Policy-Based Device and Mobility Management

Pierre Imai, Bernd Lamparter, and Marco Liebsch

NEC Europe Ltd., Heidelberg, Germany
{imai,lamparter,liebsch}@neclab.eu

Abstract. Each new generation of mobile terminals offers more and better functionality, e.g. terminal mobility, multi-homing or inter-device session mobility. Furthermore, the interaction with consumer devices, e.g. DLNA TV or stereo sets, is becoming more common. Every new feature, however, is likely to result in increased complexity for the end user: Most people do not know how to utilize all features of their mobile terminals, hence devices that offer only a reduced feature set are becoming more popular. Additionally, while the end user expects to be in control, the network operator might want to exert some influence over which features are available or trigger actions, e.g. handovers, based on contract, location, etc. The aim of our research is to offer high flexibility and functionality combined with ease of use. We designed a policy management framework for the mentioned session mobility functions which supports the user in the configuration of the functions and automates commonly performed actions.

Keywords: Policy management, session mobility, home automation.

1 Introduction

Future terminals for mobile communications will offer users a variety of services through multiple radio access technologies. Mobile communication systems will implement heterogeneous access networks, where mobile terminals can attach through to operator services or services of the public Internet. Mobility management will support reachability of the user of a mobile terminal and control low latency handover even between access networks using different technologies. Simultaneous use of multiple radio connections will be supported and allows routing of different service flows through different access networks. One argument to route a particular flow through the access network of a particular technology is for example the need for a certain Quality-of-Service (QoS), which is handled well by CDMA-based access networks of 3^{rd} Generation mobile communications systems. In case delay or jitter is not of major interest, but the need for high bandwidth access is the most important aspect for the use of a particular service, associated flows may be routed through the network of a WiMAX (IEEE 802.16) provider or Wireless LAN (IEEE802.11) access.

Many users own other communication devices besides their mobile phone. PCs can be equipped with microphone and camera and then be used for video calls. Many modern TVs or hi-fi devices can also be connected to the Internet

C. Giannelli (Ed.): Mobilware, LNICST 0007, pp. 101–114, 2009.

and support features of the Digital Living Network Alliance (DLNA) [1]. This allows the complete or partial handover of sessions between devices, e.g. from the mobile phone to the PC or only the video to the TV. For session mobility IETF documents are available [2] and it is currently specified in 3GPP [3].

Certainly, operators have specific policies regulating the route of traffic. Aiming at best performance and most satisfied users, future terminals and mobile communication systems will allow flexible configuration of such policies not only to *gold subscribers*, but any user can specify its own policy profile on the mobile terminal. Decisions about the use of a particular radio technology, service or support device, such as an external loudspeaker, will take the user's policy profile as well as network policies into account.

Related work addresses policy management for various use cases, such as for the selection of a particular access technology to handover a dual-radio enabled mobile device. The IST WINNER project [4] investigates policy-based mobility management and proposes an architecture for policy management as well as rules to support decisions about access to different resources and networks. Furthermore, the proposal considers policy rules to prioritize traffic and to enforce associated QoS to that traffic. The proposed architecture considers distributed functional components in the network, which closely collaborate with the radio resource management. The IST Daidalos project did some early investigation on local policy management, which is processed on the mobile device, while taking local as well as user and network preferences and information, such as available access networks, into account. [5] describes inter-working between an interface abstraction layer to handle heterogeneous access technologies and a mobile terminal controller, which enforces policy decisions being taken by an intelligent interface selection function. Policy decisions take user preferences and network conditions into account. These and other related work on policy management focus on clearly defined functional requirements, such as IP- and link-layer handover as well as maintenance of a certain QoS. [6], [7], and [8] for example utilize policy management solely to choose the best access for terminal handover. In [7], policies help network operators to select an access network on the arrival of a new call or handover request. [8] describes adaptive terminal middleware for session mobility.

