Service Migration Protocol for NFC Links

Anders Nickelsen, Miquel Martin, Hans-Peter Schwefel

Service migration protocol for NFC links

Anders Nickelsen[1], Miquel Martin[2], and Hans-Peter Schwefel[13]

[1] Dept. of Electronic Systems, Aalborg University, Denmark, {an | hps}@es.aau.dk
[2] NEC Europe Ltd., Heidelberg, Germany, miquel.martin@nw.neclab.eu
[3] FTW, Vienna, Austria, schwefel@ftw.at

Abstract. In future ubiquitous communication environments, users expect to move freely while continuously interacting with the available applications through a variety of devices. Interactive applications will therefore need to support migration, which means to follow users and adapt to the changing context of use while preserving state. This paper focuses on the scenario of migration between two devices in which the actual migration procedure is executed over near-field communication (NFC) ad-hoc links. The NFC link is interesting as it gives the user the perception of trust and enables service continuity in cases where mid- or long-range wireless connectivity is unavailable. Based on an experimental performance analysis of a specific NFC platform, the paper presents a migration orchestration protocol with low overhead and low delays to be used with NFC links. Experimental results allow to conclude on the sizes of application state that can be expected to be feasible for such ad-hoc NFC migration.

Keywords-near field communication (NFC); performance measurements; service migration;

1 Introduction

Future users will be equipped with different devices and expect to be able to access personal services from any device, and even change device while using an application. The process of changing the device during application use is called *migration*. A *migratory application* supports pausing, extraction of sufficient state information to resume operation on target device, insertion of state on target device and resuming of operation. The application may rely on middleware to handle control (start,resume) and transfer of state. The middleware makes the application independent of underlying platforms and networks.

Traditional migration is conducted via high-bandwidth network links (Ethernet, WLAN or GPRS/UMTS) and a central server to coordinate the migration sequence [1]. However, using ad-hoc communication between devices and in particular the RFID based Near Field communication (NFC) can be advantageous for multiple reasons: (1) The physical closeness required to create an NFC link can be interpreted as migration trigger; (2) binding the migration to the physical closeness can increase the user-trust in the migration procedure, as hijacking of

sessions by remote devices can be prevented; (3) fast connection setup times and low interference probability due to its short range can be advantageous; (4) low power consumption can make it the technology of choice in migration scenarios triggered by low battery of the source device. In addition, migrating via the NFC link may be the only option in scenarios of interruption of coverage of mid- to long-range (cellular) technologies.
Migration over ad-hoc NFC links is however challenged by throughput limitations and the potential short time-windows during which the communication is possible. Our own preliminary experiments have shown that users expect migration to finish within 3-5 seconds when using NFC. The estimates are based on familiar uses of RFID and NFC, as for instance door locks or ticketing (which require a small window to complete). The consequence is that NFC may only be feasible for a small application state sizes, and we propose a protocol targeted at maximizing the feasible state size.

Transfer of active sessions is a well-studied procedure in existing research on *mobile agents* [2] [3]. The work in this paper is based on *service migration*, where service is defined as mobile interactive applications [1] [4]. The goal of service migration is to improve the user experience by a seamless migration. This is different from the goal of mobile agents, which is to autonomously achieve a goal independent of the platform. NFC has previously been studied as a part of different application types; for service discovery in Smart Spaces [5], for tracking habits in home health-care [6] or as information carrier in marketing such as in All-I-Touch [7] and [8]. Similarly, NFC performance parameters are mostly user-experience oriented; trustworthiness [9] and usability [10] [11]. Likewise, [12] and [13] look into the security implications of NFC communications.

In this paper, we propose a migration protocol optimized for resource-constrained links, such as NFC. In addition, we determine the range of state sizes that can be transferred within the available communication windows. From this investigation, we present general performance measurements, which are also applicable in other contexts than service migration. We focus on the performance implications NFC on migration. A detailed security and investigations of complex user interactions are out of the scope of this work.
We present a migration scenario in Section 2 to illustrate how use of NFC links challenges the migration architecture. Performance results of NFC usage are presented in Section 3 to obtain approximate magnitudes of boundary conditions using NFC for migration. In Section 4 the proposed protocol for using NFC in migration is described and implementation and evaluation of the protocol are presented in Section 5. Finally, Section 6 concludes the paper.

