An Adaptive Handover Decision Algorithm for Heterogenous Wireless Networks

Mario Pink, Thomas Pietsch, and Hartmut König

Brandenburg University of Technology Cottbus
Department of Computer Science
PF 10 13 44, 03013 Cottbus, Germany
{pink,piestho,koenig}@informatik.tu-cottbus.de

Abstract. The increasingly use of wireless networks and mobile technologies has raised the desire not only to have a good quality access to the network, but also to seamlessly change the network when moving. Various handover algorithmus have been proposed to handle this situation. Unfortunately, many of these algorithms have been only evaluated in simulative environments using simplified models and network assumptions. They do not take the wide range of mobile devices with varying system parameters and capabilities into account which are offered on the market. For the practical deployment, handover algorithms are required which adapt to various device parameters and network characteristics. In this paper we present a fuzzy-based vertical handover decision algorithm which adjusts itself to the given device parameters and network capabilities. Starting with a discussion on the requirements to vertical handover, we present the algorithm and describe how it is activated during the various phases of the handover process. Thereafter we present several experiments which evaluate the accuracy of the handover decision, the quality of service guarantees for the application, and the resource consumption.

1 Introduction

Mobility has become a feature for the network access. Users wish to access the Internet from different networks, such as GSM and UMTS, or WLAN and want to stay connected while changing into another network. This requires appropriate handover procedures to maintain an connection when moving from one network to another. Handover procedures are divided into horizontal and vertical ones. Horizontal or intra-technology handovers are applied for changes between different network cells of the same technology. They are mostly handled by the core network. Vertical or inter-technology handovers are required when changing between networks of different technologies. This handovers has to be performed always by the mobile devices themselves. Therefore, vertical handovers are very complex in detail because various aspects have to be taken into account, such as different network technologies, provider domains, service uses, and the kind of the connection maintenance [8]. Regarding the latter soft and hard handover are distinguished. A soft handover can be applied when the mobile device is connected with two networks simultaneously, so that the connection is moved without

C. Borcea et al. (Eds.): MobilWare 2012, LNICST 65, pp. 136–149, 2013.

interrupt. In a hard handover the connection is shortly interrupted and re-established when moving from the old network to the new one. Currently soft handovers are rarely supported by mobile networks. So it is not possible, for instance, to move between two networks using the same IP address. In order to support a soft handover the handover decision has to be made in time to avoid that the connection is interrupted and the application quality changes. This requires that all relevant parameters for the handover decision have to be evaluated continuously. The problem is that important parameters, such as the device speed, the network parameters, the application quality of service (QoS), or the energy consumption cannot be directly measured on the wide range of mobile devices on the market and are thus difficult to determine. This makes it complicated to estimate the time when the handover should be triggered. The parameter resolutions of the network interfaces further complicate the process. This is the reason why most existing handover algorithms are usually only evaluated in simulation environments based on simplified assumptions on the network infrastructure. It is comprehensible that these algorithms do not work efficiently in practice because they assume a generic structure that does not take the different device capabilities into account. To provide practically handover procedures algorithms are required which adapt to the varying parameter settings of the devices and network interfaces. In this paper we present an adaptive fuzzy-based handover decision algorithm which aims at supporting vertical handovers between real-life networks on off-the-shelf (DOTS) devices. The algorithm uses an adaptive parameter set, which can be reduced depending on the network- and device situation. The remainder of the paper is organized as follows. In Section 2 we give a brief overview on existing handover techniques. Section 3 formulates requirements to efficient handover algorithms. Next in Section 4, we present our adaptive handover decision algorithm. The performance and the resource consumption of the algorithm are evaluated in Section 5. Some final remarks conclude the paper.

