Turkish Journal of Electrical Engineering and Computer Sciences

Volume 20 | Number 6 Article 7

1-1-2012

Context gathering and management for centralized context-aware handover in heterogeneous mobile networks

BEHROUZ SHAHGHOLI GHAHFAROKHI

NASER MOVAHHEDINIA

Turk J Elec Eng & Comp Sci, Vol.20, No.6, 2012, © TÜBİTAK
doi:10.3906/elk-1101-1042

Context gathering and management for centralized context-aware handover in heterogeneous mobile networks

Behrouz Shahgholi GHAHFAROKHI*, **Naser MOVAHEDINIA**
Department of Computer Engineering, University of Isfahan, Isfahan-IRAN
e-mails: shahgholi@eng.ui.ac.ir, naserm@eng.ui.ac.ir

Received: 23.01.2011

Abstract

Context-aware handover decision has recently been considered as a candidate for next-generation heterogeneous wireless networks. The context-aware handover methods proposed in the literature differ in some aspects, including the location of the handover decision (distributed or centralized). Depending on the location of the decision point, the appropriate part of the context knowledge should be transferred in those methods. This paper proposes a context gathering mechanism for a policy-based context-aware handover method, which implements mobile-initiated and network-assisted handover. The proposed network context gathering mechanism is based on a media-independent handover (MIH) framework and the paper justifies the usability of the extension using some analysis on signaling overhead and latency. Another part of the context is the preferences of the users, applications, and network operators, where this paper has proposed an automatic policy construction procedure to gather and employ them in generating policies. This procedure eliminates the complexity of making policies in previous policy-based context-aware methods and allows employing up-to-date network context information to dynamically modify the policies. Simulation results show better performance in terms of perceived quality for sensitive traffic.

Key Words: *Context-aware handover, context gathering, media independent handover, policy-based handover*

1. Introduction

Recent developments in heterogeneous mobile networks have led to the development of a variety of context-aware handover mechanisms during the past several years. In fact, conventional handover decision parameters, such as received signal strength (RSS) and similar link parameters, are not sufficient to acclimatize to the handover decision due to a variety of access networks with different characteristics. Therefore, upcoming handover methods should consider a wider knowledge of underlying networks. The context-aware handover is a handoff procedure that selects a target point of attachment (PoA) based not only on received signal quality, but also

*Corresponding author: Department of Computer Engineering, University of Isfahan, Isfahan-IRAN

on the knowledge of mobile nodes (MNs), access networks, users, and applications to make an intelligent and optimized handover decision [1].

Context gathering and management is a major part of context-aware handover methods. Depending on the location of handover decision (MN or access network), some of the context information is needed to be transferred. For the mobile-controlled handover (MCHO), where the decision is performed in MNs, the network context should be gathered by the MNs, while for centralized handover decisions, it is the mobile side context (preferences and requirements) that is required to be transferred to the decision point, in addition to the network context. Between that context knowledge, the on-time gathering of the access networks' information is too important due to its frequently changing nature. While some earlier studies have not stated network context gathering, some others have proposed complicated methods for this aim.

Recently, the media-independent handover (MIH) framework [2] has been proposed to provide a common interface for managing handovers and exchanging messages between different handover modules. Although MIH provides an information service to handover decision modules, the information provided by this service is mainly static. Therefore, some efforts have been carried out in the literature to extend the MIH for context gathering. However, those methods are not suitable for the centralized handover decision. Moreover, they are complex or impose huge signaling overhead on the network.

In our previous study [3], we proposed a mobile-initiated and network-assisted context-aware handover mechanism that is based on policy-based architecture. Although a network context monitor was proposed in the handover management model of [3], that model employs static network context and did not state any method for gathering and updating dynamic network context information. Choosing the centralized handover decision model of [3], 2 MIH-based mechanisms have been presented for network context gathering: the event-based handoff-aware method and the query-based method. The paper illustrates the favorability of the proposed extensions using some analytical assessments and compares them in terms of signaling overhead and context transfer latency. Those methods are also applicable to other centralized context-aware handover decisions. The previous study also suffers from the complexity of preparing policies that are made using the direct hand of the users. By providing dynamic network context accessibility through the event-based method, the other contribution of this paper is the exposition of an automatic procedure for [3] to generate the policies and renew them according to network context changes. The automatic procedure is based on the analytic hierarchy process (AHP) method and provides a reasonable method for the generation and renewal of policies.

The next section presents a brief review of related works, Section 3 describes the centralized handover decision model of [3], Section 4 includes the proposed MIH-based network context gathering extensions, Section 5 demonstrates the proposed automatic policy construction method, Section 6 shows some simulation results and numerical examples, and, finally, the paper is concluded in Section 7.

2. Related works

In this section, we briefly consider the context-aware handover methods in the literature. Afterwards, the context gathering problem in earlier studies is investigated. Finally, Section 2.3 introduces the MIH framework and investigates the employment of the MIH for network context access.

2.1. Context-aware handover decisions

Many efforts have been carried out for the context-aware handover decision. Multiple attribute decision making (MADM) methods are widely used in context-aware handover decision methods, such as in [4-7], to rank the candidate PoAs and select the target PoA according to the preferences and requirements from one side and the context parameters of candidate PoAs from the other side. For example, Balasubramaniam et al. [7] proposed an AHP-based method for target PoA selection that only employs the personal settings (as preferences) for calculation of the objective pairwise comparison matrix. Ahmed et al. [5] proposed the architecture of a context-aware mobile-initiated and controlled vertical handover decision model for multihomed mobile devices in heterogeneous networks. The proposed PoA evaluation and ranking is based on the AHP algorithm. That paper did not clearly describe the context collection method either. Similarly, the authors of [8] proposed a combined decision method that uses fuzzy logic to decide about the handover initiation and the AHP to decide about the target PoA.

Policy-based networking architecture is another technique that has been regarded to make the best handover decision based on the policies and context information of PoAs. The basis of [9] is a policy-based architecture, and a simple cost function was proposed for the evaluation and ranking of target networks according to the selected policy. Reference [10] presents a context-aware policy mechanism for adaptive connectivity management of multiaccess wireless networks. The handover decision is made on the basis of the end-to-end evaluation to select the best candidates for a connection. In our previous study [3], we proposed a mobile-initiated and network-assisted context-aware handover mechanism that is based on policy-based architecture. That proposed method presents a new policy format that exploits the fuzzy petri-net (FPN) technique for the representation of context knowledge and for the ranking of target PoAs. A FPN-based evaluation of candidate PoAs overcomes the uncertainty in context parameters, which is due to the nondeterministic nature of context parameters in rapidly varying wireless networks. In addition, the FPN-based evaluation allows the policy decision point (PDP) to employ multiple policies coincidently for assessment of candidate PoAs. The handover mechanism of [3] does not exploit the dynamic network context. Fixed policies have been constructed from the static network context and nominal value of the network quality of service (QoS) parameters. A detailed introduction of the handover decision model of [3] is presented in Section 3.

2.2. Context gathering and management

Some of the studies in the literature have considered the context gathering and management problem in their context-aware handover model. In [11], a context server was assumed in the network backbone that collects the network information from context repositories distributed in different access networks. That context information is then provided to MNs to decide about target PoAs. However, details of gathering the context from access networks were not considered in that study. In [1], a general framework was offered for the handover decision that takes the context of both the mobile network and the user into account. In the architecture, handover decision points are responsible for deciding about the vertical handover, while context collection points collect, compile, and deliver the relevant context information to the handover decision points. MNs can also download a software agent that encapsulates the compiled context information and the handoff decision algorithm. However, this method results in more complexity for MNs. In [12], the authors proposed an integrated approach for context management that assumes programmable platforms and distributed context management components in network nodes and mobile devices. That flexible architecture is able to actively deploy different handover algorithms as context changes. However, this method increases the overhead of signaling between MNs and access networks, and also increases the complexity of the MNs. The authors of [13] designed an application profile and a working profile to collect the application requirements and user intentions for the MN's handover decision. A similar method in [14] considered the user and application-related context parameters in simple rule-based decision making. However, network context gathering was not regarded in [13] or [14].

