# Mobility Support in User-Centric Networks

Fikret Sivrikaya[1], Stefano Salsano[2], Marco Bonola[2], and Marco Trenca[2]

[1] DAI-Labor, Technische Universität Berlin, Berlin, Germany
[2] Dip. Ing. Elettronica, Università di Roma "Tor Vergata", Rome, Italy

**Abstract.** In this paper, an overview of challenges and requirements for mobility management in user-centric networks is given, and a new distributed and dynamic per-application mobility management solution is presented. After a brief summary of generic mobility management concepts, existing approaches from the distributed and peer-to-peer mobility management literature are introduced, along with their applicability or shortcomings in the UCN environment. Possible approaches to deal with the decentralized and highly dynamic nature of UCNs are also provided with a discussion and an introduction to potential future work.

## 1 Introduction

Todays mobile and wireless infrastructure networks depend on highly reliable network elements connected together with high-quality links to provide global broadband connectivity. Although this architectural approach to building networks has been very successful as manifested by the billions of connected devices, it nevertheless has its drawbacks. CAPEX and OPEX, for example, are rapidly increasing while complexity in operation and management hinder the introduction of novel features. An alternative, unconventional approach to todays mainstream telecommunication standards is to adopt the user-centric networking (UCN) model, which exploits the increasing expansion of wireless access networks in order to deploy autonomic and self-organizing wireless community networks.

An empowered Internet end-user lies at the center of UCN and assists in expanding current network operation to the fringes of Internet through several portable and networked device. UCNs represent a disruption in established Internet communication models in several ways. First, any regular end-user device may behave as a supplier of Internet connectivity and services, and consequently become part of the network. In contrast the "end-to-end" principle, one of the architectural foundation of the Internet, describes a clear splitting between network and end-user systems. Second, UCNs grow spontaneously based on the willingness of users to share subscribed Internet access. Thirdly, connectivity is expected to be intermittent given that UCNs are spontaneously deployed [1].

Mobility support is just beginning to emerge as a topic of research interest in the context of user-centric networking. This paper surveys current literature for architectural designs, protocol elements, and research results that can be employed in user-centric networks. In the process, we follow the evolution from

centralized mobility management that depends on a single mobility anchor point to more distributed, and eventually user-provided, mobility support. We present a new solution called UPMT-DAM, which extends an existing host-based per-application mobility management solution and adapts it to more distributed any dynamic environments. We also identify the key research topics for mobility support in user-centric networks and outline main directions for future work in this area.

## 2 Short Primer on Mobility Management Concepts

The objective of this section is not to present mobility management approaches in detail, but to set the stage for the rest of the paper by introducing the basic concepts and common elements in general mobility management solutions. The reader is referred to the existing literature surveys for a more comprehensive overview on these concepts, e.g., [2] [3] [4].

In general, mobility management solutions try to ensure continuity of network services despite physical location changes of the communicating entities, with little or no disruption to the service.

Chan et al. [5] analyzed current mobility management solutions, categorizing them according to the following criteria:

- *Layer*: Application layer, transport layer, network layer, link layer or cross-layer mobility solutions.
- *Controlling Entity*: Network-controlled mobility, mobile-device-controlled mobility, or a combination of both.
- *Architecture*: Centralized, hierarchical and distributed (fully or partially) mobility approaches.

Considering the *Layer* criterion, *Link layer* mobility is often called *Micro-Mobility* and is related to the change of access point within the same subnet or administrative domain. It is responsible for the establishment of a radio link between the Mobile Node (MN) and the Access Point (AP), while the IP address of the MN remains the same. *Network layer* solutions provide mobility features at IP layer and do not make any assumption on the underlying access technologies. They could be provided at the network side, or both at host and network side. *Transport layer* solutions operate at the level of transport protocols, above the network level, therefore they do not require involvement of network nodes. *Higher level* or *Application layer* solutions are usually host side and allow session continuity without help from the network.

Considering the *controlling entity*, the focus is either on the end user equipment or the network side, mainly determining who takes the handover decisions and possibly sets other mobility related parameters. The common approach in cellular networks is to employ a network-controlled approach, where entities in the operator's core network take the handover decision and are in charge of managing the network resources as needed to execute the handover. In any case, the mechanism clearly works with the involvement of mobile device, in particular for

reporting local network measurements and other context information. The other approach of granting mobile device the mobility control is employed mostly, but not only, in distributed mobility solutions, as we will cover in more detail in the rest of the text.

In terms of the *Architecture* criterion in the given mobility classification, the first approaches to mobility management followed the centralized hierarchical architecture of cellular networks. In a centralized approach, all mapping information for the fixed session identifier and the changing IP address for a mobile node is kept at a centralized Mobility Anchor (MA), which also intercepts and re-routes packets directed to the MN, as depicted in Figure 1.

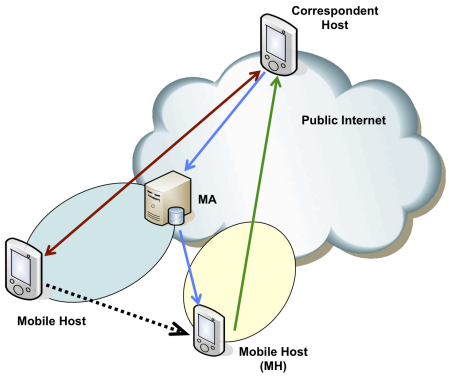


**Fig. 1.** Typical approach for centralized mobility management

The majority of currently proposed solutions and recommendations in mobility management rely on the separation between host identifier(s) and locator(s) [7]. This separation requires an anchor to maintain an association (often referred to as a binding) between the identifier and the locator, a protocol to update this association, and a data transport method between locators. While the last feature can be provided by normal transport protocols or through tunneling, (mobility) anchor management is a key function, since it provides binding management and is tightly coupled to the update protocol, which affects network performance. Depending on scenario and protocol, the mobility anchor can be a centralized entity in the network [8] or follow a distributed approach, involving the collaboration of several mobility anchors [9].

