ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΠΙΣΤΗΜΗ ΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Διπλωματική Εργασία
Μεταπτυχιακού Διπλώματος Ειδίκευσης

«Smart Caches for Mobility Support in a Publish/Subscribe Network Architecture»

Γιαννάκη Βαρβάρα

Επιβλέπων: Πολύζος Γεώργιος

ΑΘΗΝΑ, 07/2011

Table of Contents

Acknowledgements

First of all, I would like to thank Prof. George C. Polyzos for giving me the opportunity to work with him in the area of the Future Internet Networking. Participating in an ongoing research effort for a clean-slate Internet architecture was a valuable and fascinating experience.

Additionally, Mobile Multimedia Laboratory (MMlab) team, and especially Xenofon Vasilakos, for providing significant directions that assisted me throughout this thesis and sharing brilliant ideas for the research area related to this work.

My friends Christos, Giorgos, Stamos and Argyro for their support and the good moments we enjoyed during this Master's program.

Last, but not least, I want to thank my family for really caring and being next to me all these years supporting my every step.

Vera

Abstract

The fundamental paradigm of the current Internet architecture is a point-to-point, send-receive communication model between two hosts, where the host identifier (the IP address) is the central element. This approach has various drawbacks and has led to DoS attacks and spam while it hinders mobility, multi-homing and intermittent connectivity. Add-on solutions, such as Mobile IP (and related solutions), and CDNs have been introduced to mitigate these problems, even though they violate the original Internet architecture (and often the end-to-end argument).

Throughout this thesis, we investigate mobility in a future Internet architecture based on the publish/subscribe model, called PSI. PSI is an information-centric architecture that manages to decouple content from its location which helps to support mobility. Furthermore, this thesis takes mobility for this information-centric network architecture one step further in order to optimize performance without modifications to the core network architecture or applying ad hoc add-on solutions.

More specifically, we introduce the concept of ***Smart Caches (SC)*** which assists mobility management. SCs are *in-network caches* defined by the Rendezvous Point based on the topology of the network, as well as the current and the predicted (or expected) future positions of subscribers of a particular publication (i.e. data item). A SC acts both as an intermediate caching node and as a mediation assistant in the process of exchanging data between the publisher and the subscriber. This idea is in accordance with the PSI architecture and does not introduce a new component to the initial set of entities of PSI, namely publisher, subscriber and Rendezvous Point (RP). Instead, nodes that are already capable of caching are 'promoted' to also act as publishers and RPs.

We also investigated three different mechanisms for the placement of the SC. It is vital to designate as SC a node that can assist in future movements of the mobile node and enhance the existing mobility performance. The first two algorithms are quite simple and they are not based on topological knowledge of the network. The first one defines as SC the proxy of the mobile node before it moves while the second one places a SC to every AP the mobile node might move. The main algorithm that we propose here takes into account the paths from the publisher to a set of Access Points (APs) and finds the common nodes among these paths. Among these nodes we select the node that best fits most of the APs with the minimum number of hops and if there are more than one such nodes, we then decide to make them all SCs and handle the delivery of data to them from the publisher via a multicast tree.

We show with this work that Smart Caches guarantee seamless delivery of the requested data and we propose various scenarios for Smart Cache placement. Finally, we conclude with the evaluation of this idea and the different Smart Cache placement algorithms, comparing the results

against the basic mobility functionality of the PSI architecture that does not use caching elements.

Greek Abstract

Το βασικό πρότυπο της υπάρχουσας αρχιτεκτονικής του Internet είναι η από σημείο σε σημείο επικοινωνία μεταξύ δύο κόμβων, όπου το αναγνωριστικό του κόμβου (διεύθυνση IP) είναι το κεντρικό στοιχείο. Η προσέγγιση αυτή έχει πολλαπλά μειονεκτήματα καθώς δεν σχεδιάστηκε για να καλύψει τις σημερινές ανάγκες ανταλλαγής δεδομένων. Πολλά είναι τα προβλήματα που προέκυψαν στην πορεία της εξέλιξης του Internet με την υπάρχουσα αρχιτεκτονική, όπως οι επιθέσεις DoS και το spam. Η βασικότερη ωστόσο αδυναμία του είναι η δυσκολία που εισάγει στην κινητή επικοινωνία και στην απρόσκοπτη ανταλλαγή δεδομένων. Λύσεις επιπρόσθετες, όπως τα CDN και το Mobile IP, έχουν εισαχθεί για το λόγο αυτό στην βασική αρχιτεκτονική για να μετριάσουν αυτά τα προβλήματα μολονότι παραβιάζουν την αρχική αρχιτεκτονική και συχνά την σημείο-προς-σημείο επικοινωνία.

Στην διπλωματική αυτή, ερευνήσαμε την κινητή επικοινωνία σε μια νέα προτεινόμενη αρχιτεκτονική για το Internet η οποία βασίζεται στο publish/subscribe μοντέλο, στην αρχιτεκτονική του PSI. Το PSI είναι μια αρχιτεκτονική προσανατολισμένη στην πληροφορία η οποία αποδεσμεύει το περιεχόμενο των δεδομένων που ανταλλάσσεται από την τοποθεσία των χρηστών, γεγονός που εξυπηρετεί ιδιαίτερα την επικοινωνία και ανταλλαγή δεδομένων ενώ οι χρήστες κινούνται. Η διπλωματική αυτή προσπαθεί να επεκτείνει την κινητή επικοινωνία σε αυτή την δικτυακή αρχιτεκτονική έτσι ώστε να βελτιστοποιήσει την απόδοση χωρίς να εισάγει επιπλέον τροποποιήσεις στο κεντρικό μοντέλο του PSI.

Συγκεκριμένα, εισαγάγαμε την ιδέα των ***Smart Cache (SC)*** οι οποίες βοηθούν στην διαχείριση της κινητής επικοινωνίας. Οι SC είναι κρυφές μνήμες (caches) εντός του δικτύου οι οποίες καθορίζονται από το Rendezvous Point (RP) με βάση την τοπολογία του δικτύου καθώς και τις παρούσες και μελλοντικές θέσεις του κινητού χρήστη ο οποίος έχει ζητήσει ένα κομμάτι πληροφορίας. Μία SC ενεργεί τόσο σαν ενδιάμεσος κόμβος προσωρινής αποθήκευσης δεδομένων όσο και ως διαμεσολαβητής στη διαδικασία ανταλλαγής δεδομένων ανάμεσα στον publisher, κάτοχο της πληροφορίας, και στον subscriber, ο οποίος ενδιαφέρεται για την πληροφορία. Η ιδέα των SC συμφωνεί με την βασική σχεδίαση του PSI και δεν εισάγει κάποιο νέο στοιχείο στις βασικές οντότητες του PSI, τον publisher, τον subscriber και το Rendezvous Point. Αντίθετα, κόμβοι του δικτύου που ήδη έχουν τη δυνατότητα να αποθηκεύουν δεδομένα για κάποιο χρονικό διάστημα, προάγονται για να λειτουργήσουν επιπλέον ως publishers και RP.

Διερευνήσαμε επίσης στη διπλωματική εργασία αυτή, μηχανισμούς για την βέλτιστη τοποθέτηση των SC εντός του δικτύου. Είναι καθοριστικής σημασίας για την ιδέα που προτείναμε να οριστεί ως SC ένας κόμβος του δικτύου που να μπορεί να βοηθήσει σε μελλοντικές μετακινήσεις του κινητού χρήστη και να βελτιώσει την υπάρχουσα απόδοση. Οι δύο πρώτοι προτεινόμενοι αλγόριθμοι είναι αρκετά απλοί και δεν βασίζονται τόσο στην τοπολογική γνώση του δικτύου. Ο πρώτος ορίζει σαν SC τον πληρεξούσιο (proxy) του κινητού χρήστη ενώ ο δεύτερος τοποθετεί μία SC σε κάθε

Access Point (AP) που μπορεί να μετακινηθεί ο χρήστης. Ο βασικός προτεινόμενος αλγόριθμος λαμβάνει υπόψη του τα μονοπάτια από τον publisher σε ένα σύνολο από APs και εντοπίζει τους κοινούς κόμβους σε αυτά τα μονοπάτια. Ανάμεσα σε αυτούς τους κόμβους επιλέγουμε αυτόν ο οποίος είναι καλύτερος για το σύνολο των APs ενώ εάν υπάρχουν περισσότεροι τέτοιοι κόμβοι τότε ορίζουμε όλους αυτούς ως SC και χειριζόμαστε την μετάδοση των δεδομένων από τον publisher στον subscriber μέσω ενός δένδρου πολυεκπομπής.

Με αυτή τη διπλωματική αναλύουμε πώς ο μηχανισμός των Smart Cache εγγυάται σε μεγάλο βαθμό την απρόσκοπτη μετάδοση των δεδομένων που ζητά ένας κινητός χρήστης και προτείνουμε διαφορετικές ιδέες για την τοποθέτησή τους μέσα στο δίκτυο. Στο τέλος, αξιολογούμε τον προτεινόμενο μηχανισμό και τους διαφορετικούς τρόπους επιλογής της Smart Cache, συγκρίνοντας τα αποτελέσματα με το βασικό μοντέλο του PSI που δεν χρησιμοποιεί στοιχεία προσωρινής αποθήκευσης.

1. Introduction

The current Internet was initially designed to face a completely different problem than today's communications and was basically intended to serve as a resource sharing network. Thus, End-to-End, point-to-point (E2E) communication came to be a fundamental structural feature of the current Internet architecture. This end-to-end approach along with the specific implementation choices that realize the Internet today, are widely considered as the root causes of many of its limitations and inefficiencies.

Various add-on solutions such as network address translation (NAT), content delivery networks (CDNs), peer to peer overlays and Mobile IP have been proposed and implemented, resulting in the mitigation of some core limitations. However, add-ons tend to either explicitly or implicitly violate several aspects of the original design of the Internet architecture.

Furthermore, the current Internet was not designed to support today's global network of information. Users of the Internet are interested in the information provided rather in the connection with another host. This need to provide access to information and facilitate internet access to increasing number of subscribers has led the research community to reconsider the fundamentals of the current architecture and investigate information oriented architectures.

Publish/Subscribe based solutions continuously gain popularity within the research community as promising candidates for a clean slate content oriented Future Internet (FI) Architecture due to their inherent information-centrism that manages to decouple content from its location. Several research efforts on Future Internet architectures such as the PURSUIT [1, 4], PSIRP [2, 4], and CCN [3] projects investigate Publish/Subscribe as a Future Internet architectural alternative. Unlike the Internet as we know it today, Publish/Subscribe fulfills the needs of today's users who are primarily interested in content, rather than the location that content resides. Further advantages include its inherent support for multicast, its provision of anonymity and in-network processes for matching supplied to desired content. Moreover, Publish/Subscribe is inherently asynchronous and can deal with important security and trust aspects, offering resilience to distributed denial of service attacks (DDoS attacks). Finally, it balances the power between the data exchanging parties, unlike in the common send-receive paradigm applied to today's Internet that empowers the sender, posing security threats and other inefficiencies to the receiver.

The Publish/Subscribe paradigm has a further advantage when it comes to mobile environments [5]. First, anonymity and asynchrony allow such systems to adapt quickly to churn, i.e., to the continuous attachment and detachment of mobile nodes to the Access-Points (AP) of a mobile network. It is common for mobile agents to be turned off or simply be disconnected from their home network for an undefined period of time. Then, they may attach themselves to another AP of the same or even another mobile network. APs reside at the edge of the mobile network with the sole purpose of allowing mobile agents to attach themselves to the mobile network. Note that APs are not necessarily wireless access points; they may refer to any kind of a physical or software mobile layer, including wireless environments such as IEEE 802.11 and UMTS, or even to a software agent that migrates to another host, perhaps in another network.

