Multipurpose Mobility Services for the Future Internet

Francisco A. Gonzalez-Horta, [1] Pedro Mejia-Alvarez, [2] Eldamira Buenfil-Alpuche [3]

[1, 2] CINVESTAV-IPN, Computer Science Department, Mexico City

[3] UNILA, Research Department, Cuernavaca, Mor., Mexico

[1] fglez@computacion.cs.cinvestav.mx, [2] pmalvarez@cs.cinvestav.mx, [3] ebuenfil@unila.edu.mx

Abstract

Mobility and handoff management is a key problem of the Future Internet. Current solutions provide mobility services, such as seamless mobility, adaptive mobility, always-best-connected (ABC) mobility, etc. The problem is these services work separately and ignore conflicts between them. This may lead to improve one service and degrade others. Hence, we propose *multipurpose mobility* as a new holistic service that integrates multiple mobility services and supplies a *fair balance* between all the objectives to meet. As a proof-of-concept, we integrate the seamless, ABC, and adaptive mobility services, which have objectives in conflict. We formulate a Multi-Objective Handoff Optimization Problem, which grades as NP-Hard. We develop a heuristic handoff algorithm, which provides near-optimal and balanced solutions. Finally, we evaluate the algorithm through random samples of simulated handoff scenarios, which provide hit rates over 90%.

Key Words: Future Internet, handoff optimization, multipurpose mobility

1. Introduction

The Future Internet denotes a communication system that is present anywhere, anytime (*ubiquitous*), connects any user and any terminal (*universal*), supports mobility across any wireless access network (*mobile*), conveys any service over any access network and any terminal (*multiservice*), and hides heterogeneity (*seamless*) via homogeneous layers of IP software (*uniform*). Hence, the *Future Internet* [1] is ubiquitous, universal, *mobile*,

multiservice, seamless, and uniform. However, in order to achieve *multiservice mobility*, the Internet must improve a primary task called *mobility management* [2].

The problem of mobility management is the leading of packets from source to target, while source, target, or both change their network *Points of Attachment* (PoA). This problem is easy to describe, yet complex to solve. The packets delivery service must adapt to connectivity changes, while it satisfies continuity, correctness, and timeliness constraints. To face this complexity, mobility management divides into two problems: *location management*, which determines the route to reach the target at any time, and *handoff management*, which preserves the communications while end-systems change their attachment points. Location managers deal with *mobility protocols* [3]. Handoff managers deal with *handoff algorithms* [4]. We focus on mobility and handoff management.

Mobility and handoff management is extensively studied in the literature. Hence, many *mobility services* have been proposed; e.g., seamless mobility [5], autonomic mobility [6], adaptive mobility [7], always-best-connected (ABC) mobility [8], secure mobility [9], energy-efficient mobility [10], and others [11]. The problem is these services work separately and ignore conflicts between them. Consequently, one service might be improved while another is worsened, yielding an erratic and unbalanced behavior. Hence, these solutions may be seamless but not ABC, ABC but not adaptive, adaptive but not secure, secure but not power-efficient, etc. This means that *single-purpose* services will not be able to achieve multiservice mobility. When we integrate several mobility services into a service of *multipurpose mobility*, the integrated objectives may conflict with each other. In this case, a new task of the *handoff control manager* is to optimize and maintain a *fair balance* between all the objectives to meet. Since that task is not easy to achieve and it

affects the global behavior of handoff algorithms, we propose a paradigm shift from single-purpose to multipurpose mobility [12].

Despite the broad literature on mobility and handoff management, multipurpose mobility and multiobjective handoff optimization remain largely unexplored. Hence, we are interested in integrate *ABC mobility*, *seamless mobility*, and *adaptive mobility*. The purpose of ABC mobility is to keep users always connected to the most appropriate access network. This requires a mechanism to select the most suitable network and *maximize the dwelling-time in the best available connection* (DTiB). The purpose of seamless mobility is to preserve service continuity. This requires reducing the communication disruptions during handoffs, which implies to *minimize* parameters, such as the *handoff latency*, the *cumulative handoff latency* (CHoL), or the *number of executed handoffs* (nEHO). The purpose of adaptive mobility is to keep the success of all handoffs in all mobility scenarios. This requires a mechanism to determine the *success* or *failure* of handoffs and estimate the rate of successful scenarios. Adaptive mobility intends to *maximize* the *number of successful handoffs* (nSHO) or the *number of successful scenarios*. Especially, seamless mobility, ABC mobility, and adaptive mobility are mutually in conflict, and *tradeoffs* between conflicting objectives make multipurpose mobility a difficult problem. Thus, we are concerned with improving and balancing these services as much as possible.

In this paper, we formalize a Multi-Objective Handoff Optimization Problem (MOHOP) addressed to maximize DTiB, minimize nEHO, and maximize nSHO. As far as we know, there are no prior efforts providing a formalization and solution to this problem. Moreover, we classify this problem as NP-Hard. Using *deterministic heuristics*, we propose a handoff algorithm that provides near-optimal and balanced solutions in polynomial time. To verify this algorithm, we use a simple handoff simulator that creates samples of handoff

scenarios, displays the algorithm's behavior, and measures handoff performance parameters (e.g., DTiB, nEHO, and nSHO). A statistical analysis on hundreds of random samples estimates relative frequencies of acceptable solutions over 90%.

2. Problem Modeling

Let us introduce the problem modeling and the challenge of handoff management in the Future Internet with an application scenario and relevant contextual information.

2.1. Application Scenario and PoA Concept

Fig. 1 illustrates a Mobile Video-Surveillance System where end users convey real-time multimedia traffic while they change their points of attachment to the network.

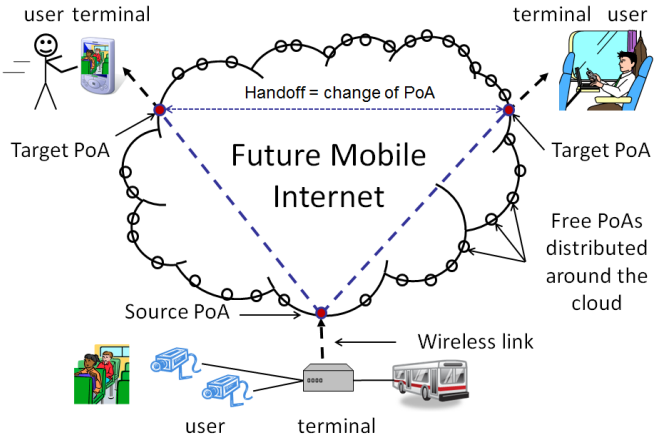

Fig. 1. Mobile Video-Surveillance System. This system uses video cameras connected to a mobile router inside a public transport vehicle in order to monitor in real-time the security of passengers. A police officer, using a tablet or laptop, checks the vehicle's internal environment. The challenge is to preserve the video quality as both officer and vehicle arbitrarily change their PoAs.

This application is a challenge to the handoff management of the Future Internet. We describe three entities that play a crucial role in this issue: *users*, *terminals*, and *PoAs*.

Users. In the Future Internet, users will be humans, sensors, actuators, machines, or objects (*things*) with the ability to collect, process, or send information to other users. In Fig. 1, the police officer and the video cameras are end users.

Terminals. Mobile terminals are hardware devices with the ability to interface a user with the communication network through attachment points. In Fig. 1, tablets, smartphones,

or laptops are terminals to the police officer, whereas mobile routers in the vehicle are terminals to the video cameras. Terminals may send data through various PoAs, either simultaneously or changing the active PoA one at a time. This choice separates two different problems, *the handoff problem*, which assumes one active PoA at a time, and *the multihoming problem*, which assumes several active PoAs simultaneously [3]. An active PoA is an attachment point that is currently transferring data to the network.

PoAs. Traditionally, a PoA is a connection point to the access network [4]. We extend this concept and define a PoA as a connection point to each layer of the network structure; i.e., a *channel* at the physical layer, a *base station* at the access layer, an *IP network* at the distribution layer, and a *service provider* at the core layer. We define a PoA as follows.

Definition (PoA). *Let C, B, I, P be universal sets of Channels, Base stations, IP networks, and Providers, respectively. A network Point of Attachment (PoA) is a tuple (c, b, i, p) such that* $c \in C$, $b \in B$, $i \in I$, *and* $p \in P$. *The universal set of PoAs is the set* $A = C \times B \times I \times P$.

We envision PoAs as *sockets* that allow terminals to establish connections with elements in the inner network. PoAs also represent *connectivity resources* distributed around the cloud. They are busy or free depending on whether a terminal has established a connection with such PoAs or not. Besides, any PoA $a \in A$ has a *desirability value* $D_a(t)$, which represents the overall valuation of *a* in different aspects (e.g., quality, cost, security). These factors are dynamic and hard to predict, so the PoA desirability may change *rapidly* and *unexpectedly*. At any specific *t* time, *the best PoA* is the one with highest desirability value.

Users, Terminals, and PoAs. Users, terminals, and PoAs intertwine directly; users need terminals to establish network connections through PoAs. Conversely, PoAs create *security associations* with terminals and users. We define binary relations between these entities.