2 Vertical Handover Algorithms

The emergence of various wireless technologies in parallel with the increasing use of multi-interface mobile devices has stimulated research on vertical handover. First work on vertical handovers was published about ten years ago [2]. Thereafter various algorithms have been proposed which apply different principles, such as *simple additive weighting* (SAW*), techniques for order preference by similarity to ideal solution* (TOPSIS), *grey relational analysis* (GRA), *analysis hierarchy process* (AHP), and *fuzzy logics* [10], which try to handle large parameter sets [7]. Over the years more and more parameters have been included in the handover decision [1][3][4][5][8][9]. To better handle the various parameters a classification of decision criteria's was proposed [7]. Such processes consume, however, a lot of resources on the mobile devices. Therefore handover decisions should be triggered only when needed to reduce resource consumption. For this, several criteria as, for instance, the round trip time (RTT) were additionally added to the decision procedure. Other approaches, in contrast, tried to reduce the set of decision criteria's. Radio signal strength (RSS)-based methods describe the expected network quality and compare the RSS of the available networks to select the best one [6]. To avoid ping-pong effects thresholds can be used. Unfortunately, the ping-pong problem still may appear if two networks have a RSS close to the threshold. To solve this problem hysteresis approaches were proposed which apply the difference to a reference signal [6]. Other approaches do not only consider the RSS. They include the estimated distance from the access points in the hysteresis derivation by comparing it with a reference cell size of a GSM/UMTS cell. Other approaches try to reduce the number of unnecessary handovers into networks with a small coverage taking the moving speed and the motion direction of the mobile device towards the network access point into account [2][6]. However, most of these approaches are only evaluated by simulations. Detailed descriptions of the algorithms are often missing. Our approach is different. It is based on parameters of off-the-shelf devices and adjusts the parameter set as well as its execution frequency at runtime depending on the resource consumption. Furthermore, it also monitors battery and temperature conditions in the environments to reduce circuit wear-out of equipment and battery aging.

3 Requirements for an Effective Vertical Handover Decision

A closer look on the numerous handover approaches reveals a wide range of different methods applied. Many of these approaches are only applicable on dedicated systems. Up to now there is no general applicable solution for handovers between heterogeneous networks which can be used on a broad range of systems. All approaches have in common that the handover process comprises the same phases. These are:

- *Pre-handover phase*: The mobile device is in a stable state. The mobility management continuously monitors the transmission quality, the application QoS, and the energy, power state of the device. When it detects a significant decrease in the connection quality it initiates the handover process.
- *Network discovery*: The mobile device scans for alternative networks preferred by the user using either physical network interfaces or dedicated web services. If alternative networks are discovered the handover decision can be started.
- *Handover decision*: The mobility management analyses the changes of network and application parameters during the movement through the current network to decide whether a real handover situation has occured. If so it starts the evaluation of the surrounding networks, otherwise it returns to the pre-handover stage.
- *Network evaluation*: The mobility management passively and actively evaluates the performance parameters of the discovered networks. When a network with a better connection quality as the current network is found it is selected.
- *Network selection*: The mobility management tries to set up new connections to the partner(s) via the selected network. If it fails, because it is behind a NAT[1]-router, it has to use an appropriate NAT traversal strategy, e.g., STUN or ICE to set up a new connection.

[1] NAT – Network Address Translation.

In this paper we focus on the handover decision stage. It is the most complex part of the handover process because it has to assess various networks and devices, to make the right decision. The question here is which parameters are really necessary for the decision? Therefore we first discuss in this section the requirements a vertical handover processes has to take into account. They can be grouped into network, user, application, and device requirements.

Network Requirements. The handover decision has to consider the parameters and properties of the discovered networks, such as link performance, handover latency, load balancing, device movement speed, and security policy.

- *Network link performance*: The performance of a wireless connection between a mobile device and an access point is determined by the RSS, the bit error rate (BER), the signal to noise ratio (SNR), and other parameters. For a handover decision, it is usually insufficient to only consider the first three parameters because the link quality is also influenced by network interferences. Therefore the network frequency distance should also be taken into account.
- *Handover latency*: A handover causes a certain delay to perform the necessary configurations, e.g., requesting an IP address from a DHCP server. This handover latency may affect the application quality, so that the delay should be considered.
- *Network load balancing*: Handovers usually cause a change in the application quality because variations in the network technology reduce or increase the traffic transmission capacity. Therefore it is important to identify a stepwise adjustment of the traffic load to the capacity provided by the selected network, e.g., latencies differences, to give the application the possibility to continuously adapt its data rate.
- *Network security policies*: Handovers implicate authentication with the new network to avoid unauthorized access to network resources. Differences in security policies and procedures of wireless products may create significant delays needed for negotiating the security requirements.

User Requirements. Handover decisions may include user preferences which indicate the performance the selected network should meet. User preferences may be determined by application requirements (real-time, non-real-time, background), service types (voice, data, video), network quality, and the cost of the service utilization.

Quality of Service Requirements. The handover decision has also to consider the maximal and the average network throughput, bandwidth limitations, and application latency. For example, an instant messenger may accept a new network with a low data rate and high latency, a VoIP application not.