Second, the multicast nature of Publish/Subscribe manages to successfully deal with a series of issues in mobile networks. Such issues include the potentially large scale of mobile agents; their tendency towards continuously detaching and reattaching themselves either gracefully or unexpectedly; the limited capabilities of mobile agents (e.g. short battery life or limited bandwidth), etc. The multicast nature of publish/subscribe helps such systems scale and utilize the resources of each agent in a more efficient way. For instance, fewer (re)transmissions lead to energy savings and gains in bandwidth utilization.

Therefore, and with respect to the increasing popularity of mobile networks, we justify the need for employing the publish/subscribe paradigm inside the very architecture of mobile networks. The Internet of today does not facilitate mobility. Its end-point centrism and the location dependency of the IP (Internet Protocol) hinder even the add-on solutions currently in use – Cellular IP being the most notable. What we propose with this thesis is an on-going effort towards a publish/subscribe mobile architecture. Our effort is undertaken within the context of the PURSUIT project, an EU-FP7 funded research project that investigates a clean-slate, information-oriented Future Internet architecture. The goal of this work is to seamlessly integrate mobility to a clean slate Publish/Subscribe Internet (PSI) architecture as envisioned in PURSUIT and PSIRP.

In the following chapters of this thesis we first discuss some notable papers regarding publish/subscribe networks in the related work section; then we present an overview of the PSI architecture, its principles and its main functions that differ from current architectures. We proceed by explaining the insights of the mobility support in PSI, outlining a micro-mobility scenario for a mobile network that functions along the lines of PSI and supports in-network caching. We then present and analyze the simulation results for different implementations of the Smart Cache mechanism that validate our ideas. Finally we conclude and refer to our future plans for validating, improving and assessing the ideas proposed in this work.

2. Related Work

There are notable studies in the literature on mobility with respect to content-centric or information-centric architectures. i3 [6] is based on indirection for solving the problems that end-point oriented communication imposes on mobility. Essentially, i3 is an IP overlay network that conceals point-to-point communication by applying a rendezvous-based model on top. Senders send packets to a specific rendezvous point and receivers issue triggers on specific packet identifiers. i3 is mobile friendly, it protects location privacy and has been shown to perform better than Mobile IP in terms of stretch and fault tolerance [7]. PSI uses similar concepts through the rendezvous and topology formation processes. Yet, unlike i3, PSI is native to the network and not an overlay approach.

ROFL (Routing on Flat Labels) [8] and DONA (Data Oriented Network Architecture) [9] use information identifiers rather than location –possibly hierarchical – identifiers. DONA employs self-certifying identifiers as a replacement to the DNS naming resolution that enable “finding” content (actually finding a location to get content from). DONA’s discovery mechanisms also provide a facility for caches, for forming multicast delivery trees and anycast by means of locating the nearest replica of the desired content. Nevertheless and unlike what we propose here, DONA cannot be credited with proposing any new ideas regarding data transfer in a mobile environment. ROFL on the other hand uses Distributed Hash Tables and their hierarchical versions. Node identifiers called “labels” are location independent. This matches the needs of a mobile environment where agents move and therefore need to be uniquely identified without respect to their current location (which may well continue to change). Mobile agents are specially treated and identified as ephemeral hosts. The latter hosts participate as little as possible to the overall architecture. This treatment reduces the significance of the overall performance impact posed by their mobility. This is a major difference from PSI, for we deliberately do not distinguish between stable and mobile agents in an effort to apply a universal property to our architecture.

CCN is another important on-going future Internet research project with a content-centric perspective in mind. Unlike ROFL, routing is based on hierarchical naming; consumers broadcast “Interest” packets that contain the name of the desired content. “Data” packets stamped with a content name that is a suffix of the name asked by an “Interest” packet is considered as satisfactory for that interest. CCN supports mobility by means of caching, by allowing nodes to transmit “Interest” packets simultaneously from multiple interfaces and by assuming an unreliable transport delivery service [10]. Nonetheless, there is no notion of a rendezvous scheme as there is in PSI, while the unstructured organization of the information and the advertisement of content by flooding will possibly have a negative impact on mobility support.

References [11] and [12] propose Publish/Subscribe architectures that use brokers which mobile agents can use in order to attach themselves and to send Publish/Subscribe requests. Perhaps the most relevant study to our work is reference [12]. Its authors describe a mobility support solution for Publish/Subscribe networks that uses a neighbor graph per broker. This graph is a data structure that contains candidate brokers for caching subscription context. Upon receiving subscription data, a broker conveys this subscription to other brokers within a single-hop distance. We believe that this poses an overhead concern regarding control messaging for propagating subscriptions within a neighbor graph and for maintaining the neighbor graph itself. We also note that despite the statements of the authors of that work, forming the neighbor graph requires topological knowledge as subscriptions carry the address of the originating broker of the mobile node. Publish/Subscribe requests in [9] are recursively propagated through their brokers to the neighbors of the latter and so on. As a result, whenever a broker identifies a match between a request and data in transit she knows where to forward the data next. The authors argue that there are no scalability issues due to this "flooding" of requests amongst brokers. We prefer to take a different approach by using a PSI compatible Publish/Subscribe mechanism which allows agents to move without suffering any loss in the data requested by them and without the need of any control messages.

3. Publish/Subscribe Architecture

3. 1 Introduction

Publish/Subscribe architectures involve three major entities: the publisher, the subscriber and the event notification service of rendezvous point [13]. Publishers hold the role of the information provider, i.e., the one who advertises the availability of specific pieces of information by means of issuing publication messages. Subscribers are information consumers, who express their interest for specific information items by issuing subscription messages. As a first step, the event notification service locates the publishers who provide information items matching the consumers' subscriptions. As a second step, the event notification service initializes a forwarding process from the information providers towards the information consumers. The publication and the subscription operations do not have to be in sync while publishers and subscribers do not have to be fully aware of each other.

The event notification service is modelled by the Rendezvous Network (RENE) and is mainly responsible for matching subscriptions with the appropriate publications; moreover, it initiates the information forwarding process. As described in [14], a RENE is composed of several rendezvous nodes (RNs), each of which is responsible for a set of publications. We refer to the RN that is responsible for a publication as the Rendezvous Point (RVP) of this specific publication. As in [14], we use (statistically) unique labels for each discrete piece of information, composed by a pair of flat, semantically free identifiers: the Rendezvous Id (RId) and the Scope Id (SId). It is this unequivocal label that implies that a specific RVP should be used rather than another one. This particular RVP matches subscriptions to publications. In addition, scoping mechanisms are used in order to control access to information from principals that have been delegated access rights to a particular scope. Note that scopes employ a hierarchical structure where sibling and parent-to-children relationships exist [15].

In the PSI architecture, there can be physical scopes such as a corporate network or logical scopes such as a social network. Scopes are identified by a second class of identifiers, the scope identifiers (SIds). A SId denotes the specific scope within which the information is reachable. Note that both RIds and SIds are independent from the endpoints producing and consuming the associated information items.

In the next chapter we analyze how the RENE is organized and give an example of how it manages publications and subscriptions in order to finally achieve the delivery of data.

3. 2 Rendezvous Network Architecture

We based our work on the Rendezvous network architecture described in [14] and in [16]. The RENE that was implemented as part of this thesis was based on the Rendezvous, Topology, Forwarding and Media architecture (RTFM). In RTFM, publications, as described in the previous section, are actually announcements for the availability of information and do not involve the delivery of information. The network tracks the available information and when a user subscribes to it, the network locates the publisher that holds the publication and instructs the publisher to deliver the data to this subscriber. The delivery of data is orchestrated in RTFM so that optimal paths are selected, computed by a topology function. The architecture separates the networking operations into three distinct functions: Rendezvous, Topology and Forwarding. The Rendezvous function is responsible for matching user subscriptions to publications and locating publishers. The Topology function maintains a topological view of the network and creates optimal delivery paths for disseminating information. The Forwarding function implements the actual data transmission towards the subscribers.

For the network analysis of the architecture we will consider three types of network elements in the system: routers, rendezvous points and user hosts. All of these nodes may perform both publications and subscriptions but they differ to the level of topological knowledge they have and to their network functionalities. This network analysis does not introduce new entities to the Publish/Subscribe architecture. Routers and hosts may perform publications or subscriptions and act in each case as either a publisher or a subscriber. All network elements are assigned a flat, semantically free, topology independent node identifier. Node Ids are unique within a network area, but they do not need to be globally unique, unlike IP addresses.

Routers are simple, low complexity, packet switching elements that maintain a topological view of the network through a link state routing protocol. Rendezvous Points (RVPs) are special network nodes that keep track of available publications. Publications and subscriptions are routed to RVPs which perform subscription-publication matching. User hosts gain access to the network through gateway access routers. Users do not have access to topology information; they communicate only with their gateway routers which proxy user publications and subscriptions to the network. Throughout this work, we will refer to these different types of network elements, as nodes that publish or subscribe to information.

Publications are identified by a pair of flat, semantically free identifiers, namely the Scope Id and the Rendezvous Id. The Rendezvous Id (RId) is a statistically unique identifier used as a label for the publication. The Scope Id (SId) is used to organize information items into larger collections. The combination of SId/RId allows applications to build relationships among information items, for instance, creating information taxonomies, suggesting the location of information or enforcing access control policies.

The forwarding function is performed with the assistance of zFilters, which are fixed size, in-packet Bloom filters as described in LIPSIN [17]. Each network link is assigned with a Bloom filter identifier called its Link Identifier (LID). LIDs are unidirectional and therefore each link is identified with a pair of LIDs, one for each direction. To create a forwarding path between two network nodes, LIPSIN encodes all LIDs in the path into a single Bloom filter using binary OR. Packets are transferred in the network using the zFilter as a source route header. When a node receives a packet, it extracts the zFilter and compares it with its outgoing LIDs using binary AND. If the result of the comparison between the zFilter and the LID matches the ANDed LID, then the node assumes that this LID is encoded in the path and transmits the packet on that link. To define the recipients of a packet along a path, nodes are also assigned a Virtual Link Identifier (VLID) pointing at themselves. Path construction includes also encoding the VLIDs of the destination nodes (one or more). When nodes receive packets, they also compare the zFilter with their VLID and if the result matches that VLID, they deliver the packet to the system's networking software. The LIDs that constitute a path form an identifier, the forwarding id (FId) for this path that is used by the forwarding service to forward the data packets.

3. 3 Inter-Process Communication

In this section we describe the publish/subscribe communication among processes in a single node and continue with the description of connectivity establishment between two nodes. Next, we describe a publish/subscribe version of a link state routing protocol that runs in network routers for managing topological information.

3.3.1. Node Inter-Process Communication

The heart of the node is the Local Rendezvous Component (LocRC) which provides a publish/subscribe inter process communication (IPC) mechanism to system processes. The LocRC stores subscriptions issued by processes and forwards received publications to processes with matching subscriptions. Network protocols are organized as publish/subscribe processes around LocRC, forming a protocol circle instead of a typical OSI protocol stack .

