

# Performance Evaluation of Service Connectivity in Mobile Augmented Reality

Tuukka Turunen, Tino Pyssysalo and Juha Röning

*Computer Engineering Laboratory, Department of Electrical Engineering, P. O. Box 4500, FIN-90014 University of Oulu, Finland*

*{Tuukka.Turunen, Tino.Pyssysalo, Juha.Roning}@oulu.fi*

**Key words:** Mobile multimedia, mobility management, adaptive protocols, inter-technology handoff, ubiquitous computing, IP mobility.

**Abstract:** The number of different networks available in the future and the ubiquitous nature of services will pose stringent requirements for interoperability and service connectivity. An architecture for mobile AR with service connectivity support will be needed to enable smooth operation of large-scale mobile AR services. One important ability of this type of architecture is the ability to switch networks without losing the service in inter-technology handoffs. This can be implemented in various ways, but many of these options cause significant delays of over three seconds, which is unacceptable in most mobile AR applications. Our approach is to manage service connectivity in the upper protocol layers, which results in significantly shorter delays in inter-technology handoffs and service provider changes. In this article, we will describe the key performance issues of an IP-based service connectivity architecture for mobile AR, present an approach for implementing inter-technology handoffs, describe an experimental system used for measuring the performance of a system based on this approach, and present and analyze the principles of service connectivity performance. The analysis will show our approach to be good for realizing large-scale mobile AR services as well as other mobile multimedia services that need to utilize several different wireless networks for connecting to services.

# 1. INTRODUCTION

Augmented Reality (AR) differs from Virtual Reality (VR) in that it incorporates virtual objects into the user’s perception of the real world [13], whereas VR consists of only virtual components. The virtual objects of AR can be sound, force feedback, graphics or text, for example. We define AR as any instance of computer-generated objects being added to the user’s perception of the real world. Mobile AR is a form of AR not fixed to any one location, i.e. the user can move freely while using mobile AR services [1].

Globally interoperable networked virtual environments, such as mobile AR systems, offer tremendous possibilities for novel value-added services. Mobile AR, however, poses stringent requirements for the enabling technologies. One important requirement is a global service architecture, and especially service connectivity, which has been frequently neglected in the current suggestions for mobile VR and AR systems. Service connectivity issues include such questions as how to find an optimal service provider, how to use the service while moving, and when and how to switch to another provider. An intelligent service connectivity architecture is needed to ensure smooth operation of services and interoperability of different systems.

To be able to handle all service providers and to be easy to use, the system architecture must provide advanced service connectivity support. One scenario of different service providers and methods of communication is presented in Figure 1. The service providers are reached via several different (wireless) networks. This necessitates inter-technology handoff [15], which means an ability of the terminal to switch between different physical network technologies. In addition to the capability of switching between networks, a service connectivity architecture is needed to provide the logical connections between the user and the service(s).

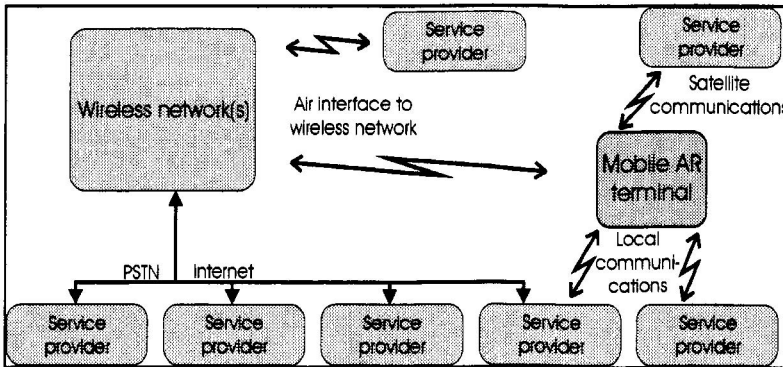


Figure 1. Service architecture of a typical mobile AR environment

Service connectivity is defined as the way and means to find an optimal service and to maintain the connection while the client and sometimes also the service provider or the content are moving. It provides a solution that enables the user to switch between service providers without a discontinuity in the service. Service connectivity research has focused on network level services in wireless environments, in mobility management [12,14], and on resource discovery [2,4,5,11]. We have extended the service connectivity research to cover highly demanding mobile systems to suit the needs of mobile AR [22,23,24]. In this paper, we will concentrate on the principles and justifications for service connectivity performance and the performance analysis of our solution [23,25].

