Proceedings of the

2nd International Workshop on Information Management for Mobile Applications

Message from the Workshop Chairs

Mobile devices pose tremendous challenges to the design and implementation of information systems suited for mobile environments. While users expect similar functionality on their smart-phone as provided on their laptop or desktop computer, the hardware and communication platforms are still limited. Especially, data-intensive mobile applications require new ways of data management, processing, and analysis. Crucial issues include energy-efficiency, limited CPU power and storage, real-time processing, small displays, and communication costs.

The International Workshop on Information Management for Mobile Applications (IMMoA'12) targets to provide a forum for researchers and practitioners to present, share, and discuss insights, advancements, and challenges in technologies and mechanisms which support the management of mobile, complex, integrated, distributed, and heterogeneous data-focused applications.

The workshop aims in particular at the challenges of managing complex data and data models in a mobile context. Mobile data management has become an important research area in the field of data management. Therefore, the 38th International Conference on Very Large Databases provides an interesting forum to discuss recent advancements in mobile data management research. The limitations but also new opportunities of mobile devices initiated new research topics which are addressed in this workshop.

We received more than ten high quality submissions of which we could accept five as full papers and and one as short paper. The papers have been peer-reviewed by three to four reviewers each. The accepted papers discuss data management techniques for distributed mobile applications and peer-to-peer systems, integration of distributed mobile data sources, mechanisms for privacy preservation, and context-based systems. The workshop program is completed by an invited talk entitled "Dynamic Context Management for Mobile Applications" given by Prof. Dr. Daniela Nicklas from Oldenburg University.

We hope that the workshop initiates inspiring and fruitful discussions and that the second edition will be as successful as the first workshop held in 2011. We would like to thank the DFG Research Cluster Ultra High-Speed Mobile Information and Communication (UMIC, `http://www.umic.rwth-aachen.de`) at RWTH Aachen University, Germany, for their support in organizing this event. Furthermore, we would like to thank the reviewers for their good work.

Thierry Delot
University of Valenciennes & Inria Lille

Sandra Geisler
RWTH Aachen University

Christoph Quix
RWTH Aachen University

Bo Xu
University of Illinois at Chicago

Workshop Chairs and Program Committee

Workshop Chairs

- Thierry Delot, Dept. of Computer Science, University of Valenciennes & Inria Lille, France
- Sandra Geisler, Information Systems, RWTH Aachen University, Germany
- Christoph Quix, Information Systems, RWTH Aachen University, Germany
- Bo Xu, Department of Computer Science, University of Illinois at Chicago

Program Committee

Yuan An, Drexel University, USA	Jochen Meyer, OFFIS, Germany
Ying Cai, Iowa State University, USA	Nathalie Mitton, INRIA Lille, France
Xin Chen, NavTeq, USA	Xinzheng Niu, University of Electronic Science and Technology of China, China
Hyung-Ju Cho, Ajou University, Korea	Aris Ouksel, University of Illinois at Chicago, USA
Christine Collet, Grenoble INP, France	Claudia Plant, Florida State University, USA
Bruno Defude, Télécom & Management Sud-Paris, France	Florence Sèdes, University Paul Sabatier Toulouse, France
Marwan Hassani, RWTH Aachen University, Germany	David Taniar, Monash University, Australia
Hideki Hayashi, Hitachi Central Research Laboratory, Japan	Masaaki Tanizaki, Hitachi Central Research Laboratory, Japan
Sergio Ilarri, University of Zaragoza, Spain	Goce Trajcevski, Northwestern University, USA
David Kensche, Thinking Networks AG, Germany	Upkar Varshney, Georgia State University, USA
Xiangyang Li, Illinois Institute of Technology, USA	Jari Veijalainen, University of Jyväskylä, Finland
Andreas Lorenz, Deutsche Telekom AG, Germany	Ouri Wolfson, University of Illinois at Chicago, USA
Doug Lundquist, University of Illinois at Chicago, USA	José-Luis Zechinelli-Martini, Universidad de Las Americas Puebla, Mexico
Sanjay Madria, Missouri University of Science and Technology, USA	Xianggang Zhang, University of Electronic Science and Technology of China, China

Table of Contents

Keynote

Dynamic Context Management for Mobile Applications

Daniela Nicklas

Carl von Ossietzky Universität Oldenburg, Germany

Abstract: With the upcoming widespread availability of sensors, more and more applications depend on physical phenomena. Up-to-date real world information is embedded into business processes, in production environments, or in mobile applications, so that such context-aware applications can adapt their behavior to the current situation of their user or environment. For this, a high variety of so-called context information has to be managed, often in an push-based way (applications register for context changes). In more and more applications, the amount, the physical source distribution, the resources of the processing devices, and/or the update rate of the incoming sensor data prevents its storage in DBMS and the use of triggers or periodic queries. This talk shows how data stream management techniques can be used to provide an efficient, comprehensive, up-to-date, and even quality-annotated dynamic context model for mobile applications.

A Hierarchical Approach to Resource Awareness in DHTs for Mobile Data Management

Liz Ribe-Baumann
Ilmenau University of Technology
Ilmenau, Germany
liz.ribe-baumann@tu-ilmenau.de

Kai-Uwe Sattler
Ilmenau University of Technology
Ilmenau, Germany
kus@tu-ilmenau.de

ABSTRACT

Data is increasingly distributed across networks of mobile nodes such as wireless sensor networks, distributed smartphone applications, or ad hoc recovery networks in disaster scenarios, but must still be reliably collected, stored, and retrieved. While such networks run in either ad hoc mode or use existing infrastructure, all of them must deal with node heterogeneity. Wireless nodes invariably have differing levels of power availability, and often varying connectivity and computing power. While many distributed hash tables (DHTs) have been designed for mobile ad hoc or heterogeneous networks, they do not consider differences in node strength, or *resource availability*, for an arbitrary number of resource availability levels. In this paper, we present a scalable, location aware, hierarchical DHT that utilizes nodes' varying resource availability levels to increase and prolong the mobile network's data storage and retrieval capabilities. Furthermore, we compare this DHT to other location aware flat and hierarchical approaches, examining their structures' suitability for nodes with varying resource availability.

1. INTRODUCTION

Today, mobile applications are no longer restricted to the classic client/server architecture relying on a backbone infrastructure. Instead, an increasing number of applications for smartphones (e.g. mobile games, content sharing) as well as wireless sensor network applications follow a serverless ad-hoc model of interaction and data exchange. Even if such applications require a server or gateway to initially fetch some data or to finally publish results, data has to be collected, exchanged, and stored for some time in the network. From a data management point of view, this poses two challenges: (1) to *efficiently* manage and retrieve data in a distributed way and (2) to *reliably* provide the data while taking possible node failures and resource restrictions (connectivity, battery power) into account.

Distributed hash tables (DHT) for mobile peer-to-peer (P2P) networks have been proposed in the past to address the first challenge. However, constructing a distributed hash table on a mobile P2P network, where nodes have restricted battery power and communication costs the peers vital energy, is a definite challenge. While traditional DHTs such as the Content Addressable Network (CAN) [24] and Chord [29] may be reliable on a network without such communication restraints, they fail to take the important differences in nodes' resource availabilities into account for resource sensitive dynamic networks. And while today's large mobile networks are based on smartphones, laptops, etc. which use an intact backbone infrastructure that nodes need not consider, the use of DHTs on ad-hoc networks should be considered for the near future.

A fundamental operation for any kind of data management task (store, update, retrieve) in a DHT is the key lookup operation. To implement this operation in a (mobile) ad-hoc network, overlay nodes also forward messages on the network layer, in which case a long distance overlay hop may require many forwarding nodes, while a short distance lookup hop may be completed with few network hops. This case brings the additional challenge of routing distances, giving the physical distance that lookups traverse a central role in a network's ability to survive its load. Thus, power and (in the future) location awareness are important for DHTs running on mobile networks.

Consider for example a large network based on smartphones, laptops and a limited number of servers that cooperatively maintain a DHT, with each node storing some of the global application data. As long as the load is balanced between the nodes, this network is inherently scalable - each of the nodes is responsible for fetching and storing a portion of the network's information and for routing and processing a portion of the network's requests. While the numerous smartphones jointly provide a large portion of the network's storage and routing capabilities, a single smartphone has a restricted amount of battery power available in a give time frame (until it is recharged). Thus, the network must find a way to incorporate each of these weaker nodes' resources in order to provide scalability without causing failure by overuse. In this paper, we address the second of the above mentioned challenges by examining several approaches to balancing maintenance and routing load according to node's resource (i.e. power) availabilities and locations. None of these approaches acts blindly with regard to either resources or locations, yet each has clear limitation with regard to how much it can incorporate. Since we could, for example, consider a node's restricted computing power or bandwidth availability instead of its power availability, we consider the abstract notion of a node's "resource availability."

The three major approaches towards addressing heterogeneous node capabilities in DHTs - hierarchical DHT structures, virtual nodes, and node movements within the identifier space - are typically not well adept to our scenario. Hierarchical approaches, typically with two tiers [3, 13, 36], offer large reductions in the load on leaf nodes but overuse peers with restricted resources which act as super peers (in order to assure the system's scalability). On the other hand, virtual nodes and node movement approaches, which vary the quantity of data at each physical node, actually introduce more maintenance overhead and churn while assuming that higher resource availability implies larger storage capacity. However, since the main communication overhead in a DHT comes from maintenance, we are most interested in balancing the network maintenance and lookup routing load to nodes' resource availabilities.

We compare four different structural resource and location aware DHT approaches in this paper: (a) a novel, multi-tiered hierarchical approach, (b) a two-tiered hierarchical approach, (c) a flat resource and location aware approach [27], and (d) a (novel) hybrid approach between (a) and (c). While all three hierarchical approaches treat the lowest level nodes as leaf nodes, approach (a) constructs multiple upper tiers, approach (b) uses location aware DHash++ [9] for the upper tier, and approach (d) uses the location and resource aware Chord extension RBFM [27] for the upper tier. We examine each network's ability to store and retrieve data, as influenced by the nodes' lifetimes and the percentage of deliverable lookups. This paper's main contributions are:

- A novel location aware hierarchical DHT for an arbitrary number of resource availability levels and
- a simulated comparison of network robustness for four flat and hierarchical resource and location aware approaches.

We discuss related work in Section 2; explain our network assumptions and foundations in Section 3; describe our novel DHTs and consider the routing complexity of the approaches in Section 4; and compare the four DHTs using simulation in Section 5.