2.3. Overview of MIH and MIH-based context gathering methods

The IEEE 802.21 working group prepared a MIH framework so that upper layers can abstract the heterogeneity aspects of different technologies and interact with them via a unified interface [15]. The MIH framework defines 3 main services through MIH_SAP for MIH users. These services are the MIH event service (MIES), MIH command service (MICS), and MIH information service (MIIS), as shown in Figure 1. The 3 main MIH services were introduced as below:

- The MIES indicates the changes in the state and transmission behavior of the physical, data link, and logical link layers, or predicts the state changes of these layers and provides support for both local and remote notifications [2]. The events are local if issued by the link layer of the node, and are remote if issued by another MIH function (MIHF) entity in another node (sending MIES messages).
- The MICS enables higher layers to control the physical, data link, and logical link layers [2]. It is utilized for gathering information about the status of connected links, and also to execute mobility and connectivity decisions in layer 2. The commands can be both local, if issued by an upper layer entity, or remote, if emitted by another entity (MIHF) in the access network [16]. Some of the commands have specifically been defined for mobilecontrolled or networkcontrolled handover. For example, the MIH_MN_HO_Candidate_Query, which is sent from the MN to the access network, is the MICS primitive that is used in the MCHO for querying about the availability of resources in candidate PoAs. Similarly, MIH_Net_HO_Candidate_Query is the MICS primitive, which is sent from the access network to the MN to request QoS constraints of the MN in network-side handover decisions. Another type of MIH commands are the ones transmitted from one entity in the access network to another entity in the access network. An example of this type of command is MIH_N2N_HO_Query_Resources. This command is sent from a MIHF entity in the serving access network to a MIHF entity in the target network to request the preparation of resources.
- The MIIS provides a framework and corresponding mechanisms by which a MIHF entity can discover and obtain network information existing within a geographical area to facilitate the handover [2]. The role of MIIS is to provide information about the available networks, operators, and PoAs via information element (IE) structures. IEs are obtained by requests from the information server (IS), using the MIH_Get_Information primitive. The information provided by this service is intended to be mainly static, primarily used by policy engines that do not require dynamic and updated information [15].

Employing the MIH framework for context access was considered in a few previous studies. The proposed method of [17] utilized the IEEE 802.21 framework to improve the session initiation protocol (SIP) handoff performance. In that study, the MIIS is used to discover the presence of new networks and to obtain information about the available networks, such as neighbor graphs and relevant context information. However, the authors did not describe how to possibly update the dynamic part of the network context in the IS. A SIP-based mobility management system was proposed in [18], which exploits the MIIS to achieve static context information of the neighboring PoAs. The available bandwidth is an important dynamic context parameter that was considered in that paper. This parameter is updated at fixed intervals by an agent that monitors each PoA through the use of the simple network management protocol (SNMP). Short intervals impose heavy signaling overhead and long intervals have to afford inaccuracy.

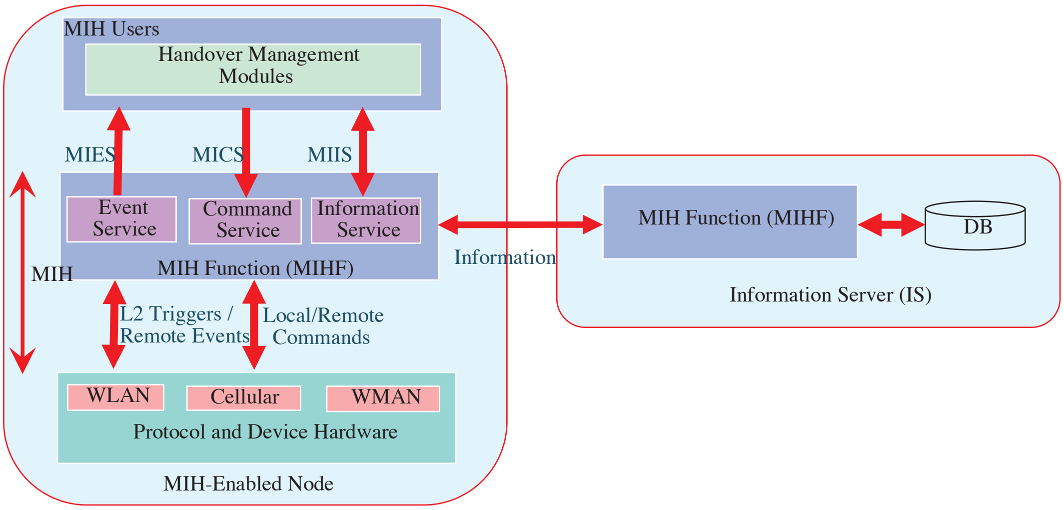

Figure 1. MIH framework.

An enhanced MIH (EMIH) framework was introduced in [19] to gather more context information. The EMIH communication model allows the handover decision to be performed in MNs or the access network. The model considers a context-aware server in the network side to dynamically identify network context and generate triggers to inform context changes to handover control modules (HCMs) for the handover decision. Similarly, a context-aware module was regarded in MNs to identify the MN's context and inform the HCM of their changes. The HCM was considered in both the MNs and access networks to provide the support for the mobile-controlled handover and network-controlled handover. Communication between the entities is provided through the EMIH function (EMIHF), while in [19] the details of necessary changes in the MIH protocol and primitives were not stated. This method is complex and seems too different compared to the standard MIH framework.

The authors of [20] extended the MIH_MN_HO_Candidate_Query and MIH_N2N_HO_Query_Resources MICS primitives defined by IEEE 802.21 to ask the neighboring PoAs about dynamic information, such as available bandwidth. In that method, the MNs ask the IS about neighboring PoAs and their static context, and then ask through the currently serving PoA for the dynamic context of neighboring PoAs. Asking from the IS is performed after each handover accomplishment (using the MIH_Get_Information primitive) and asking from PoAs is performed when a handover decision is underway (using the extended MIH_MN_HO_Candidate_Query and MIH_N2N_HO_Query_Resources). That method was proposed for mobile-controlled handovers.

In Section 4, we propose 2 novel MIH-based context gathering methods to provide the accessibility of the centralized handover decision of [3] to the dynamic context of PoAs in addition to a static one.

3. Architecture of the centralized policy-based handover decision

In this section, a brief review of the centralized policy-based handover decision method of [3] is presented. As Figure 2 shows, the policy enforcement point (PEP) is a MN, while the PDP is a node in the backbone of the underlying heterogeneous wireless network. In the PDP, the preferences of the users, applications, and operators are employed in combination with the access networks' context to prepare the policies for the policy decision maker (PDM). However, [3] did not specify the method of gathering the network context and making the policies. Instead, the network operators, users, and administrators are responsible for providing the policies from their knowledge of network context and preferences.