#### Adaptation in IP-based mobile AR

The most prominent infrastructure for the upcoming mobile AR systems is the Internet. As mentioned in the introduction, the networks needed in mobile AR are available all around us, and they merely need to be managed in a way feasible for mobile AR. The problems related to the use of Internet technologies in mobile AR are mainly caused by protocols not fit for wireless environments. This is due to the fact that the network and transport protocols were never designed for environments that involve high mobility and unreliable channels. There are, however, possibilities to correct this, as it will be pointed out in this section.

One issue of service connectivity is the need to maintain the data stream between the service provider and a mobile terminal. The continuously changing Quality of Service (QoS) of packet-connected networks makes this a challenging problem in wireless and mobile applications. The three most widely used QoS parameters that indicate the quality of a packet-switched connection are latency, jitter (standard deviation of latencies), and throughput. The flow control mechanisms of conventional streaming protocols, such as TCP, help to avoid congestion in wired networks, but perform extremely poorly in terms of throughput and jitter in wireless environments. TCP uses an exponential back-off algorithm to delay the retransmission of a data packet in case it was lost. Exponential back-off decreases congestion effectively, but waiting obviously increases latencies, which in turn increases jitter and decreases throughput [3].

In addition to back-off of retransmissions, resource reservation protocols [6] can be used to eliminate congestion in shared transmission media, but congestion is only one reason for QoS reduction. In WLAN, a Point Control Function (PCF) polls mobile stations, requiring real-time data transfer, and gives them a deterministic time slice for transmission. However, PCF cannot be used in adjunct cells. In addition, resource reservation only guarantees a certain amount of bandwidth – not the real throughput in the channel.

Adaptive protocols adapt the transmission of a data stream to suit the existing transmission conditions. Adaptation may include a need to change the data rate or packet size of the stream or to delete low-priority packets. Using adaptation, the throughput of the stream can be maximized with respect to the quality of the transmission channel. Throughput may be reduced due to congestion or interference in the channel, and the best result in such a case is achieved with a combination of resource reservation and adaptation, where the former protects against congestion and the latter against interference. Reduction of throughput increases the latencies, while the increase of jitter may be minimized. Thus, the user does not experience discontinuities when exploring the augmented environment.

There are several adaptive streaming protocols available, the most common of which are RTP and RTSP [20,21]. In all streaming protocols, adaptation is based on a feedback channel, which is usually implemented with periodic control messages, which indicate, for example, the number of lost packets, the inter-packet arrival delays, and jitter. The problem is to find a suitable period for the feedback of control messages. Too short periods lead to a high traffic load and congestion and thus a reduction of throughput. Too long periods adapt the stream too slowly, which again reduces throughput.

We have developed an adaptive protocol called Mobile Augmented Reality Adaptive Transport Protocol (MARATP) [18, 19], which is both light to implement and very fast in adaptation. Rather than utilizing the sequence numbers of the transport layer protocols, we used the data link layer error detection mechanisms to increase the speed of adaptation. At the transport layer, a lost packet is detected when one or more sequence numbers are missing in the received data stream. At the data link layer, however, each properly received data fragment is acknowledged. By making the acknowledgement time short enough, lost packets are detected considerably faster than at the transport layer.

The popularity of RTP and RTSP is based on the fact that they can be used in any network that supports UDP or TCP protocols. The same is not true of protocols operating at the data link layer, unless the protocol has been designed properly. To provide maximum support for our protocol, we used standardized meters in adaptation. All IEEE 802.11 compliant WLAN access points have an SNMP database with information about, for example, the lost data fragments. We use this standardized IEEE 802.11 aFailedCount MIB in Breezecorn's AP (1.3.6.1.4.1.710.3.2.4.2.1.5) to detect the number of lost data fragments before each transmission, which allows adaptation to be done fast and reliably.