2. RELATED WORK

The DHT forerunners such as CAN (Content Addressable Network) [24], Chord [29], and Kademlia [21] use efficient routing but were not designed to run on mobile nodes where both location and available resources play important roles. Proximity-awareness in DHTs is generally classified as proximity-aware identifier selection (PIS, such as Mithos [31] and SAT-Match [26]), proximity-aware neighbor selection (PNS, such as DHash++ [9]), proximity-aware route selection (PRS, such as Tapestry [35]), or a combination thereof and is primarily directed at reducing overall traffic or average round trip times [16]. Proximity-awareness has gained interest in many areas related to DHTs, including caching and replication protocols and hybrid overlays [10, 20].

The three main approaches for balancing load in heterogeneous DHTs are the use of hierarchical DHT structures, virtual nodes, and node movements within the identifier space. Hierarchical DHTs often group nodes by some defining characteristic such as group associations (e.g. departments within a university) or peer capabilities ("have" or "have not"). Systems with group structures such as Canon [12], Hieras [32], and Cyclone [2] tend to route lookups as far as possible within one group before forwarding them on to a different (often hierarchically higher) group. In contrast, hierarchical DHTs based on two-tier peer capabilities [3, 36], where nodes assume the roles of super-peer or leaf-peer, route lookups directly from leaf nodes to parent nodes, rendering the parent nodes fully responsible for performing lookup routing and neglecting the varying nuances of nodes' resource availabilities. A combination of the group and two-tiered capabilities structures has also been suggested in [13], such that weak peers are arranged in disparate DHTs controlled by super-peers which form their own DHT and are responsible for routing lookups to the correct group.

Virtual nodes pose a different solution, with each physical network node balancing its load independently by hosting a varying number of virtual overlay nodes, each with its own set of keys and links [15, 18]. Similar to virtual nodes, node movements within the identifier space achieve load balance by adjusting the data that each node stores [5, 11]. Nodes with low load choose new nodeIDs that are close to nodes with high load, thus taking over some of their load but creating a large amount of churn.

DHTs for mobile ad hoc networks (MANETs) pose additional challenges since each overlay hop represents multiple underlay hops on DHT nodes, underlay routing to unknown destinations often results in broadcast messages, nodes' mobility causes frequent changes in "good" routes, and high node churn due to short uptimes (and dwindling resources) requires highly dynamic protocols for overlay maintenance and data persistence. Among the first DHTs suggested for MANETs were Ekta [23] which uses underlay routing information to choose links (PNS) and make overlay routing decisions (PRS), MADPatry [34] which uses landmark nodes to form location-based node clusters that share nodeID prefixes (PIS with node movements), and CHR [1] in which geographic clusters act as nodes using geographic routing as in GHT [25]. Basically, DHTs in MANETs employ some combination of cross-layer PIS [22, 14, 30, 33], PRS [19], and PNS [7], using network layer (i.e. underlay) information to augment overlay decisions. Many of these overlays can be considered hierarchical [22, 30, 33, 34], in part due to their clustered structures.

While proximity-awareness plays a central role in the development of DHTs for wireless networks, the treatment of nodes' heterogeneity also effects the robustness of the final system, especially when considering nodes' power availability and connectivity. The numerous mentioned DHT substrates handle node heterogeneity differently - ignoring it completely or incorporating it in a flat or hierarchical manner - but there exists limited work comparing these various approaches. Furthermore, little has been done to treat nodes with varying resource availabilities in DHTs with mobile nodes on a finer scale than strong or week, as in the Chord extension RBFM [27].

3. MOBILE NODE RESOURCE MODEL

Coordinates. We assume that each node x has sufficiently correct two dimensional virtual (not necessarily geographical) network coordinates (x_1, x_2), such as used in Vivaldi [8] for determining latencies. The *physical distance* $d_{phy}(x, y) := \sqrt{(x_1 - y_1)^2 + (x_2 + y_2)^2}$ between two nodes x and y and should reflect round trip times, the number of underlay hops, or some other meaningful distance between nodes.

Resources. We model our analysis and simulations after nodes with varying power availability - from smartphones with very limited power to servers with an inexhaustible power source - but the proposed protocol need not be restricted to this use case. It only requires that each node x has a resource availability that can be expressed as an integer value $x_R \in \{0, 1, \ldots, l_{max}\}$ for some fixed maximum level l_{max}. We assume that $x_R = 0$ is the lowest possible resource level (but still operational) while $x_R = l_{max}$ implies unbounded resources. Note that resource levels must be globally defined so that a given resource level on differing node types is comparable. If we consider power availability with $l_{max} = 3$, that could mean that a handheld operating on battery power may have resource level two when fully charged, but a cell phone with a weaker battery may only reach a resource level one when fully charged.

For our simulation, we use a Zipf distribution for nodes' resource levels, reflecting trends for node lifetime found in peer-to-peer networks, where node lifetime tends to follow a heavy-tailed Pareto distribution [6, 28] (the continuous counterpart of the Zipf distribution). The probability that a random node has resource level $\ell \in \{0, 1, \ldots, l_{max}\}$ depends on the power m of the Zipf-distribution:

$$p_\ell := P(x_R = \ell) = \frac{1}{(\ell+1)^m} \cdot \frac{1}{\sum_{j=0}^{l_{max}} 1/(j+1)^m}. \qquad (1)$$

Failures. For our simulation, we assume that failures are due to nodes' resources being depleted by node activity. Each send and receive activity drains a node's resources until the node fails (based on nodes with varying power availability). We use asymmetrical drain patterns, with a send costing more than a receive, but constant drain for all but the top level, which is not drained at all. The selected resource and drain values are abstract and serve as a benchmark to compare the protocols as opposed to assessing the real word battery runtimes. To provide results with as few dependencies as possible, we take a simplistic approach to churn in our evaluation, with nodes drained until they fail but no additional nodes added. The system runs until it has been reduced to half of the original nodes.

DHT Foundation. All of the approaches we use are based on Chord [29] mainly because Chord has a rather simplistic structure that adapts well to location awareness [16] and is the basis of the location aware DHash++. Analogous to Chord, we use consistent hashing [17] to distribute keys to nodes. Each node x chooses a random (or hashed) nodeID x_{ID} from the binary key space $0 \ldots 2^m - 1$, which is viewed as a ring with key values increasing in a clockwise direction. These completely random nodeIDs ensure a scalable key distribution. Each node positions itself at its nodeID on the key ring and establishes links to its immediate predecessor and successor as well as a successor list with its r nearest successors, making repairs possible after unexpected node failures. Each node x maintains a *simple key range x.srange*, which spans the keys between its predecessor y's key (exclusive) and its own key, or $x.srange = (y_{ID}, x_{ID}]$. Thus, each key κ is assigned to the first node whose nodeID is equal to or succeeds κ on the key ring, or that node whose simple key range contains κ. The asymmetric *key distance* from a node x (or key) to a node y (or key) via their nodeIDs is:

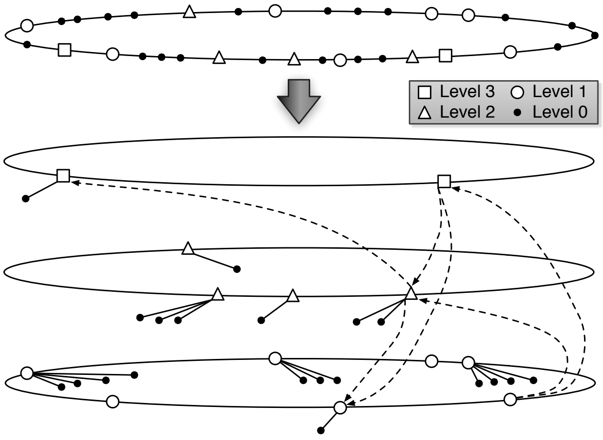


Figure 1: All resource levels shown on top key ring. Nodes within hierarchical layers below: upper layers form DHash++ overlays, links between layers, and leaf nodes assigned to upper level predecessors.

Key Distance 1. The key distance from x to y is the clockwise distance on the key ring from x_{ID} to y_{ID}: $d_{key}(x, y) := y_{ID} - x_{ID} \mod 2^m$.

4. HIERARCHICAL OVERLAY

We call our hierarchical approach Hierarchical Resource Management (HRM), where nodes are separated into different levels and maintain links within and between those levels. The lowest level, consisting of the weakest nodes, functions as a leaf level where each leaf node maintains a parent node from some upper level. Each level is linked together to form a location aware DHash++ [9], with the number of shortcut links determined by the given hierarchy level. Thus, the higher a node's resource availability, the more links it is expected to maintain, so that weaker nodes' maintenance loads are significantly reduced despite their additional load as parent nodes. We assume that all nodes play similarly important roles in data storage and retrieval, thus, we do not address heterogeneous data distribution or the necessary replication protocols in this paper.

In the following we refer to *bottom level* nodes with resource level = 0, *upper level* nodes with resource level > 0, *top level* nodes with resource level = l_{max}, and lower level nodes with resource level < l_{max}. In addition to each node's links to its predecessor and first r successors, we have three additional types of links: leaf-parent links between bottom and upper level nodes; inter-level links which connect each upper level node with its immediate successor in each of the $l_{max} - 1$ upper levels; and level fingers which provide each upper level node shortcuts within its own resource level.

4.1 Links.

Each node is responsible for the keys in its simple key range, but we also consider an upper level node's *upper key range* which contains all of its leaf nodes' nodeIDs and is integral to successful routing. For this, each node maintains upper level successor and predecessor nodes, i.e. the first successor and predecessor nodes from any upper level. Then a node x's *upper key range x.urange* consists of all keys between x_{ID} and its upper level successor's key. Note that *x.srange* and *x.urange* overlap only in x_{ID}.

ProspectiveLinks[x_R][i]

NodeID	Physical Distance
p_2ID	0.4
p_3ID	0.9
p_4ID	0.9
p_1ID	1.7

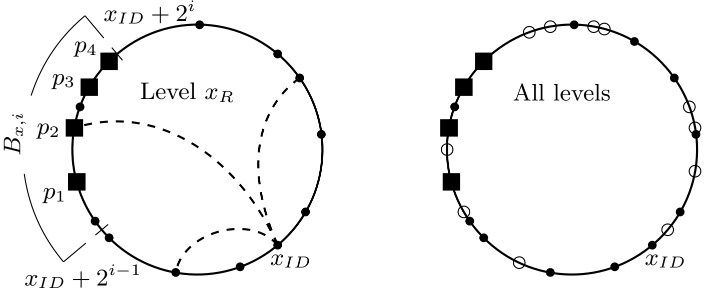


Figure 2: Key ring at right shown for node x with all nodes not in level x_R as hollow circles and at left with level x_R nodes only: six nodes in $B_{x,i}$, four of which x knows in its x_R prospective links list (squares). A level finger is established to p_2, the known node with the best physical distance to x in level x_R.

Leaf Links. Each bottom level node x ($x_r = 0$) maintains a link to its *parent node* $\pi(x)$, which is the first upper node preceding x in the keyspace. Thus, leaf nodes have parents from varying resource levels. Leaf nodes have neither inter-level links nor level fingers.