Device Requirements. Resource consumption (battery lifetime, energy consumption, and thermal effects) is another important factor of the handover decision. It is significantly influenced by the duration and the frequency of the handover.

- *Battery management*: Mobile devices are equipped with batteries which need to be recharged. These recharge cycles exhaust the battery. To avoid continuous re- and discharging processes handovers should be avoided during recharging.
- *Energy management*: In 3.5/3.9G networks methods are needed to improve the energy efficiency because the constrained energy budget of the batteries are highly loaded by the use of different network interfaces. Hence, unused interfaces should be switched off as long as possible..
- *Thermal effects*: The mobile devices lithium-ion batteries usually show their optimal performance between 4 – 20 degree Celcius. Thus ambient heat, e.g., sun light, may reduce the battery capacity and affect the lifetime of integrated circuits. Therefore complex calculations should be avoided outside this range.
- *Device movement speed*: The mobile device have to take the movement speed into account when deciding about a handover. So an handover into a network with a small cell size is not useful when moving with great speed, since another handover will be necessary shortly later. Motion analysis helps to recognize whether the mobile device is moving or not towards the network border.
- *Handover period and decision frequency*: The handover decision estimate the time remaining to complete a handover before the connection is interupted. For it, it is waken up periodically according to the approximated trend of the network and system load. The sleeping period should be as long as possible to minimize the resource consumption.

4 A Fuzzy-Based Handover Decision Algorithm

In order to support a handover decision which can be used for a wide range of mobile devices we propose a fuzzy-based handover decision algorithm. Different methods can be applied for a handover decision algorithm, such as SAW, TOPSIS, GRA, AHP, and fuzzy logic [10]. For, it we analyzed the parameter set of several mobile devices. It showed that the parameter basis is extremely imprecise. This makes it difficult to directly correlate the parameter values. For example, the RSS is represented sometimes in dBm and sometimes in a range of 0-100%. Therefore, we have decided to use the Mamdani fuzzy theory because it allows handling of imprecise parameter sets found in practice and to model non-linear functions with an arbitrary complexity.

4.1 Handover Decision Algorithm

The Mamdani fuzzy system represents value ranges using linguistic terms. In connection with a set of rules, it allows a modeling of handover decisions. At the beginning the fuzzifier maps values onto linguistic terms using membership functions. For this, we apply triangular functions to assign each value to one of the fuzzy sets *low*, *middle*, or *high*. Thereafter these terms are correlated using fuzzy rules. Unfortunately, the rule set explodes in case of large parameter sets. Therefore we first classify the parameters using metrics. Parameters of interest are: RSS, SNR, throughput, RTT, packet loss and BER, network latency, cost of network use, energy consumption, system load, temperature, device speed, motion direction, authentication latency and the amount of surrounding networks. In the *parameter selection phase* these parameters are filtered out if they do not exceed an associated threshold and assigned to one of the four classes: connection quality, quality of service, user preferences, and device state class (see Fig. 1).

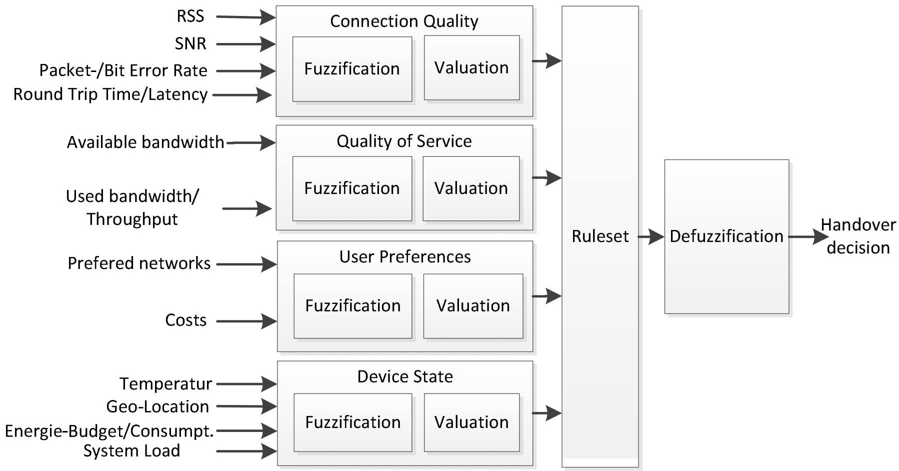

Fig. 1. Parameter classes

In the *parameter processing phase* we normalize, fuzzify these parameters, and multiply them by weights. Further, we derive additional parameters, e.g., the movement speed of the mobile device. In the following *parameter aggregation phase* we add all fuzzy values and create the fuzzy set membership function for the class metric. Table 1 shows an example of the fuzzy values for the connection quality class.