Mobile IPv6 (MIPv6) has been the focus of several studies [10]. Though the MIPv6 specification was designed to cope with only one binding per Home Address (HoA), extensions for multiple Care-of-Addresses (CoA) [11] allow a node to have multiple addresses per interface as well as having multiple interfaces [12]. In UCNs, users are expected to connect to different networks and communities, and thus the capacity to use different addresses can be advantageous, depending on the connection context, sustaining several bindings.

On the other hand, the centralized nature of the Home Agent (HA) in MIPv6 is a limitation for UCNs. Using a central anchor point reduces the signaling between peers, but can introduce overhead when used for bindings and routing. Solutions such as Hierarchical Mobile IPv6 (HMIPv6) [9] [13] and PMIPv6 [14] aim at addressing such performance issues by reducing the required message exchanges during time-critical events, e.g. handovers and bringing the anchor points closer to the node. By introducing the Mobility Anchor Point (MAP), which is a new MIPv6-enabled node located at any level in a hierarchy of routers, the amount of signaling outside the local area is minimized.

Distributed mobility management (DMM) approaches try to address the issues that centralized mobility solutions commonly suffer from. These are identified in [15] and [16] as:

- Low scalability due to the need for new mobility anchor deployment with increasing number of mobile nodes and traffic.
- Per node (and not per flow) mobility support, which may unnecessarily increase the congestion on mobility anchors, since applications not requiring mobility support cannot individually bypass the mobility anchor.
- Single point of failure, since the failure of one mobility anchor may affect many mobile nodes.
- Non-optimal routes, resulting in longer delays and unnecessary load in the core network with respect to more distributed mobility solutions.

The work on distributed mobility management is driven mainly by IETF's DMM Working Group [17], chartered in March 2012, towards standardization in this domain. We will cover DMM in greater depth in Section 4.

## 3 Mobility Management Challenges in User-Centric Networks

The user-centric networking paradigm that we introduced earlier envisions a fully decentralized control of the network, which should be operational without relying on any dedicated network entity. Wireless access points opened up by end users to the UCN community may act as gateways for several services, including Internet access and mobility support, but they should not be deemed as robust or reliable entities in general since they are not dedicated to perform those functions only and their availability may change over time.

Since the access network elements are provided by users, there is a twofold requirement that applies in general to protocol design for UCNs. On one hand there is a need to let users have control of who is using their resources; on the other hand, the solution should take into account the dynamic nature of the network. The structure of a user-provided network would have a high level of dynamicity and mobile nodes should be able to use ANs provided by different operators or by other users, and to dynamically switch among them when needed. In user-centric scenarios, users should be aware of the operating context and be able to take decisions based on their own preferences, which may drive the

requirement that handover procedures are MN based. The MNs need to measure the performance of different networks / ANs and accordingly take proper handover decisions.

We can highlight the main challenges in the design of mobility support mechanisms for user-centric network environments as follows.

- *Coping with highly dynamic environment* - unreliable nodes and links, mobility anchors possibly coming up and going down frequently.
- *Mobility anchor selection* - nodes in the user-centric network having different user and device behavior, e.g. trust level, available resources, mobility pattern, etc. that should affect the selection.
- *Handover decision* - unnecessary handovers and ping pong effects are more likely and should be avoided. Several sources of information are of potential interest in order to take the handover decision: geographical mobility estimation, context info, social mobility aspects, etc.

Those intrinsic characteristics of the UCN environment naturally call for distributed and dynamic approaches also for the mobility management. On the other hand, with the evolution of commercial operator networks towards a flattened all-IP model and the increasing traffic from mobile users, there has already been a strong interest and great deal of research on more dynamic and distributed mobility management approaches, as discussed in [18]. In the next section we will go over the DMM literature and then present a distributed, per-application mobility management solution in detail.

In the dynamic environment of UCNs, the selection of a reliable mobility anchor may be even more crucial than the decision of when to perform a handover, since the anchor selection may directly affect the reliability of connectivity and session continuity. Condeixa et al. introduce a range of scenarios for user-centric networks in [19] and affirm that a proper approach should consider a dynamic and optimized mobility control point distribution according to mobility models and considering network changes. They also identify two main blocks that present major issues to be considered for a user-centric mobility management approach:

- *Binding*: Users already have several devices, each with multi-access capabilities, and this will be even more the case in the future. IP address should not be related to user identification, and instead, used only for location procedures. The binding process should support the association of one user identification to several IP addresses. With this new association, the binding update/maintenance process would also need a reformulation.
- *Forwarding*: The mobility control element in its current form, performing both data plane and control plane functions, should be refactored considering the splitting of these functionalities. Such split would make data forwarding more flexible, since several data plane elements can be placed in different places in the network, providing the possibility of dynamically choosing the best data forwarding point for each MN.

Nascimento et al. further extend these concepts and identify the functional building blocks of mobility management in user-centric networks as device identification, binding mechanism, routing or forwarding, handover negotiation, resource management, and mobility estimation [20].

## 4 Distributed and Dynamic Mobility Management

Although the term *DMM* often refers to *Distributed Mobility Management*, mainly driven by the IETF working group on DMM [17]; it is usually used, implicitly or explicitly, also to capture the concept of *Dynamic Mobility Management*. So we first clarify our interpretation of these two complementary concepts, in line with the IETF definition of DMM. In general, *Distributed* MM contrasts with centralized MM by using a multitude of mobility anchors dispersed in the network, and removing the reliance on centrally deployed anchors to manage IP mobility sessions. Mobility anchors can still be assumed to be mostly fixed and robust in nature within this concept. *Dynamic* MM involves the additional concept of dynamic activation/deactivation of mobility protocol support (i.e. giving mobility services only to users or applications that need it) [21]. In user-centric networks, *Dynamic MM* should also capture the ability to cope with the more dynamic environment characteristics, such as frequently changing topology of anchor nodes as well as their changing resource availability and other contextual properties.

The requirements for distributed mobility management has been recently given in the Internet-Draft [16] by Chan et al. as follows.

