets as well as decrease packet losses during handovers.

1 Introduction

The rapidly growing number of Wi-Fi hotspots and worldwide deployment of new wide-area networks, such as UMTS have made it possible to develop new wireless multimedia services that can be used anywhere and anytime using any available carrier or operator. Mobile e-meeting applications that are running on portable devices with multi-access capabilities will for example allow users to stay connected and participate in virtual communities by using wide-area cellular networks or inexpensive high performance Wi-Fi connections.

Even if multi-access gives users more flexibility in communication, it also imposes new demands on network management and interoperability. When users move between different physical locations, it may become necessary due to limited coverage or bad network performance to make a handover to another network. Similarly, if a better network becomes available, a handover should automatically be initialized to the network offering the best price/performance ratio subject to the user's need.

Today, users must normally take an active part in the handover process and are often required to manually select which network to use. Moreover, during or immediately after a handover it is very common that packet losses and delays occur due to signaling propagation of new location updates. For most applications, such as HTTP or FTP, handover delay is not of vital importance, e.g. waiting one or two second extra when

downloading a web page is not critical. For real-time media on the other hand, delays and packet losses are extremely important and even a small disturbance can make a media stream unintelligible.

Research about mobility management has so far mainly focused on how to preserve communication and manage location updates. Handover management however, i.e. making fast and low delay handover decisions is still a challenging problem. A handover algorithm must for example be able to evaluate all available networks and select the best performing network as fast as possible in order to avoid interruptions in communications. This is particularly difficult as wireless performance can fluctuate rapidly due radio interference, especially if the coverage is bad.

Oscillations are another problem with handover management. If it takes time to complete a handover and if the performance of a network fluctuates, then there is always a risk that handovers are triggered back and forth between two or more networks causing instability and seriously degraded performance.

These problems raise the question of whether or not it is possible to design a handover algorithm that can:

1. Automatically select the network that is the most suitable for real-time media, i.e. the network with the least packet losses and end-to-end delay.
2. Make a handover to that network without the users perceiving interruptions in real-time media flows.
3. Make handover decisions without the users perceiving degraded performance due to oscillations.

This paper presents a new handover decision algorithm called *Competition based Soft Handover Management* CSHM, that solves these problems. In the paper it is assumed that mobile hosts have access to at least two connections simultaneously. It can also be worth to point out that handover decisions are only based on network performance. Decisions based on financial costs, such as dynamic charge models (none flat-rate) is left for future work.

The rest of the paper is organized as follows. Section 2 gives a brief introduction to previous work related to handover management. In section 3, the RMS is briefly described followed by a more extensive presentation of CSHM. In section 4, the algorithm is evaluated using the Marratech Pro prototype and in section 5, the paper is finally concluded with discussion and future work.

2 Background and related work

There have been numerous proposals for providing lossless handovers and minimizing the handover delay to support wireless multimedia. Several micro-mobility schemes have for example been proposed to complement Mobile IP [14]. Cellular IP [18] provides improved handover support in limited geographical areas by incorporating cellular principles found in traditional telecommunication networks. Another micro-mobility scheme, Hierarchical Mobile IP [15], tries to reduce the home network registration time by using a hierarchical network management structure. A difference between the work presented in this paper and research related to Mobile IP, is that CSHM is completely

implemented in the application-layer and requires no support from the networks. As mobility is managed end-to-end, CSHM can provide seamless handovers between any network (e.g. a handover between a Wi-Fi network and a UMTS network) and not only seamless handovers within a Mobile IP or Cellular IP enabled network. Another difference is that the paper focus on handover control, i.e. how to trigger handovers, rather than describing how to implement handover support.

A common way to trigger handovers is to monitor the signal strength to the base-stations and use some sort of dwell-timers, hysteresis or threshold based control algorithm [3, 13, 19]. One problem with these handover strategies is that they tend to increase the handover delay, which makes them unsuitable for real-time media.