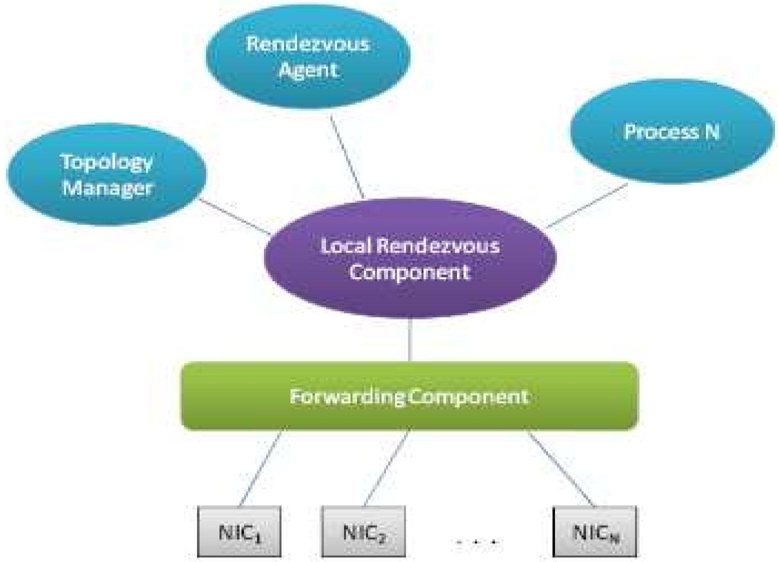

Figure 1: Internal view of a Publish/Subscribe node

Below the LocRC lies the Forwarding Component (FwdC) which is directly connected to the node's network interfaces. In order to transmit data to the network, processes have to pass publications to the FwdC via the local publish/subscribe IPC. For example, assume that node X wants to send the publication **(S, R, [data])** to node Y, with S being the Scope Id and R the Rendezvous Id. X must obtain the forwarding path pointing from X to Y, encapsulate both the path and the publication into a new publication:

(Fwd_SId, Fwd_RId, [pathXY , (S, R, [data])]),

which is then published to its LocRC. Node X's FwdC, which is already subscribed to Fwd_SId/Fwd_RId, receives this publication, extracts the encapsulated publication and transmits the packet according to the embedded forwarding path, which in this case points to node Y. The FwdC at Y receives the packet, removes the path from the data of the publication and publishes the original publication (S, R, [data]) to Y's LocRC. If there are any processes in node Y that had already subscribed to (S, R), Y's LocRC will deliver the publication to them.

3.3.2. Point to point communication

When a publish/subscribe node is connected to a communication link, that node's FwdC broadcasts a publication to a well known SId/RId pair agreed between the FwdCs, e.g. Fwd_SId/Fwd_Link_Connect_RId and announces its VLID to neighbouring nodes. When FwdCs receive such publications, they respond back with a new publication containing their own VLID. After the VLID exchange phase is completed, FwdCs set a new LID for the established link and announce the event locally by publishing the LID/VLIDs information under Fwd SId/Link Established RId. Communication with neighbouring nodes is now feasible as it only requires encoding the LID and VLID of the new node into a forwarding path. To support multi-hop communication, routers need to obtain topology information, as described in the following section.

3.3.3. Publish/subscribe topology management

The Topology Management Component (TMC) is an implementation of the network's Topology function and is responsible for providing forwarding paths for disseminating information across the network. TMCs manage topological information via a distributed link state routing algorithm. Topology management is required only in switching elements of the network, thus TMCs are installed only in network routers. A light version of a TMC is also installed to the user hosts but it only provides the proxy router as a neighbouring node. Thus the only forwarding path provided to the simple hosts is the one to their proxy, which then forwards the publication/subscription to the original node. TMCs in different machines communicate with each other by publishing and subscribing to a SId/RId pair agreed a priori, e.g. TMC_SId/TMC_RId.

A TMC is subscribed to Fwd_SId/Link_Established_RId so that it is notified whenever the node establishes a new link connection. Upon link establishment, the TMC creates a link state connectivity publication under TMC_SId/TMC_RId. Link state publications contain the router's Node Id, its outgoing LIDs and the neighbouring Node Ids. The TMC computes the path destined to the neighbouring node and sends the publication. The neighbouring TMC receives the publication, updates its local network graph and responds back with a new Link state publication containing its own link state connectivity information. TMCs forward received link state publications to their neighbours in a recursive manner so that a link state publication reaches all network routers.

As a result, the TMC of the routers and the RVP is an internal component that has the topological knowledge of the whole network graph and when provided with a source and a destination node of the element it can calculate and provide the optimal path for the delivery of the information.

3.3.4. Rendezvous System

Publications and subscriptions sent by users are routed to designated network nodes called Rendezvous Points (RVPs). RVPs keep track of available publications and perform the matching between subscriptions and publications. RVP presence is published to network routers in the same way that link state publications are flooded. An additional internal component installed in all network elements, called Rendezvous Agent (RVA), stores the location of RVPs and assists the rendezvous process. RVAs residing in nodes' access routers operate as rendezvous proxies; they capture publications and subscriptions sent by users and forward them towards the RVP. The coordinated operation of RVAs and the RVP comprises the Rendezvous System.

In the next section we describe a scenario of two users, a publisher and a subscriber, publishing and subscribing to the network to exchange data.

3.3.4.1. Publishing and subscribing

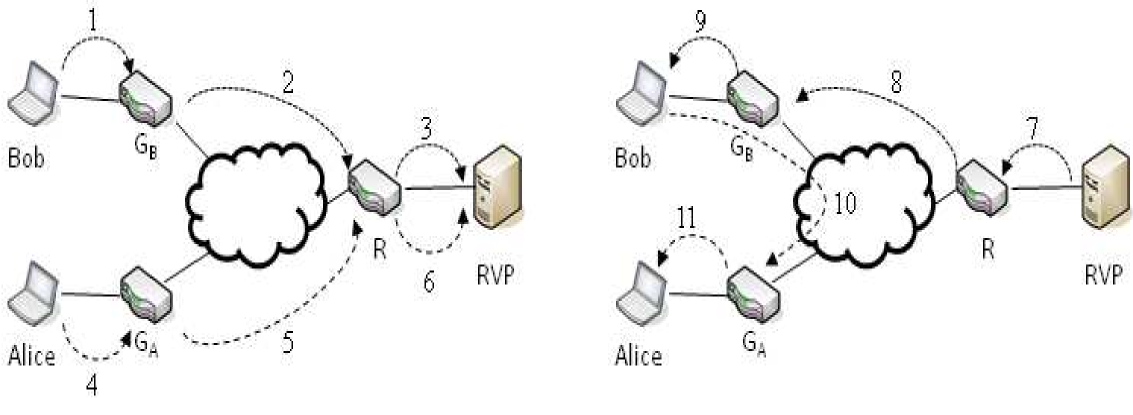

Figure 2: (a) Bob publishes some information and Alice subscribes to this information. Publication and subscription reach the RVP via router R. (b) RVP matches publication and subscription coming from routers G_B and G_A and instructs G_B to send the information to Alice. G_B then instructs Bob and data are sent first to G_A and then to Alice.

Figure 2 above shows an example of two users, Bob and Alice, connected to a publish/subscribe information centric network with a single RVP attached to router R. Bob wants to announce a publication labelled BobSId/BobRId to the network. Bob's host RVA creates a special publication:

(RVA_SId, Pub_Announce RId, [Bob, BobSId/BobRId])

and sends it to its gateway router, G_B (step 1). G_B's RVA receives the publication and records that Bob announced a publication labelled BobSId/BobRId to its local database. Then, the RVA at G_B replaces Bob's identity with its Node Id and forwards the publication

(RVA_SId, Pub_Announce RId, [G_B_nodeid, BobSId/BobRId])

to R (step 2) which in turn forwards the announcement to the RVP (step 3). At this point, the RVA running inside the RVP stores the [G_B_nodeid, BobSId/BobRId] information to its local database. This record indicates that a host connected to router G_B has announced the availability of data with the label BobSId/BobRId.

Next, Alice subscribes to the data labelled BobSId/BobRId. Alice's host RVA encapsulates the subscription into a special publication

(RVA_SId, Subcribe_RId, [Alice, BobSId/BobRId])

and sends it to Alice's gateway router, G_A (step 4). The RVA in G_A receives the subscription, stores it to a local subscription table, replaces Alice's identity with G_A's Node Id and forwards the publication RVA SId, Subcribe RId, [G_A_nodeid, BobSId/BobRId]) to the RVP Proxy Router R (step 5) which eventually forwards it to the RVP (step 6). The RVP matches the subscription sent by G_A with the publication announced by GB and requests its TMC to create the forwarding path from GB to GA.

The next step for the RVP is to send a publication to G_B instructing it to deliver the publication to G_A. The RVP creates a new publication

(RVA_SId, Pub_Instruct RId, [path (G_B, G_A), BobSId/BobRId])

and sends it to G_B (step 7). G_B's RVA receives the subscription, looks up in its local database and matches it with Bob's previous announcement. G_B's RVA updates the received forwarding path by adding Bob's outgoing LID. The zFilter now contains the path from Bob's host to Alice's gateway router. The instruction is handed to Bob (step 9) and Bob's host transmits the publication to the received path (step 10). When G_A receives the publication, it searches its local subscription table, matches the publication with Alice's subscription and forwards it to Alice's host (step 11).

4. Mobility in PSI

Publish/Subscribe Internet architectures, as discussed in previous chapters, constitute clean-slate efforts for a Future Internet. Thus, they are basically designed to support mobility with no further adjustments to the network and its components. The basic functionality is intended to be indifferent to the type of hosts, whether they are fixed or mobile. With this work, we demonstrate this feature by describing a mobility scenario and analyzing the behaviour of the basic network components of the architecture.

Our idea aims to keep untouched the fundamental functionality of a Publish/Subscribe network and its main principles, i.e., publishers, subscribers and the rendezvous point. A publisher issues a publication request in the rendezvous network, which includes the metadata and the RId of the information object that she wishes to make available in the network. Similarly, subscribers submit subscription requests for the RIds of the information objects they wish to obtain. As discussed in the previous section, both the publication and the subscription are forwarded through the rendezvous network to the Rendezvous Point (RVP). Publication and subscription may be issued asynchronously with no impact to the network process and communication. Thereafter, the RVP taking into consideration the topological knowledge of the network, sends the right FId to the publisher(s) and initiates the process of the information forwarding from the publisher that holds the desired content to the subscribers that wish to obtain the information. As one would expect, FIds are constructed with respect to the topology of the forwarding part of the network. In accordance to PSI, a separate topology manager creates the right FId which is forwarded by the RENE to the publisher(s), ensuring that data will reach the subscriber(s) (in general by multicast). An important aspect of this architectural choice is that publishers successfully transfer data to subscribers, without the network disclosing where the latter reside in the network.

The above scenario of information exchange is totally indifferent to the location of both the publisher and the subscriber. This proves actually the fact that the PSI architecture is ready to support mobile agents since all of its functionalities are both location-independent and location-agnostic. A subscriber that has requested a piece of information and decides to move while receiving the data, will still get the information requested since the topology manager will recalculate the route to the new location and will deliver the data correctly.

The problem here is that taking into consideration that any sort of message exchanged in the Rendezvous Network is either a subscription or a publication, a subscriber would inform about the new location by actually subscribing again for her pending subscriptions. This means though that the rendezvous network would initiate the forwarding process from the publisher to the subscriber all over again and the subscriber would get duplicate parts of the information. This would be a critical problem in cases of information flows that take into account the time and the sequence that the data arrive. The PSI architecture though wiil also include transport functionality for the exchange of data that will indeed handle both the duplicates and the packet losses. Nevertheless, with this work, we are trying to demonstrate a more efficient way to handle mobility in PSI, by firstly trying to avoid to base on higher level functionalities such as transport to support seamless mobility and secondly by trying to foresee where the requested data will be of demand in order to also provide better network efficiency and better response times.

This work aims to face the mobility problems posed by the PSI architecture and take things one step further by proposing a series of enhancements that will assist mobility. More specifically, we introduce the idea of Smart Caches (SC) that assist mobility management. SCs are in-network caches defined by RVPs based on the topology of the mobile network, as well as the current and the predicted (or expected) future positions of subscribers of a particular publication. A SC acts both as an intermediate caching node and as a mediate assistant in the process of exchanging data between the publisher and the subscriber.

In the following sections we will analyze the nature of the Smart Cache components and we will provide an insight to the added functionality in the rendezvous network.