Inter-level Links. Each upper level node x establishes a link $x.I[\ell]$ to its direct successor $x.I[\ell].node$ in each of the upper levels ℓ.

Level Fingers. In DHash++, each node x with nodeID x_{ID} chooses one link - or finger - $x.F[i]$ per finger interval $B_{x,i} := [x_{ID} + 2^{i-1}, x_{ID} + 2^i)$ for $i \in \{1, 2, \ldots, m\}$. The corresponding node that $x.F[i]$ points to is notated $x.F[i].node$. In our protocol, a node x only choses fingers within the same resource level, i.e. $(x.F[i].node)_R = x_R$. Furthermore, the number of fingers that a lower level node establishes varies from level to level.

A level 1 node x has as few fingers as necessary, establishing level fingers only to nodes which are closer successors in the keyspace than x's closest higher level inter-level link. That means:

$$d_{key}(x, x.F[i].node) < d_{key}(x, x.I.closestHigher). \qquad (2)$$

Where $x.I.closestHigher$ is the closest of x's higher level inter-level links, $x.I.closestLevel$ is $x.I.closestHigher$'s resource level, and $x.I.closestInt$ is the finger interval in which $x.I.closestHigher$ is found:

$$x.I.closestLevel := \underset{\ell : x_R < \ell \leq l_{max}}{\operatorname{argmax}} d_{key}(x, x.I[\ell].node)$$
$$x.I.closestHigher := x.I[x.I.closestLevel].node$$
$$x.I.closestInt := j : x.I.closestHigher \in B_{x,j}.$$

Meanwhile, level l_{max} and $l_{max} - 1$ nodes maintain fingers for each finger interval $B_{x,i} := [x_{ID} + 2^{i-1}, x_{ID} + 2^i)$ for $i \in \{1, 2, \ldots, m\}$. Nodes in additional levels ℓ with $1 < \ell < l_{max} - 1$ maintain sets of fingers of varying sizes, depending on ℓ. We let $x.Finterval \in \{1, 2, \ldots, m\}$ be the furthest finger interval in which a node x maintains a finger and $x.Fkey = x_{ID} + 2^{Finterval-1}$ its corresponding key value. For example, given five levels ($l_{max} = 4$), we might have:

$$x.Finterval = \begin{cases} x.I.closestInt, \; x_R = 1 \\ m - 1, \; x_R = 2 \\ m, \; x_R \in \{3, 4\}. \end{cases}$$

Thus, each upper level node x maintains a *finger table* with (at most) one finger for each finger interval $B_{x,i}$ with $i \in \{1, 2, \ldots, x.Finterval\}$. Note that the fewer links a level maintains, the less maintenance load is incurred and the faster messages are passed on to other (higher) levels. Lookups are thus routed quickly out of the bottom layers and dispersed between the upper layers.

Level fingers are chosen in a location aware fashion as in DHash++. Nodes' coordinates and resource levels are piggybacked on network messages, providing node information to other nodes at minimal overhead. Thus, an upper level node x choses for $x.F[i]$ that known node with resource level x_R in the finger interval $B_{x,i}$ which has the smallest physical distance to x. For this, x maintains a set of l_{max} *prospective links* lists, one for each resource level, with the ℓ^{th} prospective links list containing a list of the closest (in terms of physical distance) k nodes in $B_{x,i}$ for each $i \in \{1, 2, \ldots, m\}$ from level ℓ which are known to x. At most k nodes in $B_{x,i}$ are saved via their resource levels, nodeIDs, and physical distances, so we have at most $k \cdot m \cdot l_{max}$ saved nodes.

When x receives a message that originated at sender y, x uses y's coordinates to determine $d_{phy}(x, y)$ and update its level y_R prospective links list accordingly (see Figure 3). An i^{th}-finger request is sent to the closest entry in x's level x_R prospective links list for $B_{x,i}$, if it contains an entry (see Figure 2). Otherwise, the first successor of key $x_{ID}+2^i-1$ in resource level x_R is contacted (see Figure 3), which requires level-specific lookup forwarding (see Routing). Upon node x's receipt of a finger request response from node y, if $y_R = x_R$ and $y \in B_{x,i}$ with $i \leq x.Finterval$, then y is assigned to $x.F[i]$. If a given finger interval contains no level x_R node, then this finger entry remains empty.

Note that while node x only links level fingers to nodes in level x_R, maintaining prospective links lists for multiple levels causes little overhead while easing a node's transition between resource levels. The prospective links list entries are continually updated with fresh node information to automatically adapt the network to changing coordinates and are deleted once used for a finger request to ensure their freshness. Simulations have shown that $k = 1$ is beneficial in networks with high churn, reducing the use of failed prospective links and minimizing the lists' overhead.

Figure 1 shows the basic overlay stucture. The connected key ring on which each node establishes its predecessor and successor is shown on top, and the individual levels are shown below with the bottom level nodes assigned to their upper level predecessors (i.e. parents). Inter-level links are shown for three nodes only and level fingers were omitted.

4.2 Node Joins and Failures.

To join the DHT, a node x must have valid network coordinates, choose a nodeID and resource level, and contact

```
procedure maintainFinger(finger)
    lookupKey = myKey + getOffset(finger)
    myLevelList = prospLinkList(myLevel)
    if myLevelList.size(finger) > 0 thenlevel
        listEntry = myList.getClosestEntry(finger)
        lookupKey = listEntry.key
        myList.removeUsedEntry(listEntry)
    end if
    sendLookup(lookupKey)
end procedure
```

```
procedure suggestProspectiveLink(nodeInfo)
    finger = getFingerInterval(nodeInfo.key)
    dist = getPhysicalDist(nodeInfo.coordinates)
    level = nodeInfo.resourceLevel
    if pLList(level).contains(finger, nodeInfo.key) then
        pLList(level).update(finger, dist, nodeInfo)
    else if pLList(level).size(finger) < k or dist < pLList(level).farthestLinkDist(finger) then
        pLList(level).addNode(finger, dist, nodeInfo)
        while pLList(level).size(finger) > k do
            pLList(level).removeFarthestLink(finger)
        end while
    end if
end procedure
```

Figure 3: Maintaining fingers 1 to $m-1$; Updating prospective links lists (pLList) with $\leq k$ entries

one participating node. Once x has established links to its immediate predecessor p and successor s on the key ring, s sends its successor lists to x, which x uses to initialize its own list, and corresponding keys are transfered from s to x. Once x has completed the basic join in the overlay, it must also perform either a leaf join or upper level join (see below). The node x continually updates its *prospective links* lists and periodically performs finger maintenance (see Figure 3) to establish and maintain its fingers.

The basic reaction to node failures is as in Chord, with failed nodes also removed from the inter-level list, prospective link lists, and potentially the parent link or leaf list once their failure is noticed. An upper level node leaves gracefully by sending messages to each of its leaf nodes and its upper level predecessor informing them of their new parents/leaf nodes. Otherwise, if a leaf node's parent has fails unexpectedly, the leaf node must perform a leaf join to reestablish a parent (see below).

Upper Level Joins and Failures. The upper level join serves two purposes: establishing the upper level successor and predecessor nodes (from any upper level) and transferring the responsibility for leaf nodes. Node x uses an upper level bootstrap node to send an upper level join message, which is routed along the upper level nodes to x's upper level predecessor y, for which $x_{ID} \in y.urange$. Node y responds to x with its own upper level successor z and the list of y's leaf nodes which are now in $x.urange$. Then y informs each of these leaf nodes of their new parent node x and removes them from its leaf list.

If an upper level node's resource level is reduced to the lowest level, it becomes a leaf node and forfeits its role as parent node by transferring its leaf nodes to its upper level predecessor y. Leaf nodes ignore finger and inter-link requests, upper and leaf join requests, and upper stabilize requests. If it is observed that a node has left the upper levels, it must be removed from inter-level list, prospective link lists, upper level successor and predecessor links, and parent links (but not from successor and predecessor links).

Leaf Joins and Failures. Nodes with resource level 0 perform leaf joins to establish a live parent node. Recall that a node x's parent is the first preceding upper level node on the key ring. The message is forwarded to an upper level bootstrap node and then routed to the upper level node whose upper level key range contains x_{ID}. This parent node responds and and enters x into its leaf list.

Maintenance. Given the dynamics of mobile networks, maintenance is integral for detecting and addressing network changes. Inter-level links and level fingers are maintained similar to in RBFM with varying maintenance intervals and are automatically adapted when nodes change resource levels. Thus, each link is maintained at an interval that depend on the link's node's resource level: Bottom level links are maintained according to a reference interval t_{ref} and higher level links at varying multiples of t_{ref} for each resource level.

However, leaf and parent links as well as upper level successor and predecessor links are maintained analogously to direct successor and predecessor links, using direct maintenance messages at a fixed interval.

4.3 Lookup Routing.

Routing of lookups is not performed in a strictly greedy fashion like Chord, but rather in a series of greedy steps. Let κ be the message's destination key. Now recall that a message is destined for the node whose simple key range contains κ. One negligible piece of information is added to messages: the key of the last upper level node that handled the message (only needed for high churn scenarios). Once a node x has determined that $\kappa \notin x.srange$, its routing behavior depends on its resource level as follows:

Leaf Nodes. If a message originates at a leaf node x, it is forwarded to the parent node. Otherwise, if x receives a messages for which its parent node was not the last upper level node to have handled the message, it forwards the message to its parent node. Otherwise, it forwards the message to its successor node y if $\kappa \in y.srange$ or else to the closest preceding node from its successor list.

Upper Level Nodes. An upper level node x routes greedily to the closest preceding node in a resource level $\geq x_R$ using both its finger table and inter-level list. For lower level nodes with restricted finger tables, this means that messages to 'distant' destinations are routed to upward. For top level nodes, this means that messages are routed to the closest top level predecessor of κ.

If there is no closer predecessor node with resource level $\geq x_R$, then the message is routed back down the hierarchy by choosing the highest possible inter-level link preceding κ. If there are no such links, then $\kappa \in x.urange$ and the message is delivered directly to the node y with $\kappa \in y.srange$: y is either x's upper level successor (whose simple key range overlaps $x.srange$) or one of x's leaf nodes.

Level Routing. Finger requests and inter-level link requests use level sensitive overlay routing. These requests are not necessarily delivered to the node x for which $\kappa \in$

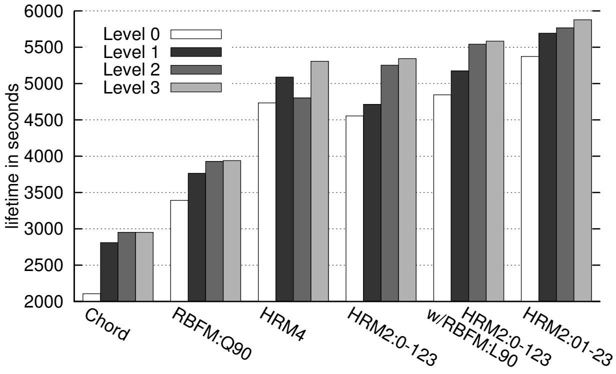


Figure 4: Mean node lifetimes per resource level until half of the system nodes have failed due to resource drain. Level 3 is also total system lifetime.