In Figure 2, we show the performance of our adaptive protocol compared to the other common transport layer protocols UDP, TCP, and RTP. The

quality of the wireless link was excellent, and the mobile station did not move during the transmission. The transmission rate was 200 packets per second, and each packet was 192-byte long. Each transmission contained 12,000 packets. 192 bytes constitute the four sensors of the Polhemus tracking device (head, left, and right hands), each of which is 47 bytes long. Additional 4 bytes are used for sequence numbering (2 bytes) and application-specific control.

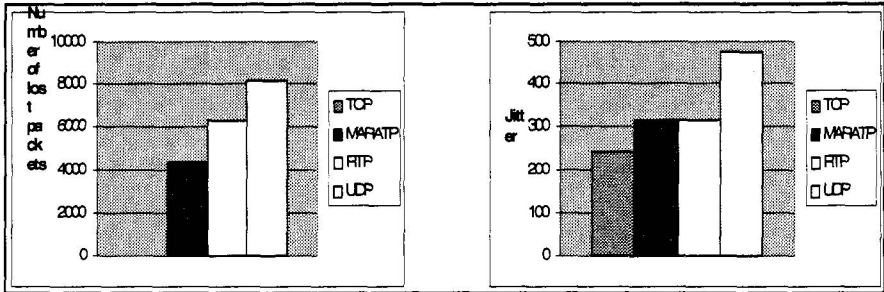


Figure 2. Comparison of the performance of IP-based streaming protocols in a static environment

It is obvious that TCP outperforms other protocols easily because exponential adaptation can find the optimal transmission rate to maximize throughput in the channel. Jitter is defined as a standard deviation of inter-packet arrival times. The small jitter indicated that TCP can provide the smoothest stream of data packets to the terminal. However, as soon as the quality of the link declines, TCP is not applicable at all, because TCP disconnects if there are too many dropped packets. The performance of UDP also deteriorates in terms of throughput. When we transmitted 24,000 packets, more than half of them were lost by UDP.

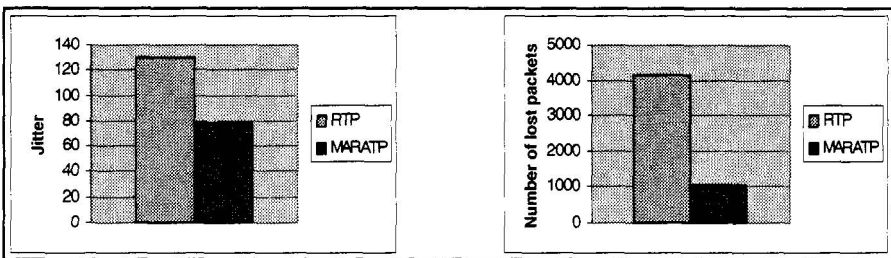


Figure 3. Comparison of the performance of RTP and MARATP in a highly mobile environment

Figure 3 shows the number of lost packets and the jitter of the MARATP and RTP protocols. In these tests, the mobile station moved around the base station, and the link quality hence varied from excellent to very poor. As mentioned, it was not possible to transmit a reasonable amount of data using TCP or UDP. When MARATP is used, the number of lost packets and jitter are significantly smaller than with RTP, mainly because of the much faster data link layer-based adaptation.

## **2. NETWORK LAYER MEDIA ADAPTATION FOR PERFORMANCE OPTIMIZATION**

In addition to the transport layer protocols, the operation of the network layer affects the performance of the system significantly. One technology designed to provide mobility support for IP networks is mobile IP [16, 17]. In this approach, separate IP addresses are utilized by assigning each device a home address to which the packets are sent and by forwarding them to the roaming address in which the device is. Although mobile IP solves the problem of switching between different networks, it is not suitable for the mobile AR architecture without modifications. This is due to its performance in switching networks, which is too slow for real-time services, and to delays caused by the need to reroute packets.