Definition (Relations Users-Terminals-PoAs). *Let U, M, and A be universal sets of Internet Users, Mobile Terminals, and Attachment Points, respectively. We define binary relations: UM Relation = {(u, m) : u ∈ U, m ∈ M, u uses m}; MA Relation = {(m, a) : m ∈ M, a ∈ A, m connects to a}; UA Relation = {(u, a) : u ∈ U, a ∈ A, u associates with a}.*

The UM Relation states a '*usage'* relationship between users and terminals. A user can use zero, one, or many terminals, and zero, one, or many users can use a terminal, thus, UM is a many-to-many relationship. The MA Relation states a '*connectivity'* relationship between terminals and PoAs. A terminal can connect to zero, one, or many PoAs, and one PoA can have zero, one, or several connected terminals, thus, the MA relation is many-to-many. The UA Relation states an '*association'* relationship between users and PoAs. It is the relationship between users and providers, which is many-to-many.

Definition (Null Elements). *There exist null elements, namely, null user $u_0 \in U$, null terminal $m_0 \in M$, and null PoA $a_0 \in A$, such that when they relate to other elements, they produce new entities. For instance, $UM(u,m_0)$ represents a user with no terminal, $UM(u_0,m)$ is a terminal with no user, $MA(m,a_0)$ is a disconnected terminal, $UA(u,a_0)$ is an offline user, and both $UA(u_0,a)$ and $MA(m_0,a)$ represent a free PoA (i.e., a PoA that is dissociated from any user and any terminal.)*

Definition (Available PoA). *Let A(a, m, u) mean PoA a is available to terminal m and user u; R(m, a) mean m reaches a; P(a, m, u) mean the provider of a has authenticated terminal m and user u; thus, $(\exists a \in A, m \in M, u \in U)$ $[A(a, m, u) \Leftrightarrow (R(m, a) \wedge P(a, m, u) \wedge UM(u, m))]$.*

Finally, communicating entities in the Future Internet must have the ability to be *identifiable* [13]. That is, there must be a *unique identifier* associated with each producer or

consumer of information [14]. Every communicating entity combines a user, a terminal, and an attachment point. We name this identification as follows.

Definition (Unique Identifier). *For each communicating entity k (producer or consumer), let $\phi(k)$ be the identifier of k represented by the tuple $(u_k, m_k, a_k) \in U \times M \times A$. Each communicating entity has exactly one identifier. That is, $(\forall i,j)(\phi(i) = \phi(j) \Rightarrow i = j)$.*

In this way, identifiers for communicating entities are unique, yet dynamic. An identifier shifts as the user changes of terminal or the terminal changes of PoA.

2.2. Handoff Definition

A *handoff*/*handover* is a process that changes the identifier of a communicating entity. Different types of handoff may occur: from (u, m_{old}, a) to (u, m_{new}, a), from (u, m_{old}, a_{old}) to (u, m_{new}, a_{new}), and from (u, m, a_{old}) to (u, m, a_{new}). In this work, we consider handoffs as changes of PoA only, from an old PoA to a new PoA, or rather, from the current PoA a_c to the best available PoA a_b. We define a handoff as follows.

Definition (Handoff). *Assume that two PoAs, a_c and a_b, are available to terminal m and user u. Before handoff, $(m, a_c) \in MA$, $(m, a_b) \notin MA$, and a_c is better than a_b according to a particular metric (e.g., desirability). If a_b becomes better than a_c, then a handoff will initiate. During handoff, a function $h: U \times M \times A \rightarrow U \times M \times A$ is applied to the communicating entity to change its identity from (u, m, a_c) to (u, m, a_b). After handoff, $(m, a_c) \notin MA$, $(m, a_b) \in MA$, and the new attachment point a_b becomes the current attachment point a_c; this closes a cycle until a new a_b appears in the scene.*

Currently, mobile terminals can change their active attachment points regardless the terminals are moving or not. Modern wireless networks are overlaid, large cells over small cells, so, several PoAs may be available in a specific place and time.

2.3. Handoff Types and Handoff Latencies

A change of PoA may involve a change of channel *c*, base station *b*, IP network *i*, or service provider *p*. Different *types of handoff* occur depending on the parameter that is changing. A terminal achieves a *layer 1 handoff* if *c* changes within the same *b*, *i*, *p*. A *layer 2 handoff* occurs if the terminal changes *c* and *b*, but preserves the same *i*, *p*. A *layer 3 handoff* happens when the terminal changes *c*, *b*, and *i*, but maintains the same *p*. Finally, a *layer 4-7 handoff* takes place when the terminal changes *c*, *b*, *i*, *p*, simultaneously. As expected, the more layers involved in a handoff, the higher is its latency and complexity. The *handoff latency* for a *k-layer* handoff, namely Λ_k, is given by $\Lambda_k = \sum_{i=1}^{k} \lambda_i$, where λ_i is the handoff latency in a specific layer *i*. The handoff latency is hard to predict, since different kinds of handoff may occur: *horizontal/vertical*, *symmetrical/asymmetrical*, *upward/downward*, *soft/hard*, *imperative/opportunistic*, *reactive/predictive*, etc. Authors in [11] provide a thorough classification of handoffs.

2.4. Problem Formulation

Let us consider a *handoff* (*mobility*) *scenario* as a particular scene where several PoAs, including the null PoA, concur and are available to terminal *m* and user *u* for a time called *scenario length*. Considering that *m* has only one active PoA at a time, we depict in Fig. 2 connectivity changes that might result from such a roaming terminal. Fig. 2 distinguishes three kinds of time intervals: *disconnection intervals* $T(a_0)$, *connection intervals* $T(a_c)$, and *handoff intervals* $\Lambda_k(a_c a_b)$. Expression $[T(a_0)]_i$ represents the *i*th disconnection interval. Expression $[T(a_c)]_i$ describes the *i*th connection interval where *m* connects to a_c (current PoA). Finally, expression $[\Lambda_k(a_c a_b)]_i$ represents the *i*th handoff interval where *m* performs a k-layer handoff from a_c (current PoA) to a_b (best PoA candidate).

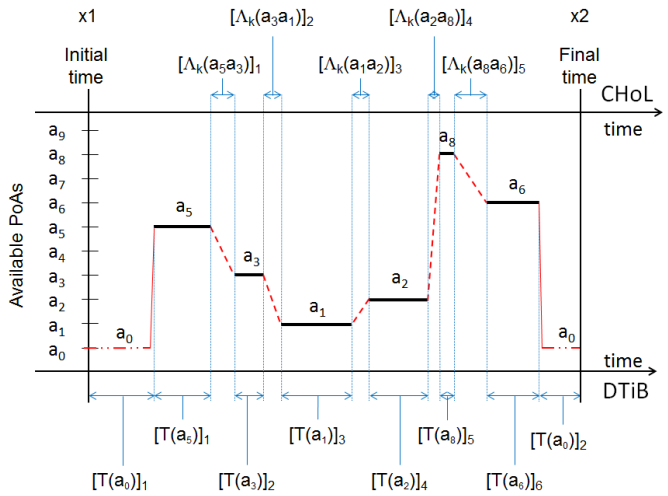

Fig. 2. Connectivity Timelines Split by PoAs. This diagram shows transitions (handoffs) from one active PoA to another. Major parameters in this model are the sum of handoff latencies CHoL, the sum of connection times DTiB, the sum of disconnection times DTiD, the number of executed handoffs nEHO, the number of disconnection intervals nDI, and the scenario length (x2 – x1).

Using nEHO and nDI, we define Cumulative Handoff Latency CHoL, Dwelling-Time in the Best DTiB, and Dwelling-Time in Disconnection DTiD, as follows.

$$CHoL = \sum_{i=1}^{nEHO} [\Lambda_k(a_c a_b)]_i \; ; \; a_c \neq a_b \neq a_0 \, , nEHO \geq 0. \quad (1)$$

$$DTiB = \sum_{i=1}^{nEHO+1} [T(a_c)]_i \; ; \; a_c \neq a_0 \, , nEHO \geq 0. \quad (2)$$

$$DTiD = \sum_{i=1}^{nDI} [T(a_0)]_i \; ; \; nDI \geq 0. \quad (3)$$

Transitions to and from a_0 are not considered handoffs; they rather represent transitions between connection and disconnection states, thus a_0 is excluded from (1) and (2). Disconnection intervals occur when there is no available PoA in such a period. Hence, the size and number of disconnection intervals do not depend on the handoff algorithm, but on the availability of PoAs in the scenario. By splitting the scenario length into disjoint intervals DTiB, CHoL, and DTiD, the following expression holds.

$$\frac{DTiB}{(x2 - x1)} + \frac{CHoL}{(x2 - x1)} + \frac{DTiD}{(x2 - x1)} = 1 \quad (4)$$

Each ratio in (4) provides a measure of the parameter in the numerator. Thus, the *rate of time in the best connection* rTiB = DTiB / (x2 – x1), the *rate of time in handoff execution* rTiH = CHoL / (x2 – x1), and the *rate of time in disconnection* rTiD = DTiD / (x2 – x1). Also notice that DTiB, CHoL, and DTiD are numbers bounded by the closed interval [0, (x2–x1)] satisfying (4); and rTiB, rTiH, and rTiD are bounded by [0, 1].