To make more accurate handover decisions, several location-aided handover strategies have been proposed in the literature [6, 8]. These studies have shown that user movements can be fairly predicted by using a history of recorded user movements, current direction and velocity of the user. However, it has been discussed that mobility prediction algorithms in general are incapable of adapting to new situations and that a small random variation can cause many mobility prediction algorithms to fail [4]. Besides, it is unclear if current technologies, for example the 802.11b can provide sufficient positioning precision [10] to make handover decisions fast enough to support real-time media.

Clearly, if packet loss during handovers could be avoided completely, it would be possible to perform speculative handovers without degrading the quality. To provide lossless handovers between heterogeneous networks, some work has recently been done to add soft handover support in layers above the network layer. RMS [11] provides for example soft handover support by allowing simultaneous use of multiple UDP sockets for data communication. Similar functionality is provided by the ADD-IP [16] mechanism in the Stream Control Transmission Protocol (SCTP) [7].

The major contribution of this paper is a new type of handover management strategy for end-to-end based soft handovers. In contrast to other IP based soft handovers schemes such as [9], CSHM is designed to use multiple IP connections simultaneously. Rather than using redundant connections only as passive backup links, the paper shows how redundancy can be used to improve network performance and how to evaluate end-to-end performance during handovers.

CSHM can also be compared with other multi-link streaming protocols, for example the work presented in [5] or the Multimedia Multiplexing Transport Protocol [12]. However, it is important to point out that the purpose of CSHM is not to increase the throughput, but rather to minimize packet delay during handovers.

3 Competition based Soft handover Management

There is a strong relationship between handover management and mobility management. While the later provides the fundamental architecture that is needed to execute handovers, handover management controls and initializes handovers. To understand how CSHM works, it is necessary to first explain how handover support is implemented in the RMS.

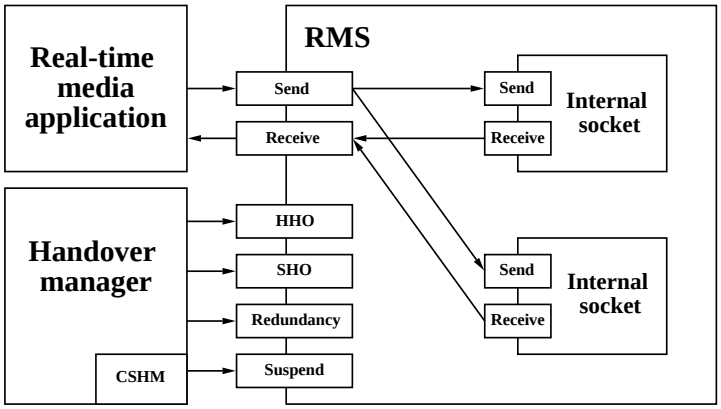


Fig. 1. An overview of the RMS architecture.

3.1 Resilient Mobile Socket

RMS is an application layer mobility scheme for streaming real time media, developed at the division of Media Technology at Luleå University of Technology. The primary purpose of the RMS is to preserve the communication and provide a more robust platform by allowing applications to suspend connections and then resume them using another (or the same) IP address.

An application that sends and receives packets over the Internet normally uses a socket, representing an end-point of a communication link to another application running on the Internet. By encapsulating multiple sockets into a new socket abstraction (RMS), any encapsulated or internal socket can fail without disturbing the applications. As each internal socket represents an entry point to each connected network, running applications will still be able to communicate if the current active internal socket becomes disconnected and another internal socket is available. In this way, a handover process in RMS refers to migrating data flows between different internal sockets.

Figure 1 shows an overview of the RMS architecture and how internal sockets are encapsulated. Note, that RMS besides functionality to send and receive packets also provides methods to control which internal sockets that should be used.

The *SUSPEND* procedure is used to hibernate on-going communication and is automatically called when all network connections are lost, i.e. no internal sockets can be used.