Table 1. Valuation of the parameters RSS, BER, SNR, motion direction, and trend

Level	RSS,BER,SNR	Motion Direction	Movement Speed	Signal Quality	Trend
High	3	approaching	3	good	3
Middle	2	Stationary	2	stable	2
Low	1	Leaving	1	critical	1

Function (1) gives an example for the connection quality vq of a WLAN. We estimate the parameter *trend* for each class metric using linear regression to recognize the remaining time available for network evaluation and selection. Then all four class metrics are correlated also using triangular functions and mapped onto their linguistic terms. In our example vq < 1.66 describes a low, 1.66 ≤ vq ≤ 2.33 a middle, and vq > 2.33 a good connection quality. Similar calculations have to be performed for the other three classes.

$$vq = \frac{w_1 \cdot RSS + w_2 \cdot BER + w_3 \cdot SNR + w_4 \cdot Motion + w_5 \cdot Trend}{\sum wi} \quad (1)$$

Finally we apply a handover decision rule set on these terms to decide about the handover. The rule set specifies if a handover decision should be taken. Table 2 shows an excerpt of such a handover decision rule set for a WLAN.

Table 2. Excerpt of a handover decision rule set

Rule	Signal Quality	QoS	User Acceptance	Device State	HO-Decision
1	high	High	Acceptable	Good	no
2	high	Low	not acceptable	middle	yes
3	low	High	not acceptable	good	yes

The handover decision algorithm is applied independently for each network interface. Fig. 2 summarizes the main steps of the algorithm.

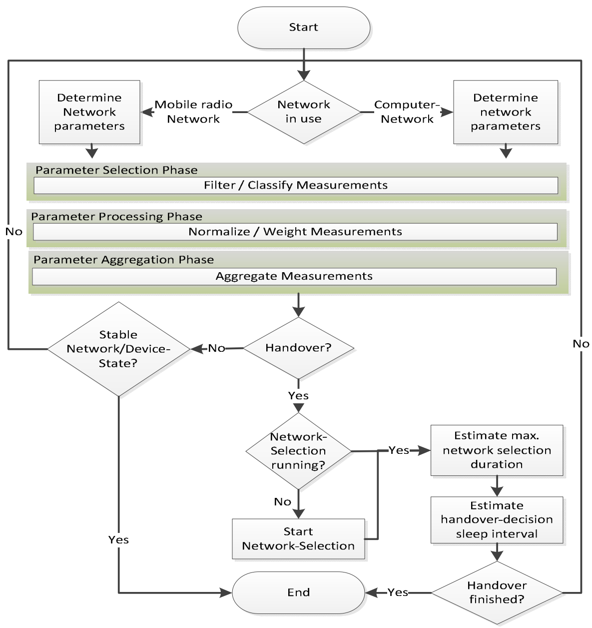

Fig. 2. Adaptive fuzzy-based handover decision algorithm

4.2 Duration of the Handover Decision

The time needed for a handover may not be sufficient in certain network situations [10]. To avoid such critical situations it is very important to determine the time t_{Init} when the algorithm should start. In addition, the maximum handover duration t_{HO} has to be determined to finish the needed handover steps before the connection aborts.

$$t_{Init} = t_{Abr} - t_{HO} \quad \text{with} \quad t_{HO} = t_{HD} - t_{NE} - t_{HS} \tag{2}$$

Hence, the handover duration t_{HO} depends on the time t_{NE} needed for the network evaluation and the time t_{HS} for selecting the network. The moment of connection abort t_{Abr} is estimated using a linear regression on a set of 10 consecutive parameter measurements. To determine it, a device dependent range adaptation of the parameters with respect to the signal characteristics and device configuration is performed. For example, the RSS range in a WLAN may have a lower limit of -90 dBm, -95 dBm, or -100 dBm on different mobile devices. For it, we developed a simple self-adaptation mechanism that adjusts the range of each parameter and threshold, e.g. using a mapping of RTT to specific RSS. Unfortunately, this adaptation takes some time. To avoid expensive configuration periods this adaptation initially starts with the average values of each parameter and threshold which are preconfigured.