Current solutions are designed and tailor-made for a very specific function and do not consider a common policy management for different kind of mobility management schemes. Session mobility, as specified for the Session Initiation Protocol (SIP) [9], has not attracted a lot of interest so far in the design of policy management concepts whereas both types of mobility management, IP mobility and Session mobility, have potential and will co-exist in future mobile communication systems. The policy management can also play a vital role for access control to resources both within the terminal and outside. In this case, policies behave similar to firewall rules, e.g. granting or denying access to specific services based on time, application or device ID, etc.

As the complexity of mobility management using heterogeneous access networks increases and multiple schemes to maintain reachability of a user's device co-exist with mobility-enabled session management along with a variety of approaches to control traffic flow routing, we endorse a powerful policy management on the mobile terminal, which takes various information sources and policy rules as input to handle all relevant tasks in a centralized component. These tasks include the selection of an appropriate mobility management scheme (IP Mobility vs. SIP mobility). The proposed policy management solution is meant to ease handling of complex rules and tasks as well as to personalize associated decisions taken by the policy management. At the same time, the level of user interactions to build a policy database on a mobile device must be adjustable to not stress unexperienced users with expected input or manual configuration.

In Section 2, we motivate and support the design of a common policy management solution for mobile devices by means of use cases and Section 3 describes some user-related aspects of policy management. Section 4 describes the proposed architecture and Section 5 describes the policy management mechanisms and language. Section 6 provides an overview about sample protocol operations between the architecture components. Section 7 concludes this paper.

2 Exemplary Use Cases

Making use of a scenario-based design, this section briefly describes three different use cases and the role of policy management on the mobile terminal. In a high-level design, the policy manager component implements a central function that takes decisions based on policy rules, states and events. Various event sources provide input to the policy manager, whereas decisions result in input to and control of action components. Event sources comprise for example a trigger from a user or an application, whereas action components comprise for example functions for mobility and session management.

A first use case targets support of inter-technology handover along with IP mobility management. As a result of a user's mobility pattern and location, the link quality of its currently used cellular radio access gets worse, which is reported from network interface control components to the policy manager. As a first step, the mobile terminal initiates discovery of alternative network technologies and associated access points as well as operator preferences regarding technologies to be used. Based on the result of the discovery, the mobile terminal performs scanning of these technologies and provides the results to the policy manager, which selects a handover target while taking the scan results, operator preferences and user policies as well as costs aspects into account. In a last step, the decision of the policy manager results in a controlled activation of the target technology and handover to the associated access network.

A further use case describes a policy decision to activate and use a second radio network interface simultaneously to benefit from the maximum radio bandwidth of each network connection and is depicted in Fig. 1. In this use case, a mobile terminal has multiple IP flows active, which first share the dedicated bandwidth of a single radio connection. An application may report the need for more bandwidth to the policy manager. Same as for the handover use case, the mobile terminal first discovers available network technologies and associated link characteristics. Based on the available bandwidth and costs as well as other policies, such as user preferences and operator policies, the policy manager may take the decision to activate a second radio network interface, for example based on the Wireless LAN (WLAN) technology, and attach it to the network. By means of an evaluation of both radio connections' characteristics, the policy manager takes the decision to distribute the applications' IP flows to different radio connections. Being driven by the policy manager, one or multiple action components may take the responsibility to coordinate flow distribution locally on the mobile terminal and in the network.