In tests performed in the University of Sheffield, England, the efficiency of mobile IP handoffs using TCP and UDP protocols was measured [8]. Different switching algorithms were tested, but even in the best case, the delay was three seconds with UDP. The performance of TCP is even worse due to the exponential back-off problem explained in Section 2.2, which causes an additional delay in transmission even after the mobile IP handoff has been completed, resulting in delays of the order of ten seconds. The tests were planned for WLAN operation, but a fixed Ethernet was utilized in them. A wireless environment causes an additional delay in the handoff, further detracting from the performance of mobile IP.

Slightly improved mobility support is included in the upcoming IPv6 [10], but even that will be insufficient for most applications requiring fast mobility, e.g. mobile AR. A simple way to provide better mobility support via fast handoffs is to utilize several physical layers and to perform handoffs in the upper layers, e.g. the transport layer. The disadvantage of this is the utilization of several IP addresses, which are currently quite scarce resources. In the upcoming IPv6, however, the number of addresses is so huge (due to the 128-bit address space) that it is no problem to allocate several addresses to one device.

Another disadvantage of the network layer media adaptation is that TCP sessions (or sessions of the upper layers) are not automatically retained. To facilitate this, server side support for session control is needed in the architecture. In practice, this is achieved by synchronizing the service according to the principles described in the Sections 4 and 5.

In addition to the switching capabilities, a modified network layer is needed to facilitate the use of the different physical media available in the future terminals utilizing the service connectivity architecture. Although service connectivity is managed in the transport layer and the layers above it, a modified network layer or separate network layers for each physical medium are needed. In Figure 4, wireless access is provided by Bluetooth, GPRS (cellular system) or WLAN. We consider this a common minimum set of the possible ways to contact the service providers in a mobile AR environment. This approach allows the wireless connection to be switched from, for instance, WLAN to cellular (GPRS) without an interruption in the service. This requires support for handoffs between the different physical network technologies, which is likely to cause delays or even interruptions of service. To make the network switch transparent for the user, a slightly modified network layer, incorporating IP with media adaptation, will be needed.

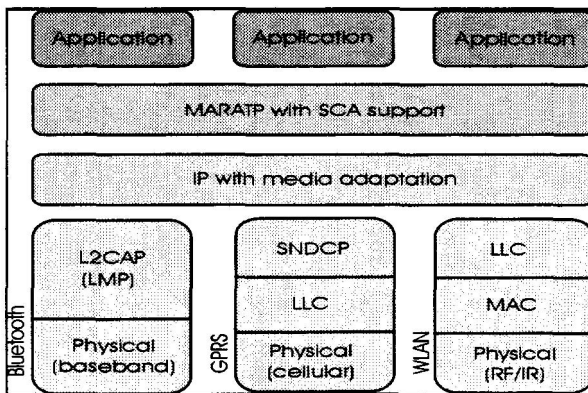


Figure 4. Adaptation to different physical media is accomplished in the network layer

The network layer (IP with media adaptation) manages simultaneous connections to service providers via separate physical channels. Another way to implement the same type of functionality is to use a separate IP layer for each of the different media. Either approach allows fast switching between the networks in the transport layer (MARATP with SCA support) or the layers above it. Our approach is to manage service connectivity in the transport layer, which is the first layer in which network independence can

be created. As mentioned above, the upcoming terminals will have a possibility to use several access media for connecting to service providers. By using them simultaneously, the delays related to handoff can be minimized from the application viewpoint.

#### System architecture example

The system is divided into three processes, which are executed in the mobile terminal. The detailed SDL [7] design of the processes has been described in [23]. DBSearch is responsible for finding databases that list service providers. SPSwitch operates the service provider switch when the lower layers decide to make a handoff to a new cell or a new network. SPSwitch uses the SPSearch process to find and locate a service provider, into which the terminal camps. It also operates on the server side, providing service (content) synchronization between the service providers upon the switching. The service provider may be in another physical network, and the switch of service providers may thus include a switch of networks as well.

In our experimental system, the processes are implemented in the Java programming language, which has several advantages, including good support for networking. The implementation is quite straightforward and compatible with the principles of SDL design. Actually, it is possible to automatically generate the code (structure) from design. We, unfortunately, only had a C language code generator available, and some manual labor was therefore required.

