

Prototyping of vertical handover enabled services

Raimo Nikkilä, Dmitrij Lagutin, Henrik Asplund and Mervi Ranta

PM&RG – Product Modelling and Realization Group
Department of Computer Science and Engineering
Helsinki University of Technology
P.O.B 5400, FIN-02015 HUT, Finland
fax +358-9-451 3293
tel. +358-9-451 3338, Raimo.Nikkila@hut.fi
tel. +358-9-451 4532, Dmitrij.Lagutin@hut.fi
tel. +358-9-4515197, Henrik.Asplund@hut.fi
tel. +358-9-451 4807, Mervi.Ranta@hut.fi

Abstract

This paper discusses the prototyping of vertical handover enabled services in the Innovation prototyping for vertical handover project (VHO project). It is a collaborative project of Ericsson Finland, Helsinki University of Technology, Radionet and TeliaSonera. In particular this paper focuses on the first results of the research of adaptive applications. The aim of the project is to discover what kind of new possibilities vertical handover offers for improved services. Furthermore, with an innovation prototype it is possible to run field tests and usability studies. The first vertical handover enabled service of the project is a music service that is called the VHO Ämppäri. It is a new version of a previously implemented mobile music service called the Ämppäri. The paper discusses the objectives, requirements and chosen approaches for the VHO Ämppäri. The most obvious special feature of vertical handover enabled services is that they are able to adapt to the changes in the network and its characteristics. The VHO Ämppäri plays music from the network server and adapts the music quality and the used bandwidth according to both the changes in the network and the preferences and context of the user.

1 Introduction

The VHO project is a collaborative project of Ericsson Finland, Helsinki University of Technology, Radionet and TeliaSonera. The aim of the VHO project is to realise real-time multiaccess in heterogeneous networks. This enables a user to utilize several networks (such as WLAN, GPRS, UMTS) in parallel. Vertical handover allows the application services to be seamlessly transferred between different network technologies.

The purpose of the VHO project is to produce working prototypes to allow experimentation on both vertical handover and adaptive applications that utilise vertical handover. The project is not meant to produce a final product, but rather to experiment the requirements and possibilities that vertical handover capability introduces for future product development.

So far, there is little understanding of how multiaccess will affect the services and end users. In the VHO project user needs and potential future applications are investigated. The aim is to discover what kind of new possibilities vertical handover offers for improved services and applications.

Furthermore, the project includes application and development of the innovation prototyping methodology that is being developed at the Product Modelling and Realisation group (PM&RG) of the Helsinki University of Technology. An important object of the VHO project is to disseminate gained experience to public research community. The innovation prototyping approach and methodology supports the aim of making the development process transparent to not only the partners but also to other interest groups (Ranta et al. 2004).

The objective of this paper is to give an overview of the architecture of the implemented VHO Ämppäri that is the first result of the application programming efforts of the VHO project. In addition to being the very first application, it provides a basis for the development of future vertical handover enabled services. Notice that the objective of the application prototyping efforts is to provide a platform and reusable components to enable easy realisation of further innovation prototypes of vertical handover enabled services. Thus, the application architecture presented in this paper or its pieces are not proposed for standardisation purposes. The adaptive application research still concentrates on experimenting the requirements for vertical handover enabled services and supporting the standardization efforts of the other, network related workpackages of the VHO project.

This paper starts with a brief overview of the concepts. Then chapter three discusses the application communication that consists of the VHO application programming interface and the control protocol for application level control. The fourth chapter explains the architecture of the entire VHO Ämppäri and introduces its components. Finally brief conclusions are presented on the results and future research.

2 An overview to the concepts

The body of this paper gives an overview of the architecture of a vertical handover enabled service, the VHO Ämppäri. Vertical handover enabled service means here an application that can adapt to vertical handover.

The VHO Ämppäri is an audio service designed as a research subject for the study of vertical handover and adaptive applications in vertical handover. MobileIPv6 [MobileIPv6] and a Simultaneous Multiple Access (Ylitalo et al. 2003) (SIMA) enabled Linux kernel form a base for the VHO Ämppäri.

The VHO Ämppäri allows a user to play audio from a remote server and change the audio stream played. The VHO Ämppäri can also adapt by changing the bit rate of the audio stream being played in event of a vertical handover.