The following policy structure has been defined for policy-based decision making:

$$P = (FC, < P, W >^{*}, < PN, PW >^{*}, Thr) \quad (1)$$

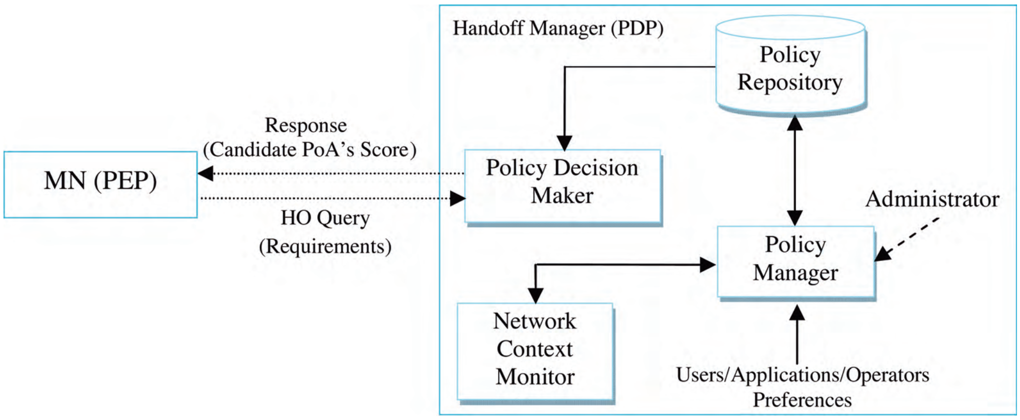

Figure 2. The policy-based context-aware handover mechanism proposed in [3].

In Eq. (1), *FC* is a set of filtering conditions for the policies regarding the properties of the handover request, such as the class of traffic that is going to be handed over, user ID, currently serving PoA, and even logical expressions on other context parameters. $< P, W >^{*}$ represents the user/application preferences as a set of parameters and their level of importance. $<$PN, PW$>^{*}$ is a set of candidate PoAs and their weights, and *Thr* is a firing threshold, which determines the importance of the policy compared to other selected policies contributing in the FPN-based reasoning method. Each policy structure is modeled as a FPN, as shown in Figure 3.

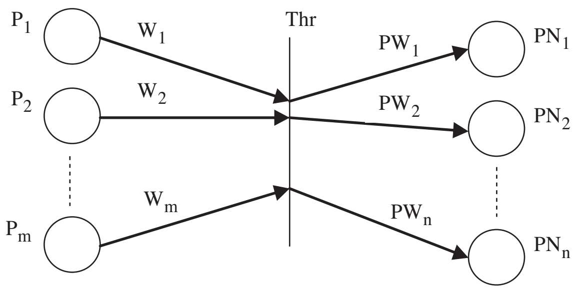

Figure 3. FPN structure constructed from a policy.

The PDM employs an approximate fuzzy reasoning algorithm on the FPN structure (which is constructed from all of the matching policies) to evaluate the candidate PoAs with respect to the requirements of the handover request as FPN inputs. Finally, the MN (PEP) utilizes these evaluation results in its local decision to select the best PoA for the handover execution.

4. The proposed MIH-based network context gathering method

In this section, we will explain 2 MIH-based context gathering methods for the network context monitor module and compare them to each other. These methods are also suitable for other centralized context-aware handovers.

Provided by the MIIS, the network context monitor is planned to obtain the static information about the PoAs of its domain and also the neighboring graph of PoAs (containing neighboring PoAs of each PoA). In addition, we propose extensions on the MIH framework to allow it to be used by the network context monitor to attain the dynamic QoS parameters of the PoAs. These extensions enable the network context monitor to prepare dynamic information, including the current available bandwidth, number of mobile users, mean frame delay, jitter, and loss of PoAs.

Two possible methods are imagined for the network context monitor to obtain the dynamic QoS parameters of the PoAs:

1. In the first method, which is called the query-based method, the network context monitor queries for the dynamic context of some of the PoAs whenever a handover request arrives at the PDM, implying that the latest context of those PoAs is necessary. The MIH_N2N_HO_Query_Resources primitive is an appropriate candidate for the implementation of this method, modifying it to include the required QoS parameters in the request packet and the values of those QoS parameters in the response packet (similar to the modifications that were proposed in [20] for query-based dynamic context access)

2. PoAs report their context changes to the network context monitor after the execution of a handover or the arrival of a new connection. Since the dynamic QoS parameters of the access networks commonly change after the execution of handovers (major changes), such an event-based context renewal is reasonable and seems to be more optimal than the above query-based method. We will investigate and compare the signaling overhead and context access latency of this method to the first method at the end of this section. To be capable of updating such information, a new MIES primitive, namely Information_Update, has been proposed. Via the proposed event, the PoAs inform the network context monitor about the latest allocated bandwidth or released bandwidth as well as modifications in other dynamic parameters when a handover procedure is completed or a new connection is established. The Information_Update event occurs in the network context monitor when a PoA sends the Information_Update indication message, and upon the occurrence of this event, the network context monitor updates the relevant context parameters in its database. The Information_Update indication message has a unique action ID from the reserved ones (AID = 8) in its header, as shown in Table 1. It includes a link identifier TLV (to indicate the PoA) and the list of modified context parameters. New TLVs have been assumed to describe each of the modified context parameters in the dynamic context list. Two of the major TLVs are allocated bandwidth TLV and released bandwidth TLV, to describe new bandwidth allocations and releases due to a recent handover. The network context monitor should register for this event in all of the PoAs of its domain, and those PoAs send their initial context after registration.

At the end of this section, we represent some analytical resolutions to compare the methods. Our analyses are performed in terms of signaling overhead and context access latency. Comparing the signaling overhead, Eq. (2) shows the mean number of MIH packets exchanged between the PoAs and the network context monitor for the first method (query-based method).

$$Number\_of\_Packets = \sum_{N_{MN}}^{i=1} (HO\_Triggers_i \times (N_n + 1) \times 2 \tag{2}$$

In Eq. (2), $HO\_Triggers_i$ shows the number of handoff requests issued by the ith MN, N_n is the mean number of neighboring PoAs of each PoA, and N_{MN} is the number of MNs. This equation comes from the fact that the network context monitor should query the up-to-date context of the currently serving PoA and its neighbors ($N_n + 1$ request and $N_n + 1$ response) when a handover request arrives. This metric for the second method (event-based) is given by the following equation:

Table 1. Information_Update indication message.

MIH header fields (SID = 2; AID = 8; opcode = 3)
Source Identifier (Source MIHF ID TLV)
Destination Identifier (Destination MIHF ID TLV)
PoA Link ID (link identifier TLV)
Dynamic Context Parameters (dynamic context list TLV)

$$Number\_of\_Packets = \sum_{i=1}^{N_{MN}} (Completed\_HO_i \times 2 + Connections_i) \quad (3)$$

where $Completed\_HO_i$ is the number of executed handovers by the ith MN and $Connections_i$ shows the number of wireless connections established by the ith MN. The number of completed handovers is multiplied by 2, since each handover is leaving a PoA and connecting through another PoA.

The number of handover requests ($HO\_Triggers_i$) is usually greater than the number of executed handovers ($Completed\_HO_i$), and even using the best handover initiation algorithms, $HO\_Triggers_i$ is sorely equal to $Completed\_HO_i$. Being so, and considering the fact that N_n is usually greater than 1, it is obvious that the signaling overhead of the second method is lower than that of the first one. $Connections_i$ is not considerable compared to $Completed\_HO_i$ in a usual mobile environment with moving MNs.

We have also compared the 2 methods in terms of context access latency. To evaluate the delay of both methods, we have considered M/M/1 queues to model delay components. We have assumed that the MIH packets (either in the query-based method or the event-based method) arrive according to a memoryless Poisson process. The capacity of the waiting queues is assumed to be infinite and the packets are served in the order of arrival (first-come, first-served queue). The simple scenario shown in Figure 4 is considered for this comparison, where the handoff manager of [3] is assumed to be in the access router (AR).

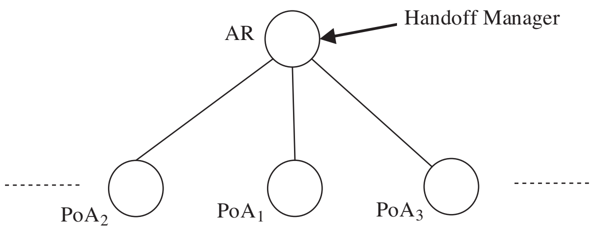

Figure 4. The chosen scenario for analysis of the context access latency.