2 Background

Figure 1 illustrates a migration scenario. The user is carrying a mobile device on which a migratory client-server application is running, for instance a video streaming application. The client-part of the application is running on top of a middleware on the mobile device; the client-side middleware collaborates with support functions on the migration server. The client-part of the application

uses special interfaces of the migration middleware such that the middleware can start, stop, pause, and resume the client-side application; furthermore, the client-side application needs to use middleware functions to store application state that must remain persistent after the migration has been executed. In case of a video streaming application, the required state information to be transferred is a URL, a time offset and potentially media relevant information regarding codecs, etc. which were received/negotiated in the beginning of the stream, for instance as a session description protocol (SDP) profile [14]. Because the video stream is not re-initialized when migrated, such information is not re-exchanged between client and server, and must thus be transferred as state. A SDP-profile is exchanged in clear-text, and its size can range from 230 bytes for a compressed video profile [15], over 860 bytes for a raw audio profile [16] and upward for more complex sessions. To avoid modification of the server-part of the application, the migration procedure is made transparent by using a mobility anchor point function (e.g. a mIP home agent [17]) in the migration server to redirect data flows during and after migration.

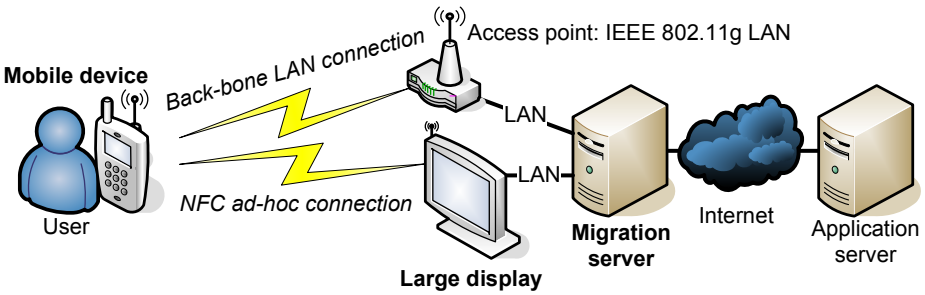

Fig. 1. A scenario of migrating the client-part of the application from the mobile device to the large display.

The client-side middleware registers devices and applications on the migration server, which orchestrates the migration through the available network connections. The users can search for suitable devices to migrate to and can trigger the migration directly from the terminal. The basic migration scenario (without ad-hoc links) from the mobile device (source) to the large display (target) includes the following activities (cf. Figure 2): 1) the user selects an application to be migrated and the large display device as target; 2) the user triggers the migration and the middleware passes the trigger event to the migration server, 3) the migration server orchestrates the migration using the client-side middleware components, which include pausing the application on the mobile device, extracting state information, transferring the state from the source device to the migration server, initializing the middleware components on the target device, transferring the state to the target (large display in Figure 1), inserting the state into a new application instance and resuming this application in the

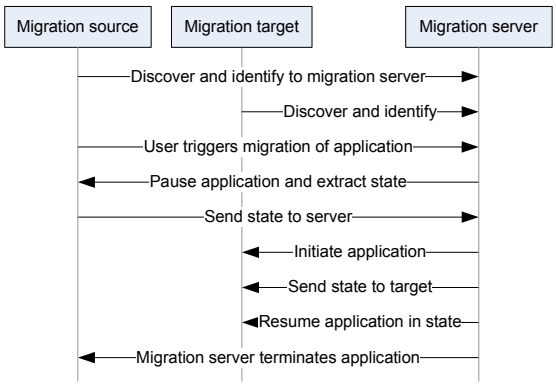

Fig. 2. Steps during a migration procedure.

original state, 4) the migration server terminates the original paused application on the source device. Note that in the general case, the trigger can be generated either by the user via a manual interaction or by the migration server through an automatic decision based on general context information [18]. We focus on the scenario of user-generated triggers. See [19] for more details.