4.3 Reactivation Interval

Another important parameter in this context is, as mentioned at the end of Section 4.1, is the interval in which the handover decision should be reactivated. This reactivation interval is determined by the type of the power supply, the full recharging time of the battery, and the average of the active and passive refreshing frequencies of the operating system, of the network interfaces, and of the device parameters. The active refreshing frequency considers consecutive parameter measurements, whereas the passive refresh frequency considers the refresh interval of parameters, e.g., incoming beacon in WLANs. If the device is powered without a battery the algorithm is stopped. Otherwise, at the beginning we use the RSS as the most important parameter for a handover decision and determine its active and passive refreshing frequency for every network interface. We take the lowest interval as the reference interval I_{ref}. Then we determine the refreshing interval P_{ref} of each parameter P and put it in relation to I_{ref}. Using these intervals we can poll the parameter values only when they are refreshed. Thus, the reactivation interval of the algorithm for the different network interfaces is determined using the average refreshing interval of the parameters selected for the handover decision and the last reactivation intervals. If the parameter values decreases unexpectedly during the last reactivation the next interval is adjusted according to the parameter value fluctuations during this interval. Otherwise, if no critical changes are observed the reactivation interval is increased up to a threshold $TH_{reactivate}$ to reduce power consumption. $TH_{reactivate}$ is estimated using a linear regression of the ongoing reactivation intervals.

4.4 Adaptive Handover Decision

To take the limited resources of mobile devices into account and to react on critical environmental influences on the mobile device, e.g., the battery temperature, it is necessary to apply a handover decision algorithm with a dynamic parameter selection. Our algorithm allows the handover decision to gradually adapt itself and to activate or deactivate parameters to reduce its calculation complexity. For it, the handover decision algorithm monitors itself and either adjusts its reactivation interval as described above or the parameter set used. To adjust the parameter set three mobile device states are distinguished: *idle*, *standard*, and *high load*. A mobile device is in *idle* state when the system load is below a threshold TH_{low} for several minutes, i.e., the CPU frequency is low and the network interfaces uses low power levels with a low transmission rates. It is in state *standard* when the system load and the network interface power level, as well as the transmission rate are between the thresholds TH_{low} and TH_{middle} for several minutes. The *high load* state is reached when the system load increases above TH_{middle}, the CPU works with the maximal frequency and the network interfaces use high power levels with high transmission rates. Depending on the state of the mobile device and its battery, various priorities are assigned to the parameters. In the state *high load* only parameters with the priority *high* can be applied, if the temperature does not exceed $TH_{Max-Temp}$ (40°C) or falls below $TH_{Min-Temp}$ (-5°C); In the state *standard* accordingly parameters with priorities *high* and *middle*, if the temperature does not exceed the temperature $TH_{High-Temp}$. All parameters can be used if the temperature does not exceed the temperature $TH_{Low-Temp}$ (25°C) in the state *idle*. When the temperature exceeds $TH_{Max-Temp}$ or falls below $TH_{Min-Temp}$ the algorithm stops. A reduced parameter set, however, decreases the handover decision accuracy. Therefore the parameter set need to be structured in a term-oriented function structure, as for example function (1), that always a correct handover decision can be made.

Table 3. Parameter priorities for WLAN and GSM/UMTS networks

Parameter	**WLAN Priority**	**GSM/UMTS Priority**	**Substitution Parameter**
RSS	high	high	RTT
BER, SNR	low	low	RTT
RTT	low	high	RSS
RSS Trend	middle	middle	RTT
Motion- Direction, Speed	middle	low	RTT, RSS
Available/Used Bandwidth	high/low	high/low	Latency
SSId , BSSId, Location Area Code, Cell-ID	high	high	-
Cost	middle	high	-
Energy Consumption/Temperatur	middle	middle	-

Table 3 gives an example for a priority assignment for WLAN and GSM/UMTS parameters. It shows that the RSS can never be ignored in any handover decision.

4.5 Substitution Parameters

Beside the adaptation of the parameter set, we also adapt the accuracy of the parameters of the handover decision using a parameter substitution approach (see Table 3). For this, we model the characteristics of a parameter using substitution parameters to increase its accuracy in time and resolution. For example, if the refreshing interval of a parameter *A* is too long we apply a substitution parameter *B* between two consecutives measurements of *A*. Thus parameter B is used until parameter *A* changes its value. Furthermore, if the parameter *A* is not available parameter *B* can be used as alternative for *A*. This improvement can be applied always or when the value of *A* falls below or exceeds a threshold. Substitution parameters are needed because several parameters cannot be determined for various mobile devices, especially for low price devices. Therefore substitution parameters can never be switched off. Even if these devices provide the parameters their refreshing interval is often too long or the parameter values provided are only average ones. For example, the *Samsung Omnia* b7610 does not supply the UMTS RSS. The *Huawei* E160 UMTS network interface is another example. It has a refreshing interval for the RSS of nearly 5 seconds. In this case the RTT has to be taken as substitution parameter to approximate the RSS.