The *HHO* (hard handover) procedure provides the opposite operation and is used to recover from a disconnection or to initiate a handover to another network. During a hard handover, the currently active internal socket is first removed before a new internal socket is created. A hard handover is typically a reactive or an unplanned operation and occurs when something unexpected happens to the system, for example when a connection is suddenly lost. Managing handovers in this case is quite simple as there is usually only one connection to choose from.

The *SHO* (soft handover) procedure provides in contrast to hard handovers, functionality to use redundancy during handovers by using multiple internal sockets simul-

taneously to send and receive packets. This technique eliminates handover delay and prevents packets from getting lost, but must be proactively initiated to be effective, i.e. initiated before the currently active internal socket becomes disconnected. Because the RMS now has access to multiple connections, a handover management algorithm must be able to evaluate and select the best available connection.

An important component in the RMS architecture is the *Handover Manager*, which can be seen to the left in figure 1. The Handover Manager is responsible for monitoring the system and triggering handovers by calling the procedures mentioned above. A difference between RMS and other mobility management schemes such as Mobile IP, is that handover decisions are always made per packet stream rather than for the whole system. This makes it possible to apply different handover strategies for different media. Audio packets can for example be sent over a Wi-Fi connection while video packets are sent over a UMTS network. Moreover, for none real-time media it may be sufficient to only use hard handovers as soft handovers usually waste bandwidth. From a handover management point of view, this kind of flexibility is extremely important as it relieves the Handover Manager from resolving conflicting handover requirements.

3.2 Competition based Handover management

To be able to use soft handovers efficiently several new problems must be solved. The perhaps most difficult problem is how to decide when to initialize soft handovers. As mentioned before, soft handovers must always be initialized proactively, i.e. triggered when at least two internal socket are available. A soft handover management scheme must consequently be able to predict when a connection is going to be lost.

Another difficult problem with soft handovers is how to minimize redundancy. As redundancy wastes resources, both in terms of bandwidth and computer resources, an efficient soft handover management algorithm should strive to minimize redundant packets and in the same time keep the network performance as good as possible.

The rest of this section discusses how CSHM addresses these problems and how handover decisions can be made by using a competition based evaluation between internal sockets.

3.3 Making proactive handover decisions

Even if it would be possible to make proactive handover decision based on mobility prediction, it is important to point out that real-time media such as Voice-over-IP requires that handover decisions are made within a couple of hundreds of milliseconds, before the playout buffer is exceeded. Considering the precision of current technologies and how much the network performance can fluctuate during a couple of hundreds of milliseconds, location-aided handovers do not seem to be a very promising approach. Besides, it is very likely that the performance of a radio network gets degraded even if the user is not moving at all, e.g. somebody closes a door or the user touches the radio antenna.

A more realistic alternative to location-aided handover is to make handover decisions based on jitter interruptions in media streams. When radio conditions are bad it is very common that packets get lost over the air interface. Link-layer approaches such as

automatic repeat request (ARQ) attempt to hide channel losses from the network layer by re-transmitting lost packets. However, as it takes time to retransmit lost packets, i.e. ARQ will increase packet delay, and since packets cannot be retransmitted forever some packets will still get lost.

When a user moves away from a network physically, it is very likely due to limited coverage that packet losses and delay occur just before a connection is completely lost. This information is used by the CSHM algorithm to proactively initialize a soft handover.

When an RMS end-point receives a packet stream from another RMS, it calculates a packet delay based on the arrival time of the current packet and the previous packet. If the packet delay exceeds a threshold value, Φ, it will send a *SHO request* to the other end-point, asking it to initialize a soft handover. In this way, the receiver sends feedback[1] to the sender, which makes the final handover decision. If an RMS is both sending and receiving packets, it will take at least two handover decision rounds before both incoming and outgoing packets are duplicated. Note that CSHM does not make any difference between a severely congested network and a network with bad radio performance. If an access network becomes congested somewhere, it may also be reasonable to initiate a handover to another network, assuming that the congested network is not shared with the other available access networks. In this case, there is a risk that redundancy makes the congestion even worse, which will negatively affect the performance of all internal sockets.