5. Smart Caches

In the previous section, we presented as main contributors to supporting mobility the fundamentals of the PSI architecture. Also, we stated that we do not aim to provide an add-on solution to the initial architecture unlike the common solutions for the current Internet architecture. Therefore we maintain the fundamental components of the network, publishers, subscribers and RVPs, and their basic functionality. The idea of Smart Caches does not introduce a new type of network element or an add-on to the PSI. The PSI architecture as described in previous sections has integrated support for mobility by design, as well as for other important aspects (anycast/multicast, caching, multi-homing, security and privacy). SCs are just an optimization and are more likely existing ordinary caches in the network, used for caching frequently requested content or any other network component with the capability to cache even a small amount of data for a small period of time.

A SC caches incoming data for a short period of time, typically for the amount of time needed for a subscriber to move from one Access Point (AP) to another. Thus, it is guaranteed that while the mobile host moves, the data packets are not lost but are instead cached at the SCs and will reach the final destination, the subscriber, as soon as she reconnects to the destination AP. Furthermore, SCs behave in the network as a delegate for the exchange of content between the publisher and the subscriber. Data packets from the publisher first reach the SC and are then sent to the host that requested the content. Thus, as it can easily be observed, a SC is a network element that actually combines the characteristics of both publisher and the subscriber. On the one hand, a SC is responsible to deliver data over a path designated by the RVP (like a publisher would do) and on the other hand, receives the data originated for the mobile host (like a subscriber) and delivers them to the new location.

Another advantage that the SCs offer to PSI is that they enhance anonymity between the involved parties. Due to the enforced FId mechanism, plain PSI – i.e. the PSI architecture with no incorporated Smart Caches – would still not disclose the identities of the parties to one another, nor would it reveal their location in the topology of the network. Introducing SCs improves anonymity as SCs mediate as transfer-proxies between publishers and subscriber; thus prospective transfer protocols may take advantage of SCs in order not to reveal the identities of any involved principals. SCs (or similar mediators) could secondarily function as parts of reliable transport protocols that would prevent parties from knowing each others' addresses and thus better balance the power between senders and receivers (unlike today's imbalance of power between senders and receivers [18]).

5.1. Smart Cache functionality

In section of the Publish/Subscribe architecture we described how the exchange of data between two nodes of the PSI network takes place. A publisher, the node that obtains a piece of information, issues a publication with a pair of identifiers, RId and SId, and this announcement reaches via the RENE the RVP. Thereafter, when a host, namely the subscriber, is interested to this piece of information sends a subscription request to the RENE. The RVP matches the subscription and the publication and sends a FId that contains the path from the publisher to the subscriber to the publisher, which then starts to send the data.

SCs intervene in the above process in order to cache data for a short period and finally to assist mobility. We should revisit the PSI mechanism for the exchange of data, in order to explain how Smart Caches are integrated in the network. The publisher continues to announce the possession of data with the unique identifiers and the subscriber still issues subscriptions with the same RId, SId to get the desired content. The RVP matches the publication with the subscription, but at this moment, instead of requesting from the topology manager one FId from the publisher to the subscriber, the RVP asks its local topology manager, who is aware of the entire network topology, to designate a SC, and then sends two different FIds. The first FId describing the path from the publisher to the SC is sent to the publisher and the second FId describing the path from the SC to the subscriber is sent to SC. We should note here that the RVP at this point, also registers to its publications list an extra publication which has the same identifiers as the one issued by the publisher, but refers to the SC as the node that offers the publication (initial publisher). Thus, the SC is now considered as an extra publisher by the RVP. When the FIds are received by the publisher and the SC, the publisher sends her data to the SC and then the SC forwards them to the subscriber. A Smart Cache caches the incoming data from the publisher for a very short period of time, typically for that long as it would take for a mobile host to move from one AP to another. After this period of time ends, the data packets are dropped and can only be found to the initial publisher of the information. This is mainly because SCs are intended to be used for supporting mobility and not to cache data. Storing data for only a short period of time means that the same node will support more mobile users.

As soon as the SC is declared from the RVP and the data are cached, one can understand that when the subscriber that requested the content moves towards a new location, she should receive the data seamlessly. We assume here that when a mobile subscriber moves to another AP, she re-subscribes for the subscriptions that are not yet completed. This means that this subscription will still reach the RVP to be matched with the publication and get the remaining data. The benefit that the SC offers is apparent at this point. The RVP has now two matching publications for the subscription that has reached it, the initial publication from the publisher and the one that the RVP self-registered for the SC. The dilemma then is to choose between these two publications; this is functionality already integrated in PSI for multiple publishers and is based on the topological knowledge of the RVP and the topology manager. Namely, the publisher that results in a shorter path between the publisher and the subscriber is selected to send the data. In this case, and taking into consideration that the SC designation is based on network topology and on the potential next movements of the mobile host, the SC is the most likely selected publisher. Thus, the subscriber gets the data that were lost while she was not attached to any AP, since these are cached and does not need to get the entire information from scratch that would have result in duplicate pieces of information. Moreover, the SC is a network element expected to be closest to the mobile host, which means that the introduction of the SC also means that we will have improved performance in the delivery of data.

We should note here that the selection of the SC is an iterative process and that when the mobile host moves and re-subscribes, the RVP will again declare a SC in order to optimize for the new location but also to assist further changes of the location of the mobile host. This SC might be the same or different from the SC originally selected, based entirely on the suitability of the SC, the network topology and the movement of the mobile node.

We can see now in more details the how the above mechanism works with a use case scenario. The following picture depicts an exchange of data between Bob and Alice, where Bob is the fixed publisher that attains the information and Alice is the mobile subscriber that desires to get the content that Bob has advertised.

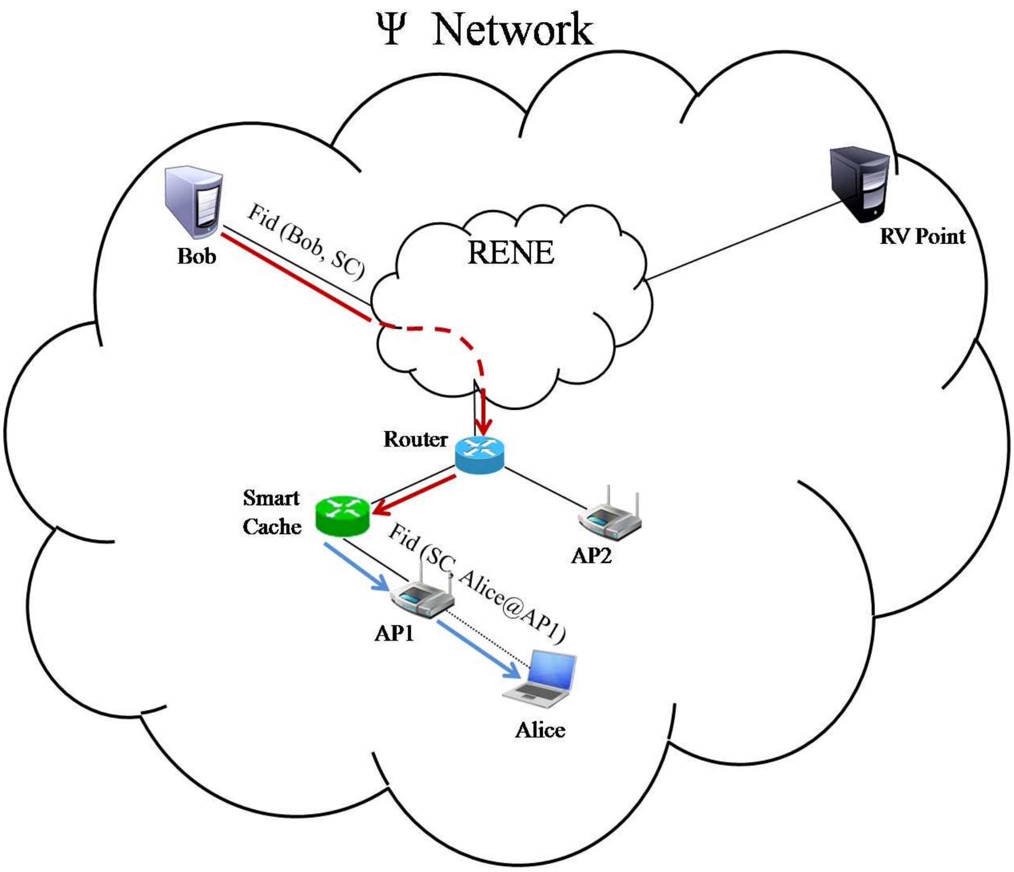

Figure 3: Bob delivers data to the SC and the SC to Alice

Figure 3 above depicts a scenario in which Bob issues a publication <Bob_SId, Bob_RId, [metadata]> under the scope and rendezvous identifiers Bob_SId and Bob_RId respectively. Let us assume for illustration here that the announcement that Bob makes with this set of identifiers concerns a file, e.g. a movie, of 4GB size. A mobile device (Alice) that is currently attached to some AP, is interested in this particular publication and therefore issues the subscription <Bob_SId, Bob_RId>. Both the publication and the subscription find their way to the RVP through the RENE of the PSI network. The latter matches the publication and subscription based on the scope and rendezvous IDs.

The RVP designates a smart cache (SC) based on the topological knowledge of the network through the topology function, and records the SC as another publisher for Bob_SId/Bob_Rid by registering the SC as another entry in its publications map. The goal is to facilitate the delivery of data to Alice and any other prospective mobile subscribers. Upon matching a subscription, the RVP uses two forwarding paths:

a. FId (Bob, SC) which is sent to Bob and refers to the path from Bob to the SC, and
b. FId (SC, Alice) which it sends to SC and refers to the path from the SC to the AP that Alice is currently attached to.

We should note here that the RVP at that point selects which of the publishers, Bob or the SC, will deliver the data to the subscriber by sending the related instruction message via the rendezvous process described in previous chapter. This means that either the SC will deliver the data to the subscriber, or Bob will deliver the data to both the SC and the subscriber. This functionality of the RVP enables us to better handle the initial state of the subscriber, before it moves to another AP, since the selection of the SC is mainly based on future positions of the mobile node; note that there might be cases that before the subscriber changes her location, the initial publisher (here Bob) may be closer to the subscriber since the SC is selected based mainly on future locations of the subscriber. Thus, selecting between the SC and the initial publisher is necessary in order to select the most suitable publisher in all cases, before and after subscriber's change of location.

As shown in Figure 3, Bob starts sending data to the SC using FId(Bob, SC), which in turn resubmits the data to Alice using FId(SC, Alice). We must note here that data are not sent in a packet of 4 GB, but are segmented into smaller chunks. The first chunk that the publisher sends contains some metadata information that includes the size of the file. The SC caches data for at least as much time as required for Alice to move to a neighbouring access point AP. Assuming a micro-mobility scenario, the RVP is able to foresee the APs that Alice might attach herself to at any point of time. Thus, it may estimate the above caching period; moreover, it may designate a SC which will be suitable not only for the current, but also for future attachment points in the mobile network.