1. *Distributed processing*
   DMM solutions must enable distributed processing to avoid traffic traversing single mobility anchor.
2. *Transparency to upper layers when needed*
   Not every application needs a stable IP address, i.e. mobility support. DMM solutions must provide such transparency above the IP layer as needed.
3. *IPv6 deployment*
   DMM solutions should primarily consider IPv6, and not just IPv4, as the target environment.
4. *Existing mobility protocols*
   DMM solutions should consider reusing and extending IETF-standardized protocols before specifying new ones.
5. *Co-existence with deployed networks and hosts*
   DMM solutions must be able to co-exist with existing network deployments and end hosts.
6. *Security considerations*
   DMM solutions should not create new or amplified security risks.
7. *Multicast considerations*
   DMM solutions should enable multicast solutions to be developed to avoid network inefficiency in multicast traffic delivery.

In [22] Bertin et al. propose a dynamic approach based on IPv6 to provide mobility support while keeping traffic as close as possible to the user in the access network. Traffic of a moving user is managed with a tunnel, but the Mobility Agent only takes part in the procedures during the handover. Access points or base stations, termed as *access nodes* in this work, support two distinct functions in this proposal: *AAN* (Anchor Access Node) performs the anchor functionality for MN's IP address traffic on the access point (or base station) to which it has been currently associated and *VAN* (Visited Access Node) is used for delivering MN's traffic sessions with IP addresses not anchored to the current access node. This distinction between AAN and VAN functions allows supporting simple, dynamic and distributed mobility management. Figure 2 illustrates an example where MN's current access node serves both a flow anchored to it (from correspondent node 2) and a flow previously anchored to another access node (from correspondent node 1).

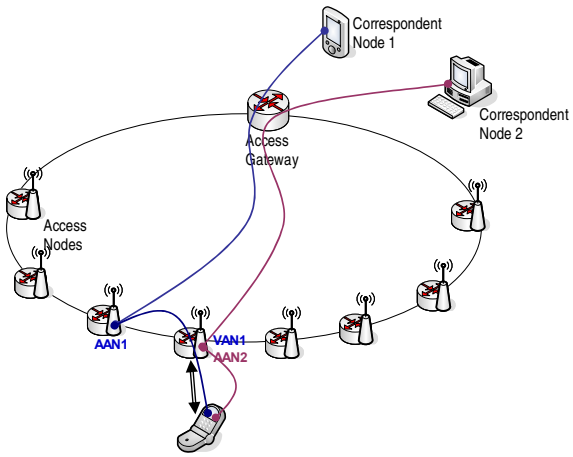


**Fig. 2.** Illustration of distributed mobility management components proposed in [22]

Chan proposes a modified PMIPv6 architecture to distribute mobility anchors over different networks in [23]. The proposal is based on the observation that traffic deriving from signaling is several orders of magnitude lower than that of data traffic. So it is proposed to distribute only the latter one, by replicating the re-routing functionalities on the access network, while keeping the control plane centralized or hierarchical (using some duplicated servers to avoid single points of failure).

A distributed mobility management scheme is presented in [24] based on a flat network architecture, replacing the hierarchical network architecture elements and reducing the signaling cost. Following a clean-slate approach, the locator/id split approach is employed for mobility support by relieving the IP addresses

overload problem, which consists in the fact that in the IP architecture the IP address is used both for identifying an end-point and for locating it.

Xu et al. introduce a new Local Mobility Anchor (LMA) selection algorithm, called Mobile Controlled Movement Tracking (MCMT), for distributed mobility management in IP networks by taking the mobility pattern into consideration for LMA selection [25]. The main idea is to constantly monitor the node mobility and attempt to locate an LMA that is stable and closest to the mobile node, with the objective of providing low latency handovers and load balancing through the selected LMA. The changing mobility characteristics are detected by the algorithm and adapted in discovering new LMAs that are more suitable for the new mobility pattern.

A nice overview of the distributed mobility management literature is given in the recent work [26], which also includes the standardization activities from both IETF and 3GPP perspectives. It is clear that DMM solutions can be exploited well in the UCN context due to the removal of centralized mobility anchor requirement; however, no existing solution is ready to be applied directly as a standalone solution, since there's still the involvement of some fixed entities for the coordination and selection of distributed mobility anchors. Nevertheless, the concepts presented here – the separation of control and data plane, replacement of the hierarchical architecture with a flat one for distributed mobility, keeping traffic closer to the user in the access network, and the LMA selection mechanisms – are all important building blocks towards truly decentralized and dynamic mobility support in user centric networks.

## 5 UPTM-DAM: A Distributed, Per-Application Mobility Management Solution

In this section we present a distributed and dynamic mobility management solution, called UPMT-DAM, which extends the basic UMPT (Universal Per-application Mobility Management using Tunnels) solution introduced earlier in [27]. Our extension of the UPMT solution is mainly towards removing its reliance on a centralized *Anchor Node* (AN), which is usually offered by a provider.

UPMT is based on "IP in UDP" tunneling and provides per-application flow management, i.e. traffic flows of different applications can be independently routed on different access networks. UPMT is well suited for Always Best Connected scenarios [28], whose basic idea relies on the automatic selection "at any time" of the "best" interface for sending and receiving data. It allows performing vertical handovers over different access technologies without session disruption. UPMT acts at application level, without changes on the TCP/IP stacks in the mobile host and is fully compatible with existing network infrastructures. Using UPMT, a host can manage its network flows separately for each application.A set of policies dynamically select the best interface to use for each flow, basing the decision on the availability / quality / cost of the different interfaces. The UPMT mechanism is seen by the Mobile Host (and by the applications therein)

exactly as a NAT service. In principle, all applications that can be run behind NAT boxes can also run using UPMT.