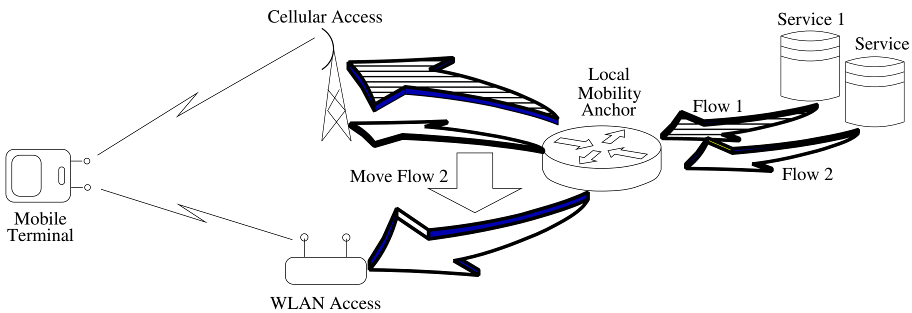

Fig. 1. Multi-homing scenario

A third interesting use case deals with session mobility, automatic detection / classification of devices, and semi-automatic policy generation (see also figure 2). The first part of the scenario begins when the user installs a newly bought DLNA-capable TV with attached camera in the home network to which the MT is also connected. The MT detects the broadcast advertisement messages of this device, and asks the user how to proceed. This process can involve the presentation of a menu from which the user can select which actions to perform under what circumstances (see 3). The specified actions are automatically validated, encoded into policy definitions and added to the policy database. This concludes the first part of the scenario. The second part starts with the user in a video call. While still on the phone, he comes home and switches on the TV set. The MT's policy management detects that the TV set has become available and automatically performs the actions associated with this event, e.g. splits the video stream from the session and moves it to the TV with associated camera.

3 User-Related Aspects

One of the most important goals of interface design is to improve the device's usability. Automatically performing common tasks, reducing the necessary user input to a minimum, e.g. by pre-selecting the user's last choice is one possibility,

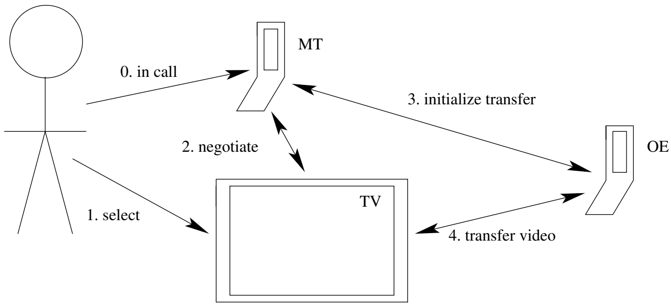

Fig. 2. This figure shows the interaction between the individual devices when the video component of a call is to be transferred to the TV set

providing only the most commonly needed options another. However, current design is often based on assumed common user preferences, not on the preferences of the actual user. By providing a means for the user to set their own policies, everyday use of the device becomes easier and more comfortable.

The average user will not be aware of the policy management concept. Instead the rules are generated according to the user's input on the GUI. Additionally expert users can insert their own rules and edit existing ones. How the policy management can influence the user interface is described in the next section. Afterwards we will discuss possibilities for detecting the user intention.

Menu-Based Selection

Even though the user's effort to interact with the device is limited to the necessary minimum, sometimes extended user interaction is necessary. For example, when a new device is found, the user might want to associate policies such as when to use this device and what actions to perform then. On a mobile phone, this interaction is normally performed via a menu (see Figure 3 on the left), where only the currently appropriate actions are presented. Menu items are normally sorted such that the most commonly selected items are easiest to reach and uncommon items are either not present at all or hidden in a sub-menu. In a simple implementation, the menu can remain static, with only the current selection being remembered for each menu. However, if the user's choice is predictable, the menu may be altered appropriately (see next Section). Depending on the user's choices, policies can be automatically created in response or entered directly by the user. In the latter case, an automatic 'sanity check' of the rules it advisable before adding them to the policy manager.

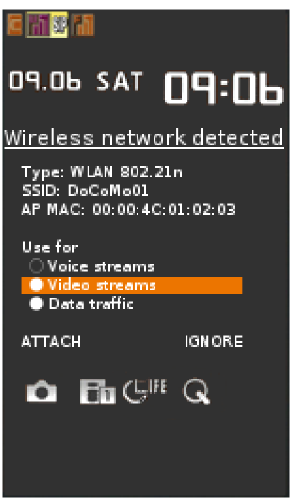

Fig. 3: Menu

Automatic Estimation of User Preferences

The better and more reliably the policy manager is able to predict the intention of the user, the easier it is to use the device. In our implementation, the prediction can be based on similar previous decisions made by the user, predefined rules generated from test user data or operator settings. A better but far more complex approach is to equip the policy management with limited artificial intelligence. This approach is considered for future work.