In particular, three measures of handoff performance are DTiB, CHoL, and nEHO. Expressions (1) and (2) show how these measures are mutually related. Nevertheless, we can write a simplified version of such equations as follows. If $\overline{\Lambda}$ is the average latency of a k-layer handoff from a_c to a_b and $\overline{T}$ is the average connection time in a_c, then

$$CHoL = (nEHO)\overline{\Lambda}\,, with\ \overline{\Lambda} > 0. \quad (5)$$

$$DTiB = (nEHO + 1)\overline{T}, with\ \overline{T} \geq 0. \quad (6)$$

$$0 \leq nEHO \leq (x2 - x1)/\overline{\Lambda}. \quad (7)$$

A goal is to minimize CHoL (5) and maximize DTiB (6). Nevertheless, minimizing CHoL involves reducing both nEHO and $\overline{\Lambda}$, and maximizing DTiB involves increasing both nEHO and $\overline{T}$. However, nEHO cannot be increased and decreased simultaneously; hence, CHoL and DTiB are mutually in conflict. Since nEHO is the control parameter for attaining a suitable balance between such conflicting objectives, we have two optimization choices: (a) maximize $\overline{T}$ (for a maximum DTiB) and minimize nEHO (for a minimum CHoL), or (b) minimize $\overline{\Lambda}$ (for a minimum CHoL) and maximize nEHO (for a maximum DTiB). We choose to maximize DTiB and minimize nEHO.

We normalize DTiB and nEHO so that we can compare them. Note that rTiB, the first ratio in (4), is already a normalization of DTiB, thus, we focus on normalizing nEHO. According to (7), nEHO bounds from above and from below, but such interval includes

both *trivial* and *nontrivial handoffs*. Trivial handoffs appear without reason or need, i.e., they are unnecessary handoffs. Thus, we claim the existence of an integer $nEHO_{max}$ that works as the *least upper bound* of (7), such that $0 \leq nEHO \leq nEHO_{max} \leq (x2 - x1)/\overline{\Lambda}$, and $nEHO_{max}$ is the maximum number of nontrivial handoffs that are necessary to make rTiB = 1. In Section 3.8, we explain how to obtain this parameter. Using this relationship, we define the *rate of executed handoffs* rEHO as $0 \leq rEHO = nEHO/nEHO_{max} \leq 1$. For particular scenarios where $nEHO_{max} = 0$, we let rEHO = 0 so that rEHO is defined. We let rEHO as a normalization of nEHO. Since rTiB and rEHO are compromised parameters, we usually study the ordered pair (rTiB, rEHO) as a joint random variable.

The third optimization parameter we study is the number of successful handoffs (nSHO) that occur in a handoff scenario. Let $[T(a_{new})]_{i+1} = (t_c - t_b)$ and $[\Lambda_k(a_{old}a_{new})]_i = (t_b - t_a)$, we say a *handoff succeeds* if (8) is true.

$$([T(a_{new})]_{i+1} \geq [\Lambda_k(a_{old}a_{new})]_i) \wedge$$
$$(\forall t \in [t_b, 2t_b - t_a])\{D_{new}(t) \geq D_{old}(t) + \Delta\,, \Delta > 0\} \qquad (8)$$

Now, we normalize nSHO so that we can compare it with rTiB and rEHO. Since nSHO $\in$ [0, nEHO], then the *rate of successful handoffs* is rSHO = nSHO/nEHO, where rSHO $\in$ [0, 1]. If nEHO = 0, we let rSHO = 1 so that rSHO is defined. A *null scenario* occurs if $(\exists c \in A)(\forall a \in A)(\forall t \in [x1,x2])[D_c(t) \geq D_a(t)]$. Typically, nEHO = 0 in null scenarios since there is no need to make handoffs if the current PoA *c* is always the best one.

We define a Multiobjective Handoff Optimization Problem (MOHOP) by describing optimization objectives and optimization constraints. Let us use the following notation. Let *S* be the universal space of mobility scenarios. For each $s \in S$, there are *random variables* *X*, *Y*, *Z*, defined on *S*, such that $X : S \rightarrow$ rTiB, $Y : S \rightarrow$ rEHO, $Z : S \rightarrow$ rSHO, where $X(s)$,

$Y(s)$, $Z(s)$ represent numerical values of rTiB, rEHO, and rSHO, respectively. The solution space for the bivariate random variable (X, Y) is the unit square represented by $\{(x, y) : 0 \leq x \leq 1, 0 \leq y \leq 1\}$ where $x = X(s)$ and $y = Y(s)$. The solution space for Z is $\{(z) : (0 \leq z \leq 1)\}$ where $z = Z(s)$. Let $S_n = \{s_1, s_2, \ldots, s_n\}$ be a random sample of n scenarios; for each $s_k \in S_n$ we get a solution point (x_k, y_k), (z_k), such that $x_k = X(s_k)$, $y_k = Y(s_k)$, and $z_k = Z(s_k)$.

Optimization Objectives: High rTiB (X) values improve ABC mobility, low rEHO (Y) values improve seamless mobility, and high rSHO (Z) values improve adaptive mobility. Thus, the optimization goal is to maximize $X(s)$, minimize $Y(s)$, and maximize $Z(s)$, $\forall s \in S$. Since $0 \leq x, y, z \leq 1$, for any solution (x, y), (z), the optimum point is (1, 0), (1) and the worst point is (0, 1), (0). In terms of Euclidean distances in the solution spaces for (X, Y) and (Z), the optimization objectives are to *minimize* the next *functions*, simultaneously.

$$dist[(x, y), (1,0)] = \sqrt{x^2 + y^2 - 2x + 1}. \quad (9)$$

$$dist[(x, y), (x + y = 1)] = (x + y - 1)/\sqrt{2}. \quad (10)$$

$$dist[(z), (1)] = 1 - z. \quad (11)$$

Note that (9) and (11) provide *near-optimal* solutions since they minimize the distance to the optimum, and (10) provides a *fair balance* between X and Y since it minimizes the distance to the line of equilibrium ($x + y = 1$). Therefore, by minimizing (9), (10), and (11) we obtain near-optimal and fair-balanced solutions. Moreover, the optimal solution (1, 0), (1) only can be obtained by null scenarios. Nevertheless, null scenarios are not common in reality, since it is rare that a single PoA remains as the best one in the whole scenario.

Optimization Constraints: A solution (x, y), (z) is *unacceptable* if $(x < 0.5 \wedge y > 0.5) \vee z < 0.5$, on the contrary, a solution is *acceptable* if $(x \geq 0.5 \vee y \leq 0.5) \wedge z \geq 0.5$. Since x, y, and z are functions of s, if a solution is acceptable then the underlying scenario s is

acceptable. For random samples of handoff scenarios S_n, with $n > 30$, we expect *relative frequencies* of acceptable solutions over 90%.

To conclude, we stress on the classification we make of this problem as NP-Hard. The MOHOP we stated above characterizes objectives in *conflict* and *nonlinear* functions such as (9) and (10). According to Kumar and Banerjee [15], optimizing conflicting objectives subject to nonlinear constraints is a NP-Hard problem. Therefore, this MOHOP is NP-Hard.

3. Solution Development

We want a computational solution to the prior optimization problem. Hence, we express the optimization problem as a computational problem.

Problem (Seamless-ABC-Adaptive Handoff). *Given S_n, for $n > 30$, we wish a handoff algorithm R with control parameters CP, such that $R(S_n, CP) = (x, y), (z)$ subject to $f[(x \geq 0.5 \vee y \leq 0.5) \wedge z \geq 0.5] > 0.9$, where f[E] is the relative frequency of event E.*

Since the MOHOP is NP-Hard, no algorithm can always produce the optimal solution; but, we can obtain suboptimal solutions within specific ranges of quality. We require optimization techniques [16] that reduce the consumption of resources, and produce fast and acceptable solutions, since the algorithm may run in mobile terminals, where battery loads, processing capacities, and storage capacities are limited resources. Once we have a computational problem, we describe models to develop a computational solution.

3.1. State-Based Handoff Model

In [17, 18, 19], we described a generic handoff control system coordinating the stages before, during, and after the handoff. We review the handoff state diagram in Fig. 3.

The handoff states work as follows. *Disconnection* is the initial state where no available PoAs are present; thus, the terminal stays disconnected from the network or connected to

the null PoA. *Connection* is the state where the terminal connects to the best available PoA; therefore, the communications perform in the best way possible. Disconnection and Connection are final states. *Preparation* state occurs when a *candidate PoA* starts to perform better than the *current PoA*, hence, the terminal prepares for a potential handoff. During preparation, the terminal keeps exchanging data with the network through the current PoA while the handoff algorithm makes crucial decisions.

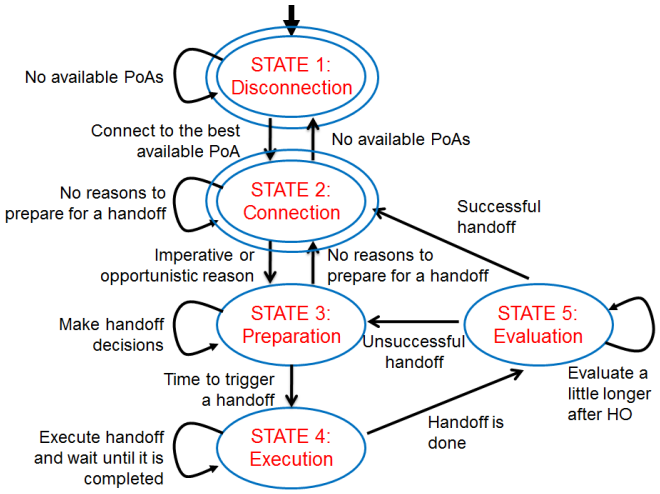

Fig. 3. Generic state-based model for handoff control. Transitions from disconnection to connection to preparation are sequential states that can rollback to previous states. However, transitions from preparation to execution to evaluation cannot rollback.