Using the Kleinrock independence approximation, the average end-to-end packet delay of a series of tandem M/M/1 queues is obtained by adding the mean waiting time of the queues [21]. Therefore, in the query-based method, the overall delay of the context access is given by:

$$T_q = \max_{PoA_i \in neighbors(PoA_i)} \{T_{AR-PoA_i} + T_{PoA_i} + T_{PoA_i-AR}\} \quad (4)$$

where PoA_c is the currently serving PoA of the MN, T_{AR-PoA_i} is the mean waiting time of modified MIH_N2N_HO_Query_Resources request packets in the AR's transmission queue before being sent, T_{PoA_i} is the mean waiting time an information query spends in the PoA_i's M/M/1 system being investigated and responded to, and T_{PoA_i-AR} is the mean waiting time of the MIH_N2N_HO_Query_Resources response packets in PoA_i's transmission queue.

The average time a packet spends in the M/M/1 system is given by Eq. (5), where μ is the mean service rate and λ is the mean arrival rate of the packets [21].

$$T = \frac{1}{\mu - \lambda} \tag{5}$$

Therefore, T_{AR-PoA_i} is obtained as below:

$$T_{AR-PoA_i} = \frac{1}{\frac{B}{L_{req}-\lambda_{req}}} \tag{6}$$

where B is the transmission rate of the wired links, L_{req} is the length of the information request packets, and λ_{req} is the mean arrival rate of the information request packets. For simplicity, we have not considered the arrival of data packets in the queuing systems of both methods. This assumption does not invalidate our comparison, as the arrival rate and statistics of these packets are the same for both methods. Therefore, the arrival rate of the information requests depends on the rate of the MNs' handoff requests arriving at the handoff manager. Eq. (7) shows λ_{req} as a function of the MN's handover triggers.

$$\lambda_{req} = (N_n + 1) \times n \times HO\_Triggers \tag{7}$$

In Eq. (7), the average rate of triggers from each MN is defined by *HO_Triggers* and $n$is the number of MNs. $(N_n + 1)$ information requests are sent to the current serving PoA and its neighboring PoAs for each handoff request, where N_n is the mean number of the neighboring PoAs.

Assuming that the mean service rate of each PoA to respond to the requests is $\mu_{service}$, T_{PoA_i} is calculated as:

$$T_{PoA_i} = \frac{1}{\mu_{service} - \frac{\lambda_{req}}{m}} \tag{8}$$

where the mean arrival rate of information requests to each PoA is assumed to be $\frac{\lambda_{req}}{m}$ on average and m is the number of PoAs in the scenario in Figure 4. The mean waiting time in the PoA's transmission queue is the last delay component of Eq. (4). This latency is given by:

$$T_{PoA_i-AR} = \frac{1}{\frac{B}{L_{resp}-\lambda_{resp}}} = \frac{1}{\frac{B}{L_{resp}-\frac{\lambda_{req}}{m}}} \tag{9}$$

where L_{resp} is the mean length of the information response packets and L_{resp} is the mean rate of the generating response packets, which is equal to the rate of arriving requests.

For the second method (event-based method), the average end-to-end context transfer latency is only the mean waiting time of the Information_Update messages in the PoAs' queue, and it is equal to:

$$T_e = \frac{1}{\frac{B}{L_{INFO\_UP}-\lambda_{INFO\_UP}}} \tag{10}$$

where L_{INFO_UP} is the mean length of the Information_Update packets and λ_{INFO_UP} is the mean arrival rate of these packets in each PoA. λ_{INFO_UP} is a function of the mean rate of the executed handovers in each PoA and the mean rate of the connection establishments through each PoA. Assuming that the mean rate of the handovers executed by each MN is *Completed_HO* and the mean rate of the established connections from each MN is *Connections*, λ_{INFO_UP} could be written as in Eq. (11) considering the average case where $\frac{n}{m}$ of the MNs are served by each PoA.

$$\lambda_{INFO\_UP} = \frac{2 \times n \times Completed\_HO + n \times Connections}{m} \tag{11}$$

Comparing Eq. (4) to Eq. (10), it is obvious that the access latency of the event-based method is lower than that of the query-based one. To analyze both methods further, we have assumed that N_n is 3, $\mu_{service}$ is 1000 packets/s, and B is 1 Mbps. The length of the MIH signaling packets varies depending on the number and type of the context parameters; however, we assume that L_{req} is 60 bytes, and $L_{resp} = L_{INFO\_UP} = 200$ bytes on average. Figure 5 shows the delay comparison versus the ratio of *HO_Triggers* to *Completed_HO*, assuming that $n = 40, m = 6$, and the mean rate of the new coming connections (*Connections*) is 0.25. As Figure 5 shows, the context access latency of the event-based method is lower than that of the query-based method, and this difference exponentially increases as the ratio of handoff triggers to executed handovers increases. This indicates that as the decision by the MNs for handover initiation is weaker, the context access latency of the query-based method increases more compared to the event-based method.

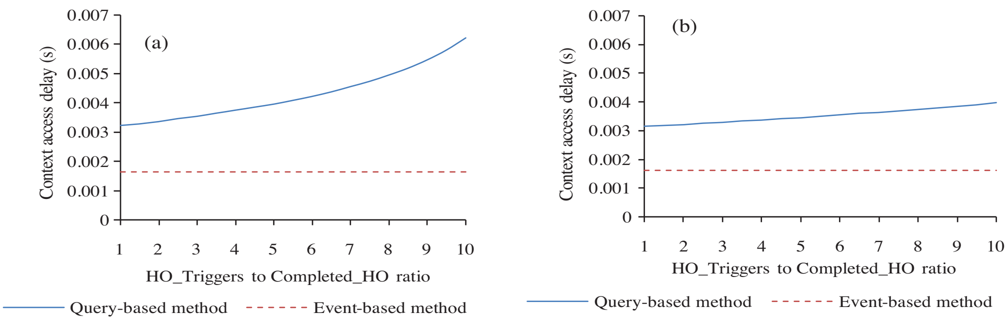

Figure 5. Delay comparison of the context gathering methods: a) *Completed_HO = 1* and b) *Completed_HO = 0.5.*

Figure 6 shows this comparison versus the number of the MNs (n), assuming that the number of PoAs is 8, *HO_Triggers* is 6, and *Completed_HO* is 2. These graphs show that as the number of MNs increases the context access latency rises, particularly for the query-based method. This delay increase is more severe in the real world, as any increment in the number of MNs results in an increase of data packets in addition to the signaling packets.

The above comparisons emphasize that the second method with the proposed MIES extension is more desirable in terms of signaling overhead and context access latency. One may assume that the query-based method is preferable due to guaranteeing accessibility to any context change (which may be caused by other reasons than handovers). However, to prepare up-to-date policies for the PDM, the network context monitor should wait for the responses from the information requests. This postponement increases the handover latency, causing additional degradations to the application that is waiting to obtain the handover decision. In contrast, the PDM is provided with the recently up-to-date policies without waiting for information via the event-based method. Moreover, as the context transfer delay in the case of the second procedure is lower than that of the first one, it is more likely that the latest context is being used.