Traditional RFID communication consists of a passive RFID tag and an active RFID reader. The reader generates a radio frequency (RF) field to request a response from the tag and the tag uses the energy in the reader's RF field to respond. One NFC entity integrates both RFID tag and reader and thus allows two NFC entities to communicate peer-to-peer. The NFC specification ([20]) allows for entities to work as traditional passive RFID tag and reader for compatibility, however, only the peer-to-peer mode is considered here, in *active* mode, where all entities generate RF fields.

NFC devices must be prepared for neighbor discovery, similar to the inquiry phase of Bluetooth and the frequency scanning phase of ad-hoc mode WLAN. An NFC device can have one of two roles; *initiator* or *target*. The initiator uses its RF field to contact targets. The target only senses for initiator RF fields and does not turn on its own RF field unless requested by an initiator as part of communication. A protocol using NFC must include at least one initiator and one target to have successful communication.

3 NFC performance measurements

This section presents an experimental performance study of an NFC link in order to obtain ranges of several performance metrics. These ranges are important for the design of the migration protocol for resource-constrained NFC links. We investigate relevant migration parameters: neighbor discovery time, transmission delay (round-trip), and throughput. The neighbor discovery time directly

subtracts from the time budget available for orchestrating migration and transferring state. The round-trip time message delay impacts duration of exchanging messages during orchestration. The throughput is used to estimate how much state data can be transferred during the remaining time budget. The setup consisted of two HTC 3600 smart phones running Windows Mobile 5 and with an 'SDiD 1010' NFC dongle [21] attached. The NFC dongles were configured in active mode with 424 kbit/s PHY data rate. A file transfer application was deployed on both devices and one application instant acted as NFC initiator while the other acted as NFC target. Since the maximum frame size was 186 bytes, fragmentation was implemented. All experiment where repeated 30 times to obtain 95% confidence intervals, which is shown with each result.

Neighbor discovery time was measured by placing the devices in range and let the initiator search for targets. The initiator blocks execution while searching for targets. The measured interval indicates the blocking time from the moment the application started searching until the target was discovered and the initiator could continue. The mean discovery time was measured to be 59 ± 0.8 ms.

Transmission delay was measured by sending the smallest possible frame size (16 bytes header, no payload) back and forth between the NFC entities. The delay is defined as a round-trip delay, i.e. as the time from starting the send procedure on one device until receiving the response on the same device. For two nodes, the average delay was measured to be 106 ± 1.8 ms.

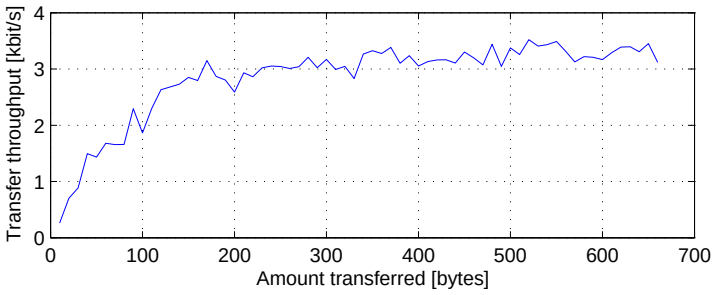

Fig. 3. Throughput measurements over NFC link

Throughput was measured by sending multiple frames continuously to mimic a large application state object. The total size of the transferred file was varied between 10-660 bytes and the time was measured from the first event until the final acknowledgment was received. Results from the experiment for different file sizes are shown in Figure 3. The figure shows that the throughput increases rapidly with increasing file sizes on the left end and then it converges to a value around 3.26 kbit/s for values above 300 bytes. The throughput degradation for small file sizes is expected as the ratio between overhead and payload is relatively high.

4 Procedure for migration over NFC

In this section, we propose an orchestration protocol that is targeted at performing migration over a resource-constrained link such as NFC. We describe the protocol details below and provide an experimental evaluation in the subsequent section.