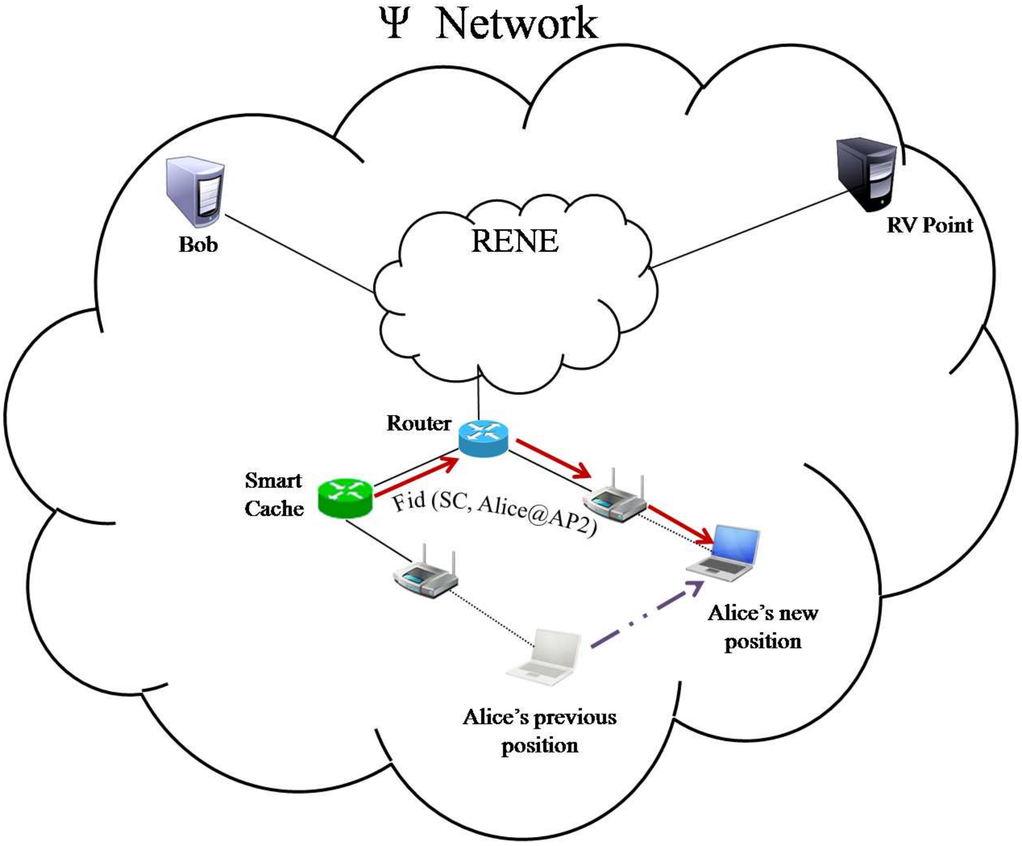

Figure 4: Alice moves while the delivery of data takes place

Alice may decide to move, attaching herself to AP2 while data are in transit to AP1 (Figure 4). Following the completion of the handoff, Alice issues a new subscription from her new position (AP2), which is delivered to RVP. The RVP publications list contains at least two different publishers whose publications match this subscription; one that corresponds to Bob and another one for the SC. The RVP instructs the best suited publisher with respect to Alice to start sending data. SC will most probably be the best suited publisher, for it was originally designated with that role in mind. Note that data are still sent to AP1 via a multicast tree until all subscriptions submitted from AP1 expire, and that includes the one initially sent by Alice though AP1. Meanwhile, SC may well be used to facilitate access to the same item for new subscribers for <Bob_SId, Bob_RId> that fall under the branch of the multicast tree that AP1 resides on.

With Alice being now attached to another AP, the RVP can either select a new SC for Alice or instruct Bob to directly send the data to Alice in AP2 using FId (Bob, Alice). Nonetheless, there is a consideration for subscribers that keep changing positions; in such cases it is sounder to maintain the same model, i.e., the RVP to select a new SC. As soon as the FIds become available to Bob and/or the SC, they can start sending data to Alice. Note that no data in transit are lost while Alice moves. This is particularly important for streaming applications where SCs can be used to redirect such data to the current position that Alice is attached to. Regarding document object data, this is usually not needed; but it can nonetheless be applied for downloading big documents such as movie files.

5.2. Smart Cache Selection Mechanisms

We have presented above how the Smart Caches contribute to the mobility support in PSI network. However, it is now worthwhile to investigate mechanisms for selecting the best suited Smart Cache that will best facilitate mobility.

We have considered two different mechanisms to select the most suitable smart cache(s): RVP forecasting (RVPf) and AP based (APb). In both cases, the RVP adds the SC as another publisher for the same content as the original publication. The two mechanisms differ with respect to who initiates this procedure and the point of time that this action is performed:

RVPf: We assume that the RVP has an up-to-date knowledge of the current topological organization of the network via a topology-manager function [16]. Consequently, RVPs can forecast the next possible position of any mobile node in micro-mobility handoffs; and thus can designate SCs in advance, i.e., before the subscribed agent moves. Upon the designation of SCs, any data that were originally sent to the AP are also sent via a multicast tree to the new SCs.

APb: The underlying idea here is that APs in wireless mobile environments can detect a reduction of the signal strength when the mobile device moves away from it. On the one hand the current AP may infer the tendency of the mobile agent to move away and re-attach herself somewhere else. On the other hand, a prospective new AP may also infer in advance that the mobile device may attach herself to this AP. Upon such movement detection, the current AP sends a control message to the RVP, requesting the reconsideration of the current SCs and perhaps the creation of new SCs. Once the RVP receives this control message, the procedure followed is identical to what is described in the previous sections (adding a new SC as a publisher to the RVP, create suitable FIds to send data via new SCs etc.).

APb adds an extra control message for every hand-over. This is a small overhead and compensated by taking load off the RVP. This is quite important, since the RVP is the node responsible for matching the subscriptions made from all the mobile devices inside its domain; overloading it could cause delays to rendezvous and thus to data delivery. Moreover, APb allows the RVP to take action only when needed, which means only when the mobile agent actually moves to another AP and does not need to run the topology function for every mobile node to see its possible next positions and create SCs. Nonetheless, APb is not suitable for non wireless mobile networks; in such cases RVPf is left as the only alternative.

The mechanisms implemented and described in the following sections are mainly in the category of RVPf in order to take advantage of the topological knowledge the RVP has and because it is quite difficult in reality for the AP to measure the signal strength of all its mobile nodes in order to evaluate if they will detach from it in order to move to another access point or not.

5.2.1. SC Placement Mechanism 1 – A simple idea

The first attempt we made in order to evaluate the use of the Smart Cache for the mobile environment of PSI was not based on any topological algorithm. We decided to place the Smart Cache at the proxy of the AP that the mobile node was attached before moving. This decision was made in order to intervene as little as possible to the existing PSI mechanism.

As described above, we replaced the simple shortest path that the data had to follow to reach the subscriber from the publisher by inserting a mediator – the SC – that was not necessarily in the path from the publisher to the subscriber. Thus, in some cases information might travel over a longer path to reach the final destination, the subscriber. This is an issue for all placement mechanisms at the initial location of the subscriber, since as we stated before the SC is placed based on potential moves. Thus, the path publisher-SC-subscriber might be longer than the path publisher-subscriber.

This case minimizes the extra SC and it also offers the capability of smart caching to the network. It is though limited to serve well movements among APs that are under the same proxy and movements of the mobile host to neighbouring APs. Otherwise, as described above, the RVP will find the initial publisher as the most suitable, which means that the path from the initial publisher will be shorter than the one of the SC.

5.2.2. SC Placement Mechanism 2 – A multicast based idea

The second idea we investigated was also a simple one and we tried to take advantage of the fact that the PSI architecture facilitates multicast. We therefore decided to select multiple smart caches and place them to every AP the mobile node would possibly move to.

We should note here that we take for granted that the Topology Manager of the RVP is totally aware of the network topology and knows the existing APs in the network graph. Therefore, when mobile nodes are detected, it is easy for the RVP to designate a SC to each AP.

A significant remark at this point is that the publisher still gets one FId that serves all possible paths to all destination APs. Thus, when the FId and the instruction to send the data are sent to the publisher, more than one of the outgoing links match with the Bloom-filter based Id and result in the information being sent to multiple destinations, forming a multicast tree.

This idea, although very simple, enables the network to firstly avoid the overhead introduced by the Smart Cache and reduces the hops needed to get the missing data to a path consisting only from one hop (from the subscriber to the AP). It also offers the capability to setup the caches at the setup of the network, avoiding extra complexity and overhead for the RVP to designate caches. Unfortunately, this scenario means that the network might get overloaded with information that might not be requested, since it is not likely for a mobile node to move to each one of the APs. Furthermore, the fact that we don’t have an algorithm provided by the RVP and the network topology means that if the Smart Caches reach their capacity, it is more difficult to find an alternative that will work in the same way as the SC placed directly at the AP.

5.2.3. SC Placement Mechanism 3 – Definitions

5.2.3.1. Neighbours

In order to proceed with the following algorithms we need to introduce the idea of Neighbours. We will name Neighbours of an AP from now on, the APs to which a mobile node attached to this AP can move directly. These APs are actually considered to be one hop away, even though the fixed network graph implies that more hops are needed to reach them.

5.2.3.2. Proximity cost

We introduce a new metric to use to the main mechanism for the selection of the Smart Cache. The idea is to find a way to discover which node of the network graph is most suitable to be the SC and would cover the movement of the mobile subscriber to most of the APs that the subscriber is most likely to move. We can assume that we have N APs and the mobile subscriber is attached to one of them. The probability for the subscriber to move to each of these APs is p. We have node i of the network, that is a candidate to be a SC, and this node is h_i hops away from AP_i . We define as Proximity Cost the following:

$Proximity\ Cost = \sum (p * h_i)\ for\ every\ i\ E\ N$

We should note here this metric is very significant for the selection of the SC. Among two nodes, a node with lower proximity cost is most suitable to become a SC since it better supports mobility to more APs with fewer hops to the network.

5.2.4. Placement Mechanism 3 – The main mechanism

One can understand from the previous approaches, that given that the mobile host will move to another AP and that the original source of data is the publisher, we want to find a way to select a SC so that when the host moves, the SC is the most suitable publisher. Since we do not actually know the next movement of the host, we want to cover as many cases (as many APs) as possible.

The main idea is based on the fact that if we don't have a SC the data will be delivered from the initial publisher, over the path that the topology manager decides that is optimal for forwarding the data to the subscriber. The proposed solution is to actually check the paths from the publisher to all APs and find the common nodes among these paths. These nodes are all candidates to be a SC. In order to select the most suitable among them, we select the node that has the minimum proximity cost. In case that there are more than one nodes with the same value for the proximity cost, we then decide to make them all SCs and handle the delivery of data to them from the publisher via a multicast tree. It could appear a case though where the paths are “parallel” and they have no common nodes. In this case we consider all the nodes of the paths as candidates to become a SC.

We should note here that we have first tried this mechanism with the topology manager having to check all the APs of the system and decide the SC that best suits all of them. We also undertook simulations for a scenario that only some of the APs were taken into consideration, namely the neighbors of the AP. In order to decide the set of the neighbors, normally historical data can be taken into consideration. These data describe the APs that the mobile hosts of the AP usually move to and actually give a smaller set of APs to check, that are those APs that the mobile host is most likely to attach herself when she decides to move.

6. Publish/Subscribe Mobility Support evaluation

6.1. Evaluation Environment

The Smart Cache mechanism introduced above as well as the different mechanisms for the placement of the Smart Caches in the Rendezvous network was implemented in Java and more specifically with the use of Java RMI in order to also support the deployment of different PSI nodes to different machines.

In order to evaluate the mobility support and its mechanisms some simplifications have been made to the above description of the RENE and the SC. First of all, we did not include in our implementation the bootstrap phase of the network since we did not consider this to be of great importance for the mobility case we are investigating. This means, that we consider the network graph fixed and we did not include the phase that the each network element connects to the graph one by one and announces its connection to the neighbours and the routers of the graph. Thus, the network routers are already aware of the network topology and the RVP is known to all the network routers. We also investigate the case that there is only one RVP, a logical assumption though since we are analyzing a micro-mobility scenario and not an inter-domain scenario.

Furthermore, we had to make some assumptions regarding the topology manager of each network element, since no specific requirements are described for this component of the RTFM architecture in the literature. We assume here that the topology manager is aware of the network graph. The topology manager knows all the network elements by their node ids, the links among them and their nature – whether they are fixed or mobile, AP, router or RVP. Thus, the topology manager when requested to find the right FId from one source to the destination calculates the optimal path based on this knowledge. In our implementation of the TMC, this optimal path is obtained using the Dijkstra algorithm.