5 Experimental Evaluation

In order to evaluate our handover decision algorithm with respect to performance and resource consumption we run three series of experiments. We used four mobile devices, a *N900* with Meego, a *Samsung* i8910 and a *Nokia* e90, both with Symbian S60, as well as a *Samsung Omnia* b7610 with Windows Mobile 6.5. The objective of the first experiment series was to prove whether our fuzzy-based decision algorithm is capable to successfully determine when to perform a handover, i.e., to investigate its ability to prevent unnecessary connection interrupts and latencies. Next we examined whether the algorithm is able to bring the available network resources in line with the quality of service demands of the application. In this case it must be also verified whether unnecessary handover decisions are avoided. Finally the resource consumption of the algorithm was measured.

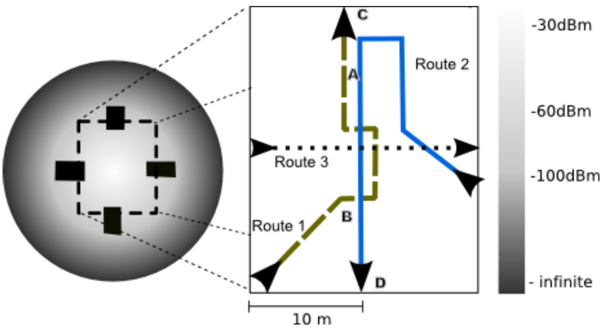

Fig. 3. Distribution of the radio signal strength of a 802.11b/g WLAN in a sub-urban area

5.1 Accuracy of Handover Decisions

To evaluate the accuracy of handover decisions we moved along different paths through a 802.11b/g WLAN in a UMTS (HSPA) covered sub-urban area with several buildings, as shown in Fig. **3**. The signal map at the left-hand side shows the distribution of the radio signal strengths among the building. The right-hand map depicts areas with good connectivity in white and worse connectivity in grey with the routes passed in the experiments.

In our experiments we moved using the mobile devices with a speed of approximately 5 km/h through the area (see Fig. 3). We passed each way ten times. At the points A and B handover decisions were initiated ($t_{Handover}$), while at the points C and D the connection was definitely interrupted ($t_{Disconnection}$).

- **Experiment 1:** In the first experiment we moved on path 1 (green line) through the WLAN network and left at point *C*. Every test run showed a critical network state with a positive handover decision to the UMTS network at point *A*. The time until connection interrupt was determined here between 4 und 7 seconds. At point *C* the connection with the WLAN access point was aborted. The handover decision worked optimally in this case because the RSS decreased continuously.
- **Experiment 2:** In the second experiment we moved on path 2 (solid line) and left the WLAN at point *D*. As in experiment 1 the RSS became critical at point *A*. The time till connection abort was estimated between 4 and 8 seconds, but immediately after that the RSS increased in the direction to *B* and the algorithm refused to handover. At point *B* the RSS falls down and the algorithm decided to change to the UMTS network. The estimated remaining time laid between 3 and 6 seconds.
- **Experiment 3:** In this experiment (solid dotted line) we moved along path 3 inside the WLAN without leaving it. Here the algorithm never indicated an handover.

5.2 Quality of Service Evaluation

Evaluations of handover decision algorithms mostly analyze the ability to avoid connection interrupts as a result of falling RSS. In addition, we evaluated how the decision algorithm guarantees the QoS required by the application. For this, we perform handovers at application level using two adapted SOCKSv5 proxies, one on the mobile device (*Samsung Omnia* i8910) and one on a PC with Gigabit Ethernet. These two proxy instances hide the IP address change from the application and communicate over TCP or UDP. In the first experiment we analyzed the quality of a video stream over a UDP connection using a customer 802.11g WLAN/16 MBit DSL (see Fig. 4) and a UMTS network of O2-Germany (see Fig. 5). The handover decision (HO) algorithm compared the QoS requirements of the video stream with the QoS capabilities of the network over a certain period of time ($t_{Handover}$). Due to the increasing data rate, it decided to handover from UMTS to WLAN after 170 seconds and during movement after the next 160 seconds from WLAN to UMTS.