The driving scenario is depicted in Figure 1. Some time period before the user puts the mobile device close to the large display to activate NFC neighbor discovery, the WLAN connection disappears so that migration needs to be performed via the NFC link. The lifetime of the NFC link is limited by the window of time the user holds the mobile device close to the large display. The goal of the proposed migration protocol is to maximize the probability that the application state can be successfully transferred within the available time window. In order to maximize use of the time window, the protocol aims at starting the transfer as early as possible. When the devices have discovered each other, the state is prepared in the source device and the size of the state is exchanged at first. After that, the state is transferred. Finally, when the state has been transferred successfully, the application in the source device is terminated.

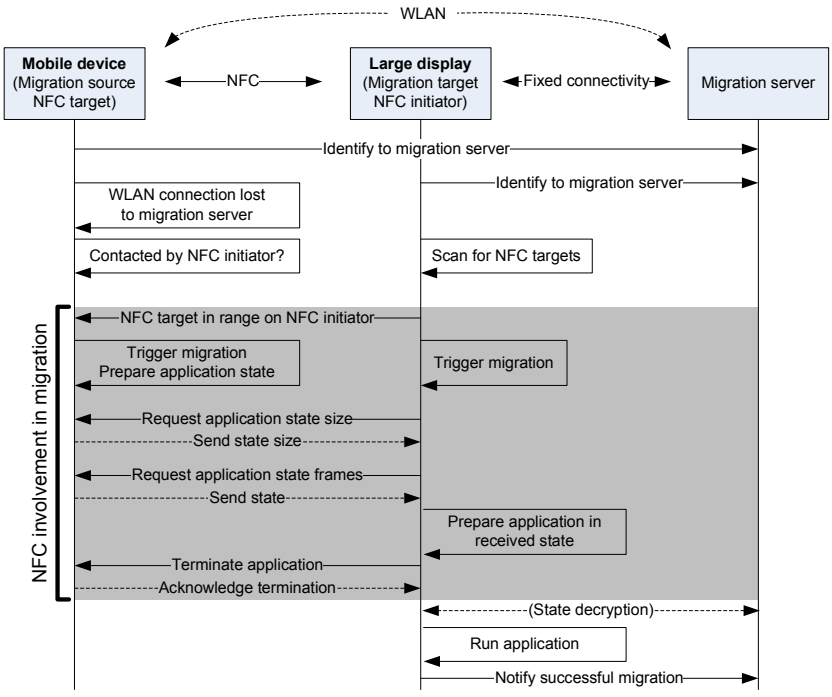

Fig. 4. Fast and low-overhead migration procedure utilizing the NFC link.

In a scenario with a back-bone connection (such as WLAN), migration decisions are made by the migration server and therefore clients spend much time waiting before transferring the state due to communication with the server. Also, when migration is controlled by the server, the transfer of the state introduces overhead as the transfer is split into an upload from the source device to the server followed by a download from the server to the target device. In our scenario where the WLAN connection has disappeared, there cannot be direct communication between source device and server. This would have to be relayed by the target device, which would be even more time consuming than normally.

In order to optimize the state transfer, delay and overhead must be reduced and interactions between source device and server cannot be performed. This requires that the decision to transfer the state needs to be made solely on the target. Moreover, the amount of signaling messages between the target and the source must be kept low due to the NFC round-trip delay. Our proposed protocol to achieve this optimization is illustrated in Figure 4 and works as follows. Both devices are assumed to have registered previously with the migration server via the WLAN connection to establish a trust relationship. The large display is configured as NFC initiator and the mobile device as NFC target. The large display actively searches for the mobile device. When the user *swipes* the mobile device close to the large display to trigger migration, the NFC initiator detects the NFC target and triggers the migration procedure on the large display.

The large display requests the application state size from the mobile device. To deliver this result, the mobile device must know which application to migrate and its state size. Several selection schemes can be employed, such as choosing the application which has the active/focused window, or the user can have selected an application to migrate manually before the swipe, e.g. when notified about the loss of connectivity. The assumption here is that the mobile device knows which application to migrate and that the application state size is known. The large display then downloads the state object from the mobile device via the NFC link. Once the download has finished, the large display sends a termination command to the mobile device, which terminates the original application. To activate the application on the large display the application must be resumed in the downloaded state. To avoid malicious stealing of applications/sessions by using migration, the state object could be encrypted and hence the large display would need to request a decryption key from the migration server or have the server decrypt the state. This way the state download can be initiated without having to establish a trust relation, which is both time consuming and would bother the user. In summary, our lean protocol relies on decentralization of decisions from the server to the migration target. The major decisions that have been decentralized are:

- Trigger the migration when 'NFC-in-range' event is detected
- Use NFC for the migration protocol and state transfer
- Use pre-defined rules or user-interaction to allow the source device to identify which application to migrate

As the role of a device in the NFC communication must be set before communication, a role selection algorithm must be in place. Selection could be based on static rules, where all devices select a certain role or have pre-assigned roles. The rule in a given scenario must ensure that roles are distributed in order to allow communication. The selection could also be based on a random-hopping scheme, where devices hop between the two roles and listen for presence of opposite roles; initiators request target responses and targets listen for initiator requests. Due to the scarceness of time in the scenario, we employ a static selection decided by the device type: Mobile devices are NFC targets and static devices are NFC initiators. The rationale is that mobile devices are typically power-constrained where as static devices are assumed to have a external power supply is in the case of the large display. As the target role requires less power than an initiator the target role is assigned to the most power-constrained device.

5 Implementation and evaluation

To evaluate the feasibility of the lean migration protocol, it was implemented using the same NFC setup as in the performance measurements in Section 3. Here, one of the mobile phones acted as the large display. An additional goal of the evaluation was to understand how much state information can be transferred within a typical swipe. The proposed migration orchestration messages GET_STATE_SIZE, GET_STATE and TERMINATE were implemented as simple string messages and the transferred state size was varied similar to the throughput measurements. The results of one experiment for each application state size in the range between 1 and 700 bytes are shown in Figure 5.

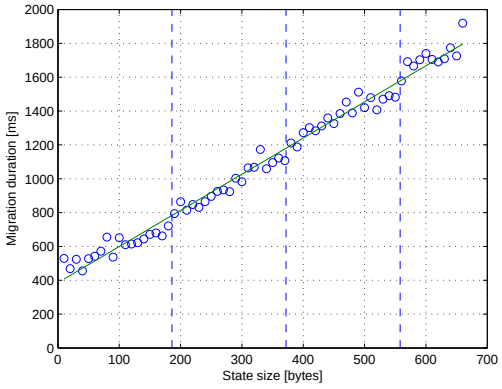

Fig. 5. Duration of a migration consisting of orchestration signaling and state transfer

The results for the migration duration show a linear behavior over state size S; a least-squares fit yields the relation $delay = 2.1\frac{ms}{byte} \cdot S + 386ms$ which is

also depicted in Figure 5. The minimum delay at $S = 0$ is due to the required 3 message exchanges. A detailed analysis of the time stamps shows that each exchange requires 106ms plus some processing delay. The inverse of the slope of the line $\frac{1}{2.1}\frac{bytes}{ms} = 3.8kbit/s$ is in the same order of magnitude as the throughput values observed in Section 3.
The observed linear behavior can be used to estimate the limit on state sizes that are feasible to transfer in a certain time window. For instance, assuming a maximum time window of 5 seconds for the transfer, the relation yields a maximum S=2197. Based on this upper limit, the orchestration entity in the target can decide whether to initiate migration when only the NFC link is available.

6 Conclusion and future work

This paper presents a solution for migration over resource-constrained links, with the specific example of RFID based near-field communication (NFC) technology. The main challenge of migrating an application over resource-constrained links is to make best use of the limited time in which the link is available. The primary steps of migration that can be optimized time-wise are device discovery, decision to trigger and message exchange during orchestration, including transferring the actual state of the application from the source device to the target device. In a general migration scenario, all these steps are coordinated by a central migration server in the network. Through an experimental setup we have shown that in some cases, migration over ad-hoc NFC links is possible and feasible. This is achieved by shifting the primary migration steps from the server to the target device. By measuring the overhead of the optimized migration orchestration protocol and the performance of NFC it is shown that the platform can be used for migrating application state sizes up to 2000 bytes within an NFC swipe window of 5 seconds.