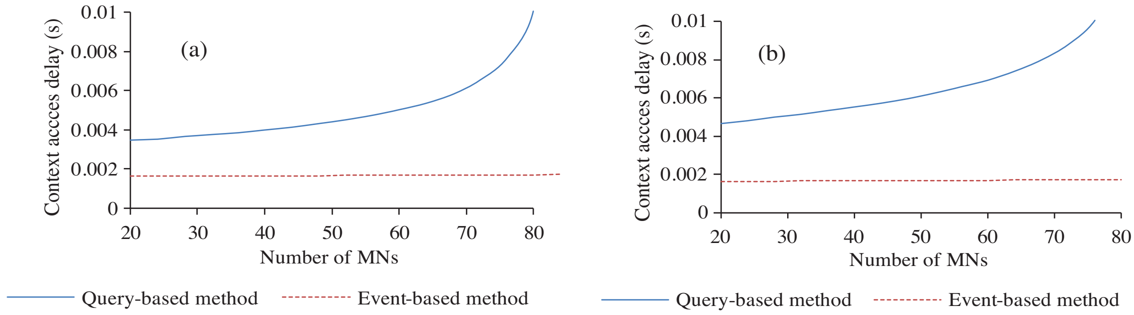

Figure 6. Context access latency versus the number of MNs: a) $\mu_{service} = 1000$ and b) $\mu_{service} = 500$.

In the next section, we will describe the procedure of automatic generation and renewal of decision policies with respect to the context information attained using the network context monitor.

5. Automatic policy generation and renewal

This section presents our proposed method, which is used by the policy management module shown in Figure 2 to automatically generate and renew the policies. Since network context is not directly applicable in [3], this approach allows the policy manager to promote the policies as the dynamic network context changes. Assuming the event-based context gathering method of the previous section, we present a procedure for renewing the policies as the context of the PoAs changes.

The context information used to construct the policies is partitioned into 2 parts; the first part is the preferences of the users/applications/network-operators and the second part is the access networks' context parameters. The preferences are provided assuming a portal where the users/operators can define their preferences in terms of QoS parameters or define preferences for different application profiles and mobile devices. The preferences are provided in 2 sections:

1. The first section includes the parameter preferences that express the importance level of the context parameters (***Param_Pref*** vector). These preferences are used in our procedure to generate the $<P, W>^*$ portion of the policies. The users/operators provide such preferences in a pairwise comparison matrix where the AHP method [22] is used to exploit the ***Param_Pref*** vector from them. Hence, the weights of the parameters are guaranteed to be between 0 and 1 and the sum of the weights to be equal to 1, as required by the FPN-based reasoning algorithm.

2. The second section includes the priorities that indicate the precedence of different policies or policy types (user/application/operator policy) among each other or the priorities between different access networks (or access network types). The priorities of the policies are used for the determination of the *Thr* portion of the policy structures, while the priorities of the access networks are utilized in combination with the PoAs' context parameters to produce the $<PN, PW>^*$ section. Users are forced to enter the priorities of networks (***NetP*** vector) between 0 and 1.

The parameter preferences may be provided along with some logical conditions (***Pref_Cond***) describing the cases where the preferences should be applied. These conditions are used in making the *FC* section of policies.

Most of the ***Param_Pref*** vectors are not dedicated to a specific serving PoA, meaning that one policy has to be provided per each available PoA, considering it as the serving PoA. Therefore, the preferred networks (*PN* in the <*PN, PW*> list) of each policy include the neighboring PoAs of the associated serving PoA, which are obtained from the network context monitor. Thus, the remaining problem with the automatic construction of the policies is the calculation of *PW* values. In the remainder of the paper, a method is proposed for calculation of those weights from the context parameters and priority of the PoAs.

The proposed method is based on a pairwise comparison of candidate PoAs using the AHP pairwise comparison matrix [22]. This matrix is constructed per each context parameter that is in the $<P,\ W>^*$ portion of policy, and its dimension depends on the number of neighboring PoAs (the number of neighbors plus 1). We define the comparison matrix by $\boldsymbol{CM}$, where $\boldsymbol{CM}_{i,j}^{C(k)}$ is the relative score of PoA_i to PoA_j with respect to the kth context parameter. AHP scoring suggests the relative scores to be scaled between 1 and 9. This relative score is obtained using the following equation:

$$\begin{aligned} &RS^{C(k)}_{PoA(i),PoA(j)} = \left(1-\frac{V^{C(k)}_{PoA(i)}}{V^{C(k)}_{PoA(j)}}\right)\times 10 && \text{if} \quad V^{C(k)}_{PoA(i)} < V^{C(k)}_{PoA(j)} \\ &\frac{1}{RS^{C(k)}_{PoA(i),PoA(j)}} = \left(1-\frac{V^{C(k)}_{PoA(j)}}{V^{C(k)}_{PoA(i)}}\right)\times 10 && \text{if} \quad V^{C(k)}_{PoA(j)} < V^{C(k)}_{PoA(i)} \\ &RS^{C(k)}_{PoA(i),PoA(j)} = 1 && \text{if} \quad V^{C(k)}_{PoA(j)} = V^{C(k)}_{PoA(i)} \end{aligned} \tag{12}$$

where $V^{C(k)}_{PoA(i)}$ is the value of the kth context parameter for PoA_i and $RS^{C(k)}_{PoA(i),PoA(j)}$ shows the relative score between 2 PoAs regarding the kth context parameter. Hence, the pairwise comparison matrix for cost parameters (those preferred to be as low as possible) with respect to the kth context parameter is as below. For value parameters (those preferred to be as high as possible), the transpose of this matrix is used.

$$\mathbf{CM}^{C(k)} = \begin{bmatrix} 1 & RS^{C(k)}_{PoA(1),PoA(2)} & RS^{C(k)}_{PoA(1),PoA(3)} & \cdots \\ \frac{1}{RS^{C(k)}_{PoA(1),PoA(2)}} & 1 & RS^{C(k)}_{PoA(2),PoA(3)} & \cdots \\ \frac{1}{RS^{C(k)}_{PoA(1),PoA(3)}} & \frac{1}{RS^{C(k)}_{PoA(2),PoA(3)}} & 1 & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix} \tag{13}$$

The $\boldsymbol{CM}^{C(k)}$ matrix should be normalized by dividing each element by the individual sum of the elements of its column. The average value of each row is then calculated to give the weight of PoA_i regarding the kth context parameter, as shown in Eq. (14).

$$PW^{C(k)}_{PoA(i)} = \frac{\sum_{l=1}^{m} \mathbf{CM}^{C(k)}_{i,l}}{m} \tag{14}$$

where m is the number of columns in the $\boldsymbol{CM}$ matrix. The final weight of each candidate PoA in the policy is obtained by a weighted averaging on the weights calculated from Eq. (14) and the probable network/network-type priorities. Eq. (15) shows this evaluation, where K is the number of context parameters considered in the policy, $W^{C(k)}$ is the weight of the kth context parameter in the ***Param_Pref*** vector, and $\boldsymbol{NetP}_{PoA(i)}$ is the priority of PoA_i if any network priority vector (***NetP***) has been determined in the preferences for this policy.

Procedure 1. Automatic policy generation.

For each ***Param_Pref*** vector

- Make an empty *Policy* structure
- Insert the identifications of the sources (user/application/device/...) that the policy belongs to into the *Policy.FC* portion
- Insert any ***Pref_Cond*** that may be provided along the ***Param_Pref*** in *Policy.FC*
- Utilize the ***Param_Pref*** to fill the *Policy.*<*P, W*>* portion
- Use the normalized priority of the policy with respect to other policies to fill the *Policy.Thr* field or 0 if there is no priority
- If an specific PoA (PoA_x) has been determined for this policy
  - Exploit the network priority and context information of PoA_x and its neighbors to prepare the *Policy.*<*PN, PW*>* using Eqs. (12) to (15)
  - Insert the $PoA_x.ID$ into *Policy.FC* portion as the serving PoA
  - Add *Policy* into the policy repository
- Else; For i = 1 to $|PoAsList|$
  - PoA_i = *PoAsList[i]*
  - $Policy_i$ = *Policy*
  - Exploit the network priority and context of PoA_i and its neighbors to prepare the $Policy_i$.<*PN, PW*>* using Eqs. (12) to (15)
  - Insert the $PoA_i.ID$ in $Policy_i.FC$ portion as the serving PoA
  - Add $Policy_i$ into the policy repository