The key focus on implementing the experimental system is related to the SPSwitch process, which is discussed in detail in Section 5. The other two processes are also implemented, but the counterparts of their interfaces to the environment are modeled as software stubs. This is due to the fact that the goal is to evaluate the performance of service connectivity - not to implement all parts of the architecture.

The experimental implementation of the service connectivity architecture allows measurement of the performance of the inter-technology handoff between different networks, e.g. WLAN (IEEE 802.11b) and GPRS (packet-switched extension of the GSM cellular network). This type of inter-technology handoff can be implemented in several different ways, some of which are discussed in [15]. We selected an approach where both networks (WLAN and GPRS) are active and, when a handoff is needed, the transition from one network to another is performed in the application layer. The system architecture of one experimental system is as depicted in Figure 5.



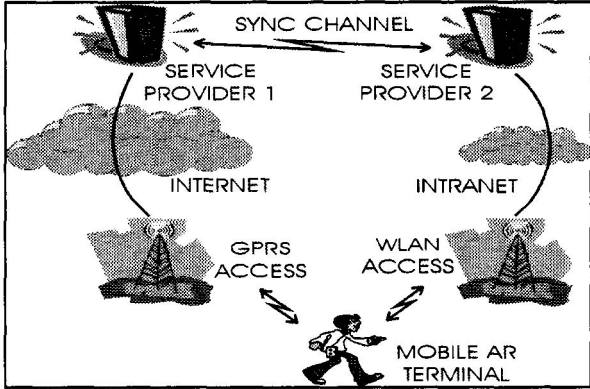


Figure 5. Example setup for performance measurements of our service connectivity architecture

As the support for multiple network interfaces is not sufficient in the operating systems that offer suitable device drivers for commercially available WLAN and GPRS products, we modeled the different physical networks in the software. This has several advantages over the usage of actual hardware components in the experimental system, such as easier measurement and simpler implementation. The main reason to use actual hardware for the experiments is to have the real thing implemented, but the advantages gained by this are minimal. We therefore decided to use software to model such things as handoff time and GPRS network characteristics.

The time to perform the handoff can be determined by different criteria, all of which involve the use of signal strength, throughput, or similar measures to decide when the handoff should take place. In our architecture, the handoff criteria are not fixed and may vary between implementations. In our experimental system, the handoff is triggered by the user, which allows the handoff performance to be measured accurately and with reliable repeatability.

When the handoff is triggered, the current service provider (e.g. service provider 1) is informed that the service is to be switched to another service provider (e.g. service provider 2) that has been selected from the list of available service providers. Then the service providers (e.g. 1 and 2) synchronize the content (e.g. via Internet), and after the synchronization, the service is transferred from the first to the second service provider (e.g. from 1 to 2) and the network used by the terminal is simultaneously changed.

### **3. SPSWITCH PERFORMANCE**

The key concepts of our service connectivity architecture can be tested with an experimental system, as described in Section 4. The key process in the architecture is SPSwitch, which takes care of switching the service from one service provider to another. The switch of service providers can be due to a number of reasons. Two of the most significant situations are running out of the service area of a service provider and running out of the coverage of the current network. In theory, a network change can be done without changing the service provider, but this is actually handled in the same way as in the case where both the service provider and the network are changed.

The SDL [7] process model, which is a finite state machine, of the SPSwitch process is presented in Figure 6. As shown by the Figure, the key functions of the process are the search and selection of suitable service providers from a list of available service providers, negotiation of the quality of service parameters, synchronization of the service, and the change itself. The input and output signals of the SDL processes are connected to the environment (system). The outputs initiate respective procedures indicated by their names, and the inputs are caused by actions of the environment.

The inter-technology handoff between WLAN and GPRS networks in both directions is simulated by the software. The switch times are not specifically optimized, as the prototype service connectivity architecture is for research purposes only. Still, the benefit of our approach is huge compared to the time spent performing the same switch with mobile IP [8,16] because the initiation time and delays caused by mobile IP are not present.