In order to evaluate the suggested mechanisms we used representative topologies of network providers as proposed by Rocketfuel [19]. From the suggested network topologies we have selected 3 for our simulations. The first one contains 7 routers, the second one 41 and the third 295 routers. In these network topologies we assume that 30% are APs and mobile nodes are attached to some of them. The mobile nodes are moving from one AP to another, performing about 1000 moves in total during the simulation time.

6.2. Evaluation Metrics

The main target of the mechanism we propose is to enhance the already existing mobility support for the publish/subscribe architecture. Thus, the solution we propose is expected to have better performance results compared to the case that we have no SC in the system. We consider here as better performance having a smaller path for the delivery of data from the publisher to the mobile subscriber. In order to evaluate our results we have introduced the following three metrics:

- **Enhancement Success:** This metric is the main performance indicator for our evaluation. It describes the percentage of the total moves of the mobile node that the data delivery path is smaller than the path that the data would follow in the case of the plain PSI architecture with no SCs.

- **Mean Path Decrease**: We introduced this metric in order to investigate how much shorter the new path for the delivery of data is. This is actually necessary, since the above metric of the Enhancement Success will account as success both the case that we only gain one hop and the case we deliver over a path that is smaller than the original by a high percentage. We consider that it is important to know the gain that the Smart Cache mechanism and its mechanism of SC placement offer. A path that is selected for the delivery of data through the SC mechanism is improved by **[(y-x)/y] %** where **x** are the number of hops of this path and **y** are the number of hops from the initial publisher to the subscriber – namely the case that no SC exists. This metric represents the mean of this path improvement for all the moves the mobile subscriber performs.

- **Subscription Capture Success**: This last metric is based on the fact that we try to avoid the additional load of the re-subscriptions of the mobile node to the RVP. When the subscription of the mobile node reaches the designated SC, we prevent the subscription to reach the RVP and instead deliver the data directly from the SC. We must note that the main goal of the SC is not to perform as an RVP as in this case but we consider this to be an improvement to the existing architecture, although it "upgrades" some network elements to become temporarily RVPs. Furthermore, all the other parts of the SC mechanism as well as the placement mechanisms do not take this option into account in order to find a solution that enhances mobility, since this would mean that an alternative network functionality is taken into consideration. This metric describes the percentage of cases over the total moves that the subscription of the mobile node did not reach the RVP but the SC first.

6.3. Evaluation Results – Analysis

Based on the metrics described above and using the three different network topologies we performed several tests to evaluate the Smart Cache mechanism and its different SC placement scenarios. We will demonstrate the results for all the different mechanisms and will try to give an explanation of the results.

6.3.1. SC Placement – Mechanism 1

This mechanism is the first idea for the Smart Cache placement where we decided to place the Smart Cache at the proxy of the AP that the mobile node was attached before moving. The following diagram shows the results for the first metric, the Enhancement Success:

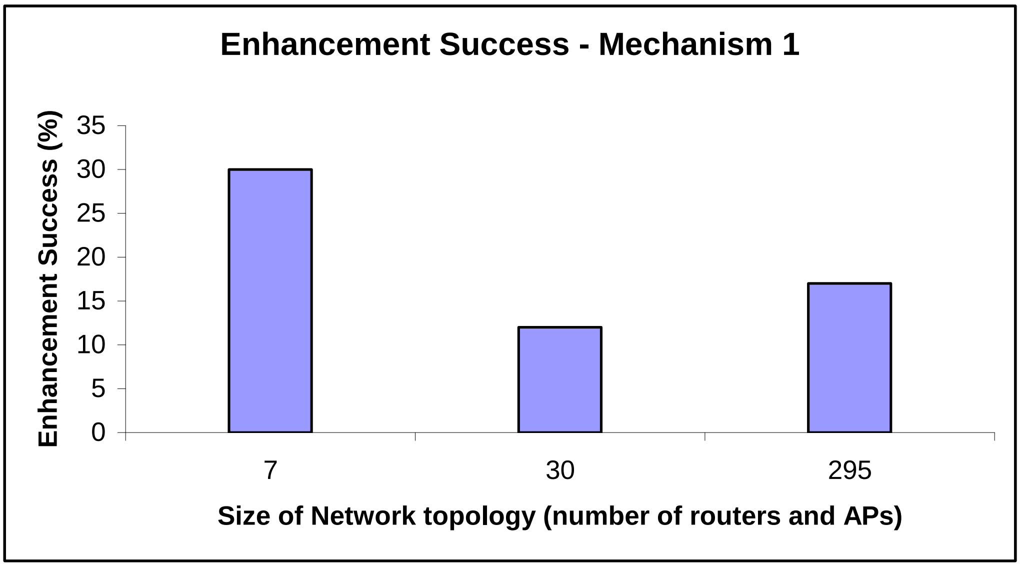

Figure 5: Enhancement Success diagram for Mechanism 1

The above results prove the fact that this method was quite unsuccessful since it has a quite low success rate. The exception of the small network topology is not significant, since in this case the fact that we have very few AP and routers, leads to the fact that these are in very low proximity and that among three APs in one case the previous AP is closer than the original publisher. In general the results here confirm that this solution could only apply in very specific cases where multiple APs have the same proxy. This is also confirmed from the second metric we introduced, the Mean Path Decrease:

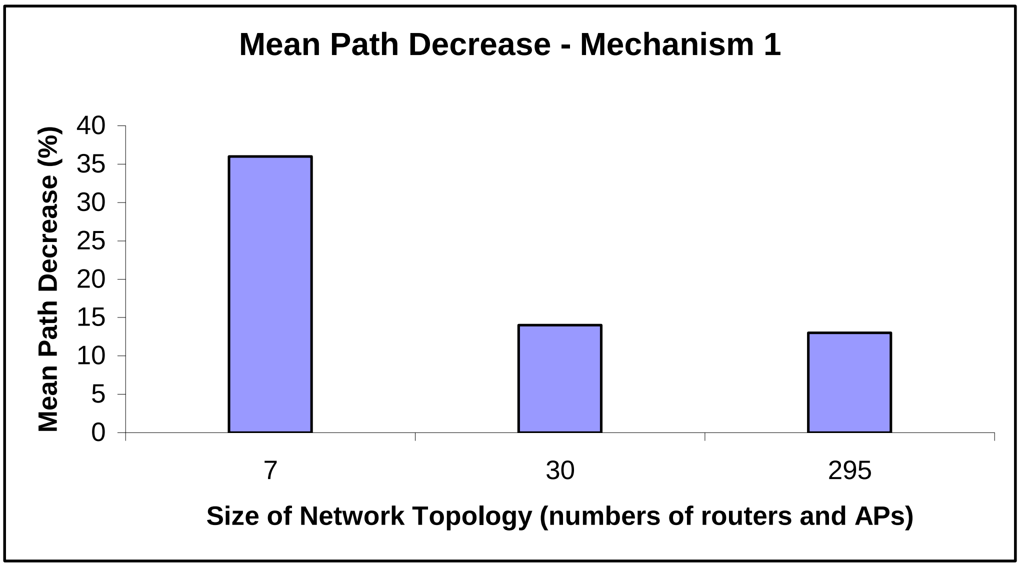

Figure 6: Mean Path Decrease diagram for Mechanism 1

Figure 6 above also describes a better performance for the small network topology but this as described above is related to the nature of the topology. The paths for delivering the data at the first topology are already consisting of a few hops, thus reducing the hops even by one has a greater impact here. In the bigger topologies we see that this mechanism has very few to offer since it improves the path by a very small percentage.

The last metric of the Subscription Caption Success was in all cases for this mechanism 0 since none of the APs had the same proxy and as a result the subscriptions reached the RVP and not the SC first.

6.3.2. SC Placement – Mechanism 2

The second mechanism we evaluated is the one where we placed a SC to every AP of the network, based on the multicast nature of the publish/subscribe architecture. The following diagrams depict the results for this mechanism:

The following shows that in this mechanism we have 100% Enhancement Success rate:

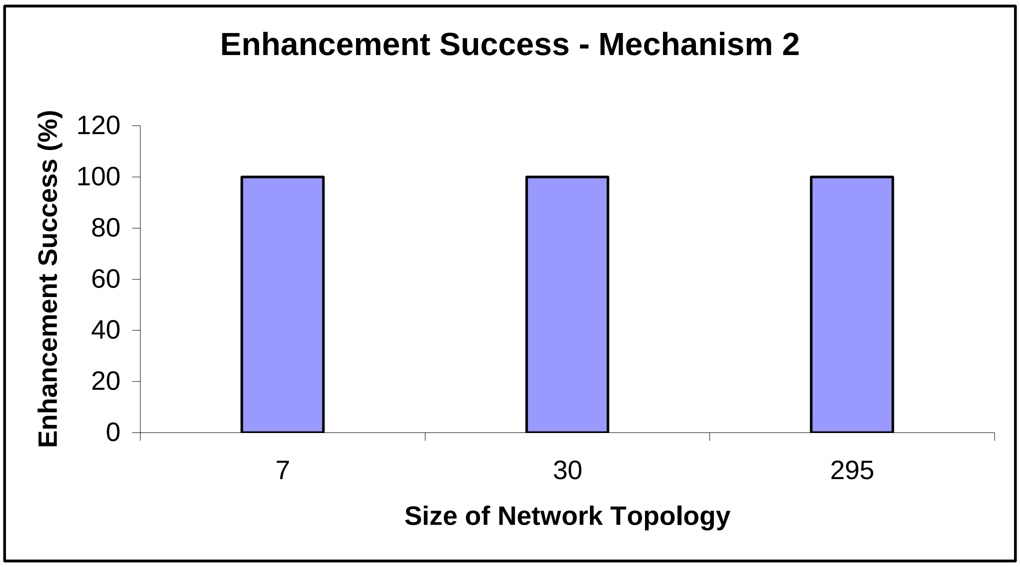

Figure 7: Enhancement Success diagram for Mechanism 2

The idea of this algorithm guarantees that when the mobile node moves there will be a SC with the data waiting for her one hop away. Therefore it is guaranteed that the selected path will always be shorter than the path from the original publisher. Additionally, the fact the SC is only one hop away predicts also the results for the mean path decrease depicted below:

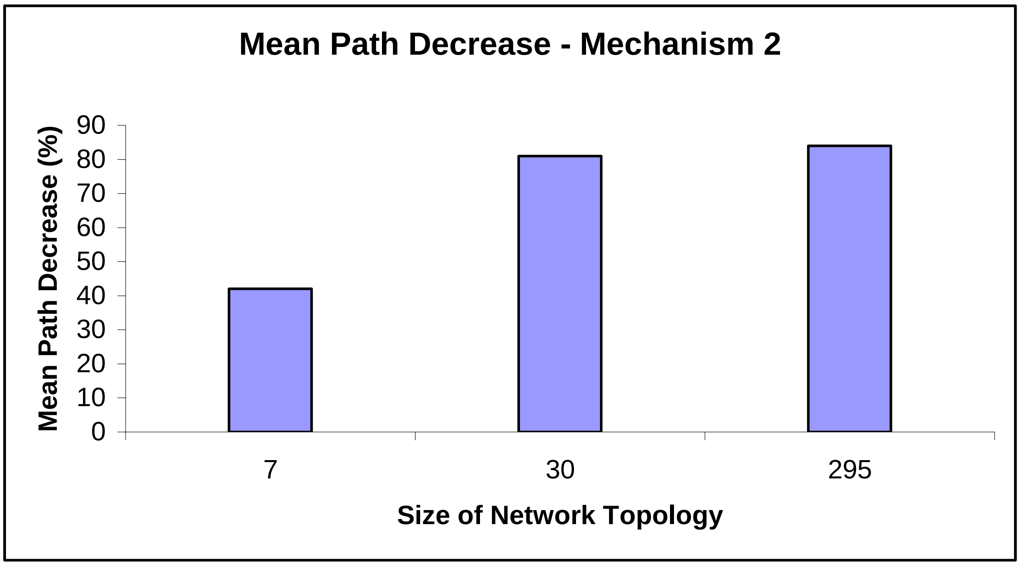

Figure 8: Mean Path Decrease diagram – Mechanism 2

One can understand that here the last metric of the Subscription Capture Success is also always 100% since the SC is always one hop away and can therefore be reached before the RVP.