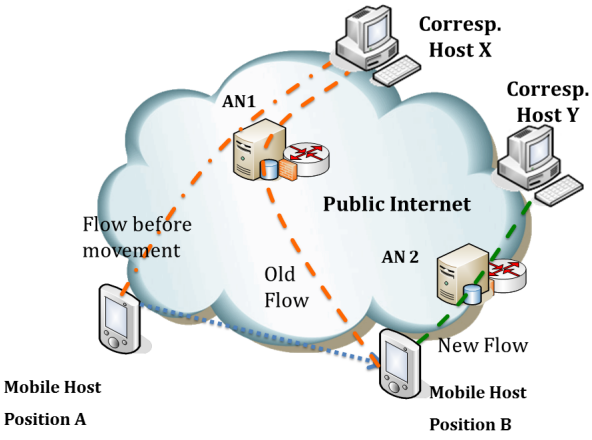


**Fig. 3.** UPMT-DAM, Multi-AN case

In UPMT-DAM, we consider a distributed approach by replicating the centralized anchor point into multiple ANs, potentially at the edges of the network, and allowing the users to select the "best" one for their purposes. Figure 3 depicts the UPMT-DAM multi-AN scenario.

Having multiple ANs creates new issues to face: how to signal their presence in the network, how to select them and how to switch the applications from one to another.

### 5.1 AN Brokering

In order to support multiple ANs, which are supposed to join and leave the UPMT overlay network over time, we introduce a new entity called "AN Broker". This can be a special AN or a dedicated node to which the other ANs signal their presence and capability as they activate UPMT functionalities (see Figure 4).

The AN Broker keeps a constantly updated list of ANs, and provides it to the UPMT clients that request it. When a new AN becomes available, it sends a registration message to the Broker, signaling its presence and features. A Keep Alive function is called to refresh the parameters at regular intervals, allowing to update the AN List. When a Mobile Host (MH) activates the UPMT functionalities, it is unaware of the presence and address of the ANs. The client first connects to an AN Broker (arrow A in Figure 4), sends an "AN List Request" message to the AN Broker and receives the updated list of ANs. From the re-

ceived list of ANs, the MH can select the best AN (or ANs) to connect to (arrow B in Figure 4) and start creating tunnels.

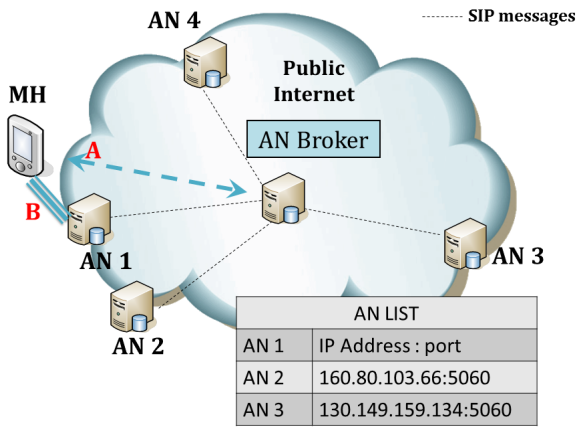


**Fig. 4.** The AN Brokering function

Note that this approach has the same effect of control and data plane separation as in [23] presented in the earlier section; the control plane employs the centralized approach by using the AN Broker, while the data traffic is dispersed over different anchor nodes. The AN Broker functionality can also be decentralized for better applicability in the UCN context, e.g. by a peer-to-peer overlay approach, which we will revisit in a more general context in Section 6.

### 5.2 Interface and AN Selection Policies

The selection of the interface and of the AN to be used is made through policies. A *policy* in UPMT-DAM includes a set of interfaces (or ANs) and the criteria to make the selection among them. The selection of interfaces and ANs is a continuous process, driven by events like interface connections and disconnection or by the updates of performance metrics gathered by the MH. The MH uses a default interface policy for the selection of the interface and a default AN policy for the selection of the AN. Moreover, independently for every application, an interface policy and/or AN policy that overrides the default behavior can be configured. A policy can be based only on information about the availability of a given interface or of an Anchor Node or it can use performance metrics dynamically gathered, for example related to packet delay, packet loss rate, estimates of available bandwidth. When the MH is equipped with several access technologies, it will establish for each interface a tunnel toward the AN that it may want to use for sending and receiving data. Starting from this moment, the MH can select, independently for each application, which tunnel (i.e. which

interface) to use to send packets, using the policy definition. For what concerns the interface selection, the following policies can be independently associated to each application:

- *Block*: The packets of the selected application will not be forwarded in any tunnel.
- *Static*: An interface is indicated and will be used for every packet of the selected application. If not available, the policy will be set as Block.
- *PriorityList*: The user gives a set of interfaces in order of preference. The first interface available on the list will be chosen. If none of them is available, the Block policy is selected.
- *Random*: A random interface between the ones available is selected and used for the application.
- *PerfThreshold*: The user provides two thresholds with the maximum allowed value of RTT and PL and a list of interfaces. The first interface among them that fulfill the requirements will be selected. If no interface fulfills the requirements, the policy will be read as a normal PriorityList.
- *VoIP*: This is a special performance policy that will select the best available combination of RTT and PL for real time applications, based on the Mean Opinion Score (MOS).

According to our previous definition, the first four are "availability based" policies, while the other two are "performance based" policies.

Once the client is connected with more than one AN, we need a mechanism to manage its flows through this multiplicity of ANs. The main objectives we aim at are to balance the load and keep traffic local. If more ANs are available, the MH should select in every moment the best AN, for example considering link-level / IP-level performances.

Currently, there are four kinds of implemented policies for the Anchor Node selection.

- *PriorityList* takes a list of IP addresses given by the user and looks for a match with the IP of the associated ANs. The first one that matches is selected as AN for the given application.
- *Any* takes whatever anchor node is available. In the current implementation, there is a default active anchor node that will be selected by the Any Policy.
- *Static* defines the IP address of an anchor node to connect to. If this address is found among the associated ANs, it will be used.
- *Random* selects randomly between the available anchor nodes.

The system reacts to events like the disconnection of interface or the update of the performance metrics according to the active policies. As far as the interface policies are concerned, the system may re-route the active flows on other tunnels. For example, if a PriorityList interface policy is active, the available interface with the highest priority will be chosen. When such a handover occurs the communication can keep on seamlessly, thanks to the presence of the AN, that hides the change of interface to the correspondent host. In this case we refer

to a "flow-level handover". As far as the AN policies are concerned, the change of the selected AN does not impact existing flows, only applies to the new flows that will be originated by the applications.