When building the actual systems, the capability to use several different physical media and protocol stack modifications is essential. In the operating system, the applications and services that use the network connections should not be bound in such a way that a switch of the network will require a lot of time for re-binding. As discussed in Section 3, the architecture of the protocol stack and the utilization of the existing layers may vary between implementations. While this has some effect on the switching performance, the key factor is simultaneous utilization of different media, which allows fast inter-technology handoffs, but requires support from the server side. This approach is, however, essential in providing a sufficiently fast service provider and physical network switch for mobile AR services.

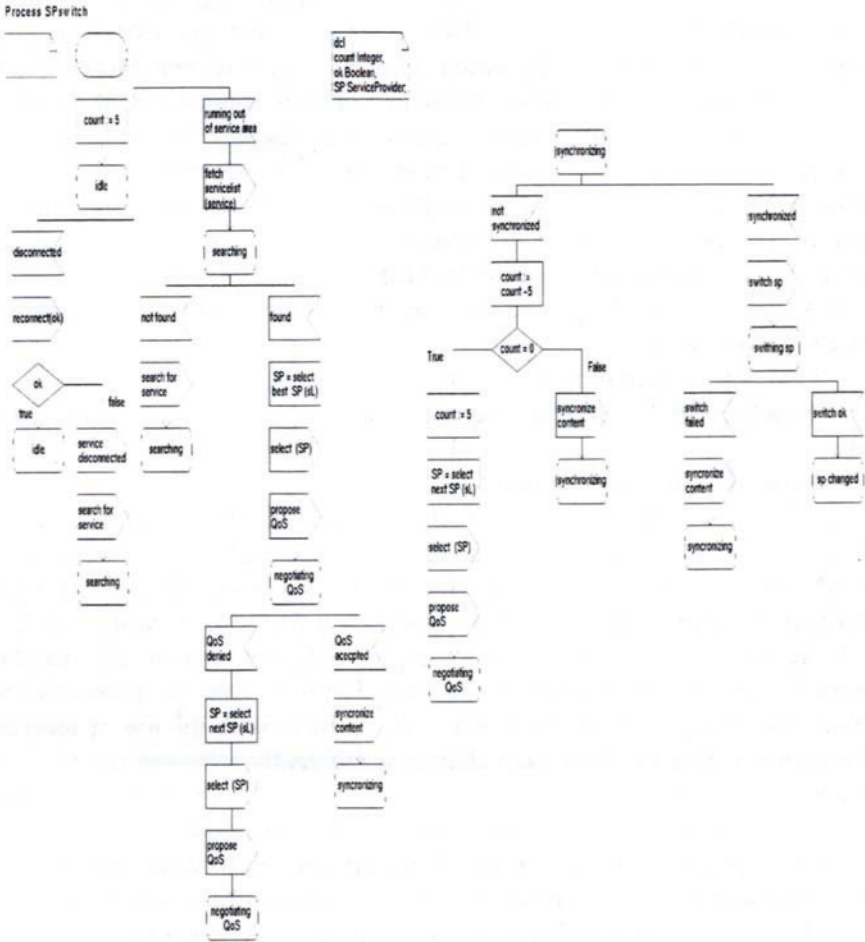


Figure 6. SDL model of the SPS switch process

#### 4. PERFORMANCE JUSTIFICATIONS

Smooth handoff at the network layer does not guarantee equally smooth handoff at the application layer, in case the application must be switched into another network. It is necessary to negotiate about the service switch with service providers in different networks before making the actual handoff. This negotiation or handshaking leads to a significant increase of network signaling, but, fortunately, the handoff is very seldom possible between more than two or three networks.

In [9], a three-level architecture for picocellular networks to support smooth handoffs is presented. Mobile Support Stations (MSS) control several Supervisor Hosts (SH), which, in turn, control several Mobile Hosts (MH) in the same way as Mobile Services switching Centers (MSC) control the Base Station Controllers (BSC) and Mobile Stations (MS) of the GSM network. MSS multicasts the same data stream to all neighboring cells of the current cell, so that the stream is available to the SH immediately after the handoff. The problem is how to control SHs in different types of network, such as WLAN and GSM, because the MSS must also change.

The service switch procedure can be divided into three different procedures (phases):

1. Find new service providers,
2. Synchronize the service content with the previous service provider, and
3. Initiate the new service provider.