VHO Ämppäri is a term used to describe the entire audio service. The VHO Ämppäri consists of four parts, Ämppäri player, Ämppäri server, Control protocol for Ämppäri and the VHO application programming interface (API). The client side of the VHO Ämppäri has the Ämppäri player as well as the VHO API. The server end of the VHO Ämppäri has the Ämppäri server. The control protocol for Ämppäri is used for control communication between the player and the server end in the VHO Ämppäri.

3 Application communication in vertical handover

In vertical handover the Ämppäri player prototype has a fundamental need of communicating with both the low-level network interfaces as well as on application level with the corresponding Ämppäri server it is connected to. Most non-vertical handover enabled applications need not know how the terminal device is connected to the network or if the connection happens to change. The Ämppäri player however requires this information to provide the optimal quality of the service based on the context and the preferences of the user. Should the capabilities of the network connection in use change so that the present quality of the audio service can no longer be maintained, the Ämppäri player must adapt to using a lower quality stream to prevent the network connection from becoming clogged by the traffic. Likewise, if a higher quality stream can be accommodated the quality of the audio stream should be increased if it pleases the context and the preferences of the user. The network interface information is highly platform dependent and can differ greatly even between seemingly similar operating systems.

For an audio stream there is always an Ämppäri server that, using the current protocols, cannot know of a vertical handover happening on the Ämppäri player. The server can detect a change in the IP address of the Ämppäri player but has no way of knowing anything about the network capabilities of the new connection. There are no trivial means for transferring this information either, in case of the VHO Ämppäri the audio data stream cannot be used to transfer control information. The protocols involved with vertical handover, namely MobileIPv6, neither have any means of transferring the detailed connection information required by the Ämppäri server. It is also possible that the system running the Ämppäri player might not want to use all of its network capacity for the audio stream.

3.1 VHO application programming interface

It is necessary for the Ämppäri player to obtain low-level information on network interfaces. Unfortunately this information is behind a cumbersome kernel interface which itself is work in progress and bound to change sooner or later. To overcome this problem an additional layer was developed between the application and the kernel interface. This layer, the VHO application programming interface (API) was designed as a library to offer a reasonable abstraction for applications and to lessen the impact for them, should the underlying kernel interface change or be replaced. The operating system level interface for vertical handover is unlikely to be standardized in the foreseeable future, this in mind, having an additional layer to hide this interface also greatly increases the portability of the Ämppäri player to other platforms.

VHO API consists of data structures and function calls necessary for the Ämppäri player or any other application to handle vertical handover while hiding the kernel level implementation entirely. The VHO API is responsible for informing the application of any changes in the network connectivity of the terminal device and providing any information concerning the network connections in an application friendly format. The entire VHO API was designed with the application in mind to allow easy implementation of future vertical handover enabled services.

The VHO API also provides tools for experimenting with vertical handover applications, these include features like logging and synchronous/asynchronous function calls to match the needs of the application.

3.2 Control protocol for application level control

Since neither the protocols used in vertical handover nor the audio stream can accommodate control information a new protocol was designed and implemented. The base of the protocol, known as just control protocol, was also designed with primarily the VHO Ämppäri in mind. An Ämppäri specific module was then implemented for the control protocol, known as control protocol for Ämppäri. The structure of the protocol base is highly modular allowing it to be used for almost any service, not just the VHO Ämppäri, requiring external control.

The control protocol base provides an external control connection separate from any possible data streams. A network connection like an audio stream in the VHO Ämppäri is not meant for upstream traffic and including the control information within the audio stream would only be considered an evil hack. The best solution for the VHO Ämppäri in its current state was the control protocol for Ämppäri. The control protocol is entirely an application level protocol working on top TCP/IP. It uses traditional client-server architecture and the control protocol server keeps the client connections separate allowing several instances of the Ämppäri player to work on a single Ämppäri server without any ill effects.

Using Simultaneous Multiple Access (SIMA) it is possible to use the control protocol on the most stable connection available while using the connection with the highest throughput for the data stream. This way, should the data connection be lost the control connection still remains and can be

used to quickly form a new data connection. System running the Ämppäri server does not require any vertical handover or simultaneous multiple access features as all effects resulting from them are limited to the client and handled on the server end using the control protocol for Ämppäri.