### 5.3 Load Balancing and System Reliability

Since the user-centric networking environment may typically involve ANs with limited capacity, we introduce a mechanism to make sure that this limit is not reached, giving ANs the ability to refuse connections. To do this, each AN can set a maximum number of MHs. If the number of associated MHs reaches this value, the AN will refuse all the following associations, until some associated MH leaves.

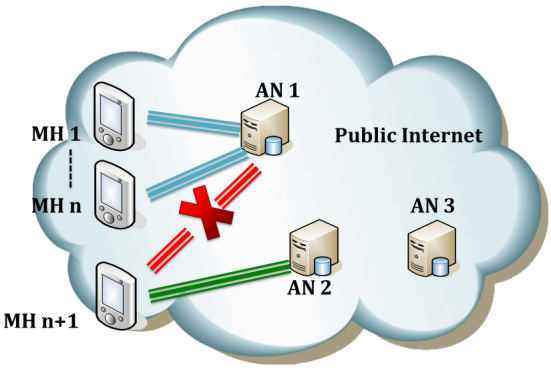


**Fig. 5.** Depiction of a MH refused by an AN

Refusing clients can also be based on AAA considerations, for example refusing "bad" users on a black list, or allowing only users from a given subset to use an AN (e.g. only from Operator A and not from Operator B). In the current implementation, the black-list is simply read from a configuration file, as a list of users SIP IDs to refuse. As shown in Figure 5, if the MH receives a negative reply, it skips the association to that specific AN and tries with the next one on the list, if present.

### 5.4 Handover Management

In the existence of multiple ANs, we also need to take in consideration how to manage handovers, i.e. the change of selected AN for a specific application. We note that under our architecture it is not possible to reroute an active flow, anchored at a given AN, to a different one without session disruption for our

Correspondent Hosts (CH) that are UPMT-unaware. In fact, by changing the AN, the "virtual NAT" operation is made using a different public IP address, and the applications running on the CH will see a different source sending packets. While obviously we can perform seamless handovers across tunnels connected with the same AN, we are not able to perform flow-level AN handovers. The AN handover is a different type of handover that we may refer to as "application-level AN handover". It simply means that an application will change the selected AN to be used by new flows, starting from a given time. As shown in Figure 6, if the old AN remains available, the pre-existing flows will keep going through it while the new ones will rely on to the other AN.

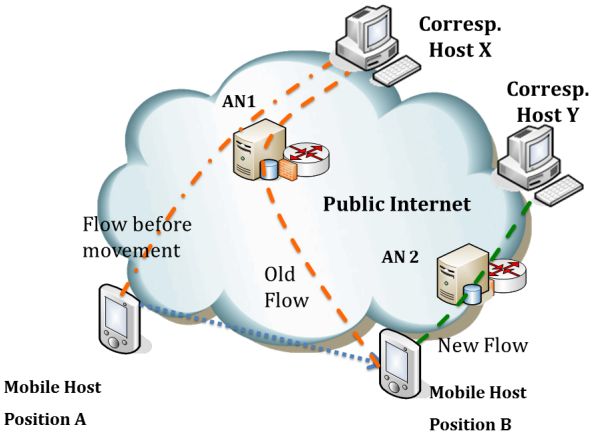


**Fig. 6.** A MH starts using a new AN.

There are several events that could lead to performing an application-level AN handover, and these are processed by the AN policies of the application, which will select the new AN if needed. For example, an application-level AN handover could be recommended when some operating conditions change (e.g. the MH moves or the used AN gets overloaded) and another AN could grant better performance. In this case the AN policies should trigger the application-level AN handovers as defined above. At the current status of our implementation, the AN policy can react to the failure of an AN, or to the occurrence of a new AN higher in the priority list. In order to change the selected AN among two active ones, the defined policy for a given application can also be manually edited. If that happens, the new flows will be routed via the new AN.

### 5.5 UPMT-DAM Deployment

The proposed solution has been fully implemented and the developed components are available under the GPL license [29]. We have deployed UPMT-DAM

in a testbed and performed a set of experiments to verify the functionality of the system. As for the mobile hosts we have implemented a Linux version and an Android version of the UPMT-DAM MH. The Linux version is based on Linux kernel 2.6.35.4-upmt, but porting to later kernels should not be a problem. The Android version has been developed for an HTC Desire HD (ARM Snapdragon S2 processor) with Android 2.3.7 based on CyanogenMod 7.1.0. As physical hosts, we use laptops equipped with built-in Wi-Fi card and 3G USB stick. As for the Anchor Nodes and AN Broker nodes, they are desktop Linux PCs running the same Linux kernel as the MHs. They can be physical Linux hosts or guest Linux virtual machines (VMs) running in a host server. In the experiments we considered two ANs running in two guest virtual machines on a server at Tor Vergata University in Rome and two ANs running on guest virtual machines on a server at TU Berlin. We used as access networks the campus WiFi networks at Tor Vergata, and at TU Berlin the wired campus Ethernet and a public 3G network provided by a network operator. In the testbed, we use netem in order to add delay and loss over given interfaces (or even selectively for given flows) and synthetically recreate impairments that can happen in real networks. For example we can add some delay on a given interface of the MH to simulate the delay on the access network even if the real delay between the MH and the AN is limited as we are performing a simple lab experiment.

We have verified the MH to AN broker interface and the ability of the MH to choose an AN according to the AN policy, to associate with an it and setup the tunnels over all its physical interfaces. Then we have verified the flow-level handovers (among tunnels towards the same AN), both considering "make before break" handovers, where the new tunnels are available before the old ones becomes unusable, and "break before make" handovers in which the connectivity over a tunnel is lost before a new tunnel is available toward the AN.