First, the algorithm determines whether the reason for preparing a handoff is a *necessity* or an *opportunity* (*why*). Second, it selects the *best PoA candidate* from a list of candidates (*where*). Third, it selects a handoff *method* according to the type of handoff in progress, the running application, and the handoff objectives (*how*). Fourth, it decides *who* is going to take the handoff in progress to a final state; this is important since there may be several control managers distributed in the network. Finally, it decides *when* to trigger the actual handoff; this is, perhaps, the most important decision. Once a control manager decides to trigger a handoff, there is no way to roll back the process; it will execute the handoff at the stated time. The *Execution* state performs the actual change of PoA; i.e., the physical and logical disconnection from an old PoA and reconnection to a new PoA. After the handoff,

the *Evaluation* state takes some time to assess the handoff and determine its success or failure. Unsuccessful handoffs could lead to experience a disastrous Ping-Pong effect [20].

3.2. Desirability of PoAs

To model the concept of *best available PoA*, we develop the notion of *desirability*. Desirability represents a measure of how attractive a PoA is at a given time. *Desirability is a utility function that combines multiple variables to produce a single numerical value for one specific PoA at one specific time*. Each variable in the function is *correlated* to a feature of the PoA, such as its performance, quality, preference, cost, energy, security, etc. In [19] we provide a rich set of variables for desirability functions.

PoA desirability is dynamic and dependent on many factors, such as the network operating conditions, the time of the day, the type of running application, the user preferences, the user mobility, the geographic location of wireless overlays, etc. This implies that the best available PoA changes with time, perhaps abruptly and stochastically or perhaps smoothly and deterministically. In fact, the behavior of the best PoA can be both, deterministic or non-deterministic at different times. In addition, users or providers may assign suitable *weights* to each variable in order to show their preferences within the function. Thus, *the best available PoA has the highest desirability value.* We classify decision variables as positive or negative according to their correlation sign. The set V of variables correlated to PoA desirability splits into two disjoint subsets, the set V^+ of positively correlated variables and the set V^- of negatively correlated variables. Examples of positively correlated variables are system bandwidth, signal strength, system security, battery load, and cell size. Increments in these variables produce improvements in PoA desirability and decrements yield degradations in desirability. On the other hand, examples

of negatively correlated variables are bit error rate, rate of lost packets, connection price, transmit power, and distance to the base station. Increments in these variables make the PoA less desirable and decrements yield a more desirable PoA. Therefore, the *desirability function* is a balance between desirability and undesirability.

Definition (Desirability Function). *Let $D_a(\boldsymbol{v}; t_k)$ be the desirability of the attachment point a evaluated at t_k time, where $\boldsymbol{v} = (v_1, v_2, \ldots, v_q)$ is the vector of q variables, which are considered to evaluate the desirability of the PoA. The desirability function is:*

$$D_a(\boldsymbol{v}; t_k) = \sum_i (E + W_i)\, log(v_i^+[t_k]) - \sum_j \left(E + W_j\right) log\left(v_j^-[t_k]\right). \quad (12)$$

W_i and W_j are weights associated with each positively or negatively correlated variable, such that W_i, $W_j \in [0, 1]$ and $\sum W_i = \sum W_j = 1$. E is a constant scaling factor so that "small" changes in variables reflect "big" changes in desirability. Finally, $v_i^+[t_k]$ and $v_j^-[t_k]$ are the values of the positively and negatively correlated variables evaluated at t_k.

The desirability function $D_a(\mathbf{v}; t_k) : \Re^{q+1} \rightarrow \Re$ maps q variables and one parameter control t_k to a single real value representing the desirability of a. We use logarithms as normalization functions, so that we can perform homogeneous operations with heterogeneous variables. For simplicity, we make $D_a(\mathbf{v}; t_k) = D_a(t_k) = D[a, t_k]$ whose *domain* is the time discrete interval $x1 \leq t_k \leq x2$, such that $t_k = x1 + k\delta$ and $0 \leq k \leq \lfloor (x2 - x1)/\delta \rfloor = n$, where δ is the step time at which the desirability function is evaluated, and $(x2 - x1)$ is the scenario length, a.k.a. total sampling time TST.

3.3. Desirability Thresholds

The *range* of desirability is $(-\infty, +\infty)$ but we bound this range with thresholds. Thresholds divide the desirability range into quality regions. Desirability values below a lower threshold L are unable to carry on communications; a PoA in this situation is unavailable or

unreachable. On the contrary, the higher the desirability is above *L*, the better is the PoA. We sustain this condition until an upper threshold *U*, with $U > L$. We consider that any PoA with desirability values above *U* is the best available PoA. Therefore, below the lower threshold (*red* region) and above the upper threshold (*green* region) there is no way to say if a PoA is worse or better than another one is. Any PoA in the red region ($D_a(t) \leq L$) is the worst and any PoA in the green region ($D_a(t) \geq U$) is the best. Only in the handoff (*white*) region, the region between *L* and *U*, it is possible to compare desirability values to decide which PoA is better. Thus, handoffs perform only in the handoff region.

3.4. Desirability Curves

The PoA desirability curves are constructed from a sequence of $n+1$ data points ($D_a(t_0)$, $D_a(t_1)$, …, $D_a(t_n)$) obtained when the desirability function is evaluated at discrete times t_k. A *polygonal curve of desirability* would result from connecting the consecutive points with line segments. Nevertheless, we prefer to use a smooth *desirability fitted curve* $D_a(t)$ as a way of easing data visualization and inferring values of the function where no data are available. Since we have no hard data to create such a curve, we model it with polynomial functions (including roots or quotients of polynomials), transcendental functions (including trigonometric, logarithmic, and exponential functions), or combinations of both. This way, we create a large variety of desirability curves for experimental purposes. Anyhow, our algorithm ignores the mathematical expressions representing the desirability curves.

3.5. Mobility Scenarios

A handoff/mobility scenario is a data structure (N, D, W, L, U, δ) where *N* is the number of desirability curves considered simultaneously in the scenario, $N \geq 2$. *D* is the set of mathematical expressions $D_a(t)$ for $a = 1, 2, \ldots, N$ representing desirability curves. *W* is the

rectangular window bounding the display and analysis of desirability curves; the opposite coordinates (x1, y1) and (x2, y2) determine this window. *L* is the lower threshold. *U* is the upper threshold. δ is the step time (or dot time) to plot desirability curves.

3.6. Time-Based Vs. Space-Based Scenarios

In space-based scenarios, a terminal moves across a service area split by cells and performs handoffs within specific overlap zones (see Fig. 4 left). Correspondingly, in time-based scenarios, the desirability of cells changes with time and the *crossing points* between desirability curves represent times to make handoffs (see Fig. 4 right).

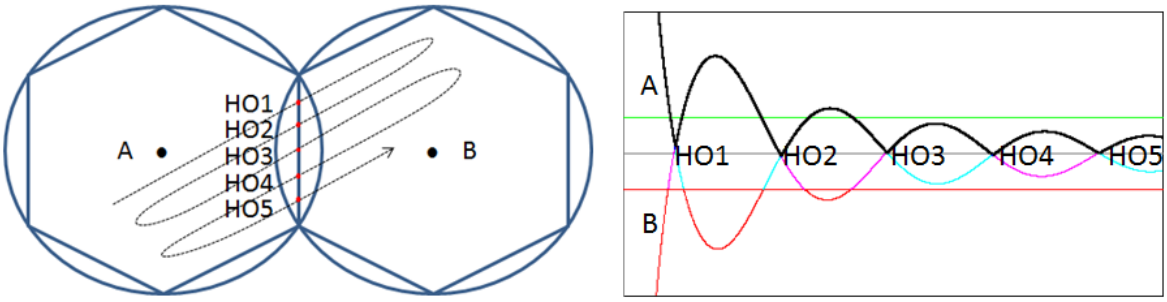

Fig. 4. Space-based and time-based scenarios. The space-based scenario depicts five handoffs performed while a terminal crosses through cells A and B. Similarly, the time-based scenario shows two curves modeling the desirability of A and B as the terminal moves. The crossing points in desirability curves represent different times to perform handoffs. The best available PoA is colored in bold black.

Space-based scenarios display geographic mobility but not desirability changes. Conversely, time-based scenarios show desirability changes but not geographic mobility. Since desirability curves may change even if the terminal is static, we prefer to use time-based scenarios in order to represent handoffs with both, static and mobile terminals.

Note the conflict between nEHO and DTiB. If nEHO is reduced in order to improve seamless mobility, then DTiB will also be reduced yielding degradation in ABC mobility.

3.7. Proactive Vs. Reactive Handoff Strategies

Two types of handoff strategies are proactive and reactive [21]. Fig. 5 depicts proactive and reactive handoffs and distinguishes the spent time in each handoff state.