For each independent ***NetP*** vector (that has not been entered with ***Param_Pref***)

- Make an empty *Policy* structure
- Insert the identifications of any sources (user/application/device/...) that the policy belongs to into the *Policy.FC* portion
- Insert any ***Pref_Cond*** that may be provided along the ***NetP*** vector in *Policy.FC*
- Use the normalized priority of policy with respect to other policies to fill the *Policy.Thr* field or 0 if there is no priority
- Fill the *Policy.*<*P, W*>* portion for all of the defined context parameters with equal weights (s.t. $\sum$W = 1)
- If a specific PoA (PoA_x) has been determined for this policy
  - Insert the $PoA_x.ID$ into *Policy.FC* portion as the serving PoA
- Exploit the normalized priority of each *PoA* in ***NetP*** to prepare the $Policy_i$.<*PN, PW*>* using Eq. (15)

Add *Policy* into the policy repository

$$\begin{aligned}
PW_{PoA(i)} &= \frac{\sum_{k=1}^{K}(PW^{C(k)}_{PoA(i)} \times W^{C(k)}) + \mathbf{NetP}_{PoA(i)}}{2}, && \text{if } \mathbf{NetP} \text{ is available} \\
PW_{PoA(i)} &= \sum_{k=1}^{K}(PW^{C(k)}_{PoA(i)} \times W^{C(k)}), && \text{if } \mathbf{NetP} \text{ is not available} \\
PW_{PoA(i)} &= \mathbf{NetP}_{PoA(i)}, && \text{if } \mathbf{Param_pref} \text{ is not available}
\end{aligned} \quad (15)$$

The procedure of generating different parts of a policy is explained in Procedure 1.

In addition to the generation of new policies, the policy manager is responsible for the renewal of the relevant policies as the context parameters of any PoA changes (choosing event-based context update method). The renewal procedure is shown in Procedure 2.

Procedure 2. Automatic policy renewal.

For each *Policy* in the policy repository

- If *Policy.*<*PN, PW*>* contains any PoA in *PNs* that has had its context changed
  - Exploit the network priority and recent context of its *PN*s to recalculate the $Policy_i$.<*PN, PW*>* using Eqs. (12) to (15)

Notice that the proposed policy generation and renewal is reasonable since it performs the same as the AHP method in the cases of certainty. Assuming 1 FPN, which is constructed from a policy, the inputs to this FPN are all 1 under certainty conditions, and applying the FPN reasoning algorithm of the previous study to this FPN returns the same scoring result as the AHP method. This is due to the fact that the weighted sum of the FPN inputs is 1, meaning that the score of each PoA is its relevant weight, PW in policy, which is obtained from the AHP method.

6. Simulation results

This section presents the simulation results of the proposed extensions of [3]. The simple scenario of Figure 7 has been simulated in NS2. In this scenario, MN_1 is receiving an MPEG video stream (Flow 1) from CN while MN_2 is receiving both an MPEG stream (Flow 3) and a constant bit rate (CBR) flow (Flow 2) from CN using UDP connections. The Wi-Fi access points (APs) are assumed to cover a range of 50 m and are connected through 100-Mbps links to the AR. Both video streams are assumed to play back at a speed of 30 frames/s with a 300-kbps bandwidth on average and the same traffic requirements, while the bit rate of the CBR flow is considered to be 8 Mbps.

We assume 3 parameter preferences specified by $User_1$ and $User_2$ as the users of MN_1 and MN_2, and also a general preference for all of the video traffic by the network administrator as shown in Eq. (16). The preferences are denoted in terms of 3 context parameters, namely bandwidth, packet delay, and packet loss.

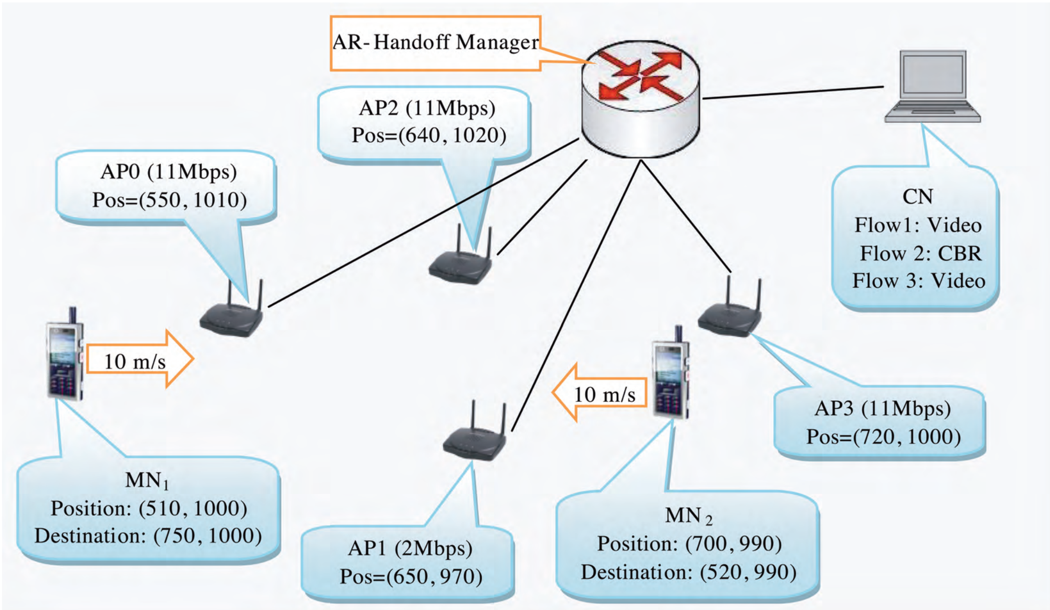

Figure 7. Evaluation scenario.

$$\begin{aligned}
&\textbf{Parame_Pref}_{\text{User1, Video}} = [\overset{\text{bw}}{0.6}\ \ \overset{\text{delay}}{0.3}\ \ \overset{\text{loss}}{0.1}];\text{bw} \in [200\text{K}\ \ 500\text{K}]\,;\ \text{delay} \in\ [1\text{m}\ \ 10\text{m}];\ \text{loss} \in [2\%\ \ \ 10\%\ ]\\
&\textbf{Parame_Pref}_{\text{User2, Video}} = [\overset{\text{bw}}{0.6}\ \ \overset{\text{delay}}{0.3}\ \ \overset{\text{loss}}{0.1}];\text{bw} \in [200\text{K}\ \ 500\text{K}]\,;\ \text{delay} \in\ [1\text{m}\ \ 10\text{m}];\ \text{loss} \in [2\%\ \ \ 10\%\ ]\\
&\textbf{Parame_Pref}_{\text{User2, CBR}} = [\overset{\text{bw}}{0.7}\ \ \overset{\text{delay}}{0.05}\ \ \overset{\text{loss}}{0.25}];\text{bw} \in [2\text{M}\ \ 9\text{M}]\,;\ \text{delay} \in\ [10\text{m}\ \ 100\text{m}];\ \text{loss} \in [0.1\%\ \ \ 2\%\ ]\\
&\textbf{Parame_Pref}_{\text{Video}} =\ [\overset{\text{bw}}{0.2}\ \ \overset{\text{delay}}{0.6}\ \ \overset{\text{loss}}{0.2}];\ \text{bw} \in [100\text{K}\ \ 400\text{K}]\,;\ \text{delay} \in\ [5\text{m}\ \ 20\text{m}];\ \text{loss} \in [5\%\ \ \ 10\%\ ]
\end{aligned} \tag{16}$$

The minimum and maximum values that have been presented for each parameter, along with the preferences, will be used for fuzzifying the requirements of the applications that request handover (inputs to the FPN reasoning algorithm), as discussed in [3]. The values of the PoAs' context parameters at the beginning of the simulation are shown in Table 2, considering that, initially, MN_1 is connected through AP_0 and MN_2 is connected through AP_3.