4 Architecture

The architecture of our approach is build around the policy management component (see Figure 4). The policy management receives events from the event components when there is happening something and it sends triggers to the action components. Those components might be the real source of an event and the real target of a trigger, but they could also receive information from other sources, like in the case of DLNA, where the major components run on remote devices and communicate through the DLNA component on the device.

Additionally components for the user interface, for communication with the network, and for storing rules and other things are necessary.

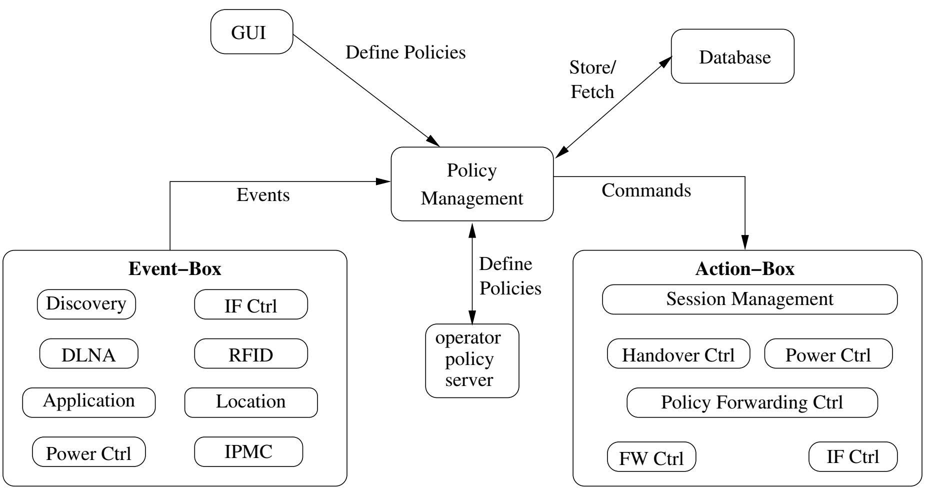

Fig. 4. Component architecture

In the following paragraphs the components are described briefly.

Policy Management: This is the central component which takes the decisions. A detailed description is given in section 5.

GUI: The graphical user interface enables the user to select, define and adapt policies in an easy, yet powerful way. More details can be found in section 3.

Network: The network operator can push policies to the terminal. A new or changed policy can then trigger the change of the access network. This allows an operator e.g. to achieve load balancing across his access networks. It might be considered advisable that the user is able to check and/or override the rules provided by the operator.

Database: The database stores policies, networks, devices, base stations, etc.

Event-Components: Sources of Events towards the Policy Management

The components of this group send events to the policy management. Usually the components communicate with other entities on the same device or in the network, get information from there and then send an event. There are also situations where the policy management tells a component to perform some action, e.g. scanning the network, and then send back an event, e.g. the result of the scan.

Discovery: Discovery of network interfaces, devices, capability as well as network preferences. This component uses DLNA and other technologies to discover devices in the environment. Furthermore, it uses ANDSF ([10], sect. 4.8.2.1) and IEEE 802.21 Information Services to discover available network technologies, handover candidates as well as network preferences.

DLNA: This component provides an interface to home electronics devices which support DLNA, like TV sets. E.g. the event "new DLNA device found" is sent when a corresponding message from a device is received.

Application: Applications might send information about their internal state or the demand for more networking bandwidth. Examples are the phone application and the chat application.

IF Ctrl: Network interface (IF) control to provide indications to the policy management about decreasing network link quality and to report results of a scan for infrastructure access points of a particular or a set of network technologies. This component may utilize the event services of the IEEE 802.21 Media Independent Handover standard.

RFID: The RFID reader/writer is a local device. The corresponding component is a daemon process which reads tags and sends this information as an event.

Location: This component sends location information events. The location can be determined by GPS, WLAN, bluetooth, and RFID.