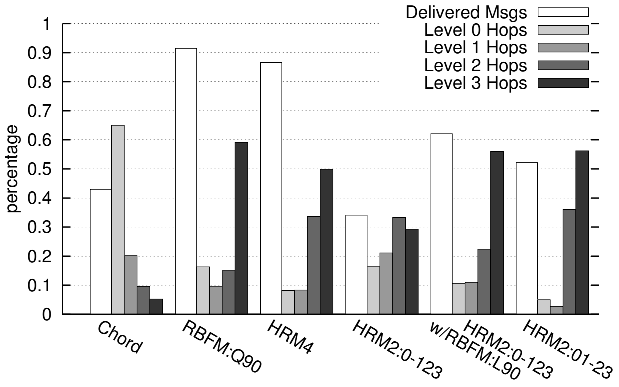


Figure 5: With node failures: The percentage of delivered application messages and percentage of total lookup hops resolved per resource level.

$x.srange$, but rather to the first successor of κ in a given level ℓ. A node y considers itself the request's destination if $y_R = \ell$ and κ is between the sending node's nodeID and y_{ID}. If not, y forwards to its level ℓ inter-level link if it succeeds κ, or to its closest known preceding link in level ℓ.

Lookup Hop Length Theorem 1. *Given a network with N nodes, the expected upper bound for the number of overlay hops required for a lookup from any node to the successor node of any key is $O(\log(N))$ hops.*

Proof. To show that routing terminates and find an upper bound on the routing hops, we consider the farthest possible lookup route. Since it takes at most one hop to route from a leaf to a parent node, we assume that each message originates at an upper level node. Furthermore, since it takes at most one hop to reach the node y with $\kappa \in y.srange$ from κ's upper level predecessor, we need only determine the number of hops necessary to reach κ's upper level predecessor.

If the originating node, its successor, or its upper level successor are the destination, then we are done. Otherwise, the message is passed upwards until it reaches the first level in which the destination key is a predecessor of the current node x's $x.Fkey$, i.e. $d_{key}(x, \kappa) < d_{key}(x, x.Fkey)$. The message is passed up at most $l_{max}-1$ levels, from the bottom to the top level.

So assume that κ is a predecessor of $x.Fkey$ and will thus be routed within level $\lambda := x_R$ on level fingers until it has reached either the destination node or the closest predecessor node within level x_R. Then, as shown in [27], the routing complexity within level x_R is at most $O(\log(x.N))$, where $x.N$ is the number of nodes in level x_R between x and $x.Fkey$. Let y be κ's closest predecessor node in this level.

Assuming the message has not reached its destination, the message is passed at most once to each of the other upper levels and routed analogously (starting at higher levels first). So we have a total of $l_{max}-1$ remaining hops between other upper levels. However, since we know that y is the closest predecessor node to κ in level λ, we can use the expected number of network nodes between any two level λ nodes (some constant c) as an upper bound on the number of nodes on which remain to route over in each level. Thus, we expect at most $O(\log(c))$ routing hops in each remaining level $\ell > 0, \ell \neq \lambda$, giving us an expected total of at most:

$$\begin{aligned} 2 + 2(l_{max} - 1) + &O(\log(x.N)) + (l_{max} - 2)O(\log(c)) \\ \leq &2 \cdot l_{max} + \hat{c} + O(\log(N)) \\ = &O(\log(N)) \end{aligned}$$

for with some constant $\hat{c}$. □

Note that our simulation results did show an increase in routing hops compared with Chord (1-2 hops), as expected due to the maximum $2 \cdot l_{max}$ hops up and down the hierarchy.

5. EVALUATION

In order to assess the various approaches' capabilities to store and retrieve data, we observe node and network lifetimes, the ratio of successfully delivered lookups, and the ratio of forwarded lookup hops per resource level. Additionally, as a reflection of each approach's suitability to mobile scenarios, we compare the average physical distance of lookup routing hops. We use two simulation configurations, one without node failures and one in which nodes' resources are drained upon message send and receives, terminating the simulation once half of the nodes have failed (as described in Section 3). We consider here only systems with four resource availability levels ($l_{max} = 3$), leaving considerations about the ideal number of resource levels for future work. We compared the HRM overlay with Chord, RBFM, and three two-tier hierarchical overlays based on Chord, RBFM, and HRM. We found the variations caused by different RBFM finger maintenance intervals and stretch constants insignificant compared to the variations between approaches, so we chose to use fixed RBFM configurations.

5.1 Setup

The simulations were performed in OmNET++ using the OverSim overlay framework [4], using and extending the functionality of the existing Chord implementation. The results are based on 10000 nodes with random coordinates divided into four resource availability levels based on the power two Zipf distribution from (3). The base measurement time was 10000 seconds, but the simulations with node failures were terminated once half of the nodes failed. Nodes started with resource values 800, 200, and 100 for levels 2, 1, and 0 (level 3 nodes are not drained due to their inexhaustible resources) and drained by 0.2 and 0.1 resource

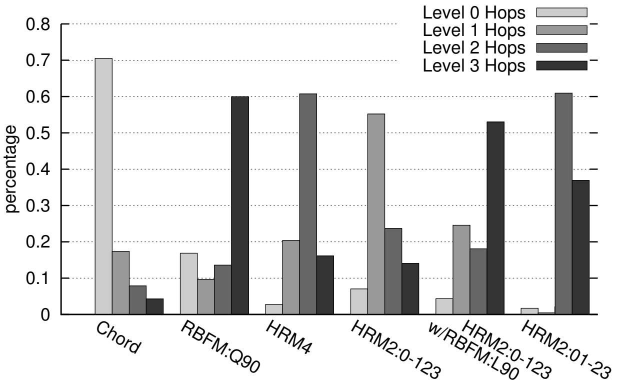


Figure 6: Without node failures: Percentage of total lookup hops resolved per resource level. Each approach delivered more than 99% of lookups.

units for every sent and received message, respectively. In addition to maintenance messages (including stabilize messages every 20 seconds to node successors and parent nodes), dummy application lookups were sent from each node to a random key every 30 seconds. The application lookups were observed for the number of hops, hop distances, and successful delivery. The lower the rate of delivery, the less reliable the DHT stores and retrieves data.

5.2 Hybrid hierarchical DHT

The three different two-tier configurations we simulated in order to compare HRM with classical two-tier approaches consist of parent and leaf nodes. Nodes from each of the four resource availability levels are assigned to either a super peer or a leaf peer hierarchical layer. For HRM2:0-123, level 0 nodes are assigned to the leaf layer and level 1,2, and 3 nodes to the super peer layer. Within the super peer layer, nodes are unaware of their varying resource levels and choose their fingers based only on physical distance. Similarly, HRM2:01-23 assigns level 0 and 1 nodes to the leaf layer and level 2 and 3 nodes to the super peer layer.

In order to add additional resource awareness, we combine HRM with RBFM to obtain a hybrid solution: in HRM2:0-123 with RBFM nodes are again arranged with level 0 nodes in the leaf layer and level 1,2, and 3 nodes in the super peer layer. However, here the super nodes are aware of their different resource levels and choose fingers in a resource and location aware fashion as in RBFM. Thus, the upper level nodes build a simple RBFM overlay on which the bottom level nodes are hung as leaf nodes. The two-tier simulations were configured with linear finger maintenance and for HRM2:0-123 with RBFM using the stretch constant $c = 90$.

5.3 Results

Further simulation configurations include: a relatively infrequent finger maintenance period of 120 seconds for Chord for reduced load, an RBFM overlay with quadratic finger maintenance and stretch constant 90 (RBFM:Q90) (see [27]), and HRM as suggested in this paper with four hierarchical levels and linear finger maintenance (HRM4) with finger intervals (as described in Section 4):

$$x.Finterval = \begin{cases} x.I.closestInt, \ x_R = 1 \\ m, \ x_R \in \{2, 3\}. \end{cases}$$

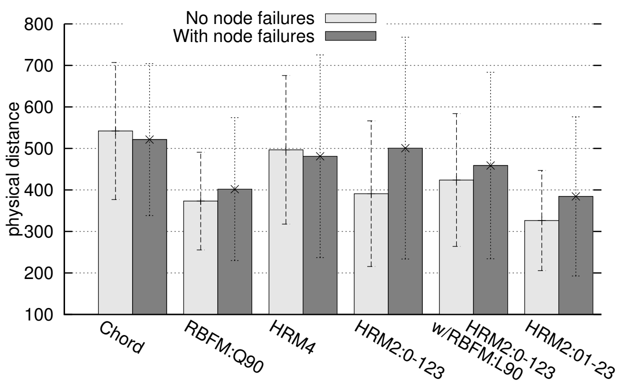


Figure 7: Mean physical distance and standard deviation per lookup routing hop.

The lifetimes of each of the systems before half of its nodes fail, as shown in Figure 4, vary strongly. Despite a very infrequent finger maintenance interval for Chord, set here intentionally to 240 seconds to reduce maintenance load, it cannot compare to the alternative approaches. Note that these results depend on the varying maintenance loads which we do not further discuss for the sake of brevity, but which produce a major portion of network load. Note that while HRM4 performs significantly better than RBFM, the two-tier approaches perform even better, most likely due to the higher average hop length of lookups in HRM (approximately one hop more per lookup). While Figure 4 demonstrates how well the traditional two tier approaches prolong node lifetimes, Figure 5 shows that these two-tier approaches' performance suffers given high failure rates, reducing the success rate of lookups to under 65%. Considering only the node and network lifetimes together with the percentage of delivered lookups, RBFM and HRM4 clearly outperform the other approaches, with HRM4 providing the longer lifetimes.

Figures 5 and 6 also provide an overview of the lookup hop load distribution among the resource levels with and without node failures. Only successful lookups are included in these figures. Figure 6 clearly shows how strongly each approach prefers a single resource level for its lookup hops and demonstrates how important it is to design an overlay with the nodes' resource availabilities in mind. For example, RBFM's lookup hop distribution is more suitable for networks in which top level nodes can handle unlimited load while HRM4 is more suitable for systems with strong nodes in level 2 that should be used to reduce top level load.