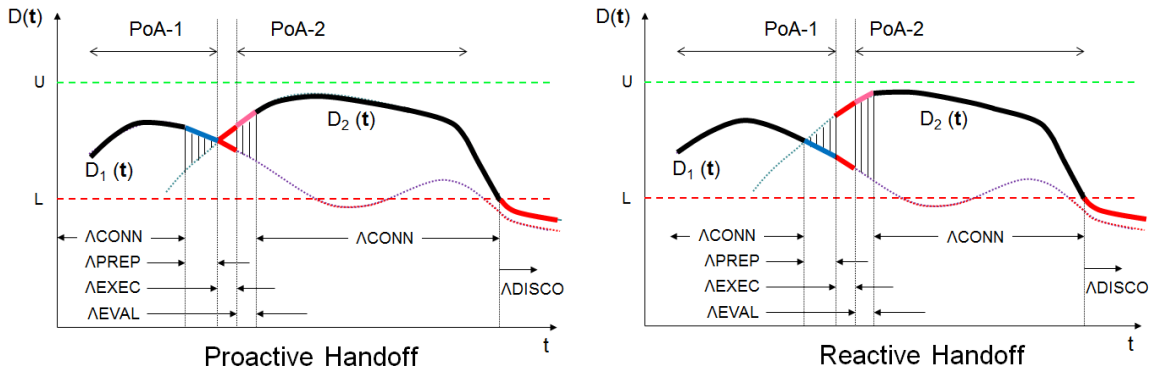

Fig. 5. Proactive and reactive handoff strategies. The proactive strategy triggers a handoff as soon as the best PoA candidate (PoA-2) improves the current PoA (PoA-1), i.e., at the crossing point. The reactive strategy does not trigger a handoff until a PoA candidate (PoA-2) has proven to be consistently and sufficiently better than the current PoA (PoA-1).

The time in connection state is in bold black (ΛCONN). The interval in preparation is in bold blue (ΛPREP). The handoff execution latency (ΛEXEC) and disconnection intervals (ΛDISCO) are in bold red. The handoff evaluation latency (ΛEVAL) is in bold pink.

An advantage of proactivity is that it can provide a better DTiB by initiating handoffs with more anticipation. However, its drawback is that it can produce more unsuccessful handoffs since the candidate PoA never proves to be better than the current PoA; it just proves to have a tendency to improve the current PoA. For this reason, we believe a reactive strategy is more suitable for random desirability curves, improving nSHO at the cost of improving DTiB. Similarly, we think a proactive strategy is more appropriate for deterministic curves, improving DTiB at the cost of improving nSHO. Since we assumed PoA desirability is highly unpredictable, we follow a *reactive* strategy.

Note the conflict between DTiB and nSHO. If DTiB is increased in order to improve ABC mobility, then nSHO will be reduced, yielding degradation in adaptive mobility.

3.8. Handoff Performance Parameters

We provide more details about rTiB, rEHO, and rSHO. First, note that we can split the scenario length (TST) into disjoint time intervals since the handoff state machine (Fig. 3) is

deterministic. Observe that (13) holds for any handoff scenario, where Σ(ΛDISCO) = DTiD, Σ(ΛCONN) is dwelling time in connection, Σ(ΛPREP) is dwelling time in preparation, Σ(ΛEXEC) = CHoL, and Σ(ΛEVAL) is dwelling time in evaluation.

$$TST = \Sigma(\Lambda DISCO) + \Sigma(\Lambda CONN) + \Sigma(\Lambda PREP) + \Sigma(\Lambda EXEC) + \Sigma(\Lambda EVAL). \quad (13)$$

However, DTiB (dwelling time in the best PoA) has two interpretations depending on the handoff strategy: in *proactive* strategy, DTiB = Σ(ΛCONN) + Σ(ΛPREP) + Σ(ΛEVAL); in *reactive* strategy, DTiB = Σ(ΛCONN). In both cases, DTiB represents the time the current PoA is the most desirable attachment point and rTiB = DTiB/TST, rTiB ∈ [0, 1].

Second, $nEHO_{max}$ is the number of crossing points ToX between the current PoA *c* and the best candidate PoA *b*. ToX = |{ $t \in [x1, x2] : D_c(t) = D_b(t)$ }|, considering that *b* becomes *c* after every crossing point. ToX is computed for each scenario under test. A handoff algorithm should not make more handoffs than ToX in order to stay always in the best available PoA. Thus, 0 ≤ rEHO = nEHO/ToX ≤ 1, but if ToX = 0 then rEHO = 0.

Third, the constraint in (8) implies that (ΛEVAL) ≥ (ΛEXEC), that is, a successful handoff will occur if $(\forall t \in \Lambda EVAL)$ $[D_{new}(t) > (D_{old}(t) + \Delta), \Delta > 0]$ is true and Δ is an adaptive hysteresis margin. The rate of successful handoffs in *s* is rSHO = nSHO/nEHO and 0 ≤ nSHO ≤ nEHO. We let rSHO = 1 if nEHO = 0. We say a *scenario is successful* if rSHO ≥ 0.5. Given a random sample S_n, we expect high frequencies of successful scenarios.

Note the conflict between nEHO and nSHO. If nEHO is reduced in order to improve seamless mobility, then nSHO will also be reduced degrading adaptability.

In summary, the bivariate variable (rTiB, rEHO) measures the balance between seamless and ABC mobility, and (rSHO) measures the performance of adaptive mobility. The next proposition gives new insights about the relationship between rTiB and rEHO.

Proposition (Correlation and Causation). *The random variables rTiB and rEHO have correlation, but not necessarily causation.*

The proof begins with the remark that the handoff state machine splits the scenario length according to (13). This equation is rTiD + rTiC + rTiP + rTiH + rTiE = 1, where rTiD = Σ(ΛDISCO)/TST (rate of time in disconnection), rTiC = Σ(ΛCONN)/TST (rate of time in connection), rTiP = Σ(ΛPREP)/TST (rate of time in preparation), rTiH = Σ(ΛEXEC)/TST = CHoL/TST (rate of time in handoff execution), and rTiE = Σ(ΛEVAL)/TST (rate of time in evaluation). Since CHoL = nEHO·$\overline{\Lambda}$ from (5) and nEHO = rEHO·ToX, we have:

$$rTiD + rTiC + rTiP + rEHO\left(\frac{ToX \cdot \overline{\Lambda}}{TST}\right) + rTiE = 1. \quad (14)$$

If we take a reactive strategy then rTiB = rTiC and (15) holds, but if we use a proactive strategy then rTiB = rTiC + rTiP + rTiE and (16) holds.

$$rTiB + rEHO\left(\frac{ToX \cdot \overline{\Lambda}}{TST}\right) + [rTiD + rTiP + rTiE] = 1. \quad (15)$$

$$rTiB + rEHO\left(\frac{ToX \cdot \overline{\Lambda}}{TST}\right) + rTiD = 1. \quad (16)$$

Both (15) and (16) show a correlation between rTiB and rEHO (along with other factors). The correlations seem to be linear, but ToX, $\overline{\Lambda}$, TST, rTiD, rTiP, and rTiE are also random variables, thus rTiB and rEHO have a *nonlinear correlation*. This is empirically confirmed by the correlation coefficient estimated in Section 5.4.

We now explore the *cause-effect* relationship between rTiB and rEHO. According to (15) and (16), one variable along with other factors, determines the other variable. Hence, a change in rEHO does not necessarily produce a change in rTiB, and vice versa. Therefore, we cannot establish a cause-effect relationship between rTiB and rEHO, unless we consider

proactive strategies (16) and scenarios without disconnections (i.e., rTiD = 0). If this is the case, then rTiB and rEHO have *bidirectional causality*.

4. Multiobjective Handoff Algorithm

This algorithm makes a terminal stay most of the time in the best available PoA, while holding the least number of handoffs and the largest number of successful handoffs.

4.1. Algorithm R (Relative Desirability Algorithm)

Algorithm R is *deterministic, reactive, heuristic, adaptive,* and *autonomous*. Deterministic, since it always produces the same output for the same input. Reactive, since it follows a reactive handoff strategy. Heuristic, since it uses deterministic heuristics to decide where and when to hand over. Adaptive, since it changes its behavior according to the case of imperative or opportunist handoffs. Autonomous, since it does not demand user interventions. Control parameters are established offline, and once the user sets an initial performance tune up, no more user interventions are required.

Our heuristics state that only if a candidate PoA is *consistently* and *sufficiently* better than the current PoA, a handoff to that candidate will provide sufficient benefits to the user.

Definition (candidate PoA). *Considering a reactive strategy, if c is the current PoA and b is an available PoA such that* $\Delta R(t_k) = D_b(t_k) - D_c(t_k) > 0$*, then b is a candidate PoA at* t_k*.* $\Delta R(t_k)$ *is called relative desirability.*

Definition (Sufficiently Better). *If c is the current PoA and b is a candidate PoA at* t_k *such that* $\Delta R(t_k) > \Delta > 0$*, then b is sufficiently better (SuffB).* Δ *is called hysteresis margin.*

Definition (Consistently Better). *If c is the current PoA, b is a candidate PoA, and if* $\Delta R(t) = D_b(t) - D_c(t) > 0$ *for all* $t \in [t_p, t_q]$ *where* $(t_q - t_p) \geq SP > 0$*, then b is consistently better (ConsB). SP is a dwell-timer called stability period.*

Definition (Best PoA candidate). *If c is the current PoA, b is a candidate PoA, and if* $\Delta R(t) = D_b(t) - D_c(t) > \Delta > 0$ *for all* $t \in [t_p, t_q]$ *where* $(t_q - t_p) \geq SP > 0$, *then b is the best PoA candidate (i.e., the best PoA candidate is a candidate PoA, which is SuffB and ConsB).*

Heuristic (Where to hand over). *The first best PoA candidate to appear is where the active PoA should go in order to preserve or improve user communications.*

Heuristic (When to hand over). *Trigger a handoff from the current PoA to the best PoA candidate as soon as the first best PoA candidate appears. But, if the current PoA c is close to a disconnection (i.e.,* $L < D_c(t_k) < L + \varepsilon$ *and* $\varepsilon > 0$*) and at* t_k *the best candidate is not yet been found, then make an urgent handoff to the best candidate at* t_k.