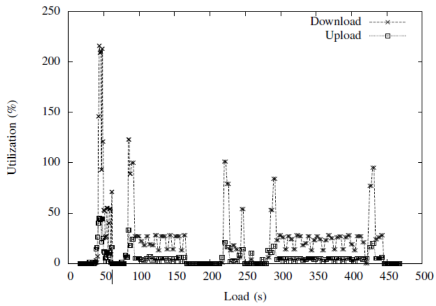

Fig. 4. *VideoStream*: HO-decision for UMTS

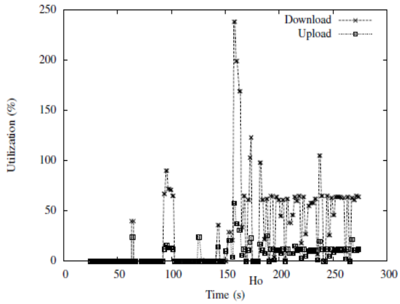

Fig. 5. *VideoStream*: HO-decision for WLAN

The second experiment analyzed the network load when requesting web pages using a relayed TCP connection. Fig. 6 and Fig. **7** show the network load for WLAN and UMTS, respectively. Here the algorithm decided against a handover because the QoS requirements increased only for a short period of time, between 10 and 20 seconds.

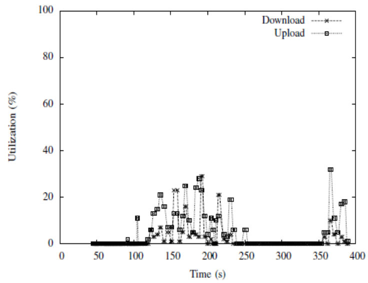

Fig. 6. *Web request*: no handover (WLAN)

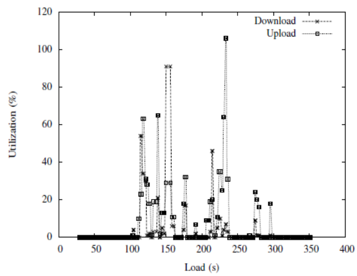

Fig. 7. *Web request*: no handover (UMTS)

Finally we analyzed the behavior when transmitting a large file. Fig. 8 shows the relayed TCP transmission with a maximum throughput starting in a UMTS network.

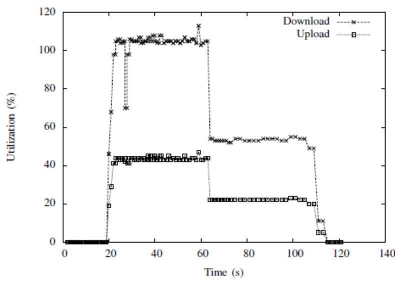

Fig. 8. *File download*: HO-decision to WLAN

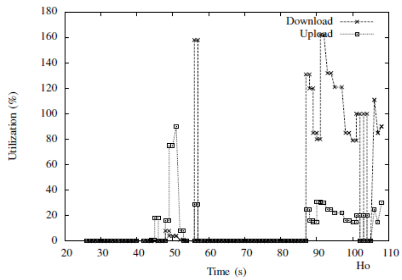

Fig. 9. *File download*: HO-decision to UMTS

Here the algorithm decided after 60 seconds to trigger a handover into WLAN. Fig. 9 shows an opposite situation. A file download starts in the WLAN and is switched to UMTS after 90 seconds because of the better QoS capabilities of UMTS.

5.3 Algorithm Resource Consumption

The resource consumption is important for the algorithm in practice. To estimate the resource consumption we executed the algorithm 1000 times applying three different gradual levels and measured the CPU time: *high* level for high priority parameters and high accuracy, *middle* for average parameters, and *low* for low priority parameters and accuracy, (low level comprises only RSS, SNR, BER).