Average lookup lengths ranged from 6.5 to 8.5 hops, with the hierarchical approaches tending to have around one hop more than the other approaches, presumably from the final and/or initial hops to and from leaf nodes. Note the high load on level 1 nodes for the two-tier approach HRM2:0-123 in Figure 6. We infer that this causes a high rate of node movement between the super peer and leaf layers, triggering HRM2:0-123's low deliverabiliy rate. Similarly, HRM4 distributes more load to level 2 nodes at the cost of their average lifetimes (relative to the other levels) as seen in Figure 4.

The average physical distance of single lookup hops shown in Figure 7 reflects what we expect of the approaches' routing hops' distances: while RBFM:Q90 has a large pool of nodes from which to choose links with strong and physically close nodes, HRM4's nodes have less choice for their links which are drawn from specific (less populated) hierarchical layers. HRM4 actually performs only slightly more location-aware than the completely location-naive Chord - reducing its usability for mobile network scenarios with ad hoc routing where location-awareness is integral: multi-tiered hierarchical approaches for MANETs require additional location-awareness such as PIS or PRS. On the other hand, level 2 and 3 nodes in HRM2:01-23 choose links from these two upper levels based only on distance, providing upper nodes with physically close links. Thus, depending on the tolerable lookup failure rate and desired distribution of load between the upper level nodes, HRM2:01-23 and RBFM are best suited for systems where the physical routing distance plays a central role.

6. FUTURE WORK

Several novel and established location and resource aware overlay protocols were compared in this paper using two very specific scenarios. While the multi-tiered hierarchical approach HRM4 provided the best combination of node lifetime and lookup deliverability, its location-awareness is not suitable for MANETs without the further integration of PIS or PRS (proximity-aware identifier/route selection). RBFM demonstrated the most stable behavior with respect to node lifetime, lookup deliverability, and location-awareness, often outperforming two-tiered approaches. The two-tier approaches performed well on many measures, but failed to provide the robustness required for a system with high churn rates. Thus, the improvement of these approaches' robustness could provide promising resource and location aware alternatives. The evaluation's observations build a foundation for future work with more complicated and realistic scenarios, for example with upper bounds on the permissible load per time unit for varying resource levels, or the development of protocol-specific replication geared toward each approach's specific strengths and weaknesses.

7. REFERENCES

[1] F. Araujo, L. Rodrigues, J. Kaiser, C. Liu, and C. Mitidieri. Chr: a distributed hash table for wireless ad hoc networks. In *ICDCS '05*.

[2] M. Artigas, P. Lopez, J. Ahullo, and A. Skarmeta. Cyclone: A novel design schema for hierarchical dhts. In *P2P'05*.

[3] M. S. Artigas, P. G. Lopez, and A. F. Skarmeta. A comparative study of hierarchical dht systems. In *LCN'07*.

[4] I. Baumgart, B. Heep, and S. Krause. Oversim: A scalable and flexible overlay framework for simulation and real network applications. In *P2P'09*, 2009.

[5] A. R. Bharambe, M. Agrawal, and S. Seshan. Mercury: Supporting scalable multi-attribute range queries. In *SIGCOMM '04*, 2004.

[6] F. Bustamante and Y. Qiao. Friendships that last: Peer lifespan and its role in p2p protocols. In *Web Content Caching and Distribution*. 2004.

[7] C. Cramer and T. Fuhrmann. Proximity neighbor selection for a dht in wireless multi-hop networks. In *P2P '05*.

[8] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: a decentralized network coordinate system. In *SIGCOMM'04*.

[9] F. Dabek, J. Li, E. Sit, J. Robertson, M. F. Kaashoek, and R. Morris. Designing a dht for low latency and high throughput. In *NSDI*, 2004.

[10] M. El Dick, E. Pacitti, and B. Kemme. Flower-cdn: a hybrid p2p overlay for efficient query processing in cdn. In *EDBT '09*, 2009.

[11] P. Ganesan, M. Bawa, and H. Garcia-Molina. Online balancing of range-partitioned data with applications to peer-to-peer systems. In *VLDB '04*, 2004.

[12] P. Ganesan, K. Gummadi, and H. Garcia-Molina. Canon in g major: Designing dhts with hierarchical structure. In *ICDCS'04*, 2004.

[13] L. Garces-Erice, E. W. Biersack, P. A. Felber, K. W. Ross, and G. Urvoy-Keller. Hierarchical peer-to-peer systems. In *ICPDCS*, 2003.

[14] J. Garcia-Luna-Aceves and D. Sampath. Scalable integrated routing using prefix labels and distributed hash tables for manets. In *MASS '09*, 2009.

[15] P. B. Godfrey and I. Stoica. Heterogeneity and load balance in distributed hash tables. In *INFOCOM*, 2005.

[16] K. Gummadi, R. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica. The impact of dht routing geometry on resilience and proximity. In *SIGCOMM '03*.

[17] D. R. Karger, E. Lehman, F. T. Leighton, R. Panigrahy, M. S. Levine, and D. Lewin. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web. In *STOC*, 1997.

[18] D. R. Karger and M. Ruhl. Simple efficient load balancing algorithms for peer-to-peer systems. In *SPAA'04*, 2004.

[19] R. Kummer, P. Kropf, and P. Felber. Distributed lookup in structured peer-to-peer ad-hoc networks. In *DOA*. 2006.

[20] B. Maniymaran, M. Bertier, and A.-M. Kermarrec. Build one, get one free: Leveraging the coexistence of multiple p2p overlay networks. In *ICDCS '07*, 2007.

[21] P. Maymounkov and D. Mazieres. Kademlia: A Peer-to-peer Information System Based on the XOR Metric. In *IPTPS*, 2002.

[22] G. Millar, E. Panaousis, and C. Politis. ROBUST: Reliable overlay based utilization of services and topology for emergency MANETs. In *Future Network and Mobile Summit '10*.

[23] H. Pucha, S. Das, and Y. Hu. Ekta: an efficient dht substrate for distributed applications in mobile ad hoc networks. In *WMCSA '04*, 2004.

[24] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content addressable network. In *SIGCOMM'01*, 2001.

[25] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker. Ght: a geographic hash table for data-centric storage. In *WSNA '02*, 2002.

[26] S. Ren, L. Guo, S. Jiang, and X. Zhang. Sat-match: a self-adaptive topology matching method to achieve low lookup latency in structured p2p overlay networks. In *IPDPS'04*, 2004.

[27] L. Ribe-Baumann. Combining resource and location awareness in dhts. In *LNCS*, 2011.

[28] S. Saroiu, P. K. Gummadi, and S. D. Gribble. A measurement study of peer-to-peer file sharing systems. In *MMCN*, 2002.

[29] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM'01*, 2001.

[30] Z. Tian, X. Wen, Y. Sun, W. Zheng, and Y. Cheng. Improved bamboo algorithm based on hierarchical network model. In *CCCM '09*, 2009.

[31] M. Waldvogel and R. Rinaldi. Efficient topology-aware overlay network. *SIGCOMM*, 2003.

[32] Z. Xu, R. Min, and Y. Hu. Hieras: A dht based hierarchical p2p routing algorithm. In *ICPP'03*, 2003.

[33] A. Yu and S. Vuong. A dht-based hierarchical overlay for peer-to-peer mmogs over manets. In *IWCMC '11*, 2011.

[34] T. Zahn and J. Schiller. Madpastry: a dht substrate for practicably sized manets. In *ASWN '05*, 2005.

[35] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph. Tapestry: An infrastructure for fault-tolerant wide-are location and routing. Technical report, UC Berkeley, 2001.

[36] S. Zoels, Z. Despotovic, and W. Kellerer. Cost-based analysis of hierarchical dht design. In *P2P '06*, 2006.

Towards an ambient data mediation system

Kim Tâm Huynh
PRiSM Laboratory
University of Versailles
Saint-Quentin-en-Yvelines
Versailles, France
kth@prism.uvsq.fr

Béatrice Finance
PRiSM Laboratory
University of Versailles
Saint-Quentin-en-Yvelines
Versailles, France
beatrice@prism.uvsq.fr

Mokrane Bouzeghoub
PRiSM Laboratory
University of Versailles
Saint-Quentin-en-Yvelines
Versailles, France
mok@prism.uvsq.fr

ABSTRACT

In this paper, we address the problem of integrating many heterogeneous and autonomous tiny data sources, available in an ambient environment (AmI). Our goal is to facilitate the development of context-aware and personalized embedded applications on mobile devices. The originality of the approach is the new ambient mediation architecture which provides declarative and dynamic services, based on rules/triggers. These services provide facilities to develop and deploy ambient applications over devices such as smartphones. This paper reports also on our first experimental prototype, combining Arduino+Android.

1. INTRODUCTION

Over the last 20 years there have been some significant progresses on the miniaturization of hardware components and wireless networks. The number and capabilities of mobile devices, wireless sensors and sensor networks open new research fields and applications. Terms such as the "Web of sensors", the "Internet of Things" and "Ambient Intelligence (AmI)" emphasize the trend towards a tighter connection between the cyberspace and the physical world.

Today, we are witnessing an unprecedent explosion of mobile data volumes (i.e. ambient data). According to a study from ABI Research [1], in 2014 the volume of mobile data sent and received every month by users around the world will exceed by a significant amount the total data traffic for all of 2008. In 2011, 1.08 billion of mobile phone users have a Smartphone and in the near future they will be surrounded by many sensors/actuators.

In his survey, Sadri [18] defines AmI as "the vision of a future in which the environments support the people inhabiting them. For example, instead of using mice, screens and keyboards, we may communicate directly with our clothes, household devices,..." The identified key features of AmI are: embedment, intelligence, context-awareness, personalization, adaptation and anticipation. It is also mentioned that "AmI can provide sophisticated support for everyday living, but the information capabilities it may use for this purpose can also potentially provide an invisible and comprehensive surveillance networks – walls literally can have ears". It inevitably opens up issues of privacy risk, acceptance and security.

In many ambient environments, data arrives as streams or as alerts/notifications and is only relevant for a period of time; its interpretation depends on the user's context and preferences. For instance, an information about a free parking place can be relevant for a user if this information is recent and if the parking place is nearby the user's location. Another example is the heat setting to the right temperature in the room where a given person is in and accordingly to his/her own comfort preferences.

In the database community, a lot of work has been devoted to efficiently monitor huge amount of data streams coming from sensors that continuously push their data to a fixed centralized system, without being concerned in privacy, mobility, context-awareness and reactivity. But as soon as a sensor is linked to my personal life (e.g, my home location, my traveling itinerary, etc), the applications using the captured data may become intrusive in my private life. Moreover, in the opposite, as soon as I leave the smart environment, I may lose the ambient capabilities that may support my everyday life (e.g. tension and heart beat measurement, mandatory presence in a certain place). Consequently, the ambient environment and applications are considered as undesirable constraints in some cases and helpful tools in others. Since my ambient environment is changing over the time and over the space (e.g. at home, at work, at the hospital), the query processing should adapt itself to these two dimensions. As Feng said [10] "AmI imposes strong user-centric context-awareness requirement on data management", but also strong system requirements in terms of hardware constraints (i.e. energy consumption, wireless communication).