3.3 VHO API and control protocol for faster application development

While both the control protocol and the VHO API were initially created for the VHO Ämppäri, both implementations were made with future services and reusability in mind. Neither the VHO API or the base of the control protocol are in any way VHO Ämppäri specific and could be used as a basis for any future vertical handover enabled service.

Aside from reusability an important aspect of both the VHO API and the control protocol is to allow experimentations on vertical handover enabled services. While both are equally important to the VHO Ämppäri they are also unrelated. Both solve a different problem for vertical handover enabled services and either could be replaced with a different solution without affecting the other. This way the VHO Ämppäri is not considerably bound to any previous design decisions and can easily be improved or experimented with.

The detailed descriptions of both the VHO API as well as the control protocol base or the control protocol for Ämppäri are beyond the scope of this paper. Detailed architectures of both are to be published in future publications covering the VHO project.

4 General overview of the VHO Ämppäri

The figure 1 shows the architecture of the VHO Ämppäri. In the figure dashed lines mean a control stream and straight lines mean a data stream, thus data and control streams are separated. On the left hand side there is an overview of the components, which are described in more detail on the right hand side. Items marked in bold are developed within the VHO project, other parts like the network utility Netcat (Netcat, A network utility, Ipv6 Branch) and the audio player Madplay (Madplay, A command line MP3 Player) are existing tools.

VHO Ämppäri includes following parts as shown in figure 1 from top to down:

1. Ämppäri server includes a Connection Control Server and an audio handler. These are responsible for serving audio files.
2. Ämppäri player plays the audio and communicates with the Ämppäri server.
3. Control protocol for Ämppäri is used for control communication between server and player.
4. VHO application programming interface for gathering data about the currently used network and network changes.

In addition VHO Ämppäri utilizes VHO operating system modules. These are responsible for changing networks according to the current policies and available networks.