Table 2. The values of the PoAs' context parameters at the beginning of the simulation.

	Available bandwidth	Mean frame delay	Mean frame loss rate
AP_0	10.7 Mbps	1 ms	0.01
AP_1	2 Mbps	1 ms	0.01
AP_2	11 Mbps	1 ms	0.01
AP_3	2.7 Mbps	3.5 ms	0.07

Assuming no priority between policies and access networks, the required policies shown in Table 3 have been made from the above context using Procedure 1.

Assuming similar context parameters, the event-based context gathering extension and related policy generation and renewal method is compared to the previous method of [3]. As the previous method does not provide dynamic context to the PDM, the fixed policies that have been constructed from nominal values of the relevant context parameters are used. Those policies that are the same as the ones in Table 3 do not change during the simulation. However, for the proposed extension, the policies change during practice as the QoS parameters of the PoAs change.

Table 3. Some of the policies constructed at the beginning of the simulation before any handover.

FC			<P, W>*			<PN, PW>*			
Current PoA	Traffic class	User ID	Bandwidth	Delay	Loss	AP_0	AP_1	AP_2	AP_3
AP_0	Video	$User_1$	0.6	0.3	0.1	0.308	0.168	0.524	
AP_3	Video	$User_2$	0.6	0.3	0.1		0.23	0.66	0.11
AP_0	Video	-	0.2	0.6	0.2	0.32	0.28	0.40	
AP_3	Video	-	0.2	0.6	0.2		0.39	0.53	0.08

For video flows, the peak signal-to-noise ratio (PSNR) levels of the video frames have been compared, while for the CBR flow, the throughput and the number of lost packets is the comparison measure. Figure 8 shows the PSNR level of 700 frames of both video flows under the proposed method and the previous method. Short-term PSNR degradations are due to the handover procedure and occur in both methods. However, using the previous method, $User_1$ experiences long-term PSNR degradations after handover from AP_0 to AP_2, which also affects the quality of Flow 3 for a while before MN_2 exits the coverage area of AP_2. Figure 9 shows the delay variations of the video frames for both methods, where the graphs demonstrate the same phenomena due to the above mentioned reasons.

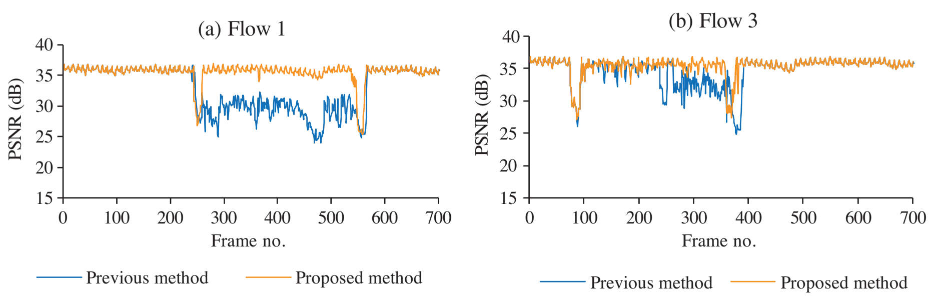

Figure 8. PSNR comparison of video flows under the proposed and the previous methods: a) Flow 1 received by MN_1 and b) Flow 3 received by MN_2.

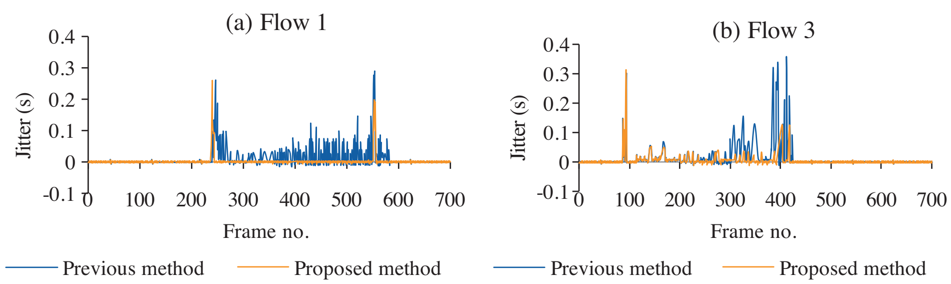

Figure 9. Delay variations of video frames under the proposed and the previous methods: a) Flow 1 received by MN_1 and b) Flow 3 received by MN_2.

Figure 10 shows the throughput for the CBR flow, which also experiences more degradation from the eighth second of simulation, when MN_1 hands over to AP_2. In the case of the CBR flow, 1241 packets were lost using the previous method, which is much more than the 980 lost packets under the proposed method. Successive packet loss may occur due to an inappropriate handover decision by MN_1. However, the harmful effect of this handover is more severe for delay-sensitive applications (videos) than loss-sensitive traffic.

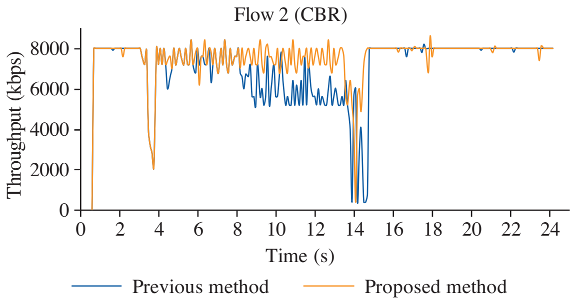

Figure 10. Throughput comparison of Flow 2 under the proposed and previous methods.

For further investigations, we present a numerical analysis for the 2 mentioned methods in the remainder of this section. After the start of the simulation, the first handover request comes from MN_2 for its video flow due to degrading user perception quality (refer to [3] for details on UPQ trigger). The handover request contains the following requirements: bandwidth = 300 kbps; acceptable delay = 5 ms; and acceptable loss rate = 5%. Using Eqs. (17) and (18), the inputs to the relevant FPN of Figure 11 are obtained as below for value and cost parameters, respectively.

$$\begin{array}{lll} P_i = \frac{req_i - l_i}{u_i - l_i} & \text{if} & l_i < req_i < u_i \\ P_i = 1 & \text{if} & req_i \geq u_i \text{or} l_i = u_i \\ P_i = 0 & \text{if} & req_i \leq l_i \end{array} \tag{17}$$

$$\begin{array}{lll} P_i = 1 - \frac{req_i - l_i}{u_i - l_i} & \text{if} & l_i < req_i < u_i \\ P_i = 0 & \text{if} & req_i \geq u_i \\ P_i = 1 & \text{if} & req_i \leq l_i \text{or} l_i = u_i \end{array} \tag{18}$$

The PDM applies the reasoning algorithm on this FPN, and the output values for the candidates (shown in Figure 11) are reported to MN_2 to select the best PoA from this ranking with respect to its local decision (AP_2 is selected in our simulated case). After completion of the handover, AP_2 sends an Information_Update message to update the dynamic context parameters in the handoff manager.

In the next stage, MN_2 requests the handover of its CBR flow due to the Link_Going_Down event. The same procedure is performed for this request and AP_2 is selected for this flow, as the changes in context of AP_2 (from the previous handover) are not considerable. However, when MN_1 requests the ranking of the candidate PoAs for the sake of degradations in its video quality, the policies in the policy manager have changed considerably due to the earlier Information_Update message from AP_2. Table 4 shows that the relevant modified policies, using the importance level of candidate PoAs from the FPN reasoning algorithm, are 0.87 for AP_0, 0.36 for AP_1, and 0.11 for AP_2. Employing these values and the local decision based on the RSS level, MN_1 hands over its video flow to AP_1. This selection is different from the selection of the similar method that only considers static context (such as the nominal data rate of the PoAs) due to selecting AP_2 for MN_1 video flow. However, MN_1 does not enjoy a steady quality of video play back, as the service quality of AP_2 is diminished owing to serving MN_2 concurrently. This effect is obvious for the video frames of Flow 1 (about 300 video frames) in the simulation results, as shown in Figures 8 and 9.