Fig. 6 depicts the adaptive behavior for opportunistic and imperative handoffs. Adaptability to opportunistic or imperative handoffs means that the algorithm must automatically vary the values for Δ and SP, according to the vertical distance between the intersection point of desirability signals and the lower threshold *L*. Minimum Δ (mΔ) and minimum SP (mSP) are *configuration parameters* fixed by the user or provider. This way, *preparation latency* (ΛPREP) gradually reduces as the crossing point occurs near *L*, and gradually increases as the crossing point surpasses *L*.

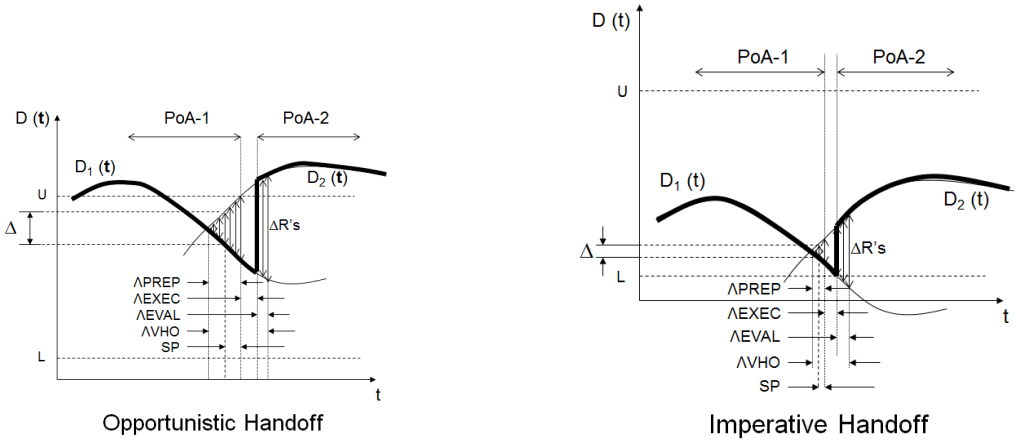

Fig. 6. Opportunistic and Imperative Handoffs. In the opportunistic case, the algorithm tests the candidate under hard conditions (large values for Δ and SP). In the imperative case, the algorithm tests the candidate under mild conditions (small values for Δ and SP).

Opportunistic or imperative handoffs depend on the handoff scenario, thus, the algorithm adapts to different scenarios. To make Δ and SP adaptable to a handoff scenario, we divide the handoff region into a number of adaptability levels of equal size. If we let the space between levels be 0.5, then the number of levels in the handoff region is (*U*–*L*)/0.5, considering $U > L$ and *U*, *L* integers. If the crossing point between the current PoA and a candidate occurs at t_k, then the corresponding adaptability level is given by $level = \lceil (D_{current}(t_k) - L)/0.5 \rceil$; therefore, Δ = (*level* × mΔ) and SP = (*level* × mSP).

4.2. Configuration Parameters

mSP: minimum SP. It is used to obtain an SP value according to SP = *level* × mSP. SP is an adaptive dwell-timer used to determine if a candidate PoA is consistently better. To test consistency for at least one step time, mSP ≥ δ.

mΔ: minimum Δ. It is used to obtain a Δ value according to Δ = *level* × mΔ. Δ is a hysteresis margin used to determine if a candidate PoA is sufficiently better. To test sufficiency, the algorithm requires that mΔ > 0.

ExL: average handoff latency ($\overline{\Lambda}$). Although we may consider *instantaneous handoffs* by making ΛEXEC = 0, in real scenarios, this is not possible. We estimate the average handoff latency by measuring the minimum latency of a layer 1 handoff, then the maximum latency of a layer 4-7 handoff, and then computing the average. We assume this parameter is a real positive number such that ExL ≥ δ.

EvL: average evaluation latency. Evaluation latency is the spent time evaluating the handoff. We consider the average evaluation latency, but meeting EvL ≥ ExL is required.

4.3. Pseudo code of Algorithm R

Inputs: Scenario (*D*[*a*, *t*], *x1*, *x2*, *L*, *U*, δ); Control (*mSP*, *mΔ*, *ExL*, *EvL*).

Outputs: *rTiB*, *rEHO*, *rSHO*, *DTiB*, *TST*, *nEHO*, *ToX*, *nSHO*.

We describe the main process in twenty steps labeled from A to T.

A. [Initialize.] Set *curr* ← 0 (current PoA, 0 = disconnected). Set *best* ← 0 (best available PoA, 0 = disconnected). Set *t* ← *x1* (scenario initial time). Set *DTiB* ← *nEHO* ← *nSHO* ← 0. Set *TST* ← (*x2*–*x1*). Set ToX ← getToX (subroutine that pre-analyzes the given scenario and determines the number of crossing points in the handoff region).

B. [End of analysis?] If *t* > *x2*, the algorithm terminates; answers are:

    a. *rTiB* ← *DTiB*/*TST* (*rTiB* ← 0 if *TST* = 0)

    b. *rEHO* ← *nEHO*/*ToX* (*rEHO* ← 0 if *ToX* = 0)

    c. *rSHO* ← *nSHO*/*nEHO* (*rSHO* ← 1 if *nEHO* = 0)

C. [Get best PoA and its region.] Set *D[a, t]* ← *max(D[1, t], D[2, t], …, D[N, t])*. Set *best* ← *a* (the best available PoA). Set *regionB* ← "white" if $L \le D[best, t] \le U$. Set *regionB* ← "red" if $D[best, t] < L$. Set *regionB* ← "green" if $D[best, t] > U$.

D. [Is current PoA disconnected or connected to the best one?] If (*curr* = 0 OR *curr* = *best*) then if (*regionB* = "red") then *curr* ← 0 (disconnect if no PoA available) else *curr* ← *best*, *DTiB* ← *DTiB* + δ (connect to the best one or remain connected to the best one, increment DTiB). Next, set *t* ← *t* + δ, and then return to step B.

E. [Find region of current PoA.] Set *regionC* ← ("white" | "red" | "green") if ($L \le D[curr, t] \le U$ | $D[curr, t] < L$ | $D[curr, t] > U$).

F. [Is current PoA in "green" region?] If (*regionC* = "green") then set *DTiB* ← *DTiB* + δ, *t* ← *t* + δ, and go back to B. (Any PoA in the green region is the best one).

G. [Is current PoA in "red" region?] If (*regionC* = "red") then set *curr* ← 0, *t* ← *t* + δ, and go back to step B. (Any PoA in the red region is unavailable or disconnected).

H. [Get *level*, Δ, and SP.] (The current PoA is in the handoff region and it is not connected to the best one, thus prepare for handoff.) Set *level* ← $\lceil (D[curr, t] - L)/0.5 \rceil$, Δ ← *level* × *mΔ*, and *SP* ← *level* × *mSP*. Set timers: *t1* ← *tk* ← *t*.

I. [Get relative desirability at tk.] Set *ΔR(tk)* ← (*D[best, tk] – D[curr, tk]*).

J. [Is ΔR(tk) < 0?] If *ΔR(tk)* < 0 then set *DTiB* ← *DTiB* + δ, *t* ← *tk*, *t* ← *t* + δ, and go back to B. (There are no reasons to prepare for a handoff).

K. [Do we still have time to compare?] If (*D[curr, tk] – L*) < ε then go to step N. (The current PoA is quite close to a disconnection, thus trigger a handoff urgently).

L. [Is ΔR(tk) < Δ?] If *ΔR(tk)* < *Δ* then set *tk* ← *tk* + δ, *suffB* ← **false**, and return to I. (The candidate is not sufficiently better). Otherwise, set *suffB* ← **true**.

M. [Is (tk – t1) < SP?] If (*tk – t1*) < *SP* then set *tk* ← *tk* + δ, *consB* ← **false**, and return to I. (The candidate is not consistently better). Otherwise, set *consB* ← **true**.

N. [Initiate handoff execution.] (The candidate is sufficiently and consistently better than current). Set *nEHO* ← *nEHO* + 1, *t1* ← *tk*. (Increase nEHO and start timer).

O. [Make handoff from current to best.] Call procedure *make-ho* (*curr*, *best*). (Execute the physical and logical handoff and wait until it terminates).

P. [Is handoff completed?] If (*tk – t1*) < *ExL* then set *tk* ← *tk* + δ and return to O.

Q. [Initiate handoff evaluation.] Set *new* ← *best*, *old* ← *curr*, *t1* ← *tk*. (Start timer).

R. [Perform handoff evaluation.] If *D[new, tk] > (D[old, tk]* + Δ) then *SHO* (successful handoff) ← **true**; else *SHO* ← **false** and *tk* ← *t1* + *EvL* (evaluation ends).

S. [Is evaluation complete?] If (*tk – t1*) < *EvL* then set *tk* ← *tk* + δ, return to R.

T. [Is handoff successful?] If *SHO* is true, then set *nSHO* ← *nSHO* + 1, *t* ← *tk*, *DTiB* ← *DTiB* + δ, *t* ← *t* + δ, and go back to step B; otherwise, set *t* ← *tk*, *t* ← *t* + δ, and go to H.

Algorithm R follows the handoff state machine in Fig. 3. We associate *disconnection* when *curr* = 0, *connection* when *curr* = *best*, *preparation* when *curr* ≠ *best*, *execution* when *best* is *suffB* and *consB*, and *evaluation* when *new* and *old* are compared. Notice that Δ and SP change according to an adaptability level, thus they modify the conditions for sufficiency and consistency, which the algorithm uses to trigger a handoff. Step K performs an *urgent handoff*. In this step, the algorithm skips the triggering conditions when there is no time to validate a candidate. This option follows the heuristic: *Better to make handoffs to non-validated candidates, than keep connected to PoAs that are losing connectivity*.