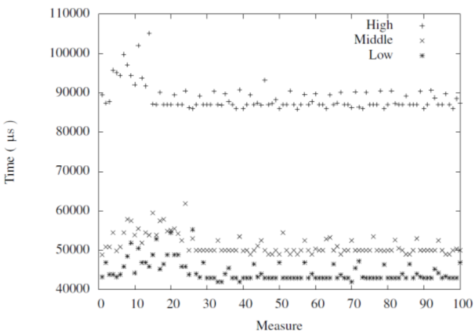

Fig. 10. Gradual levels CPU performance

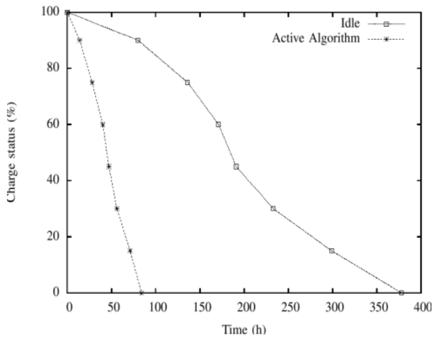

Fig. 11. Energy consumption

We repeated the measurement (see Fig. 10) for every level on the *Samsung* i8910. It showed the algorithm needed 89-94ms for handover decisions at high level. At middle level 48-50ms for decisions and 43-44ms at low level, respectively. It showed that the calculation efforts for high accuracy are two times higher than that for low accuracy. The difference between middle and low accuracy is about 15%. Next we analyzed the energy consumption of the algorithm. Unfortunately, it is not always possible to determine the battery current and voltage on every mobile device. The *Samsung* i8910 device distinguishes only 8 levels for the battery capacity: 100, 90, 75, 60, 45, 30, 15, and 0%. Therefore, we use an energy model to determine the energy consumption of the algorithm using the number of algorithm runs per battery level. At first, we determine the energy capacity, runtime of each battery level using the idle energy consumption with, without network interface during the whole battery runtime. At second we consider the runtime difference for each battery level when the CPU continuously executes the algorithm and create a energy consumption metric. Figure 11 show the energy consumption with a maximum accuracy in a 3 second interval.

6 Conclusions

In this paper we presented a fuzzy-based handover decision algorithm for vertical handovers on off-the-shelf devices. The algorithm uses an adaptive parameter set, which can be reduced depending on the network and system situation. It estimates the remaining time to evaluate the discovered networks, to select the best of them, and to handover the connection to this network. We evaluated the applicability of the algorithm in various real-life experiments on mobile devices. Nevertheless, the parameter weights, the reactivation of the algorithm and the estimation of connection abort need to be further improved to forecast an upcoming handover. As next step, we improve our algorithm using Q-learning.

References

1. Marquez, J., Calafate, C.T., Cano, J.C., Manzoni, P.: An overview of vertical handover techniques: algorithms, protocols and tools. Computer Communications 34, 985–997 (2011)
2. Theodore, S.S., Antonis, S.M., Miltiadis, E.A., Michalis, E.T.: Vehicle Velocity Estimation Based on RSS Measurements. Wireless Personal Communications 40(4), 523–538 (2006)
3. Nasser, N., Hasswa, A., Hassanein, H.: Handoffs in Fourth Generation Heterogeneous Networks. IEEE Communications Magazine 10(44), 96–103 (2006)
4. Kassar, M., Kervella, B., Pujolle, G.: An overview of vertical handover decision strategies in heterogeneous wireless networks. Computer Communications 31(10), 2607–2620 (2008)
5. Nasif, E., Tara, S., Sibel, K., Kemal, F.: An Overview of Handoff Techniques in Cellular Networks. International Journal of Information Technology 2(2), 1305–2403 (2005)
6. Lee, H., Kim, D., Chung, B., Yoon, H.: Adaptive Hysteresis Using Mobility Correlation for Fast Handover. Communication Letters IEEE 12, 152–154 (2008)
7. Morales, J.D.M., et al.: Performance comparison between MADM algorithms for vertical handoff in 4G networks. In: Proceedings of the 7th Conference on Engineering, Computing Science and Automatic Control, Tuxtla Gutierrez, Mexico, pp. 309–314 (2010)
8. Yan, X., et al.: A Survey of Vertical Handover Decision Algorithms in Fourth Generation Heterogeneous Wireless Networks. Journal Computer Networks 54, 1848–1863 (2010)
9. Navarro, E.S., Wong, V.W.S.: Comparison between Vertical Handoff Decision Algorithms for Heterogeneous Wireless Networks. In: Proceedings of the Vehicular Technology Conference, Melbourne, Australia, pp. 947–951 (2006)
10. Sharma, M., Khla, R.K.: Fuzzy Logic Based Handover Decision System. Journal of Adhoc, Sensor & Ubiquitous Computing 3(4), 21–29 (2012)