It is important to point out that triggering handovers based on interruptions in media streams can only be applied if packets are sent with regular intervals, i.e. packets are sent in a specific pattern. To manage handovers for other (none real-time) media, the Handover Manager periodically scans the routing table for changes. In case a soft handover has not already been initialized, the Handover Manager will for example trigger a hard handover if the currently used network adapter disappears from the routing table. Similarly, to determine if a new network adapter performs better than the current one, the CSHM algorithm can be configured to automatically trigger a soft handover when a new network adapter appears in the routing table.

3.4 Filtering out duplicate packets

If packets are not lost over the network, the receiver will get duplicate copies of each packet when redundancy is enabled. Even if many multimedia applications are designed to handle forward error correction (FEC) and duplicate packets, it can dramatically decrease the performance of the applications. In group communication applications, like Marratech Pro, it is very common due to lack of ubiquitous multicast to use a server/reflector to re-distribute packets to other participants. Hence, sending multiple copies of each packet will undesirably increase the load on the server.

To prevent this from happening, a mechanism is needed to filter out duplicate packets and automatically turn off redundancy when performance becomes satisfactory again.

[1] RMS provides an in-band signaling protocol, which can be used to exchange control information between peers.

By encapsulating all redundant packets into a new packet containing a sequence number, the first packet received for a given sequence number is forwarded to the application and all other copies are dropped. One advantage of using this *first-come-first-serve* scheme is that it can significantly improve the network performance during a handover. If for example two networks are performing badly, it may still be possible to merge the bad networks into one good network.

Algorithm 1 Competition based Soft Handover Management

Ensure: the best performing internal socket is always used
1: $dwellTimer \Leftarrow 0$
2: **loop**
3: **if** $packetDelay > \Phi$ **then**
4: enableRedundancy()
5: $dwellTimer \Leftarrow 0$
6: **end if**
7: **if** $dwellTimer > \Delta$ **then**
8: $isocket_{default} \Leftarrow selectWinner(Contribution_{isocket_1}, .., Contribution_{isocket_N})$
9: disableRedundancy()
10: **end if**
11: increase($dwellTimer$)
12: **end loop**

3.5 Selecting a new default internal socket

To minimize redundant packets, CSHM uses a dwell-timer that expires after a predefined amount of time, Δ. Assuming that a new SHO request has not been received, i.e. the dwell-timer has not been reset, redundancy will be disabled after the dwell-timer has expired.

The CSHM algorithm is summarized in algorithm 1. One important difference between CSHM and other handover algorithms [3, 13, 19] is that the new default connection is not decided before the handover. During the handover, each receiver calculates in percent how much each duplicated stream (internal socket) contributes to the merged stream. This new metric is called *packet contribution* and can be viewed as a combination of packet losses and delay in respect to all other duplicated streams. The internal socket that got the highest packet contribution is selected as the new default internal socket after the dwell-timer has expired.

The whole handover process can be viewed as a *competition* where the threshold, Φ, determines when the competition starts, the dwell-timer, Δ, when the competition ends, and packet contribution who the winner is. A competition may not necessarily result in a handover as it is possible that the currently selected internal socket wins. This means that CSHM can also be used to improve network performance without actually switching networks.

4 Evaluation

To evaluate CSHM, a working prototype has been built by integrating RMS with Marratech Pro [1], a commercially available e-meeting software providing tools for synchronous interaction by combining audio, video, chat and a shared white-board.

Extensive use of Marratech Pro has shown that audio is the most sensitive of all involved real-time media [17]. This evaluation has therefore focused on exploring the relationship between different Φ and Δ settings and the effect on GSM audio quality. The following sections describes the prototype, the experimental test-bed and present the results.