5. Simulation Results and Discussion

In this Section, we evaluate the algorithm's performance using a simulation tool that easily creates a variety of time-based handoff scenarios.

5.1. Handoff Simulation

Algorithm R directly works with a blend of time-domain desirability signals, one signal per available PoA. Hence, we need a simulation tool to create a variety of time-based scenarios, run algorithm R under such scenarios, and measure performance parameters rTiB, rEHO, and rSHO for each scenario. For these reasons, we built a simple simulation tool, which we are able to share with the research community to verify our results or simulate particular scenarios. This simulator graphically displays the behavior of the handoff algorithm. The simulator outcomes can be filed at the end of each session.

Fig. 7 depicts an example scenario where four crossing points are present, but only the first one meets the triggering conditions. This shows how our algorithm prevents handoffs towards temporarily better or not sufficiently better networks. This example scenario achieves a solution (0.6237, 0.25) (1.0), which we consider is a very good solution.

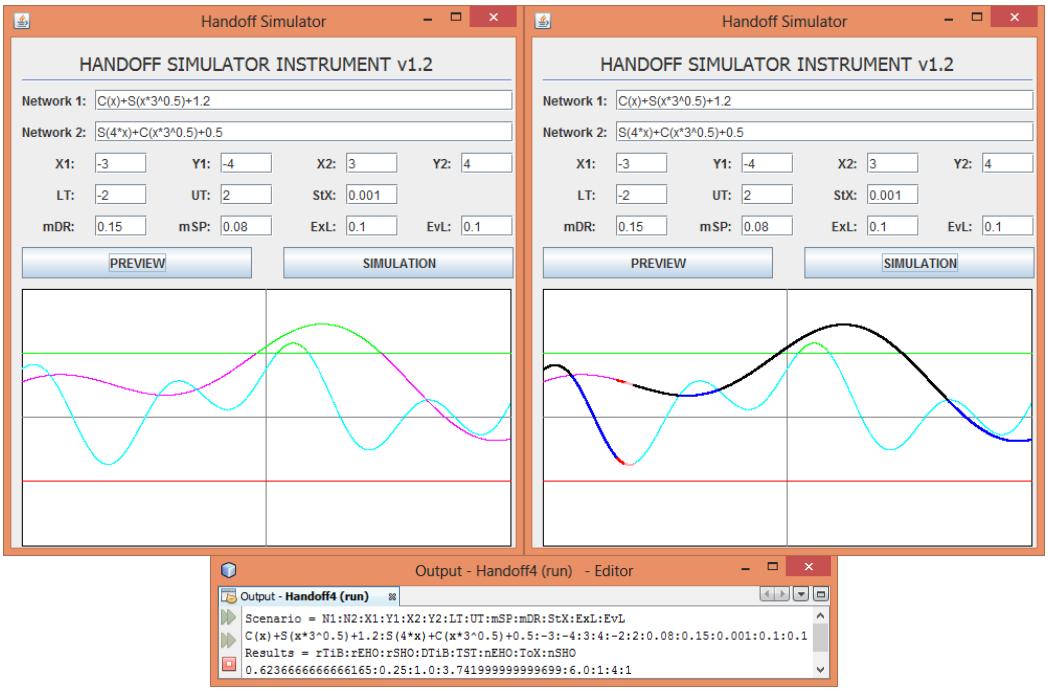

Fig. 7. Handoff Simulator Instrument. (Top left) The screen shows the scenario in preview mode. (Top right) This screen shows the scenario in simulation mode. (Bottom center) This screen shows the console output.

The simulator works in *preview* or *simulation* mode. Preview mode allows visualizing the time-based handoff scenario and setting the algorithm's control parameters up. This mode counts the number of crossing points in the scenario. Simulation mode displays the current PoA passing through every handoff state. The simulator draws the connection state in *black*, preparation in *blue*, evaluation in *pink*, execution and disconnection in *red*.

5.2. Statistical Experiment

Here is the way we chose S_n. We invited several users to try a session test with the handoff instrument. We asked each of them to create at least 30 *random* scenarios experimenting with different desirability functions. We maintained the same experimental conditions between tests, by not allowing one result to influence the way the user creates the next scenario; for this purpose, we banned the user to see the console output. We requested more than 30 samples per user because we observed that after 30 points, the frequency distribution begins to show an identifiable *statistical regularity*.

5.3. Statistical Results

Let us show the distribution of 249 sample points (x_k, y_k), (z_k) collected in the experiment. Appendix B.2 of [12] presents the original data files of this statistical experiment. Fig. 8 (left) shows the distribution (rTiB, rEHO) and Fig. 8 (right) the distribution (rSHO).

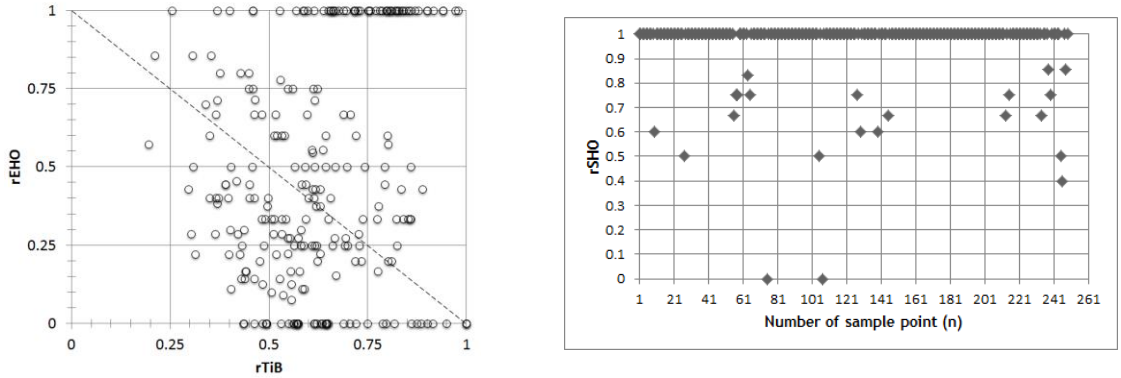

Fig. 8. (Left) Scatter plot of sample points (x_k, y_k). (Right) Scatter plot of the corresponding sample points (z_k). The average sample point is located at (0.6185, 0.4703), (0.9655). The following frequencies apply to existing events: $n[z=0] = 2$, $n[z=0.4] = 1$, $n[z=0.5] = 3$, $n[z=0.6] = 3$, $n[z=0.67] = 4$, $n[z=0.75] = 6$, $n[z=0.83] = 1$, $n[z=0.86] = 2$, $n[z=1] = 227$.

Additionally, Table I presents a summary of hit rates for particular events: acceptable solutions, successful scenarios, very good solutions, and harmful solutions. These results provide empirical evidence that support the algorithm performance.

TABLE I
SUMMARY OF RESULTS & GOALS ACHIEVEMENT

Experiments / Events [E]	249 random sample points $f[E] = n[E]/n$	Performance Goals
Acceptable solutions $(x \geq 0.5 \vee y \leq 0.5) \wedge z \geq 0.5$	225/249 = **90.36%**	**> 90%**
Successful scenarios $z \geq 0.5$	246/249 = **98.79%**	**> 95%**
Very good solutions $(x \geq 0.5 \wedge y \leq 0.5 \wedge z \geq 0.5)$	110/249 = **44.18%**	**> 40%**
Harmful solutions $(x < 0.5 \wedge y > 0.5 \wedge z < 0.5)$	0/249 = **0%**	**< 1%**

The tendency to produce balanced solutions is seen in Fig. 8 (Left), where the densities of solution points that appear above and below the line of equilibrium are nearly symmetrical. Likewise, the tendency to produce near optimal solutions is seen through high rates of acceptable solutions (90.36%) and successful scenarios (98.79%).

5.4. Discussion

Fig. 8 (Left) shows some points having the same value in rEHO but different value in rTiB. This means that rEHO is affected not only by rTiB, but also by other factors. This confirms the lack, in general, of a causal relationship between rTiB and rEHO. Moreover, the nonlinear correlation of rTiB and rEHO in (15) is confirmed since the Pearson's correlation coefficient applied to data sets (rTiB, rEHO) is 0.1247, which is far from linearity (±1).

The simulation results support the optimization goals that we established for algorithm R. Moreover, algorithm R provides acceptable solutions in *linear time*, since the scenario length linearly bounds the execution time of the algorithm. Conversely, the algorithm shows a tendency to decrease the density of solutions as they approach to the optimum; e.g., $f[(x \geq 0.75 \wedge y \leq 0.25 \wedge z \geq 0.75)] = 15/249 = 6.02\%$. This behavior is expected since the problem is NP-hard and heuristic optimization produces suboptimal solutions.

By selecting a reactive handoff strategy instead of a proactive strategy, we are giving preference to rSHO over rTiB. As a result, high rates of successful scenarios (>98%) and low rates of harmful solutions (<1%) were obtained. However, if we changed to a proactive strategy, rTiB will improve since rTiB for proactive is greater than rTiB for reactive, but rSHO will surely get worse. On the other hand, rEHO is controlled directly by the triggering conditions for consistency (mSP), sufficiency (mΔ), and urgency (ε). These conditions are designed to execute handoffs only when necessary. From (16), we expect that proactive handoffs yield better control for rEHO and rTiB with *bidirectional causation*.