To determine the start time of the service switch procedure, similar criteria as in cellular networks can be used. Although, for example, both WLAN and GSM networks provide RSS (Received Signal Strength) information, there is no standard way to fetch it from the network interface card. Actually, some cards do not support data fetching at all. Another criterion could be the location of the user, assuming that the networks can provide this information in the future. The problem with the use of location information is that the RSS may change significantly although the location remains unchanged. In any case, the handshaking between service providers must be started earlier than the actual network level handoff.

The first phase of the service switch is quite straightforward, and the user gets an acknowledgement of whether the service can be found in the new network or not. In the simplest case, this is enough. As soon as the handoff has occurred, the new service provider continues from where the old provider left off. The new service provider can start the data stream, which is not forwarded through the SH before the handoff has taken place. After the handoff, it is a matter of dozens of milliseconds before the stream reaches the mobile host.

Synchronization in the second phase is much more complicated. The problem is that although we know the average latencies between the old service provider, the SH, and the new service provider, the latencies are still non-deterministic because of network jitter. This is one reason why network jitter must be minimized by, for example, the MARATP protocol. To get rid of the effect caused by jitter, either a large buffer in the MH or a very complicated synchronization protocol between service providers is required. It is simpler to increase the buffer size, so that the application can run even when the synchronization between service providers is active. When the

handoff occurs, the MH must be able to cut off the part of the stream that has been received previously.

The third phase requires initiation of the service provider or actually a notification to a SH to start packet forwarding to the MH. This is also a fairly straightforward procedure.

The performance of the handoff should not be evaluated in units of time but rather in units of size. It is obvious that the bottleneck in the service switch is the handoff between inter-technology networks and the delays caused by the upper layer protocols due to the change of the physical connection. The amount of signaling determines the efficiency of the handoff at the level where the user perceives it. This may require connections to be established between service providers and their synchronization, which may take several seconds in the worst case of, for example, TCP handshaking. The performance of service provider synchronization is also related to the properties of the connection between the service providers. This is why content synchronization in our service connectivity architecture is performed before the handoff is inevitable.

## **5. CONCLUSIONS**

An architecture that provides service connectivity support for mobile AR is needed to enable smooth operation of global (or at least large-scale) mobile AR services. The networks available to the future mobile AR systems are numerous and the services need to be ubiquitous. This sets high requirements for interoperability and service connectivity. One important issue to be solved in order for this type of architecture to be built is the switching of physical networks without losing the service.

This type of inter-technology handoff can be implemented in various ways, but many of these cause significant delays of over three seconds, which are unacceptable in most mobile AR applications. Our approach is to manage service connectivity in the upper protocol layers and to utilize several physical (and data link) layers simultaneously, which allows significantly shorter delays in inter-technology handoffs and service provider changes.

In this article, we have described the key performance issues of an IP-based service connectivity architecture for mobile AR, presented our approach for implementing inter-technology handoffs, described an experimental system for measuring the performance of our approach, and presented and analyzed the principles of service connectivity performance. The analysis confirms our approach as good for realizing large-scale mobile

AR services as well as other mobile multimedia services that need to utilize several different wireless networks for connecting to the services.

## 6. ACKNOWLEDGEMENTS

The authors acknowledge the Nokia Foundation, Digia and Infotech Oulu for their support.