IPMC: Inter-Policy Management Communication, providing interaction between policy managements located on different device, e.g. PCs.

Power Ctrl: Reports current power state and consumption.

Action-Components: Destination for the Commands from the Policy Management

The components of this group receive triggers from the policy management and then fulfill a task. Usually this can be done without further control from the policy management. However, in some failure situations, it might be useful to inform the policy management with an event so that e.g. a message can be displayed.

Session Management: Controls the handover of sessions or parts of a session to another device. The component gets commands like "split video from the current video session and hand it over to the TV-set".

Handover Ctrl: Enables Handover control and registration for Cellular and IP Mobility Management, This component may have a direct interface to the IF Control component to align IP Mobility with the use of an appropriate network interface.

Policy Forwarding Ctrl: Controls routing tables, forwarding information bases and policy routing engines to perform according to the specified routing policies. Furthermore, this component may represent an abstract interface with a component, which performs protocol operation with the network to negotiate routing policies.

IF Ctrl: Network interface (IF) control to initiate controlled scanning for access points in the network infrastructure, (de-)activation of a network interface or handover to a selected access point. This component may utilize the command services of the IEEE 802.21 Media Independent Handover standard.

Power Ctrl: Set power states to balance performance vs. battery drain.

FW Ctrl: Change the rules of the integrated firewall.

5 Policy Management

The policy management component is the central component of our system. It has interfaces to the Event components to receive events and to the Action components to send triggers. Basically said, the policy management subsequently takes an event from the input queue, checks which policies apply, and sends out triggers.

In order to allow complex policies, we distinguish between policy and policy rules. A rule is a simple statement consisting of a condition and a statement. A policy can then consist of several concatenated rules.

5.1 Policies

With a policy the user or the operator defines what should happen when a certain event arrives. Most policies are set by the user, but the operator might also want to influence the behaviour of the terminal. E.g. the operator might want to shift traffic to a network with less load. In the following we provide a few example policies.

For terminal mobility policies can mandate the usage of WLAN or give it a higher priority than UMTS. Also policies might encourage or forbid the change between access technologies. These policies can be influenced by the location, e.g. home or office. Policies for session mobility define the concrete usage of devices found in the close environment. E.g. a known TV in the home should be used for video when the user comes into the room. Policies like that are also influenced by the circumstances, e.g. if the current call is confidential or if the TV is already used by somebody else. If a new device is found, the policy management checks if there is a rule for this type of device. According to the rule the decision can be to ignore the device or to interrupt the user and ask for a decision on what to do with the device now and in the future.

The policies defined by a network in the area of vertical handover are very similar to the discussed user policies. Additionally the rules could also depend on time/date or position (e.g. defined by cell id). This allows the operator to balance the traffic between access networks.

In contrast we don't foresee policies defined by the network which influence the session handover to another terminal.

Policies that are directly related to low-level functionality can be considered an immutable part of the operating system and are not subject to influence by user or operator.

The definition of policies can be either left directly to an advanced user or, for the average user, they can automatically be generated by the UI based on templates for predefined common events and actions. For example, the user could be presented with a menu where he can associate actions, e.g. 'connect to internet' or 'use my big TV screen for all video calls' with events such as 'coming home' (i.e. home WLAN SSID detected) or 'incoming call'. Concrete design issues of a UI that is suitable for both advanced and average users are not discussed in this paper, but considered for future work.

5.2 Policy Management Layout and Mechanism

The policy management acts on events. When an event is reported by a component, the policy management examines all its stored policies whose event field matches. All rules are processed in the order stored, and the actions of all rules that evaluate as true are executed in sequence. After each rule the state is updated appropriately. Once all matching rules have been executed, the policy management enters the idle state until a new event is received.

Variables and state information are locally stored within the policy management. The management can execute functions provided by other components to retrieve information and to execute actions. This includes informing the user via e.g the GUI and retrieving input. The policy management can additionally access the database which provides persistent storage.

The policy rules themselves are also stored within the database. Rules can be provided and modified by the GUI and the network operator, but pass through the policy management for validation before being stored in the database.