## 6 P2P for Fully Decentralized Mobility

The last step in distribution of mobility management further to the edge of the network gets its main inspiration from a full peer-to-peer (P2P) networking approach. There is growing interest in P2P overlay networks, which are already used for many purposes as file sharing, gaming, storage and processing applications [30]. Lua et al. [31] made a comprehensive survey on the most used P2P solutions, which are divided into two categories: structured and unstructured. Structured P2P overlay networks do not grow randomly, but follow a controlled pattern, and the contents are placed in an efficient way. The most known structured topologies are Chord [32], CAN [33], Pastry [34] and Tapestry [35]. Structured P2P overlay networks use Distributed Hash Table (DHT) so that every content can be found in a low number of logical hops. This does not ensure that the delay will also be low, as the physical distance between peers could be not so small as the overlay distance. Moreover, since the load is equally shared among all the peers, if some of them are resource-limited there is the risk to create some bottlenecks. The network has to monitor and maintain the

state of its peers' presence with background signaling. If the joining/leaving rate is too high, lookups may fail due to the fact that topology and availability information gets outdated quickly. All these solutions have good reliability and fault-resistance features. A drawback of DHT-based overlay systems is that they can suffer from security issues, in particular when malicious peers participate in the network [36] [37].

In unstructured P2P overlay networks the peers organize themselves in a random topology, in a flat or hierarchical manner without any control. The most famous unstructured overlay networks are Freenet [38], Gnutella [39], FastTrack [40], KaZaA [41] and BitTorrent [42]. The unstructured approach is less efficient as it relies on flooding, random-walks or expanding-ring search, and could reach the time to live before finding a rare content, but provides shorter lookup time for widely replicated contents. Furthermore, flooding searches provide great resistance to the changes of the network when a peer joins or leaves, but generates a high load on the network. In order to avoid this drawback, other solutions employ a hybrid structure [43] or rely on the presence of "super-peers" with more bandwidth and processing-power that make search more efficient on behalf of other peers, as in FastTrack.

Mobility Management could take advantage of the distributed, self-organizing and scalable nature of P2P overlay networks, in order to share the workload over the peers avoiding the problems of a centralized approach, as depicted in Figure 7. Farha et al. [44] propose such a P2P approach that could provide robustness, scalability and availability to the system. The peers virtualize the MA and FA functionalities of Mobile IP and form a mesh network, with a structured topology based on the Chord ring. All mobile nodes connect to the MAs in the Chord ring following a parent/child relationship. In this solution the peers are fixed nodes with enough resources to support more MNs at once. MNs do not participate in the Mobility Management, as they are thought to be resource limited (i.e. battery, bandwidth) but, if needed, they could be promoted to the ring if the capacity of the fixed nodes is exceeded.

Every MN has two identifiers: one permanent (i.e. hash of MAC address), given by a permanent MA (pMA), and one temporary (i.e. hash of new IP address) given by a temporary MA (tMA). As the MN moves, his temporary identifier changes, while the permanent remains constant. A permanent MA keeps track of the visited MAs. In order to avoid longest routing of data packets due to the overlay proximity, there is also a bootstrap MA, chosen on physical proximity criterion. The strong point of this solution is that mobility management is transparent to MNs and CHs, there is no need for tunnels, no triangular routing and no single point of failure. As a drawback, there is the high signaling associated to the lookup of the MN, and to the joining/leaving of MAs.

A similar solution is given by Shou-Chih Lo in [45], which employs a P2P network overlay for the organization of mobility anchors (HAs in Mobile IP). Users' address binding are hashed in the P2P network, which can be queried using P2P lookup mechanisms. The authors rightly argue that it would be impractical to provide a single P2P network for worldwide usage. Therefore they opt for a

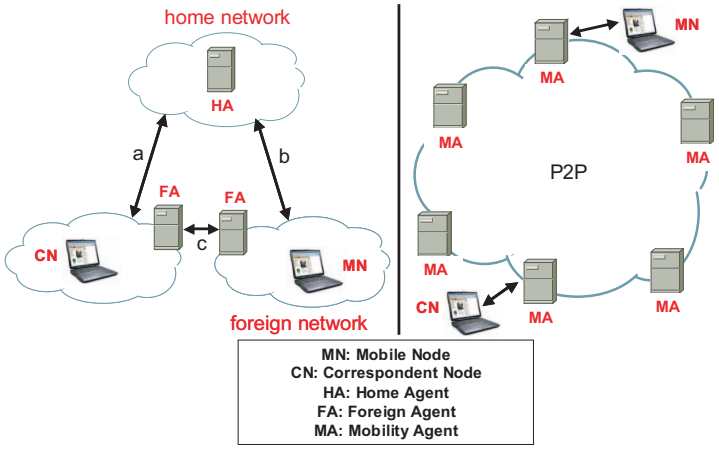


**Fig. 7.** Traditional vs. P2P mobility management [44]

multi-operator model, where each operator constructs their own P2P network. In this multi-operator environment, depicted in Figure 8, each MN belongs to a specific P2P network, which can be queried by correspondent nodes through the DNS. Once the *home P2P network* of the MN is identified through the DNS query, the MN is located through the P2P lookup as in a single operator (i.e. single P2P network) case.

Mobility anchors are organized in a two-level hierarchy, and the signaling remains local unless the MN changes its domain. Domain-level mobility anchors are represented in the figure as GFA (*Gateway Foreign Agent*). A GFA that currently has MN in its service range is selected as a temporary HA of the MN. In order to balance traffic, every HA has the same chances to become the permanent HA of a node, and if it is overloaded, it can refuse the association, which will then be handled by another HA. As peers in the P2P network (HAs) are not supposed to join and leave frequently in this solution, the signaling of binding renewals between MNs and temporary HAs, and between temporary and permanent HAs are performed with different intervals. A caching strategy with a hot list is proposed to speed up the search time of the permanent HA associated to the MN.

In [46] by Gonen et al. mobility is managed by the MNs (the peers) in a transparent way for the network. The authors change the perspective on the MNs, usually seen as resource limited devices, as they consider that recent smart phones have sufficient resources to manage different functionalities, so that they could rely on their own resources for mobility. The MN is supposed to have several access technologies and interfaces with a separate MAC address and protocol stack for each. A distributed lookup server gives information on the position of the peers. Soft handover is performed informing the CN on the change of interface and by multicasting the packets on both until it is concluded. If the CNs are

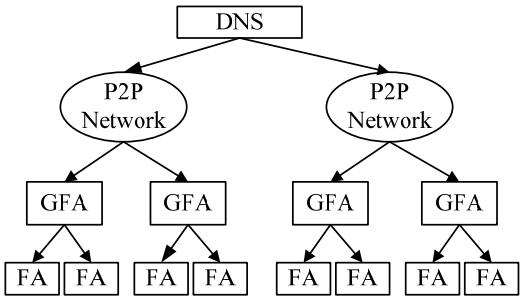


**Fig. 8.** P2P-based mobile IP in multi-operator environments [45]

not provided with such technology, backward compatibility for moving peers is provided by P2P mobility proxies that act as the HA in MIP. As more MNs have these functionalities, the role of proxies decreases. The location management is performed relying on a Lookup Server. This solution has good scalability properties since it relies only on mobile nodes and thus the network does not need an upgrade of certain entities as the number of users grows.