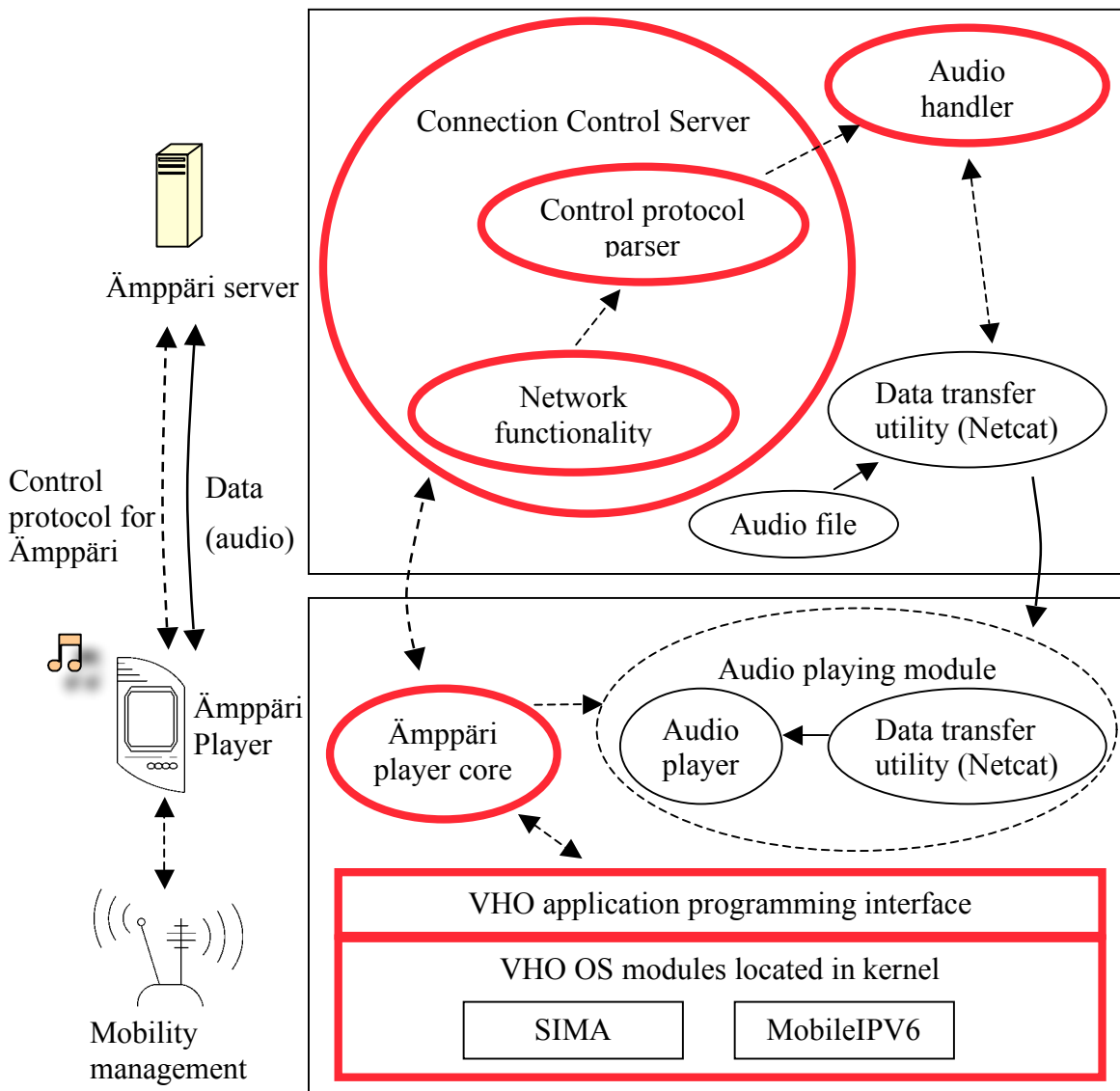


Figure 1 - Architecture of the VHO Ämpäri

4.1 Transferring and playing the audio

Studying audio or network applications is out of scope in this project, thus audio transfer and playback is handled using existing applications. Using existing applications adds flexibility and makes easier to exchange and reuse components in the future.

Netcat network utility was chosen for transferring audio. Netcat is a simple utility that can read and send data from and to the network. It was chosen because it is simple to use and it can be easily controlled from external applications.

The working principle of the data transfer is simple. On the server side Netcat reads audio file from the disk and serves it to the network. On the player side Netcat connects to the server and feeds audio data to the external audio player.

Madplay is used as an external audio player. Madplay is a command line audio player for Linux and Unices.

Any audio player, which can read audio from the standard input, can be used with VHO Ämppäri. Currently, the MP3 (MPEG-1 Layer 3, ISO/IEC 11172-3) file format is used for transferring audio. Other file formats can be used as long as they support reading from the middle of file.

4.2 *The server side components*

The Ämppäri server includes the Connection Control Server that handles control communication with the client and an audio handler that serves the audio streams to the Ämppäri player.

This kind of solution makes easy to add support to new services since most of the functionality within the Connection Control Server is service independent. Most difficult part is in finding the suitable data transfer method and an application for playing new content.

4.2.1 *Connection Control Server*

Connection Control Server communicates with the Ämppäri player using the control protocol for Ämppäri. It provides information about services and available files to the player. In addition, it forwards service related commands, like file requests, to the appropriate handlers.

When an audio file is requested, request is parsed and available port is reserved. This port will be used for serving the file. Each new data request requires a different port for data transfer, thus the Connection Control Server keeps track of used and unused ports.

Reserved port along with other information like filename and bit rate is stored in the shared memory and forwarded to the audio handler, which is executed as an external process.

To avoid re-encoding audio files in real time, server contains separate audio files for each bit rate. It would be possible to re-encode files for the requested bit rate, but this solution would require much more processing power with large amount of users.

Connection Control Server is mostly service independent, thus it is easy to add support for new services by implementing a handler for the new service.

The audio handler

Audio handler is responsible for serving the audio files to the player. It is written in Perl because it is easier to capture output from external programs from a Perl script.

Upon execution audio handler reads identification number to the shared memory from the command line and reads parameters from the shared memory. Audio handler plays files using a Unix dd utility (dd, A POSIX standard utility, IEEE Std 1003.2) and Netcat, e.g. `'dd if=/song-01-128.mp3 | nc -l -p 1234'`

Audio handler also checks shared memory for the new commands. New request for the same file may arrive if user presses pause, or the vertical handover happens and the same file is requested with a different bit rate. In that case, skip attribute of the dd utility is used to serve the file from the specific point.