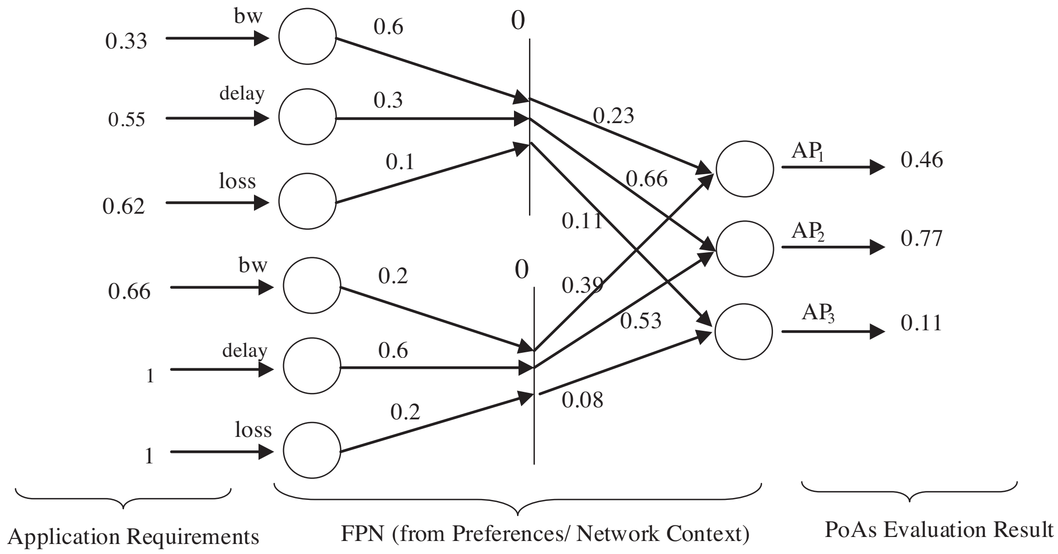

Figure 11. The FPN instance constructed for MN_2 handover request.

Table 4. Two of the policies modified after the handover of MN_2 flows.

FC			<P, W>*			<PN, PW>*			
Current PoA	Traffic class	User ID	Bandwidth	Delay	Loss	AP_0	AP_1	AP_2	AP_3
AP_0	Video	$User_1$	0.6	0.3	0.1	0.7	0.2	0.1	
AP_0	Video	-	0.2	0.6	0.2	0.62	0.3	0.08	

7. Conclusion

This paper presents 2 extensions for our previously proposed context-aware handover method. First, the query-based and event-based network context gathering methods have been proposed under the MIH framework to exploit the dynamic context of PoAs for target PoA selection. Analyzing the system in terms of context access latency, we have shown that event-based extension is more desirable compared to the query-based method. In addition, we have proposed procedures required for the automatic construction, maintenance, and renewal of policies from mobile-side and network-side context parameters. These procedures make the previous work more feasible for real implementations. The extended method has been compared to the previous one via some simulations and numerical examples. The simulation results have shown better performance for sensitive traffic using our proposed extensions.

References

[1] C. Prehofer, N. Nafisi, Q. Wei, "A framework for context-aware handover decisions", 14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications, pp. 2794-2798, 2003.

[2] IEEE, IEEE Standard for Local and Metropolitan Area Networks: Media Independent Handover Services, IEEE 802.21, 2008.

[3] B.S. Ghahfarokhi, N. Movahhedinia, “A context-aware handover decision based on user perceived quality of service trigger”, Wireless Communications and Mobile Computing, Vol. 11, pp. 723-741, 2011.

[4] W. Zhang, “Handover decision using fuzzy MADM in heterogeneous networks”, IEEE Wireless Communications and Networking Conference, Vol. 2, pp. 653-658, 2004.

[5] T. Ahmed, K. Kyamakya, M. Ludwig, “Architecture of a context-aware vertical handover decision model and its performance analysis for GPRS - WiFi handover”, Proceedings of the 11th IEEE Symposium on Computers and Communications: IEEE Computer Society, pp. 795-801, 2006.

[6] S.F. Yang, J.S. Wu, H.H. Huang, “A vertical media-independent handover decision algorithm across Wi-Fi and WiMAX networks”, The 5th IFIP International Conference on Wireless and Optical Communications Networks, pp. 1-5, 2008.

[7] S. Balasubramaniam, J. Indulska, “Vertical handover supporting pervasive computing in future wireless networks”, Computer Communications, Vol. 27, pp. 708-719, 2004.

[8] M. Kassar, B. Kervella, G. Pujolle, “An overview of vertical handover decision strategies in heterogeneous wireless networks”, Computer Communications, Vol. 31, pp. 2607-2620, 2008.

[9] F. Zhu, J. McNair, “Multiservice vertical handoff decision algorithms”, EURASIP Journal on Wireless Communications and Networking, Vol. 2006, doi: 10.1155/WCN/2006/25861, 2006.

[10] J.Z. Sun, J. Riekki, J. Sauvola, M. Jurmu, “Towards connectivity management adaptability: context awareness in policy representation and end-to-end evaluation algorithm”, Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia, pp. 85-92, 2004.

[11] J.M. Kang, H.T. Ju, J. Hong, “Towards autonomic handover decision management in 4G networks”, Autonomic Management of Mobile Multimedia Services, Vol. 4267, pp. 145-157, 2006.

[12] Q. Wei, K. Farkas, C. Prehofer, P. Mendes, B. Plattner, “Context-aware handover using active network technology”, Computer Networks, Vol. 50, pp. 2855-2872, 2006.

[13] C.P. Hong, C.C. Weems, S.D. Kim, “An effective vertical handoff scheme based on service management for ubiquitous computing”, Computer Communications, Vol. 31, pp. 1739-1750, 2008.

[14] T.H. Kang, C.P. Hong, Y.S. Kim, S.D. Kim, “A context-aware handoff management for seamless connectivity in ubiquitous computing environment”, International Conference on Pervasive Systems & Computing, pp. 128-134, 2006.

[15] A. De La Oliva, A. Banchs, I. Soto, T. Melia, A. Vidal, “An overview of IEEE 802.21: media-independent handover services”, IEEE Wireless Communications, Vol. 15, pp. 96-103, 2008.

[16] F. Cacace, L. Vollero, “Managing mobility and adaptation in upcoming 802.21 enabled devices”, Proceedings of the 4th International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots, pp. 1-10, 2006.

[17] T. Al Mosawi, D. Wisely, H. Aghvami, “A novel micro mobility solution based on media independent handover and SIP”, IEEE 64th Vehicular Technology Conference, pp. 1-5, 2006.

[18] Y.C. Yee, K.N. Choong, A.L.Y. Low, S.W. Tan, “SIP-based proactive and adaptive mobility management framework for heterogeneous networks”, Journal of Network and Computer Applications, Vol. 31, pp. 771-792, 2008.

[19] Y. Wang, J. Yuan, Y. Zhou, G. Li, P. Zhang, "Vertical handover decision in an enhanced media independent handover framework", Wireless Personal Communication, Vol. 52, pp. 615-636, 2010.

[20] Q.B. Mussabbir, W. Yao, Z. Niu, X. Fu, "Optimized FMIPv6 using IEEE 802.21 MIH services in vehicular networks", IEEE Transactions on Vehicular Technology, Vol. 56, pp. 3397-3407, 2007.

[21] N.C. Hock, Queueing Modelling Fundamentals, Chichester, John Wiley & Sons, 1996.

[22] T.L. Saaty, "How to make a decision: the analytic hierarchy process", European Journal of Operational Research, Vol. 48, pp. 9-26, 1990.