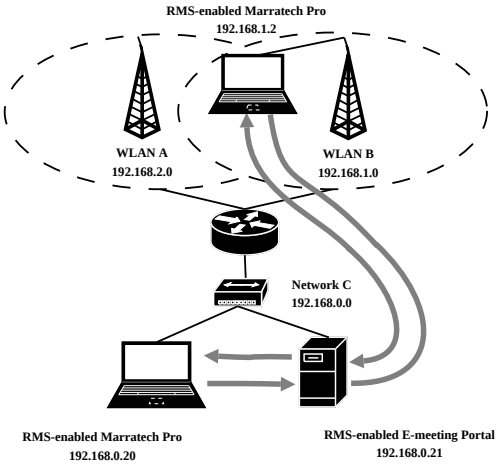


Fig. 2. The test-bed. The arrows illustrates the logical packet flow.

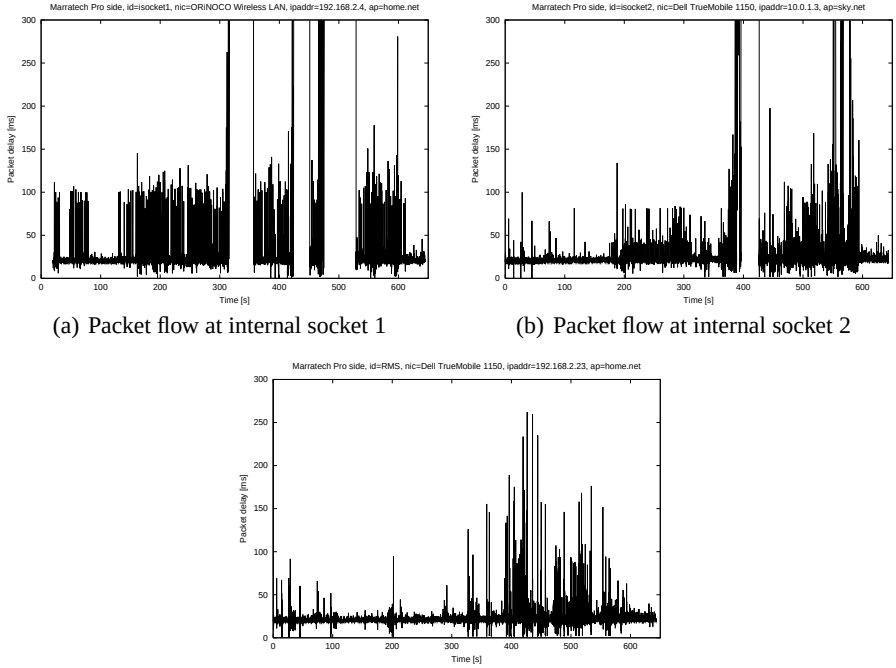
4.1 Implementation

The main part of the RMS is implemented in Java JDK 1.4 under Microsoft Windows XP. The Java Native Interface was used to implement functionality not supported by the Java platform. The IP Helper API [2] available in Windows was used to access the routing table and to detect new or disconnected network adapters.

Marratech Pro was modified by replacing the standard Java DatagramSocket with the RMS. Since Marratech Pro clients either uses IP-multicast or a media gateway called the e-meeting Portal to distribute packets, it was also necessary to replace the standard Java DatagramSocket in the e-meeting Portal. The CSHM algorithm was implemented as a part of the Handover Manager mentioned in section 3.1.

4.2 Methodology

The Marratech Pro based prototype has been tested and used together with a commercial GSM/GPRS network and several 802.11b Wi-Fi networks. Unfortunately, as the



(a) Packet flow at internal socket 1

(b) Packet flow at internal socket 2

(c) Merged packet flow

Fig. 3. Packet flows during the experiment.