5.3 Policy Definition Language and Examples

All policies processed by the policy management are written in a script-like language as defined by the following rules (EBNF):

```
<rule>    ::= "if (" <event> [<tests>] ") {" <actions> "}";
<tests>   ::= "&&" <test> [<tests>];
<test>    ::= <entity> "==" <entity> | <entity> "!=" <entity> |
              <entity> ">"  <entity> | <entity> "<"  <entity> |
              ["!"] <entity>;
<entity>  ::= <function> | <variable> | <constant>;
<actions> ::= <action> [ ";" [ <actions> ] ];
<action>  ::= <function> | <variable> = <expression> | <rule>;
```

In the following we give a few examples for policies derived from the use cases.

```
# Rule for the multihoming use cases
if (event == appl_ind && appl_ind.req == optimize) { net_ctx = link_opt; disc.net_pref()
      }

# Rule for the case of device recognition
if (event == device_found && type == unknown) { gui.ask_user(device) }

# Rule for the case of reading a tag and decide to transfer a session
if (event == tag_read && tag_id == X1 && dlna.device_available(Y1)) { sessionmgmt.
      transfer(Y1) }
```

6 Protocols

The following flows give an overview of the interactions between the components. The central component is the policy management which adheres to the following design principles: It works as an extended finite state machine which can handle multiple states in parallel. During processing of a use case, the states can change and variables can be set. The processing of one use case can be divided into several chunks. Between the chunks the policy engine might work on other use cases. When a new event comes in, the processing continues at the stored state.

We will detail each of the use cases presented in section 2 by a flow in the following sections.

6.1 Protocol Operation for Device Recognition and Session Mobility

The protocol is split into two parts: In the first the device is detected for the first time and the user is asked for a decision on which actions are to be performed when the device is detected again in the future. In the second part the device is detected again and the video is split from the session and handed over to the TV.

1. After plugging the new TV into the LAN, the policy management receives a DLNA message
2. The database is queried and
3. answers with device unknown
4. Thus the policy management decides to make the device available to the user
5. The device's capabilities are gathered from the device via DLNA

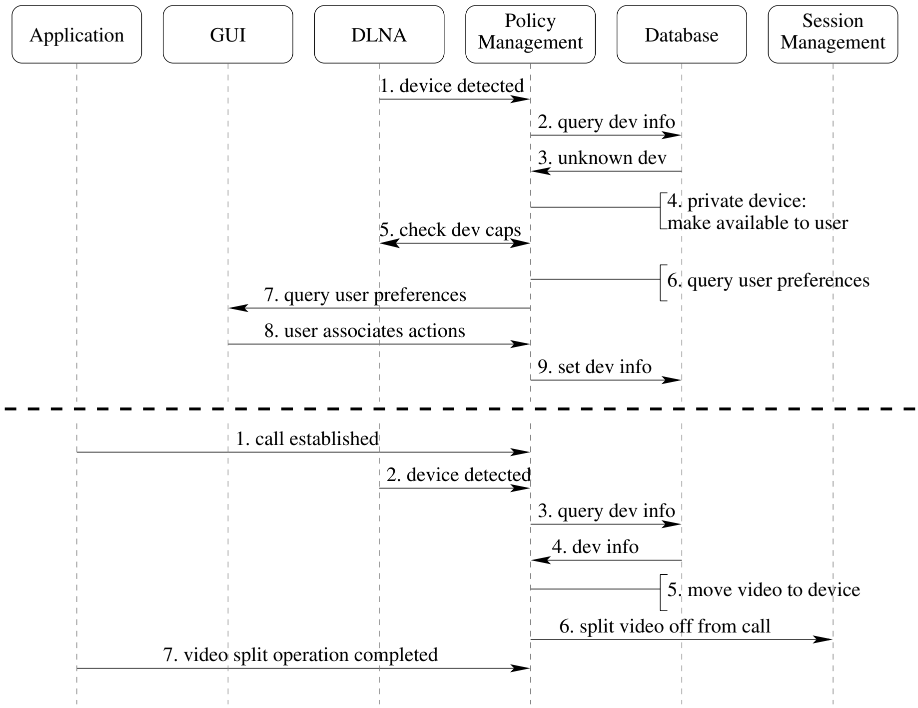

Fig. 5. Device recognition and session mobility

6. The policy management decides to ask the user which actions should be associated with the device
7. and sends the corresponding message to the GUI
8. After retrieving the information, the GUI sends new policy rules
9. The device information is stored in the database, the rules are added to the policy management

Later the user arrives back at the device while in a video call.