Future work should include investigation of how feedback to the user during migration (auditory/visual) may increase the average length of the available time window. Moreover, for simplicity reasons, we have only addressed one-way migration (from mobile device to large screen). The optimized migration protocol can be generalized to handle migration both ways by introducing a few additional handshake messages during connection establishment. The performance of such a general protocol should also be investigated.

Acknowledgments

This work was partially supported by the EU ICT FP7 project 'Open Pervasive Environments for iNteractive migratory services – OPEN', see www.ict-open.eu. The Telecommunications Research Center Vienna (FTW) is supported by the Austrian Government and by the City of Vienna within the competence center program COMET.

References

1. F. Paternò, C. Santoro, and A. Scorcia, "User interface migration between mobile devices and digital tv," in *HCSE'08*, p. 292, Springer-Verlag.
2. A. Fuggetta, G. Picco, and G. Vigna, "Understanding code mobility," *IEEE Transactions on software engineering*, vol. 24, no. 5, pp. 342–361, 1998.
3. A. Carzaniga, G. Picco, and G. Vigna, "Is code still moving around? Looking back at a Decade of Code Mobility," in *ICSE'07*, pp. 9–20, IEEE Computer Society.
4. R. Bandelloni, G. Mori, and F. Paternò, "Dynamic generation of web migratory interfaces," in *MOBILEHCI '05*.
5. Z. Antoniou and S. Varadan, "Intuitive mobile user interaction in smart spaces via nfc-enhanced devices," in *ICWMC '07*.
6. R. Iglesias, J. Parra, C. Cruces, and N. G. de Segura, "Experiencing nfc-based touch for home healthcare," in *PETRA '09*.
7. F. Kneissl, R. Rottger, U. Sandner, J. Leimeister, and H. Krcmar, "All-i-touch as combination of nfc and lifestyle," in *NFC '09*.
8. S. Karpischek, F. Michahelles, F. Resatsch, and E. Fleisch, "Mobile sales assistant - an nfc-based product information system for retailers," in *NFC '09*.
9. M. Massoth and T. Bingel, "Performance of different mobile payment service concepts compared with a nfc-based solution," in *ICIW '09*.
10. I. Cappiello, S. Puglia, and A. Vitaletti, "Design and initial evaluation of a ubiquitous touch-based remote grocery shopping process," in *NFC '09*.
11. H. Mika, H. Mikko, and Y.-o. Arto, "Practical implementations of passive and semi-passive nfc enabled sensors," in *NFC '09*.
12. P. Schoo and M. Paolucci, "Do you talk to each poster? security and privacy for interactions with web service by means of contact free tag readings," in *NFC '09*.
13. G. Madlmayr, J. Langer, C. Kantner, J. Scharinger, and I. Schaumuller-Bichl, "Risk analysis of over-the-air transactions in an nfc ecosystem," in *NFC '09*.
14. M. Handley, V. Jacobson, and C. Perkins, "SDP: Session description protocol (RFC-4566)," *Request for Comments, IETF*, 2006.
15. M. Ulvan and R. Bestak, "Analysis of Session Establishment Signaling Delay in IP Multimedia Subsystem," *Wireless and Mobile Networking*, pp. 44–55.
16. M. Elkotob and K. Andersson, "Analysis and measurement of session setup delay and jitter in VoWLAN using composite metrics," in *MUM'08*, pp. 190–197.
17. C. Perkins, S. Alpert, and B. Woolf, *Mobile IP; Design Principles and Practices*. Addison-Wesley Longman, 1997.
18. M. Bauer, R. Olsen, M. Jacobsen, L. Sanchez, M. Imine, and N. Prasad, "Context management framework for MAGNET Beyond," in *Workshop on Capturing Context and Context Aware Systems and Platforms, IST Summit*, 2006.
19. Martin, M. et al, "Migration Service Platform Design," tech. rep., ICT-OPEN EU FP7 project, 2009.
20. ECMA, *340: Near Field Communication: Interface and Protocol (NFCIP-1)*. http://www.ecma.ch/ecma1/STAND/ecma-340.htm: European Association for Standardizing Information and Communication Systems, 2004.
21. http://www.sdid.com/products1010.shtml.