GSM/GPRS network performed badly[2], it was impossible to transmit real-time media over it. Besides, as the network was shared with other users, it was hard to interpret the results and make repeatable experiments. It was even difficult to repeat the experiment by moving around between purely isolated 802.11b networks as it was impossible to move exactly the same in each experiment. One solution to this problem would be to repeat the experiment until a statistical certainty is obtained. However, as this can be very time consuming, it was decided to use some other method.

Another possibility would be to use a network simulator, but as this would require a re-implementation of both CSHM and RMS in the simulator it was finally decided to emulate different traffic flows instead. By saving a trace file for each internal socket and then replay the trace files it was possible to test how different Φ and Δ settings affected the merged stream. It was particularly interesting to investigate packet losses and how many redundant packets that were received as well as how many times the playout buffer[3] was exceeded.

Figure 2 illustrates the test-bed that was used to generate the trace files. The test-bed consists of three hosts and two partly overlapping Wi-Fi networks connected to a

[2] The round-trip time was larger than one second.

[3] Marratech Pro uses a dynamic playout buffer between 0 and 125 ms.

Table 1. Data from the experiment at the Marratech Pro end-point.

	Internal socket 1	Internal socket 2	Merged stream	Emulated
Packets received	23448	27699	30557	30557
Total packet contribution	26.5%	73.5%	-	-
Packet delay $\geq$ 50 ms	286	290	114	132
Packet delay $\geq$ 125 ms	57	53	22	22
Lost packets	7468	3217	359	359

Table 2. Data from the experiment at the Portal end-point.

	Internal socket 1	Internal socket 2	Merged stream	Emulated
Packets received	24867	29319	30909	30909
Total packet contribution	5.1%	94.9%	-	-
Packet delay $\geq$50 ms	350	280	166	166
Packet delay $\geq$125 ms	68	27	27	27
Lost packets	6049	1597	7	7

shared network. Wi-Fi connectivity was provided by two Apple AirPort with built-in NAT routing and two Lucent Orinoco Wi-Fi adapters attached to a laptop. Each Wi-Fi adapter was associated with different Wi-Fi network. The E-meeting Portal was run on a AMD Athlon 1.2 GHh computer and the others were run on Intel Pentium III 1.2 GHz computers. Microsoft Windows XP Professional was used as the operating system on all computers.

4.3 Results

The trace files were generated by moving around physically with one laptop in the test-bed and sending GSM audio between the two Marratech Pro clients. By disabling the CSHM algorithm temporarily and using redundancy during the whole experiment, it was possible to get full trace files for both internal sockets.

Figure 3 shows the packet delay for each internal socket at the Marratech Pro side as well as the packet delay for the merged packet stream. Similar results were obtained for the Portal end-point.

As can be seen in figure 3(a) and 3(b), *internal socket 1* lost connectivity three times while *internal socket 2* lost connectivity only one time. Since all disconnections occurred at different times, it was possible to merge *internal socket 1* and *internal socket 2* to one packet stream without the user noticing any disconnections at all. Moreover, note that the packet delay for the merged stream is significantly reduced compared with *internal socket 1* and *internal socket 2*. Apparently, all copies of a specific packet were not always lost even if the packet loss rate was high for both internal sockets.

Table 1 and table 2 summarize statistics from the experiment for the Marratech Pro and the Portal end-point. At the Marratech Pro side, *internal socket 1* contributed in total with 73.5% of all packets received and at the Portal side *internal socket 1* contributed with 94.9% of all packets sent to the Portal end-point. Note that the Portal end-point

Table 3. Relationship between Φ, duplicated packets and lost packets. Δ = 100 ms.

	Φ=Infinity	Φ=21 ms	Φ=25 ms	Φ=50 ms	Φ=100 ms
Packets received	30909	30885	30868	30862	30799
Packet delay $\geq$125 ms	22	22	22	22	22
Lost packets	7	31	48	54	117
SHO requests	0	2498	1924	1104	541
Duplicated packets	23306	3103	1882	870	363

Table 4. Relationship between Δ, duplicated packets and lost packets. Φ = 50 ms.