1. When the call is established, the phone application notifies the policy management
2. Once the user is close enough to the device to detect it, a notification is sent.
3. The database is queried for information about the device
4. and sends back the stored information
5. The policy management decides to move the video stream to the TV
6. and orders the session management to split the video from the session and move it to the TV
7. The application notifies the policy management about the completed split of the video

6.2 Protocol Operation for Multi-access Flow Distribution

The following protocol operation follows a use case for multi-homing, where multiple radio network interfaces are attached to the network infrastructure to perform flow distribution. Starting point is that multiple network applications are running on the mobile terminal using a single 3G connection. Due to a request from an application to optimize available bandwidth, the policy manager decides to attach a second network interface to the infrastructure and share the bandwidth of both connections.

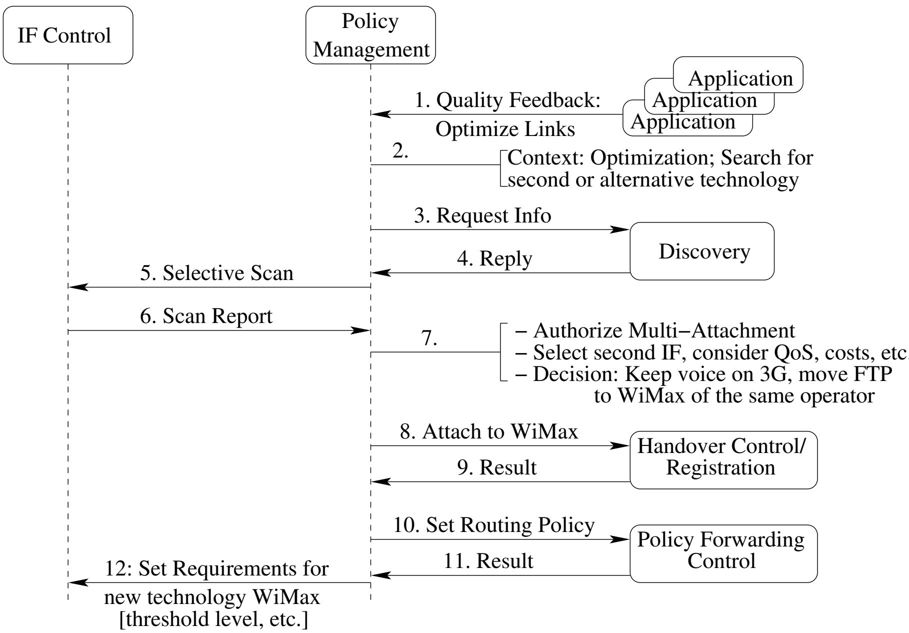

Fig. 6. Simultaneous attachment and flow distribution

1. The policy management receives an indication from at least one active application to optimize bandwidth and link quality
2. The policy management processes the request to optimize links and decides to search for alternative or additional access technologies.
3. The policy management requests the Discovery component for operator preferences regarding technology use
4. The discovery component replies with valid preferences settings.
5. The policy management requests the Interface Control component to perform a scan according to the operator's technology preferences
6. The interface control component reports the scan results to the policy management