Peer-to-peer approaches may represent an indispensable component for the design of dynamic and decentralized mobility management solutions in user-centric networks. The arrangement of distributed mobility anchors in a P2P fashion, as opposed to their coordination through central or hierarchical controlling entities, better captures the UCN scenario and characteristics. When used in combination with the DMM approaches, P2P-based solutions may provide full decentralization of mobility support entities and functions. An important aspect to consider here is the performance of the solution, since mobility management has typically much more stringent latency requirements than other traditional uses of P2P architectures.

## 7 Conclusion and Future Research Directions

The user centric network vision poses many new challenges to existing networking protocols, including mobility management. We have reviewed those challenges and presented some background work from the relevant literature on mobility management, with a focus on distributed, dynamic, and peer-to-peer based fully decentralized solutions. We have also presented a new solution proposal that can be utilized as a component for mobility support in user-centric networks.

Going over the mobility related state-of-the-art and the specific challenges in user-centric network, we observe that there are many available partial solutions that can collectively be employed and integrated as components of a truly decentralized and user-centric mobility management solution that can cope with the dynamic characteristics of UCNs. In fact, the next paper of this book [47] presents such solution that uses the PMIP solution as a baseline and introduces additional mobility support elements based on distributed mobility management

and peer-to-peer networking concepts, in order to deal with the changing topology and availability of user-provided mobility anchors. However, a through performance analysis, analytical or experimental, of such distributed and peer-to-peer solutions in the UCN environment is still an open research question.

On the other hand, there are new research opportunities in improving the reliability, robustness and performance of mobility management solutions in user-centric networks. Designation of which user-provided network entities should perform anchor point functionality, and which entities should be involved in the decentralized coordination of those anchors are some crucial design considerations with potential for further research. Another important aspect is the dynamic selection and update of the mobility anchor for each flow. Both of these issues become even more interesting and challenging due to the trust and incentive mechanisms that have to be taken into account in UCN protocol design and operation.

The UCN paradigm mainly focuses on the automatic proliferation of user-provided network resources and operation, but the concept does not exclude the involvement of existing centralized network architectures, especially those of commercial telecom operators. For example, the EU Project ULOOP [48] includes, from the beginning, the role of network operators in its original concept and scenarios. This creates a nice mixture of robust and fixed network entities as well as many dynamic, less-reliable user-provided entities, with commercial interests of operators in addition to the trust and incentive aspects in the UCN community. The search for a mobility solution in such scenarios should not only consider the design of a technical solution, but even more the socioeconomic dimension and interoperability aspects of the problem.

Finally, since users are the core elements of the UCN concept, the research on user mobility patterns also becomes a more important dimension for mobility support in UCNs than in legacy network architectures. Mobility estimation with high accuracy could make a big difference for arranging in advance the user-provided network entities and resources for a more seamless mobility support. There is already substantial research in mobility modeling and their implications on various network functions, including mobility management, but the user-centric paradigm provides a different setting where any network entity can potentially be mobile with a variety of mobility patterns. More on mobility estimation aspects in the context of DMM and UCNs can be found in [49].

## References


1. Sofia, R., Mendes, P., Moreira, W., Ribeiro, A., Queiroz, S., Carvalho, L., Jamal, T., Chama, N., Junior, A.: User-provided networks: living-examples, challenges, opportunities. SITI Technical report SITI-TR-11-03, March 2011.
2. Akyildiz, I.F., Jiang Xie, Mohanty, S.: A Survey of Mobility Management in Next-generation All-IP-based Wireless Systems. IEEE Wireless Communications, vol.11, no.4, pp.16-28, Aug. 2004.
3. Saha, D., Mukherjee, A., Misra, I.S., Chakraborty, M.: Mobility Support in IP: A survey of Related Protocols. IEEE Network, vol.18, no.6, pp.34-40, Nov.-Dec. 2004.