	Δ=Infinity	Δ=50 ms	Δ=100 ms	Δ=200 ms	Δ=0.5 s	Δ=2 s
Packets received	30909	30725	30862	30871	30868	30891
Packet delay $\geq$125 ms	22	22	22	22	22	22
Lost packets	7	191	54	45	48	25
SHO requests	0	1131	1104	1078	1075	1016
Duplicated packets	23306	237	870	1653	2519	11873

only had one network connection during the experiment and hence only one internal socket. The result presented in table 2 shows how the *internal socket 1* and the *internal socket 2* located at the Marratech Pro side were perceived at the Portal side.

As can be seen in table 1 and table 2, the emulated stream corresponds quite well with the merged stream obtained from the experiment. The merged stream can also be viewed as the base case or the optimal case as redundancy was always used. Ideally, a Φ and Δ setting should result in a similar stream, but with less redundant packets.

4.4 CSHM performance

The CSHM parameter space was explored by locking one parameter, either Δ or Φ and tuning the other parameter. The goal with this investigation was not to obtain an optimal parameter setting, but rather to get a better understanding of the CSHM algorithm.

Table 3 and 4 show the relationship between, Δ, Φ, lost packets and duplicated packets for the Marratech Pro end-point. Similar results were obtained at the Portal end-point. The numbers presented in table 3 and 4 are average values from six test runs. As can be seen in table 3, a small Φ value resulted in many SHO requests, which consequently resulted in more duplicated packets and hence less lost packets. Each GSM packet was sent with approximately 20 ms delay and setting Φ close to 20 ms resulted in 2498 SHO requests. When Φ was set in the range between 0 and 100 ms, the playout buffer was exceeded 22 times, which is exactly the same performance as the base-case, i.e. the optimal performance.

The relationship between Δ, lost packets and duplicated packets was investigated by locking Φ to 50 ms and adjusting the Δ parameter. As expected, a large Δ value resulted in more duplicated packets and hence less lost packets. Since redundancy improved the performance during the experiment, a large Δ also resulted in fewer SHO requests.

By studying the trace files it was observed that if the packet arrival jitter was low and packet losses were concentrated in terms of time, it was efficient to use a low Φ value and a big Δ value. If on the other hand the packet arrival jitter was high, then it made more sense to use a higher Φ value to prevent the CSHM algorithm from always being active.

5 Discussion

In the introduction it was asked whether or not it is possible to develop a handover decision algorithm that can:

1. Automatically select the network that is the most suitable for real-time media, i.e. the network with the least packet losses and end-to-end delay.
2. Make a handover to that network without the users perceiving interruptions in real-time media flows.
3. Make handover decisions without the users perceiving degraded performance due to oscillations.

In brief, the key to solve all these problems is to utilize multiple network connections simultaneously. The first problem is for example solved by using redundancy to compare each network connection and automatically select the connection with the least packet losses and end-to-end delay. As the use of a new internal socket does not affect the performance of the currently used socket, there is no risk that the performance gets degraded because of a handover. As an implication, it is no longer important to reduce the handover frequency, i.e. the users will not perceive any performance degradation when trying a new network.

The second problem is solved by merging multiple packet streams into one stream. This technique can also be used to decrease packet delay and reduce packet losses without performing a handover to another network. The results presented in the paper indicate that CSHM can be used to merge badly performing networks to one good network. However, if redundancy is going to be used as proposed in the paper, it is important to be able to control and minimize redundant packets. The results suggest that CSHM can be used to solve this problem or at least to reduce redundant packets for GSM audio traffic.

Regarding the oscillations, i.e. the third problem, the CSHM algorithm does not directly eliminate the oscillations as it is still possible that handovers are triggered back and forth between several networks, i.e. multiple SHO requests are triggered. However, the users will not perceive degraded performance due to the oscillations as the host receives packets from both the old and the new network during the handover. Rather than repeatedly switching between two badly performing networks, CSHM uses redundancy to improve the performance until some of the networks become stable again or until there is only one working connection left.