7. The policy management selects candidate technology types according to link requirements after having authorized simultaneous use of multiple network interfaces. Due to applications' different link requirements, the policy management decides to keep voice traffic on the currently running 3G interface while the ftp download, which runs in the background, should be moved to a WiMAX interface.
8. The policy management requests the Handover Control/Registration component to attach to the selected WiMAX access network
9. The handover control/registration component reports the result of the attachment to the policy management
10. The policy management requests the policy forwarding control component to set the forwarding rules according to the policy management's decision to use the 3G interface for voice and the WiMAX interface for ftp traffic
11. The policy forwarding control component reports the results of the setting request to the policy management
12. The policy management updates the interface control component with quality threshold levels for the WiMAX technology
13. The policy management is idle

7 Conclusion

This paper describes the use of policy based management for session mobility. As session mobility includes terminal mobility between wireless base stations with different technology, multi-homing, and the handover of sessions or parts of multimedia sessions to other devices, the document starts with the description of use cases in this areas. From the use cases an architecture, a policy language, and protocols are derived.

An important factor of such a scheme is the usability. Therefore the user interface plays an important role. Finally a prototypical implementation shows the feasibility of the scheme and potential problems.

We identified the interworking of user defined policies and policies set by the operator as a potential conflicting scenario.

Future Work

The next steps are an analysis of the mentioned conflict between user and network policies. This is not only a technical conflict, but also a conflict of the interests of the user and the operator. While the user would like to be free in his decision what network and device he uses, the operator wants to control his network e.g. balance the load between access networks.

Another conflict arises when there are multiple devices with an own Policy Management. If the policies defined on those devices might contradict each other, the behaviour might be surprising of the user. E.g. it might occur that a session is moved frequently between devices or a session ends unexpectedly. Even if the devices negotiate policies with each other in order to avoid those cases, it might be hard to explain the user why an action defined on a device does not work as expected.

The current implementation just adds new rules to the end of the list. Although this works fine in our demo setup, it is clear that a smarter sorting is needed which considers interdependencies between the rules. This problem is similar to adding rules to a firewall.

Furthermore we consider adding limited machine learning capabilities to the policy management. This might increase the adaptability of the device to the user's need and therefore make the device easier to use.

References

1. DLNA Digital Living Network Alliance, `http://www.dlna.org/`
2. Sparks, R.: The session initiation protocol (sip) refer method. RFC 3515 (2003)
3. 3GPP. IP Multimedia Subsystem (IMS) service continuity. TS 23.237, 3rd Generation Parnership Project (3GPP) (September 2008)
4. Mihovska, A., Jijun, L., Mino, E., Tragos, E., Mensing, C., Vivier, G., Fracchina, R.: Policy-based mobility management for heterogeneous networks. Mobile and Wireless Communications Summit 2007, 1–6 (July 2007)
5. Wetterwald, M., Filali, F., Bonnet, C., Nussbaum, D., Gauthier, L., Banchs, A., Bernados, C., Liebsch, M., Melia, T., Lafouge, C., Ribeiro, J.: A flexible framework for the support of heterogeneous wireless networks. IST Mobile and Wireless Communications Summit (June 2006)
6. Murray, K., Mathur, R., Pesch, D.: Policy based mobility management in heterogeneous wireless networks for smart space environment. In: 3rd annual Conference on Information Technology and Telecommunications IT&T, Letterkenny, Ireland (October 2003)
7. Murray, K., Mathur, R., Pesch, D.: Intelligent access and mobility management in heterogeneous wireless networks using policy. In: ISICT 2003: Proceedings of the 1st international symposium on Information and communication technologies, Dublin, Ireland, pp. 181–186. Trinity College Dublin (2003)
8. Ohta, K., Yoshikawa, T., Nakagawa, T., Isoda, Y., Kurakake, S.: Adaptive terminal middleware for session mobility. In: International Conference on Distributed Computing Systems Workshops, p. 394 (2003)
9. Rosenberg, J., Schulzrinne, H., Camarillo, G., Peterson, J., Sparks, R., Handley, M., Schooler, E.: SIP: Session Initiation Protocol. RFC 3261 (2002)
10. 3GPP. Architecture enhancements for non-3GPP accesses. TS 23.402, 3rd Generation Parnership Project (3GPP) (September 2008)