As seen before, smartphones and the underlying applications are, under some restrictions, good support for everyday life. However, their repetitive development from scratch is time and money consuming, it makes the software evolution quite difficult, in particular because components updates are frequent. We claim that an embedded data management system for AmI may significantly contribute to ease the development and maintenance of such applications.

The contribution of this paper is to propose an ambient mediation system (called CAIMAN for *C*ontext-aware d*A*ta *I*ntegration and *M*anagement in *A*mbient e*N*vironments) which :

- facilitates personalized and contextual integration and monitoring of heterogeneous data streams through continuous query execution;
- enables applications to dynamically sense and control, according to some preferences, the ambient environment of the user, which is changing over time and space;
- enables the user to keep some control over his personal data as the monitoring is done exclusively on his personal device.

In the remaining of this paper, Section 2 defines the requirements and the constraints imposed on the design of an ambient mediation system, and presents the architecture of CÄIMAN. In Section 3, the ambient mediation approach is described and illustrated through a scenario. In Section 4, we detail the main components of our system and report on our experimental prototype combining Arduino+Android. Section 5 concludes with some open issues.

2. MOTIVATIONS

To develop ambient applications, there is a need of an ambient data mediation system (ADMS) which allows interoperability between a set of dynamic and loosely-coupled ambient data sources. An ambient data source is a (fixed or mobile) communicating object which generates or consumes continuous (or discontinuous) flows of data. Among such objects, we can distinguish a wide spectrum of sensors and mobile phones as well as any other data services which can push specific data to the applications. In addition to these data sources, there exist other ambient objects called actuators, that do not exchange data, but simply perform some actions on other objects. Notice that a single physical object can play both the role of a data source and actuator. All ambient physical objects are abstracted by software services which encapsulate them and make visible their capabilities, especially their data exchange protocol.

The design choices of our system have been motivated by the requirements of AmI applications in general, and mobile/ubiquitous users/equipments in particular; the key issue being the continuity of ambient services whatever the dynamic changes are. In this section we review them and compare our design choices with existing related work. The proposed CAIMAN architecture is built on the basis of these choices.

2.1 The requirements

An ambient information system (AmIS) is a set of data flows provided by a collection of ambient objects to achieve the needs of AmI applications (e.g. intelligent home, intelligent city, health care, mobile social network, etc). Some AmIS objects can play the role of a mediator which is able to integrate and interpret data of many ambient data sources, as well as to perform actions over their environment. Most of the AmIS data may persist only a few seconds or minutes in the system, unless the application or the user decides otherwise for various reasons. The main specific requirements imposed to the design of an ADMS are the following :

- Data sources are heterogenous. They may be fixed or mobile and arbitrarily connect and disconnect from the mediator, during variable intervals of time. Data sources have different capacities in terms of storage and computation.
- The mediator can dynamically connect to the sources when and as long as they are active (i.e. visible over the wireless network and ready to provide data).
- The mediator should provide, for each application, the capability to define its data requirements in terms of event types, so offering a concept of a virtual schema similarly to conventional mediators, and a mechanism which handles continuous queries.
- The mediator should be able to aggregate data flows originated from the same source and integrate data flows originated from different sources on the basis of specific rules provided by the applications. As in conventional mediators, data heterogeneity should be transparent to the user, adaptors are aware of data transformation.
- The mediator should adapt itself to the user's context by continuously searching for the appropriate data sources. It should also satisfy user's preferences in terms of data delivery, relevance to domain of interest, privacy, etc.
- The mediator should be aware of energy consumption and manage consequently the connections to the sources and the usage of its resources.

These requirements clearly distinguish an ambient mediator from a conventional one [21] where the mediation schema and the sources are known in advance. Here the environment is dynamic as data sources enter and leave continuously the field of detection. The personalized and contextual integration and monitoring of heterogeneous data streams rely on continuous query evaluation.

2.2 Related work

In this section, we list the CAÏMAN main objectives and compare them with existing related works.

The first goal of CAÏMAN is to provide a high-level declarative approach which permits user applications to interoperate over distributed ambient objects. The expected data streams are relatively small in their length/size. Many formalisms for event streams processing and querying have been proposed, see [9] for a good survey. In Data Stream Management Systems (DSMS) [4], many CQL-like query languages which extend SQL have been defined. They are based on the concept of window used to manage and filter data streams in a declarative way [5, 14, 8, 13]. In Complex Event Processing (CEP), some formalisms based on composition operators (i.e. sequence, conjunction, disjunction,etc), or time-based automata are used. The goal of CEP is to detect event patterns (i.e. situation) with temporal constraints in data streams. Today, the two approaches are seen as complementary [9, 16]. Both approaches focus on events detections but none on the events reactivity which is an important feature of AmI applications, e.g. the ability to identify the context during which active behavior is relevant and the situations in which it is required. Both approaches assume that the data (i.e. events) are continuously pushed to a centralized system known in advance. These assumptions do not fit with our constraints, as the push mode consume a lot of energy.

In Sensor Databases such as TinyDB [15], data is acquired in a pull mode to avoid battery consumption. The query (i.e. Tiny SQL) is sent through the network and evaluated in a distributed mode. Sensors are active only when they are queried. The advantages of this approach is its adaptability to the features of hardware devices and to their constraints. The sensor network can contain a large number of sensors. However, the sensors are homogeneous, they all have a TinyOS and there is no mechanism of source discovery because the sensors are all known in advance.

The second goal of CAÏMAN is to make the ADMS aware of the user's context and user's preferences. Again, a lot of work has been devoted to context and preference-aware queries by the database community. Traditionally, the integration of context and preferences in queries is made in two ways [10]. The query pre-processing consists in enriching the query with context or preference informations before executing it. The query post-processing ranks the query's answers according to these informations. Unfortunately, this mechanism works only for one-time queries and not for continuous queries in which the notions of pre and post-processing do not exist. Indeed, in traditional DBMS, data are permanent and queries are transient. In DSMS, data are transient and queries permanent as they are continuously evaluated over the transient data. To our knowledge, no solution has been proposed to handle the context and the user preferences on continuous queries.

In existing context-aware frameworks [6], the context manager is generally represented by a centralized server which is in charge of collecting context information, interpreting and providing them to the client applications. However in pervasive environments, there are frequent disconnections and low connectivity, making this architecture not robust enough and adequate for this type of application. In the literature, only few systems [7, 11] have proposed a local context server to overcome this problem. However, in [11], the context sources are known in advance and correspond to built-in sensors. Conversely in [7], the authors have developed a sentient object model for ad-hoc mobile environments where the context is only used to adapt the application behavior. It doesn't allow to enhance application data with contextual information. For instance, many applications need to add the location to the data produced.

2.3 The CAÏMAN architecture

The overview of the CAÏMAN architecture is depicted in the Figure 1. The resource discovery component facilitates objects discovery and handles dynamic connections and disconnections to these objects. AmIS objects should be able to rely on their own battery, so short-range wireless communication such as Bluetooth are assumed in CAÏMAN as these personal area networks are known to have a low consumption of energy. Once, a data source is discovered the data collectors are responsible for acquiring the data. Data sources do not push their data continuously, but rather sporadically in response to the mediator request. This requests is done only if the data source can serve the needs of the applications which have been deployed on top of the mediator. The originality of our ADMS is to offer an hybrid approach combining both the push and the pull modes.

In our environment, we assume that sensors/actuators remain passive most of the time with a default behavior unless someone requests a service (i.e. light off, that can be turned on). All sensors/actuators implement some generic functions (e.g. services) and some that are optional. Sensors can only send data during a period of time fixed by the mediator, enabling the sensors and actuators to fall automatically asleep when they have finished their duty and thus turn back to their default state.

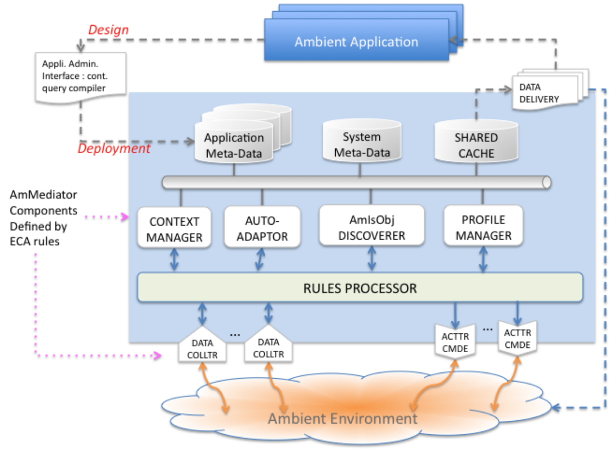


Figure 1: CAÏMAN

As the mediator should fit into lite clients such as smarphones and function in a complete autonomy, no system functionality is delegated to a central server. We don't rely on a global data source registry that might not be available at all time. Meta-data exchange between AmIS objects should be done instead at runtime and the context and profile managers should be local.

CAÏMAN provides a declarative language which allows to describe most of the system and application semantics which is based on the ECA (Event-Condition-Action) paradigm used in active databases [17]. Thus, the rule processor is a core component of our system. It will be detailed in the next section. Notice that in this paper our goal is not to propose yet another query language nor a complex context-aware model, but rather to select a subset of existing formalisms, keep them simple and tractable as much as possible to fit into lite clients.

3. THE AMBIENT MEDIATION APPROACH

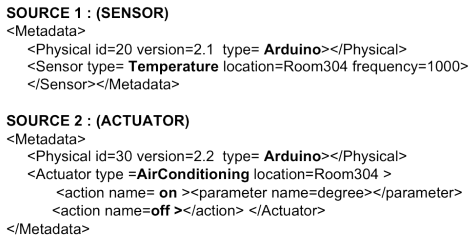
In this section, we detail our mediation approach. First, we describe the different types of ambient sources on which the CAÏMAN is built on, then the virtual mediation schema is presented. To understand the approach, we illustrate it with a scenario. In this scenario, Paul is a student and lives at the university residence. He wants to benefit of an intelligent home behavior, by automatically controlling the air conditioning of the room where he is located according to his preferences. He also needs to organize his evenings and wants to be notified about interesting cultural events located not far from his current location. In order to do so, Paul will have to deploy two AmI applications which have been specified in a declarative manner by some designers. During the deployment, the declarative description will be used to instantiate the virtual schema of the mediator, i.e. the application meta-data as described in the Figure 1.