4.3 *Ämppäri Player*

The Ämppäri player uses control protocol for getting information about the available files and making the file requests. Audio is played by controlling an external audio player. The Ämppäri player gets information about network bandwidth and vertical handover events from the VHO application programming interface.

Upon initialization the Ämppäri player opens a control connection to the server and gets information about available audio files. It shows a window for selecting audio file with basic controls like play, pause and stop.

4.3.1 *Playing the audio*

Player requests an audio file using control protocol. When the Ämppäri player receives an acknowledgement from the server, player executes Netcat and external audio player to play the audio.

To improve quality, buffering is used when playing audio. Buffering is needed because in a vertical handover situation it may take some time to establish a new connection. Also, with a low bandwidth connection like GPRS, audio skips without buffering even at low 24 kbps bit rate.

Used buffer size depends of the used bit rate and the network. With high-speed networks buffers are larger since they can be filled quickly.

4.3.2 *Handling vertical handover when audio is playing*

When a vertical handover happens, the event handler within Ämppäri player gets a notification from VHO application programming interface. The event handler then queries bandwidth of the new network and if the audio is playing, changes bit rate of the audio accordingly.

If the network characteristics have changed enough to warrant change in the bit rate, event handler for vertical handover within the player request same file with another bit rate from the Connection Control Server via control protocol.

Connection Control Server parses the message, forwards the new bit rate to the existing audio handler, and replies to the player that everything went ok.

When the player receives a reply from Connection Control Server, player kills existing audio player. After the data connection between player and audio handler is lost, audio handler in the server side starts a new Netcat to serve the file from the same spot using a new bit rate. Now player starts a new audio player to play audio at changed bit rate.

5 **Conclusions**

Using ready-made components is very important in this kind of work. It makes the development much faster and adds flexibility. If some component does not work as expected, it can be replaced with another with similar functionality. Ready-made components also make VHO Ämppäri more portable since much software such as audio players are available for the virtually any operating system.

Separating the control and data connection proved to be an efficient solution for the VHO Ämppäri. Playing and transferring the audio could be done using existing tools. The solution of external control, i.e., control protocol for the Ämppäri, proved to be robust even in error situations. Implementing future vertical handover enabled applications is also eased by the high modularity of the control protocol base.

Use of the VHO API provided considerable flexibility for the VHO Ämppäri. With minor alterations the VHO API could be used to simulate a vertical handover capable network and vertical handover events without the need of actually constructing a real vertical handover capable network.

Currently the VHO API contains considerable overlapping within function calls and data structures. It is still under research what is the most suitable interface for vertical handover enabled applications.

One important question remains: “What to do when network is completely unavailable, or is too slow to transfer even low quality audio?” One possible solution is to keep a small cache of users favourite songs in the local memory if there is enough storage space.

As further research a minimal yet complete vertical handover capable kernel interface should be considered to allow easy portability between different operating systems and mobility solutions.

The VHO Ämpäri uses MobileIPv6 as a mobility solution which, in the future, could be replaced by the host identity protocol (HIP), which offers a more mature form of mobility. This change would have only a minor effect on the actual VHO Ämpäri as all of the low level mobility is hidden behind the VHO API.

6 References

MobileIPv6, draft-ietf-mobileip-ipv6-24, Description of the mobile IPv6 protocol, <http://www.ietf.org/internet-drafts/draft-ietf-mobileip-ipv6-24.txt>

Jukka Ylitalo, Tony Jokikyyny, Tero Kauppinen, Antti J. Tuominen, Jaakko Laine: Dynamic Network Interface Selection in Multihomed Mobile Hosts. 36th Hawaii International Conference on System Sciences (HICSS-36 2003)

Netcat, A network utility, IPv6 Branch, http://freshmeat.net/projects/netcat/?branch_id=40599

Madplay, A command line MP3 Player, <http://www.underbit.com/products/mad>

MPEG-1 Layer 3, Popular audio format, ISO/IEC 11172-3

dd, A POSIX standard utility, IEEE Std 1003.2

Host Identity Protocol, Mobility and Multihoming, Internet draft, draft-nikander-hip-01

Ranta, M., Asplund, H., Saarinen, T. and Sahi, S. 2004. Scenarios and use cases in innovation prototyping of VHO Play. To be published in the proceedings of WWRF11 meeting, Oslo, Norway, 10-11 June 2004.