Finally, we propose (14) as the *equation of state* for handoffs. This equation correlates multiple handoff variables, which are independent of the handoff algorithm configuration parameters. Hence, this equation can be applied to any type of 5-states handoff scheme.

6. Previous Work

Far from being a comprehensive review on multiobjective handoff optimization, this Section focuses on works that motivated and loomed the concept of multipurpose mobility. The vast literature on mobility and handoff management reveals an exhaustive work on single-purpose mobility services, but also a remarkable gap in multipurpose mobility. Many single-purpose solutions, such as [5-10, 22, 23], perform rather well or even optimally, since they focus on single objectives and ignore conflicts and tradeoffs with other mobility services. It would be unfair and guileful to compare algorithms that optimize different objectives. At present, there is no an algorithm that optimizes the same three objectives achieved by algorithm R, thus, we cannot conduct a fair comparative study. However, we expect this work serves as a template or blueprint to create new MOHOPs and better multiobjective handoff algorithms.

The multipurpose mobility vision is inspired by the remarkable work of Tripathi [24] and Nasser [11]. Tripathi (1997) was the first author to consider handoffs that may achieve multiple desirable features. Nasser (2006) extended this list of desirable handoff features. Following the methodology in [17], we associate a mobility service with a desirable feature, which associates with a purpose, which associates with objectives, which are subjects to goals or constraints. In this way, the integration of multiple mobility services naturally leads to formulate MOHOPs. Finally, some works propose specific techniques to trade off conflicting objectives such as minimize *discovery latency* and *discovery energy consumption* [10], maximize *throughput* and *fairness* in channel assignment [25], maximize *throughput* and minimize *ping pong effect* [26], balance *overall load* and maximize *battery lifetime* [27], etc. Yet, these works are still ignoring the integration of multiple mobility services into a single solution.

7. Conclusion

Handoffs are mechanisms to support the quality of mobile communications. They are inevitable and are not yet optimized to achieve many objectives. Present handoffs have focused on providing single-purpose mobility services; nevertheless, the Future Internet demands a paradigm shift towards multipurpose mobility. The major challenge of multipurpose mobility is the optimization of multiple conflicting objectives subject to nonlinear constraints. Although this problem is NP-Hard, we showed that computational solutions are able to yield near-optimal and fair-balanced outcomes in polynomial time.

In this paper, we integrate seamless mobility, ABC mobility, and adaptive mobility, where the optimization variables nEHO, DTiB, and nSHO are mutually in conflict. The multiobjective handoff algorithm we propose is based on deterministic heuristics. We define acceptable solutions and successful scenarios, and obtain statistical data supporting the hypothesis of an algorithm with hit rates over 90%.

As future work, we are improving the simulation tool in different ways, e.g., it will automatically create thousands of random handoff scenarios with many overlapped desirability curves. We are also working on improvements to our current handoff algorithm. We are exploring new heuristics to increase the current rates of acceptable solutions. A key challenge of future work is to add new objectives of different services to the problem.

References

[1] M. Conti, S. Chong, S. Fdida, W. Jia, H. Karl, Y. D. Lin, P. Mähönen, M. Maier, R. Molva, S. Uhlig, and M. Zukerman, "Research challenges towards the future Internet," *Comput. Commun.*, 34, pp. 2115-2134, 2011.

[2] I. F. Akyildiz, J. Xie, and S. Mohanty, "A survey of mobility management in next-generation all-IP-based wireless systems," *IEEE Wirel. Commun.*, pp. 16-28, Aug. 2004.

[3] M. Zekri, B. Jouaber, and D. Zeghlache, "A review on mobility management and vertical handover solutions over heterogeneous wireless networks," *Comput. Commun.*, 35, pp. 2055-2068, 2012.

[4] X. Yan, Y. A. Sekercioglu, and S. Narayanan, "A survey of vertical handover decision algorithms in fourth generation heterogeneous wireless networks," *Comput. Netw.*, 54, pp. 1848-1863, 2010.

[5] M. Satyanarayanan, M.A. Kozuch, C.J. Helfrich, and D.R. O'Hallaron, "Towards seamless mobility on pervasive hardware," *Perv. & Mob. Comput.*, 1, pp. 157-189, 2005.

[6] J. M. Kang, H. T. Ju, J. W. K. Hong, "Towards autonomic handover decisions management in 4G networks," in *Proc. MMNS*, 2006, pp. 145-157.

[7] R. Langar, N. Bouabdallah, R. Boutaba, and B. Sericola, "Proposal and analysis of adaptive mobility management in IP-based mobile networks," *IEEE Trans. Wireless Commun.*, Vol. 8, No. 7, pp. 3608-3619, Jul. 2009.

[8] M. Louta and P. Bellavista, "Bringing always best connectivity vision a step closer: challenges and perspectives," *IEEE Commun. Mag.*, pp. 158-166, Feb. 2013.

[9] A. Izquierdo and N.T. Golmie, "Improving security information gathering with IEEE 802.21 to optimize handover performance," in *Proc. ACM MSWiM*, 2009, pp. 96-105.

[10] F. Siddiqui and S. Zeadally, "An efficient wireless network discovery scheme for heterogeneous access environments," *Intl. J. Perv. Comp. Comm.*, Vol. 4, No. 1, pp. 50-60, 2008.

[11] N. Nasser, A. Hasswa, and H. Hassanein, "Handoffs in fourth generation heterogeneous networks," *IEEE Comm. Mag.*, pp. 96-103, Oct. 2006.

[12] F. A. Gonzalez-Horta, "Cognitive handoff and mobility for the Future Internet: modeling and methodology," Ph.D. dissertation, Dept. Electronics, INAOE, Puebla, Pue., Mexico, May 25, 2012.

[13] H. Tuncer, S. Mishra, and N. Shenoy, "A survey of identity and handoff management approaches for the future Internet," *Comput. Commun.*, 36, pp. 63-79, 2012.

[14] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: vision, applications and research challenges," *Ad Hoc Netw.*, 10, pp. 1497-1516, 2012.

[15] R. Kumar and N. Banerjee, "Multiobjective network topology design," *Appl. Soft Comput.*, 11, pp. 5120-5128, 2011.

[16] P. N. Ngatchou, A. Zarei, W. L. J. Fox, and M. A. El-Sharkawi, "Pareto multiobjective optimization," in *Modern Heuristic Optimization Techniques*, Ch. 10, K. Y. Lee, M. A. El-Sharkawi (Eds.), IEEE Press, Hoboken, New Jersey: John Wiley & Sons, Inc., 2008, pp. 189-207.

[17] F. A. Gonzalez-Horta, R. A. Enriquez-Caldera, J. M. Ramirez-Cortes, J. Martinez-Carballido, and E. Buenfil-Alpuche, "Towards a cognitive handoff for the future Internet: Model-driven methodology and taxonomy of scenarios," in *Proc. IARIA COGNITIVE*, 2010, pp. 11-19.

[18] F. A. Gonzalez-Horta, R. A. Enriquez-Caldera, J. M. Ramirez-Cortes, J. Martinez-Carballido, and E. Buenfil-Alpuche, "Towards a cognitive handoff for the future Internet: A holistic vision," in *Proc. IARIA COGNITIVE*, 2010, pp. 44-51.

[19] F. A. Gonzalez-Horta, R. A. Enriquez-Caldera, J. M. Ramirez-Cortes, J. Martinez-Carballido, and E. Buenfil-Alpuche, "A cognitive handoff: Holistic vision, reference framework, model-driven methodology and taxonomy of scenarios," *Intl. J. Adv. Netw. & Serv.*, Vol. 4, No. 3&4, pp. 324-342, 2011.

[20] L. Barolli, F. Xhafa, A. Durresi, and A. Koyama, "A fuzzy-based handover system for avoiding Ping-Pong effect in wireless cellular networks," in *Proc. ICPP-W*, 2008, pp. 135-142.

[21] I. Al-Surmi, M. Othman, and B. M. Ali, "Mobility management for IP-based next generation mobile networks: Review, challenge and perspective," *J. Netw. Comput. Appl.*, 35, pp. 295-315, 2012.

[22] L. Ni, Y. Zhu, B. Li, and Q. Deng, "Optimal Mobility-aware Handoff in Mobile Environments," in *Proc. IEEE 17th ICPADS*, 2011, pp. 534-540.

[23] C. P. Lin, H. L. Chen, J. S. Leu, "A Predictive Handover Scheme to Improve Service Quality in the IEEE 802.21 Network," *Comput. & Elect. Eng.*, 38, pp. 681-693, 2012.

[24] N. D. Tripathi, "Generic adaptive handoff algorithms using fuzzy logic and neural networks," Ph.D. dissertation, Virginia Polytechnic Institute and State University, August 21, 1997.

[25] J. Rezgui, A Hafid, R.B. Ali, and M. Gendreau, "Optimization model for handoff-aware channel assignment problem for multi-radio wireless mesh networks," *Comput. Netw.*, 56, pp. 1826-1846, 2012.

[26] A. Singhrova and N. Prakash, "Adaptive vertical handoff decision algorithm for wireless heterogeneous networks," in *Proc. IEEE HPCC*, 2009, pp. 476-481.

[27] S. Lee, K. Sriram, K. Kim, Y.H. Kim, and N. Golmie, "Vertical handoff decision algorithms for providing optimized performance in heterogeneous wireless networks," *IEEE Trans. Vehic. Tech.*, Vol. 58, No. 2, pp. 865-881, Feb. 2009.