6.3.3. SC Placement – Main Mechanism

This mechanism is based as we described in a previous section on the fact that we designate as a SC the node that better fits the future movement of the mobile node. This solution checks the paths from the publisher to a set of APs and finds the common nodes among these paths. Among these nodes the one that has the minimum proximity cost is selected and if there are more than one nodes with the same value for the proximity cost, we then decide to make them both SCs and handle the delivery of data to them from the publisher via a multicast tree.

We should note here that we have first tried this mechanism with the topology manager having to check all the APs of the system and decide the SC that best suits all of them but we have also performed the same tests having an initial set of neighbours for each AP. The set of neighbors is a randomly selected set of APs of the graph, consisting 30% of the APs. The following diagrams depict the results for both of the cases:

As far as the results for the enhancement success are concerned, one can observe that we have very good results for all three network topologies. The encouraging difference with the previous topologies is that we have better results for the bigger topologies. This occurs, because at the main topology the node that is suitable for all the APs and the mobile node's moves, might not be the most suitable one for every single move compared with the initial subscriber. In the small topology, the cases that this occurs are slightly more because of the network topology. We should also note here that the results are slightly worse for the case that we have an initial set of neighbors, since there are cases that the mobile node moves to another AP, not included in this set. This could be rectified though if we had some historical data for the moves that the MNs attached to AP perform.

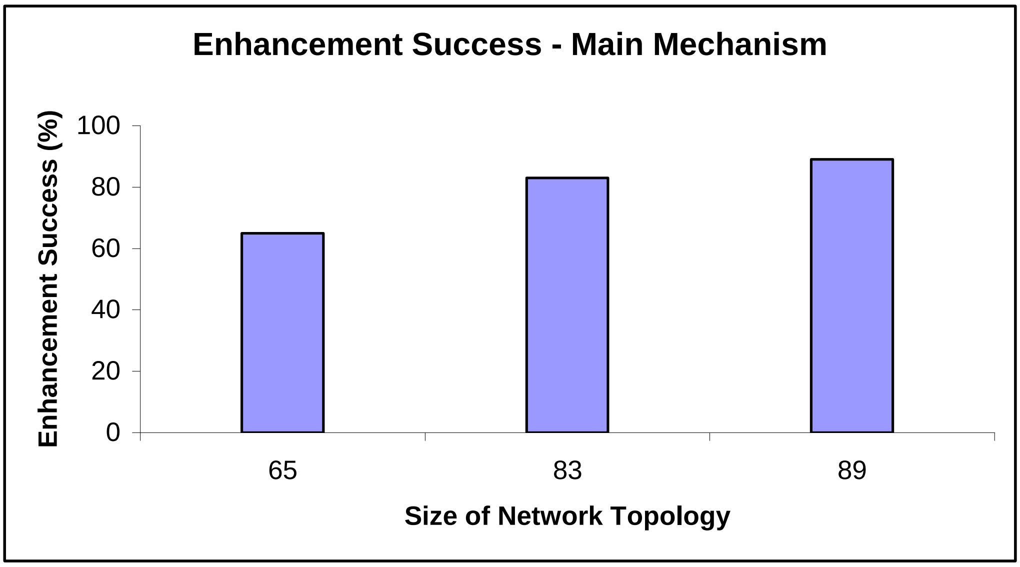

Figure 9: Enhancement Success diagram for Mechanism 3 (without neighbours)

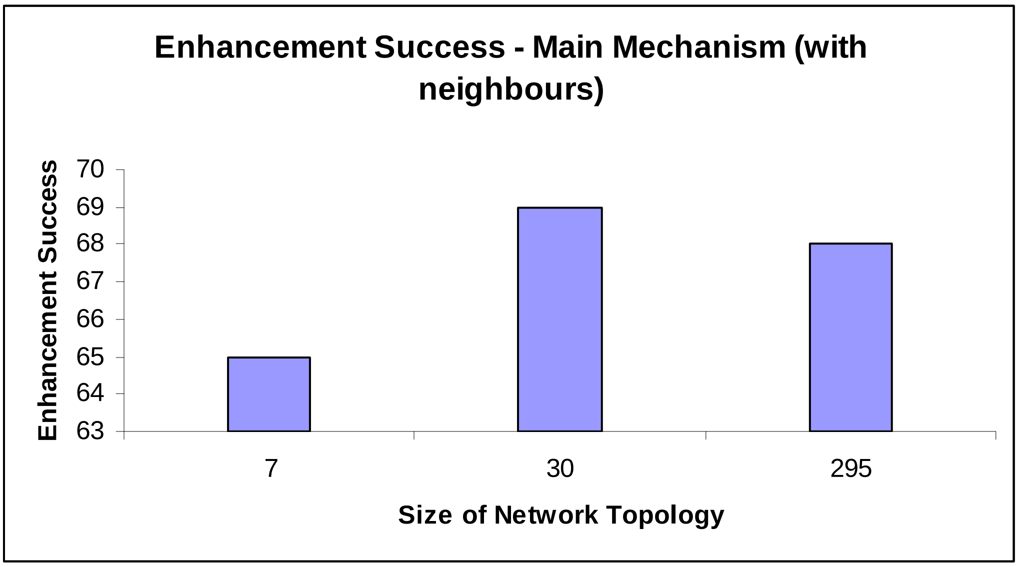

Figure 10: Enhancement Success diagram for Mechanism 3 (with neighbours)

The following depicts the results for the path improvement introduced by this placement mechanism:

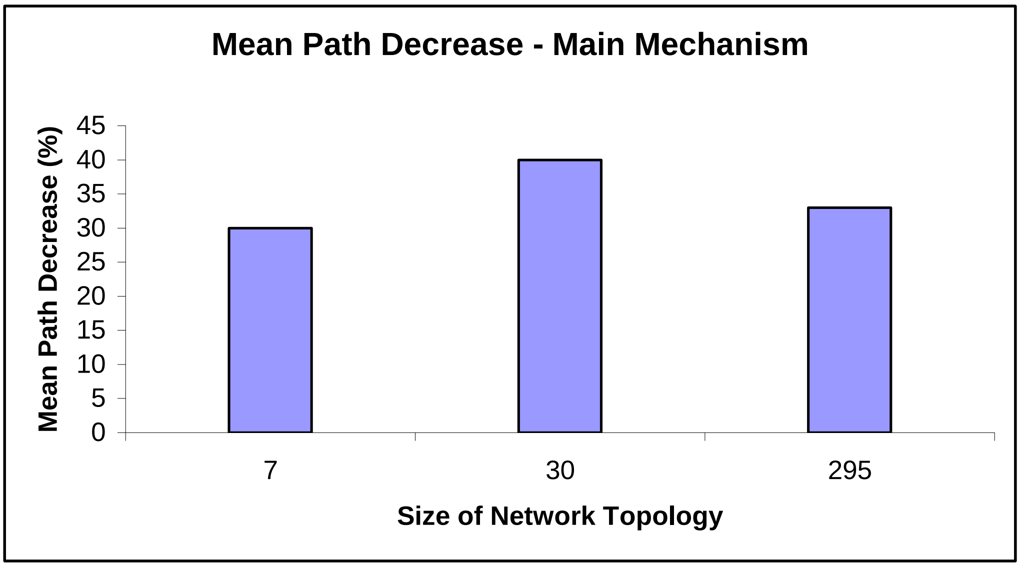

Figure 11: Mean Path Decrease diagram for Mechanism 3 (without neighbours)

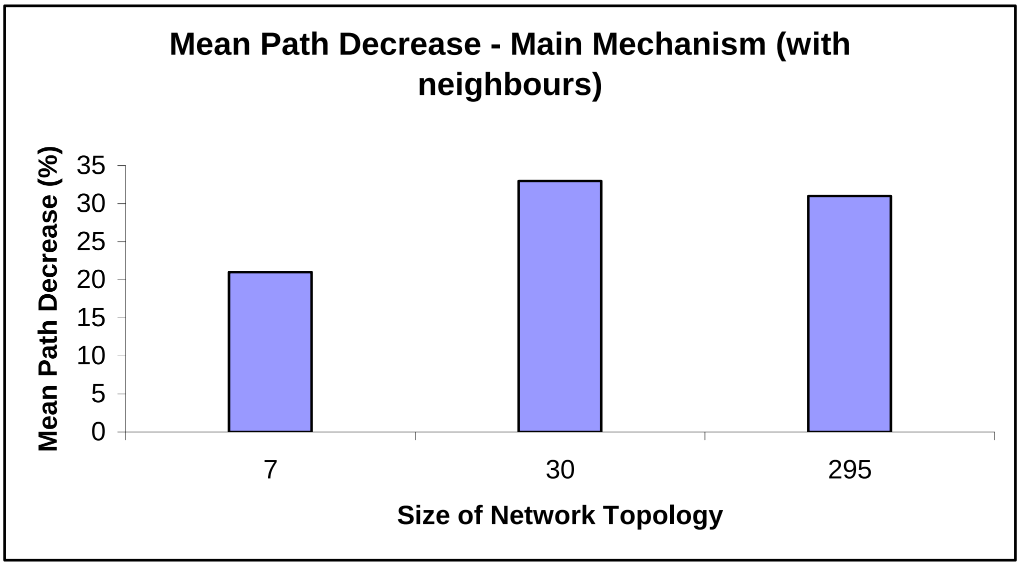

Figure 12: Mean Path Decrease diagram for Mechanism 3 (without neighbours)

It is quite interesting for this mechanism to also report the results for the last metric of the Subscription Capture Success:

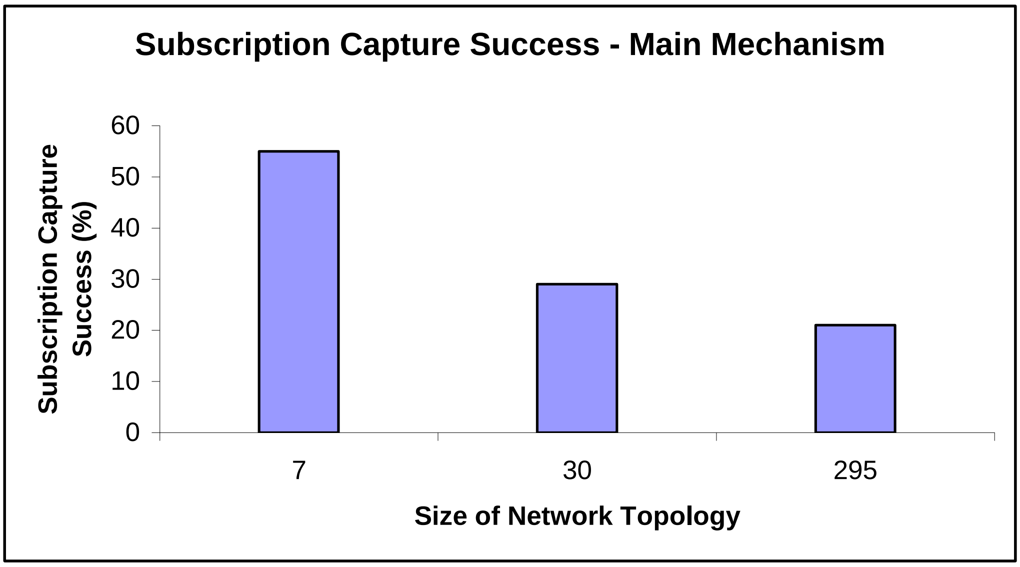

Figure 13: Subscription Capture Success diagram for Mechanism 3 (without neighbours)