6 Acknowledgment

This work was done within the VITAL project, which is supported by the Objective 1 Norra Norrland - EU structural fund programme for Norra Norrland. Support was

also provided by the Centre for Distance-spanning Technology (CDT) and Mäkitalo Research Centre (MRC).

References

1. Marratech AB. , 2004. <http://www.marratech.com>.
2. Microsoft IP Helper API. , 2003. <http://msdn.microsoft.com>.
3. S. Aust, D. Proetel, N. A. Fikouras, and C. Görg. Policy based Mobile IP Handoff Decision (POLIMAND) using Generic link Layer Information. In *IEEE 5th International Conference on Mobile and Wireless Communication Networks (MWCN'02)*, 2003.
4. R. Chellappa, A. Jennings, and N. Shenoy. A Comparative Study of Mobility Prediction in Fixed Wireless and Mobile Ad Hoc Networks. In *IEEE International Conference on Communications (ICC 2003)*, 2003.
5. S. Dhananjay and P.T. Goff. Multiple IP Links for Improving Throughput and Reliability in Mobile Environments. In *INFOCOM*, 2002.
6. F. Erbas, J. Steuer, K. Kyamakya, D. Eggesieker, and K. Jobmann. A Regular Path Recognition Method and Prediction of User Movements in Wireless Networks. In *VTC Fall 2001, Mobile Technology for Third Millennium*, 2001.
7. R. Stewart et al. Stream Control Transport Protocol, 2001. IETF RFC2960.
8. F. Feng and D.S. Reeves. Explicit Proactive Handoff with Motion Prediction for Mobile IP. In *IEEE Wireless Communications and Networking Conference (WCNC´04)*, 2004.
9. S. Kashihara, K. Iida, H. Koga, Y. Kadobayashi, and S. Yamaguchi. End-to-End Seamless Handover using Multi-path Transmission Algorithm. In *Internet Conference 2002 (IC'02)*, 2002.
10. C. Komar and Ersoy C. Location Tracking and Location Based Service Using IEEE 802.11 WLAN Infrastructure. In *The Fifth European Wireless Conference Mobile and Wireless Systems beyond 3G*, 2004.
11. J. Kristiansson and P. Parnes. Application-layer Mobility support for Streaming Real-time Media. In *IEEE Wireless Communications and Networking Conference (WCNC´04)*, 2004.
12. L. Magalhaes and R. Kravets. MMTP: Multimedia Multiplexing Transport Protocol. *ACM SIGCOMM Computer Communication Review*, 31(2):220–243, 2001.
13. Y. Min-hua, L. Yu, and Z. Hui-min. The Mobile IP Handoff Between Hybrid Networks. In *IEEE 13th International Symposium on Personal, Indoor and Mobile Radio Communication (PIMRC'02)*, 2002.
14. C. Perkins. IP Mobility Support, 1996. IETF RFC2002.
15. H. Soliman, C. Castelluccia, K. Malki, and L. Bellier. Hierarchical MIPv6 Mobility Management, 2002. Internet Draft, IETF. Work in progress.
16. R. Steward and et al. Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration, 2003. Internet Draft, IETF. Work in progress.
17. Kåre Synnes, Peter Parnes, and Dick Schefström. Robust Audio Transport using mAudio. Research Report, ISSN 1402-1528, ISRN LTU-FR–99/04–SE, Luleå University of Technology, 1999.
18. A. Valkó. Cellular IP: A New Approach to Internet Host Mobility. *ACM SIGCOMM Comp. Commun. Rev.*, 29(1):50–65, 1999.
19. H.J. Wang. Policy-Enabled Handoffs Across Heterogeneous Wireless Networks. Technical Report CSD-98-1027, 1998.