4. Bolla, R., Repetto, M.: A Comprehensive Tutorial for Mobility Management in Data Networks. IEEE Communications Surveys and Tutorials, 2013.
5. Chan, H.A., Yokota, H., Xie, J., Seite, P., and Liu, D.: Distributed and Dynamic Mobility Management in Mobile Internet: Current Approaches and Issues. Journal of Communications, vol. 6, no. 1, pp. 4-15, February 2011.
6. Kong, Ki-Sik, et al.: Mobility management for all-IP mobile networks: mobile IPv6 vs. proxy mobile IPv6. IEEE Wireless Communications, 15.2 (2008): 36-45.
7. Li, T.: Recommendation for a Routing Architecture. RFC 6115, February 2011.
8. Perkins, C.E.: Mobile networking through mobile IP. IEEE Internet Computing, 2.1 (1998): 58-69.
9. Soliman, H., Castelluccia, C., ElMalki, K. and Bellier, L.: Hierarchical Mobile IPv6 Mobility Management (HMIPv6), RFC 5380 (Proposed Standard). October 2008.
10. Kong, K.S., Lee, W., Han, Y.H., Shin, M.K., You, H.: Mobility Management for All-IP Mobile Networks: Mobile IPv6 vs. Proxy Mobile IPv6. Wireless Communications 15(2), pp. 3645, 2008.
11. Wakikawa, R., Devarapalli, V., Tsirtsis, G., Ernst, T., Nagami, K.: Multiple Care-of Addresses Registration. Internet-Draft, Work in Progress, October 2009.
12. Ernst, T., Montavont, N., Wakikawa, R., and Kulandinithi, K., Ng C.: Motivations and Scenarios for Using Multiple Interfaces and Global Addresses. Internet Draft, Work in Progress, July 2007.
13. Castelluccia, C.: HMIPv6: A hierarchical mobile ipv6 proposal. SIGMOBILE Mobile Computing and Communications Review, 4(1):4859, 2000.
14. RFC 5213, Proxy Mobile IPv6.
15. Kuntz, R., Sudhakar, D., Wakikawa, R., Zhang, L.: A Summary of Distributed Mobility Management. IETF Internet Draft, Work in Progress, May 2011.
16. Chan H. (Ed.): Requirements for Distributed Mobility Management. IETF Internet Draft, Work in Progress, November 2013.
17. IETF Charter for Working Group DMM. Available online: https://ietf.org/wg/dmm/charter/.
18. Bertin, P., Bonjour, S., Bonnin, J.: Distributed or Centralized Mobility?. IEEE Global Telecommunications Conference (GLOBECOM 2009), Nov. 30 - Dec. 4 2009.
19. Condeixa, T., Matos, R., Matos, A., Sargento, A., Sofia, R.: A new perspective on mobility management: Scenarios and approaches. Second International ICST Conference on Mobile Networks and Management, Sep 2010.
20. Nascimento, A., Sofia, R.C., Condeixa, T., and Sargento, S,: A Characterization of Mobile Management in User-centric Networks. 11th international conference on Next Generation Wired/Wireless Advanced Networking, New2AN 2011.
21. Chan, A.H. (Ed.): Problem statement for distributed and dynamic mobility management. IETF Internet Draft, Work in Progress, March 2011.
22. Philippe, B., Bonjour, S., Bonnin, J-M.: A Distributed Dynamic Mobility Management Scheme Designed for Flat IP Architectures. IEEE New Technologies, Mobility and Security (NTMS'08), 2008.
23. Chan, H.A.: Proxy Mobile IP with Distributed Mobility Anchors. IEEE GLOBECOM Workshops (GC Wkshps), 2010.
24. Yu, L., Zhijun, Z., Tao, L., Hui, T.: Distributed Mobility Management Based on Flat Network Architecture. Proceedings of WICON'2010. pp. 1-6, 2010.
25. Xu, Y., Lee, H.C.J., Thing, V.L.L.: A local mobility agent selection algorithm for mobile networks. IEEE International Conference on Communications, ICC'03, vol.2, pp.1074-1079, May 2003.

26. Zuniga, J.C., Bernardos, C.J., De La Oliva, A., Melia, T., Costa, R., Reznik, A.: Distributed mobility management: A standards landscape. IEEE Communications Magazine, vol.51, no.3, pp.80-87, March 2013.
27. Salsano, S,. Bonola, M.: UPMT Universal Per-application Mobility management using Tunnels. IEEE GLOBECOM 2009, 30 Nov 4 Dec 2009, Honolulu, Hawaii
28. Gustafsson E. et al.: Always Best Connected. IEEE Wireless Communications, February 2003.
29. UPMT home page. Available online: http://netgroup.uniroma2.it/UPMT
30. David, L.-N., Balakrishnan, H., and Karger, D.: Analysis of the evolution of peer-to-peer systems. Proceedings of the twenty-first annual symposium on Principles of distributed computing. ACM, 2002.
31. Lua, E.K. et al.: A survey and comparison of peer-to-peer overlay network schemes. IEEE Communications Surveys and Tutorials 7.1-4 (2005): 72-93.
32. Stoica, I. et al.: Chord: a scalable peer-to-peer lookup protocol for internet applications." Networking, IEEE/ACM Transactions on 11.1 (2003): 17-32.
33. Ratnasamy, S. et al.: A scalable content-addressable network. Vol. 31. No. 4. ACM, 2001.
34. Rowstron, A., and Druschel, P.: Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. Middleware 2001. Springer Berlin Heidelberg, 2001.
35. Zhao, B.Y. et al.: Tapestry: A resilient global-scale overlay for service deployment. Selected Areas in Communications, IEEE Journal on 22.1 (2004): 41-53.
36. Castro, M. et al.: Secure routing for structured peer-to-peer overlay networks. ACM SIGOPS Operating Systems Review 36.SI (2002): 299-314.
37. Wallach, D.S.: A survey of peer-to-peer security issues. Software SecurityTheories and Systems. Springer Berlin Heidelberg, 2003. 42-57.
38. Clarke, I. et al.: Freenet: A distributed anonymous information storage and retrieval system. Designing Privacy Enhancing Technologies. Springer Berlin Heidelberg, 2001.
39. Gnutella, http://en.wikipedia.org/wiki/Gnutella
40. FastTrack, http://en.wikipedia.org/wiki/FastTrack
41. KaZaa, http://en.wikipedia.org/wiki/Kazaa
42. BitTorrent, http://en.wikipedia.org/wiki/BitTorrent
43. Loo, B.T. et al.: The case for a hybrid P2P search infrastructure. Peer-to-Peer Systems III. Springer Berlin Heidelberg, 2005. 141-150.
44. Farha, R. et al.: Peer-to-peer mobility management for all-ip networks. IEEE International Conference on Communications, ICC'06. Vol.5, pp. 1946-1952, June 2006.
45. Lo, Shou-Chih.: Mobility management using P2P techniques in wireless networks. Journal of information science and engineering 23.2 (2007): 421-439.
46. Gonen, E.K., Xu, Hua, Pranav, J.: A Peer-to-Peer architecture for mobile communications. Wireless Communication Systems, 2005. 2nd International Symposium on. IEEE, 2005.
47. Peters, S., Pardo, D.P., and Zhou, Q.: Mobility Management in ULOOP. Springer LNSN Book "User-Centric Networking Future Perspectives", 2014.
48. User-centric Wireless Local-Loop (ULOOP). EU FP7 Project. Available online: http://uloop.eu.
49. Sofia, R., Sargento, A., Condeixa, T.: Mobility Estimation in the Context of Distributed Mobility Management. Springer LNSN Book "User-Centric Networking Future Perspectives", 2014.