3.1 The Ambient Sources

Three types of data sources [12] are considered: (1) physical sources (e.g. GPS built-in sensors, smartphone, external temperature sensor such Arduino), (2) virtual sources (e.g. user, agenda alerts, SMS, emails, contacts) and (3) logical sources which combine physical and virtual sources with information from databases. These sources can be either fixed (e.g. already embedded in a mobile device where the mediator is), or dynamic (i.e., another smartphone, a sensor/actuator). Fixed ambient sources are known in advance and always connected to the mediator, e.g.built-in sensors.

Dynamic ambient sources correspond to sources that appear and disappear to the mediation system over time due to the source mobility itself or due to the mobility of the user which embeds a mediator on his personal smartphone. If a smartphone is close to a mobile device, a communication can be established and some messages can be exchanged as long as the device is reachable. If suddenly it disappears due to the user mobility, some messages can be lost. Moreover, the user himself can be an ambient data producer as he/she behaves like any other sensor (intelligent sensor). For instance, the user is discovering a broken window and wants the mediation system to inform automatically the technical staff in charge of repair.

In the remaining of the paper, we focus more on the dynamic data sources. Each one exports its capabilities (e.g., metadata) in an XML document as depicted in the Figure 2. Each dynamic source corresponds to a physical device characterized by an 'id', a 'type' (e.g., Arduino, Android) and a version number.



```
SOURCE 1 : (SENSOR)
<Metadata>
   <Physical id=20 version=2.1  type= Arduino></Physical>
   <Sensor type= Temperature location=Room304 frequency=1000>
   </Sensor></Metadata>

SOURCE 2 : (ACTUATOR)
<Metadata>
   <Physical id=30 version=2.2  type= Arduino></Physical>
   <Actuator type =AirConditioning location=Room304 >
       <action name= on ><parameter name=degree></parameter>
      <action name=off ></action> </Actuator>
</Metadata>
```

Figure 2: Sources Description

3.2 A declarative approach

The declarative approach used in CAÏMAN is ECA. ECA rules are a generalization of several methods to achieve active behavior, such as triggers and production rules. ECA rules are evaluated in three steps: (1) when an occurrence of an *event* is detected, (2) the system evaluates the *condition* under which the event is considered relevant, and (3) if it is verified, the rule *action* is executed. The separation between E-C-A is important for many reasons and has been emphasized by the active database community [17].

When designing a mediator based on the ECA paradigm it is important to carefully take into account the life cycle of a rule and the dimensions related to its semantic execution. Indeed this knowledge is mandatory for those designing an application. Moreover by separating the dimensions, there is more flexibility, different behaviors can be proposed for specific applications. In the Figure 3, we summarize them. Here we shortly explain some of the dimensions. The *event detection and composition* and the *visible DB states* will be explained in the next section.

There exist many modes of *event consumption,* among them we selected two modes more suitable to our environment:

1. **recent**: only the most recent instances of any event are considered; older events are discarded. It is most suitable for fast-changing environments in which new events supersede old ones.

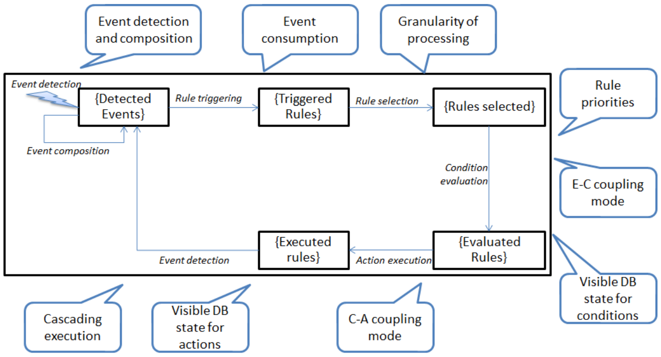


Figure 3: Active Rules & their Execution Semantics

2. **chronicle**: the oldest instances are considered and then discarded; i.e. events are consumed in a chronological order. It is preferred when there is a causal dependency between events.

The *granularity of processing* defines whether the rule processor reacts after the detection of each event (instance-oriented processing) or after a detection of a set of events (set-oriented processing). An example of instance-oriented processing is to call emergency after detecting a critical situation like the unconsciousness of a person. In order to avoid a false alarm, we can wait for multiple events before calling the emergency. In CAÏMAN, this latter dimension can be offered by defining windows on the events arrival. The *rule priorities* determines how rules are selected among a conflict set of rules (i.e. rules triggered by the same event). The EC and CA coupling modes indicates when condition (resp. action) is evaluated after event detection (resp. condition evaluation). The different options are: immediate, deferred, or detached. CAÏMAN proposes immediate coupling modes. In our system, we do not consider *cascading execution* because we assume that our actions have no side effect.

In our experimental prototype, only one semantics is implemented for the rule processor. Rules are evaluated in parallel and no cascading executions is allowed, but the event consumption and the granularity of processing can be parameterized.

3.3 The Ambient Mediation Schema

The CAÏMAN mediation schema is composed of: (1) a set of events types, corresponding to the data flows required by ambient applications, and events detectors, (2) a context model and a default user profile, and (3) a set of personalized and contextual continuous queries (defined as ECA rules).

3.3.1 Events & Event Detectors

An event type can be either simple *(SE)* or complex *(CE)*. A complex event type is a combination of other simple or complex events types. These event types are defined by the application designer.

Each ***event type*** *(SE & CE)* is defined by a set of attributes:

- *name*: name of the event type,
- *lifespan:* default time interval during which the event instance is valid,
- *aggrFunction*: function which aggregates events to produce a complex event. For simple events, there is no aggregation function.

Each event type can be instantiated at runtime according to data flows arriving to the mediation system. These ***event instances (SE & CE)*** are defined by a set of attributes:

- *value*: event instance value,
- *source*: source name that captured the event instance,
- *raisingDate*: moment when the event instance is produced/observed by its source,
- *systemTime*: moment when the event instance is detected by the mediation system,
- *lifespan:* time interval during which the event instance is valid after its raisingDate,
- *raisingLocation:* geo-location where an event instance is produced/observed by its source.

The *lifespan* is a metadata which can be provided by the event source or assigned by the application. Event instances are relevant during a limited period of time. Pervasive environments can cause delays between the raising date of an event instance and the time for treating this instance. Consequently the validity V of an event e_i is defined by:

$$V(e_i) = \begin{cases} 1 & if\, raisingDate(e_i) + lifespan(e_i) < currentTime) \\ 0 & otherwise \end{cases}$$

The *raisingLocation* is a useful notion for many location-aware applications. Indeed, the location can influence the relevance of a given event instance. For example, an event “flood” detected far away from a user can be irrelevant for him.

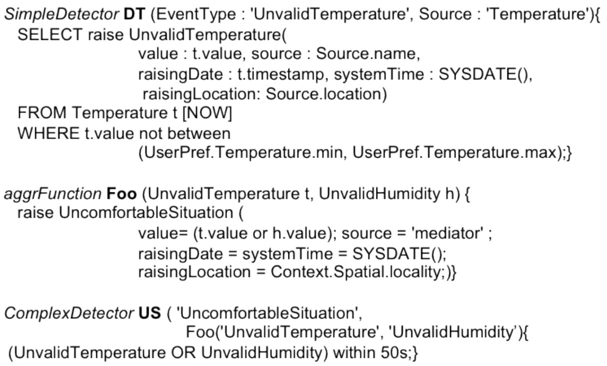
Once event types are defined, one should specify how and when event instances are created or captured. This is done by specifying event detectors. Depending on the event type and on the target data source, an event detector may be defined in various ways: a listener, a lookup function or any other procedure able to transform a specific signal into a semantic event.

For our scenario, the designers have separately defined three simple event types : *UnvalidTemperature*, *UnvalidHumidity* and *Advertisement*, with respectively a default lifespan of 5 min, 5 min and 1 week, and one complex event type *UncomfortableSituation* with its associated aggregate function *Foo* as depicted in the Figure 4. For each event, the designer must define a detector. Here, we only give the simple event detector *DT* on temperature expressed as a CQL query and the complex event detector *US* expressed as a CEP-like manner. Others are omitted as they can be expressed in a similar way.

3.3.2 Context Model

According to [20], there exist six different models for representing the context information. Some models like the ontology-based model is very expressive and allows powerful context processing. However in CAÏMAN the model used is the simple key-value model as it should be embedded.

Following our previous work [3], we define a **context** by five dimensions : spatial , temporal, environmental, equipment and user state.



```
SimpleDetector DT (EventType : 'UnvalidTemperature', Source : 'Temperature'){
  SELECT raise UnvalidTemperature(
                    value : t.value, source : Source.name,
                    raisingDate : t.timestamp, systemTime : SYSDATE(),
                    raisingLocation: Source.location)
  FROM Temperature t [NOW]
  WHERE t.value not between
                    (UserPref.Temperature.min, UserPref.Temperature.max);}

aggrFunction Foo (UnvalidTemperature t, UnvalidHumidity h) {
  raise UncomfortableSituation (
                    value= (t.value or h.value); source = 'mediator' ;
                    raisingDate = systemTime = SYSDATE();
                    raisingLocation = Context.Spatial.locality;)}

ComplexDetector US ( 'UncomfortableSituation',
                          Foo('UnvalidTemperature', 'UnvalidHumidity'){
 (UnvalidTemperature OR UnvalidHumidity) within 50s;}
```

Figure 4: Simple & complex event detectors

1. The *spatial dimension* is an important characteristics of mobile and pervasive environments. Indeed, depending on how much the user is mobile, the system will react differently. For instance, if a user is highly mobile, he will not have time to establish proper connections with all equipments around him. The other important aspect is the location. It can be expressed in many ways depending on the application: GPS coordinates, an address, or a locality label (e.g. Room 305B, Administration Building). The spatial information can be provided by GPS built-in sensors or mobile networks or derived from Google Maps or databases.
2. The *temporal dimension* is an important information that can be used for personalizing an application. For instance, a user can be interested to receive events only in the morning. Designers can change the notion of moment in the core context, e.g. when the morning begins and ends. This information can be provided by the phone clock (i.e. date, time) or derived from context definition (i.e. moment).
3. The *environmental dimension* concerns all the sensors describing the environment of the user (e.g. temperature, humidity, luminosity). This information is important when the environment is fixed, for example a smart home.
4. The *equipment dimension* characterizes all information about the media used by the user to interact with the application: the used device (e.g. type, battery autonomy, memory storage, computing power) and the connectivity (e.g. type of connection, rate). This dimension is important to adapt dynamically the system accordingly with the equipment constraints (e.g. low battery, uncertain connectivity).
5. The *user state dimension* allows to know if a user is available or not, and how he feels. In our case, we are more interested in the user availability which can have an impact on the system behavior.

Designers have to provide the context model used by their application and the set of context rules to the mediator that can compute the current context. For that, a list of context models is proposed with default context rules. For example, if the application designer is interested in the location information, there exists a rule associated with this dimension which captures the GPS coordinates from the smartphone GPS sensor every minute. However designers can also write their own context rules and submit them to the mediator.