## REFERENCES

- [1] R.T. Azuma, "The Challenge of Making Augmented Reality Work Outdoors", Proceedings of the First International Workshop on Mixed Reality, Tokyo, Japan, pp. 379-390, 1999.
- [2] D. Awduche, A. Gaylord, A. Ganz, "On Resource Discovery in Distributed Systems with Mobile Hosts", Proc. of Mobicom '96, pp. 50 –55, 1996.
- [3] H. Balakrishnan V. Badmanabhan S. Seshan R. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links", IEEE/ACM Transactions on Networking 6, pp. 756-769, 1997.
- [4] M. Barbeau, "Service Discovery in a Mobile Agent API Using SLP", Proc. of IEEE Global Telecommunications Conference, pp. 391 – 395, 1999.
- [5] P. Bellavista, A. Corradi, C. Stefanelli, "A Mobile Agent Infrastructure for Terminal, User, and Resource Mobility", IEEE NOMS 2000, pp. 877 – 890,2000.
- [6] R. Braden L. Zhang S. Berson S. Herzog S. Jamin, "Resource Reservation Protocol (RSVP) – Version 1 Functional Specification", IETF RFC 2205, 112 p, 1999.
- [7] J. Ellsberger, D. Hogrefe, A. Sarma, SDL – Formal Object-oriented Language for Communicating Systems, Prentice Hall Europe, 312 p, 1997.
- [8] N.A. Fikouras, K. El Malki, S.R. Cvetkovic, C. Smythe, "Performance of TCP and UDP During Mobile IP Handoffs in Single-Agent Networks", IEEE Wireless Communications and Networking Conference, pp. 1258 – 1262,1999.
- [9] R. Ghai & S. Singh. "An architecture and communication protocol for picocellular networks", IEEE Personal Communications 1 (3), pp. 36-46, 1994.
- [10] Internet Protocol version 6 (Ipv6) Specification, IETF, RFC2460, 1998.
- [11] E. Jung, Y-J Park, C. Park, "Mobile Agent Network for Supporting Personal Mobility", Proc. of Twelfth International Conference on Information Networking, pp. 131 – 136, 1998.
- [12] G. Liu, G. Maguire, "Efficient Mobility Management Support for Wireless Data Services", Proc. of VTC'95.5 p, 1995.
- [13] P. Milligan, H. Colquhoun, "A Taxonomy of Real and Virtual World Display Integration", Proc. of the First International Workshop on Mixed Reality, pp. 5-30, 1999.
- [14] "Mobile Radio Interface layer 3 – Specification", ETSI Technical Specification 04.08, 446 p.
- [15] K. Pahlavan, P. Krishnamurthy, A. Hatami, M. Ylianttila, J. Mäkelä, R. Pichna, J. Vallström, "Handoff in Hybrid Mobile Data Networks", IEEE Personal Communications Magazine, pp. 34 – 47, Vo1.7 No.2, 2000.
- [16] C. Perkins, "IP Mobility Support", IETF RFC 2002, 79 p, 1996.

- [17] C. Perkins, "Mobile IP", Addison-Wesley, 304 p, 1998.
- [18] T. Pyssysalo, T. Repo, T. Turunen, T. Lankila, and J. Röning, "CyPhone – Bringing Augmented Reality to Next Generation Mobile Phones", Proc. of DARE'2000 Conference, 10 p, 2000.
- [19] T. Pyssysalo, T. Turunen, "Supporting Mobility in Networked Augmented Reality", Will appear in Proc. of the 5th World Multi-Conference on Systematics, Cybernetics and Informatics, Orlando, USA, July 22-25, 6 p. 2001.
- [20] H. Schulzrinne S. Casner R Frederick V. Jacobsen, "RTP: A Transport Protocol for Real-Time Applications", IETF RFC 1889,75 p, 1996.
- [21] H. Schulzrinne A. Rao R. Lanphier, "Real Time Streaming Protocol (RTSP)", IETF RFC 2326,92 p, 1998.
- [22] T. Turunen, J. Röning, S. Ahola, T. Pyssysalo, "A Wearable Computer for Mobile Augmented Reality Based Controlling of an Intelligent Robot", Proc. of SPIE 4197 conference Intelligent Robotics and Computer Vision XIX: Algorithms, Techniques, and Active Vision, 10 p, 2000.
- [23] T. Turunen, T. Pyssysalo, J. Röning, "Design of a Service Connectivity Architecture for Mobile Augmented Reality", Workshop on Future Network Technologies, Lappeenranta, Finland, pp. 46 – 55,2000.
- [24] T. Turunen, T. Pyssysalo, T. Lankila, J. Roning, "Realization of Mobile Augmented Reality Based Services in Cellular Systems", Proc. of the International Symposium of Advanced Distributed Systems, 7 p, 2000.  
T. Turunen, T. Pyssysalo, J. Röning, "Service Connectivity Architecture for Mobile Augmented Reality", Proc. of SPIE Electronic Imaging Conference, 2001.