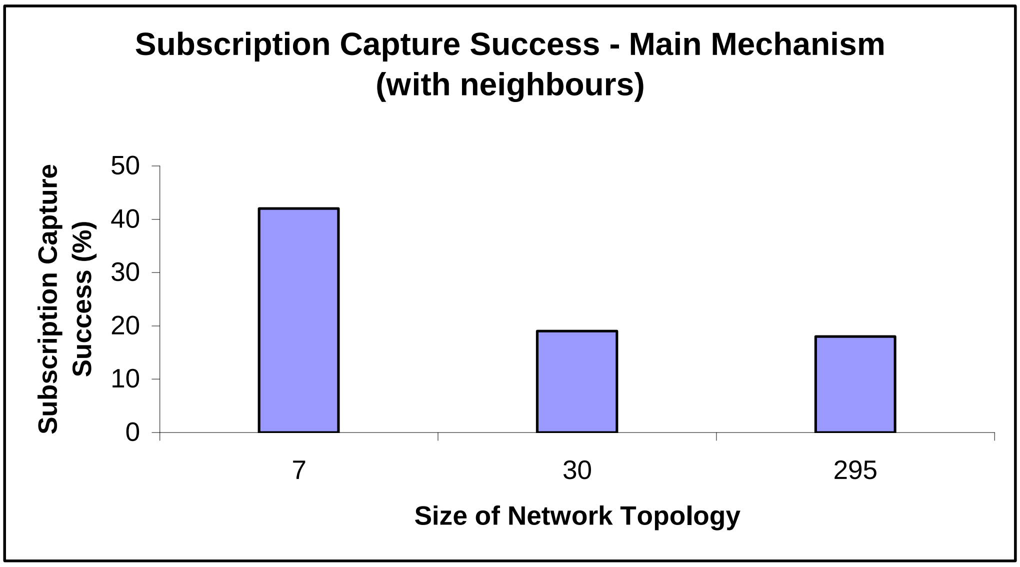

Figure 14: Subscription Capture Success diagram for Mechanism 3 (with neighbours)

It is obvious here that in some cases we managed to prevent the subscription of the mobile node to reach the RVP but the success rates were not very significant. We should note here that this is quite expected, since the Smart Cache mechanism in general targeted mainly to leave the functionality of the network elements attached and was not part of the notion to upgrade other network elements. It is an extra functionality, provided by the SC and a positive side effect to observe but not a criterion to select a mechanism.

In the next picture we show a diagram that concerns all the above mechanisms and depicts the importance of our main mechanism.

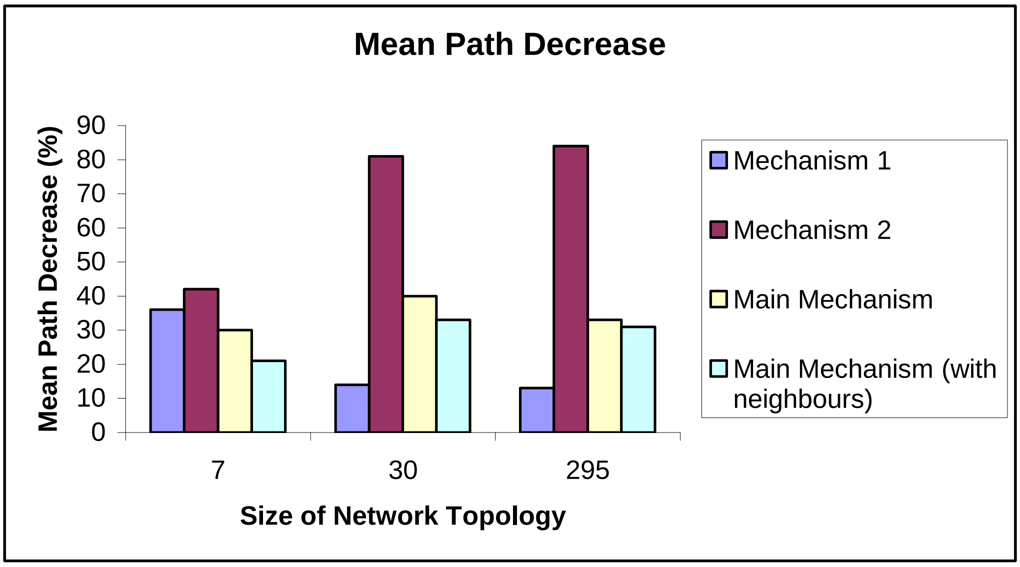

Figure 12: Mean Path Decrease diagram for all different mechanisms

The figure above shows that the second mechanism has the best results as far as mean path decrease is concerned. This though is quite misleading, because mechanism 2 might result in the overloading of the network with data that might not be requested again. Furthermore, this mechanism is based on the multicast feature of the PSI but since the delivery of the data is performed based on zFilters, it is quite possible to have false positives.

Thus, the two cases of our main algorithm have the best results with the minimum impact in all kinds of network topologies and manage to deliver the data over a path that is significantly improved compared to the path that the data are delivered when no Smart Cache exists.

7. Conclusions and Future Work

Throughout this thesis, we investigate mobility in a future clean–slate Internet architecture based on the publish/subscribe model, using the PSI architecture, and we propose a solution to enhance performance without changing the main functionality of the network elements of PSI. We therefore introduced the idea of a caching element to the network, namely the Smart Caches that will assist mobility.

SCs are *in-network caches* defined by the Rendezvous Point based on the topological knowledge of the network, as well as the current and the predicted (or expected) future positions of subscribers of a particular publication (i.e. data item). A SC acts both as an intermediate caching node and as a mediation assistant in the process of exchanging data between the publisher and the subscriber. This idea is in accordance with the PSI architecture and does not introduce a new conceptual component to the initial set of entities of PSI, namely publisher, subscriber and RP. Instead, nodes that are already capable of caching are 'promoted' to also act as publishers and RPs.

We also investigated three different mechanisms for the placement of the SC. It is vital to designate as SC a node that can assist in future movements of the mobile node and enhance the existing mobility performance. The first two algorithms are quite simple and they are not based on the topological knowledge of the network. The first one defines as SC the proxy of the mobile node before it moves while the second one places a SC at every AP the mobile node might move to. The main algorithm that we propose here takes into account the paths from the publisher to a set of Access Points (APs) and finds the common nodes among these paths. Among these nodes we select the node that better fits most of the APs with the minimum number of hops and if there are more than one such nodes, we then decide to make them both SCs and handle the delivery of data to them from the publisher via a multicast tree.

We simulated the PSI network topologies to evaluate the Smart Cache mechanism and the different mechanisms for the Smart Cache selection. These results concluded that we have achieved to improve the performance for mobility in terms of the path length needed for the data to reach the new location of the mobile node.

It is worthwhile to note that even if we have one packet of information to be received by the mobile node, our solution offers a significant improvement to the existing PSI architecture since it guarantees that even if this packet of information is lost, when requested from the new position it will reach the mobile node by a shorter path. This might be even more valuable in cases that the information is requested from a publisher that is not in the same domain (Autonomous System) with the subscriber.

Nevertheless, we should note that it offers a great advantage when bigger chunks of information or flows of information are in question. The main reason for this is that it guarantees that only the information that the mobile node lost while moving will be delivered. Duplicate packets are mainly avoided without the assistance of a transport layer functionality since we do not request these packets at all.

The evaluation of our proposal allows us to consider that we have already achieved an improvement for the mobility support in PSI. What remains to be done though is to consider a better evaluation of the mechanisms and their implications. Furthermore, we always consider throughout this work that each link costs the same for the ISP. It should be investigated, whether the results are the same if we introduce a more realistic model. The caching cost might be one of the parameters that could define cost.

Another parameter that we took as an assumption is that each mobile node moves from AP to AP randomly. In reality though, the mobile node moves to different APs with different probabilities. The idea of neighbours allowed us to simulate this case by having a preselected set of APs, where the mobile mode is more likely to move. We did not base this set on historical data of movement of the APs nodes. It remains to be investigated how these extra data, that would better predict future positions, would affect our results.

Finally, it is quite important to decide and investigate whether it would be a good solution to enhance the Smart Cache placement algorithms so that they take into account the RVP. The question here is whether we need to take off the RVP the load of the multiple subscriptions of the mobile node, or we do prefer to avoid changing the functionality of the network elements.

8. References

[1] PURSUIT: Web site, 2010. http://www.fp7-pursuit.eu/

[2] PSIRP : Web site, 2010. http://www.psirp .org

[3] CCN: Web site, 2010. http://www.ccnx.org

[4] Fotiou, N., Nikander, P., Trossen, D., and Polyzos, C., G: Developing Information Networking Further: From PSIRP to PURSUIT". Proc. 7th International ICST Conference on Broadband Communications, Networks, and Systems, 2010

[5] Katsaros, K., Fotiou, N., Polyzos, C., G., Xylomenos, G: "Overlay Multicast Assisted Mobility for Future Publish/Subscribe Networks," ICT Mobile Summit, Santander, Spain, June 2009

[6] Stoica, I., Adkins, D., Zhuang, S., Shenker, S., Surana, S: Internet indirection infrastructure. IEEE/ACM Trans. Netw. 12(2), 205-218, 2004

[7] Zhuang, S., Lai, K., Stoica, I., Katz, R., Shenker, S: Host mobility using an internet indirection infrastructure. Wirel. Netw. 11(6), 741-756, 2005

[8] Caesar, M., Condie, T., Kannan, J., Lakshminarayanan, K.,Stoica, I: Rofl: routing on flat labels. In: SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications, pp. 363-374. ACM, New York, NY, USA, 2006

[9] Koponen, T., Chawla, M., Chun, B.G., Ermolinskiy, A., Kim, K.H., Shenker, S., Stoica, I: A data-oriented (and beyond) network architecture. In: SIGCOMM '07: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications, pp. 181-192. ACM, New York, NY, USA, 2007

[10] Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M.F., Briggs, N.H., Braynard, R.L: Networking named content. In: CoNEXT '09: Proceedings of the 5th international conference on Emerging networking experiments and technologies, pp. 1-12. ACM, New York, NY, USA, 2009

[11] Sourlas, V., Paschos, G., Flegkas P., Tasioulas, L: Mobility support through caching in content-based publish/subscribe networks. 10th IEEE/ACM International Conference on Cluster, Cloud and GRId Computing, 2010

[12] Gaddah, A. Kunz, T., "Extending mobility to publish/subscribe systems using a proactive caching approach". Mobile Information Systems 6 (2010) 293-324, 2010

[13] Eugster, P.T., Felber, P.A., Guerraoui, R., Kermarrec, A.M: The many faces of publish/subscribe. ACM Comput. Surv. 35(2), 114-131, 2003

[14] Tsilopoulos, C., Makris, D., Xylomenos, G., "Bootstrapping a Publish/Subscribe Information Centric Network", to appear in Future Network & Mobile Summit, 2011

[15] Ain, M., Tarkoma, S., Trossen, D., Nikander, P: "FP7-PSIRP Deliverable 2.2: Conceptual Architecture of PSIRP Including Subcomponent Descriptions", p. 38, 2008

[16] M. Sarela, T. Rinta-aho, S. Tarkoma, RTFM: Publish/Subscribe Internetworking Architecture, In ICT Mobile Summit, 2008

[17] P. Jokela, A. Zahemszky, C. Esteve Rothenberg, S. Arianfar, and P. Nikander, LIPSIN: line speed publish/subscribe inter-networking, In ACM SIGCOMM 2009

[18] Fotiou, N., Marias, F., G., and Polyzos, C., G: Security Aspects of a Clean Slate Information Oriented Internet Architecture, Proc. 21st International Tyrrhenian Workshop on Digital Communications, 2010

[19] Rocketfuel : An ISP Topology Mapping Engine, http://www.cs.washington.edu/research/networking/rocketfuel