3.3.3 Profile Model

The survey [19] highlights the difficulty of choosing the good representation for the user preferences: qualitative or quantitative. The quantitative approach allows a total order between preferences but is not intuitive because this implies that the user put weights on his preferences. While the qualitative approach is very intuitive but makes difficult the usage because there is not necessarily a total order between preferences. In CAIMAN the user preferences considered are viewed as dynamic criteria by the application. Thus there is no order between preferences as all preferences must be considered by the application.

Following the definition given in [3], a **user profile** is organized into several dimensions, possibly decomposed into sub-dimensions. Each dimension and its sub-dimensions contain a set of attributes and their values on which preferences are expressed. We retain four important dimensions:

1. *Personal data* contains all information about the user (e.g. his name, his address, his birthday). This dimension may also contain data on social groups to which the user belongs to (e.g. student, professor).

2. *Domain of interest* is generally the central dimension of the user profile, it represents the user domain of interest and preferences. For example, the domain of interest may be the types of events the user is interested in and the preferences on how/when event instances are received or treated.

3. *Resource discovery* contains the user preferences on the remote resources (i.e. type of resource, associated data collectors, related security issues and all meta data useful to understand data stream semantics). An example can be receiving events only from sources located in the same place as the user.

4. *System adaptation* groups user preferences on how the system should adapt to the user context or to its own behavioral parameters. An example can be to disable the resource discovery component during the night. A set of preference rules can also be defined to adapt the system behavior in case of low battery, that is either disabling some functionalities or changing the frequency of the captured data.

In the same way as the context, designers specify a default user profile for their application. The Figure 5 describes the application default profile set by the designers for our scenario. We have gathered the profile of each application into a global one, but in reality each application has its own. $name and $$name variables represent respectively a string or a set of strings.

```
Profile
Resource Discovery :
    sensor.location = Context.Spatial.locality;
    actuator.location = Context.Spatial.locality;
DomainsOfInterest :
    on Temperature  t:
              if( Context.Temporal.Time between(8.00,20.00) )
                        then { min = $1, max = $2} else { min = $3, max = $4 };
    on Advertisement a:
              condition : (a.value.topic in $$5)
                AND distance(Context.Spatial.Locality, a.raisingLocation)<$6};
End Profile
```

Figure 5: User Profile

For each application, the designer has specified the resource discovery constraints. Indeed, to control the temperature in a room, we are only interested by sensors and actuators located in the same room where the user is currently located. Some variables have a default value set by the designer. If not, they will be filled by the user when installing his application on his mobile device. For instance, the default temperature variables are ($1=18°, $2=22°) for the day time and ($13=16°,$4=18°) for the night. In the same manner, advertisements relevant to a user are those within $6='15km'. Others variables are set by the user. For instance, Paul decides to change the default value $6 to 25km, and set the $$5 to ('music','sport').

3.3.4 Application ECA Rules

The last task of the designer consists in defining his set of ECA application rules. For example, let's consider the following rules defined in the Figure 6 that model our scenarii. As said before rules are contextual and personalized. Thus, the context and the user profile can be either explicitly used by designers in their rules and in particular in the condition, or implicitly used by the system during the runtime when specified within the user profile.

The first rule of the smart office scenario explicitly uses the user profile and in particular the UserPref.Temperature.min and UserPref.Temperature.max variables described above. As these variables are context-dependent (i.e. day or night), their values are changing over time. So, they will be instantiated just before evaluating the rule condition.

In the second rule, a user receives advertisements. Here no condition is specified; however one will be added by the system at runtime since an implicit preference has been defined on this event type in the default profile. The reason why the condition is not directly written in the rule, is to allow the user to change his condition at any time. Here, users receive advertisements relevant to their personal topics and within the required distance. When an event is relevant for the user, he is informed by email.

```
Define ECA_Rule Application_SmartOffice_R1
   EVENT : UncomfortableSituation u [range 5 min]
   CONDITION : AVG(u.value) not between
                UserPref.Temperature.min and  UserPref.Temperature.max
   ACTION : activate ( airConditioning', 'on', UserPref.Temperature.min+2)

Define ECA_Rule  Application_Culture_R1
   EVENT : Advertisement a [Now]
   ACTION : inform( User, 'mail', a)
```

Figure 6: Rule Examples

Up to now, we have given some intuitions of our rule language, now the semantics of rules is briefly described.

The granularity of processing (e.g. instance or set-oriented) is defined as a window. The default window [Now] is a time-based window of size 1 and corresponds to the instance-oriented. Others windows are defined as in any continuous query language and correspond to the set-oriented. For instance, *"range 5 min"* represents all events that appear within the last five minutes. Notice that we do not allow unbounded windows. The visible state of conditions and actions is defined by the window and the event type. The event consumption mode is defined as a fixed parameter of the mediator and is taken into account in event detectors that specify how to raise events.

Some generic actions are possible and are defined by a template. The first action *"inform($who,$how,$what)"* informs a user or a group of users, or an application through a delivery mode (i.e. local, wireless, email, SMS), of a particular message or event. We separate messages from events, messages are generated to users and definitively leave the system, by opposition to events that will be filtered and reinjected later in a mediator that can interpret them. The action *"activate($type, $service, $serviceparameters)"* activates a service with some parameters on a specific actuator type that satisfies some user preferences.

4. THE CAÏMAN MEDIATION SYSTEM

In this section, we briefly describe the behavior of the main components of our system at runtime. Thus, we assume that AmI applications have been deployed on the smartphone, and that their default profile, detectors and rules have been transmitted to the mediator and uploaded in the application metadata database.

4.1 Data Collectors & Actuator Commands

As the data sources are heterogeneous, a set of generic data collectors and actuator commands are needed to allow communications with the mediation system. The *Data Collectors* are used to collect and transform any kind of heterogeneous data so it can be understood and integrated in the mediator. Each source is bound with one instance of a data collector that is responsible for querying and controlling this data source. All communications are asynchronous. Another issue also considered is the data transformation as the data provided by a source is not necessarily compatible to the coding, format, unit and scale of the expected data at the mediator level. The data collector is in charge of transforming these raw data into events, thus it activates the simple event detector associated with the source.

Actuator Commands allows the mediator to transform commands into real actions on the environment through actuators services.

4.2 Resource discovery & Bindings

Due to the numerous communicating objects that enter and leave the field of detection because of the user mobility, there is a need of a *Resource Discovery* component which is continuously aware of the equipment's environment.

Ambient data sources may connect and disconnect arbitrarily. As the mediator cannot rely on a centralized resources registry, the resource discovery service is defined as a seeking function which detects the surrounding objects, identifies them through some metadata exchange and establishes connections to them if they are relevant at least for one active application deployed on the mediator.

As connections/disconnections can be frequent, the resource discovery component stores a history of all the discovered sources with their version number. This version number is useful to know if the source has changed since the last time it was discovered. This is to avoid unnecessary metadata exchanges. The history plays the role of a local resource registry which can be deleted at anytime, since it can be rebuilt at runtime.

Before communicating with a data source, one should know if the information it contains is relevant. For doing so, a matching is made between the source metadata and a part of the mediation schema. At the same time the context and profile information is used to select only the sources in a given location and at a given time, according to the preferences of the application or the user. Once the source is selected and a communication established, a dynamic binding enables to link the data source to the mediator, by instantiating the suitable data collector.

4.3 Profile & Context Manager

As many context-aware systems [6], the mediation system proposed makes a separation between the context acquisition, the context processing, and the context information usage. In fact, data collectors are in charge of retrieve context raw data, the context manager processes these data to infer context information which will be stored and used by applications and in particular application rules. These context information are also used by the profile manager to derive the active profile (i.e., all profile information which is valid for this current context). Indeed, the user profile is composed of a static part and a dynamic part. In the first one, the profile information doesn't change frequently while the second one depends on a context that can change rapidly. As we have seen in the Figure 5, the user profile can be contextual and it will be the role of the context and profile managers to keep the active profile up to date for the application. For simplicity reasons, an assumption is made on defining a contextual profile. Only If-then-else statements are allowed in order to avoid conflicts between contextual predicates. Only one active profile is valid at any instant of time. Notice also that all variables can be changed at any time by the user, via a simple interface on his smartphone.

4.4 Rule Processor

The *Rule Processor* is an idempotent service to which ECA rules with their associated detectors are submitted. As said earlier, it is important to follow rule execution semantic as described in the Figure 3. The query processor relies on a multi-threaded execution framework. The approach we follow is, in a way, very similar to what has been proposed by Krämer [14] and especially their SweepArea that models a dynamic data structure to manage a collection of events of the same type. ECA rules act as continuous queries over collection of events and react over their environment when a situation is encountered.

As events of the same type can be used in many rules, events are not removed when used for triggering a rule. Thus, a garbage collector is necessary and events are removed from the dynamic structure when their lifespan has expired. In order to avoid triggering multiple times a rule with the same events, each rule has a context summarizing the past execution. When the rule processor is looking for the rules that can be triggered, it uses the context which is also computed in a continuous way. Only the most recent event is kept for each dimension.

4.5 The prototype

Smartphones as well as computers cannot really sense the world. For AmI environments, there is a need for tools for making computers that can sense and control more of the physical world than any other desktop computer. This is the role of the Arduino [2] platform to sense the environment by receiving input from a variety of sensors and to affect its surroundings by controlling lights, motors, and other actuators. A first prototype combining Arduino and Android validating the approach has been implemented. Many sensors and actuators have been prototyped. The CAÏMAN mediator and many simple AmI applications can be deployed on an ANDROID platform. For the time being, data collectors and actuators are operational, as well as the resource discovery. Simple and complex event detectors, as well as simple ECA rules are supported. When the validity of events expires, the garbage collector automatically deletes the events. We are currently integrating the context and the user profile within the rules. The user agrees or not to install an AmI application on top of CAÏMAN. He can turn it on/off whenever he wants to. He can also checked the types and the number of events, the mediator is currently monitoring.

5. CONCLUSION

In this paper, we have presented the different requirements posed by ambient environments and proposed an ambient mediation system called CAÏMAN. As we have seen, a declarative approach based on ECA rules is proposed. The main originality is that it combines in an elegant way the context, the user profile and the continuous queries together. Some dimensions of the profiles can be integrated and changed at anytime by the user. Another important contribution of our work is that it is based on personal mobile devices and local computation that better fulfill the user privacy. To our knowledge CAIMAN is the first ambient mediation system embedded in a smartphone offering such functionalities.

Some open issues still remain to be considered such as how to adapt the system if the context is critical (i.e., low battery), what to do when many mediators are acting in a conflicting way on specific resources, how an event that has been sent several times by different data sources can be discovered to avoid repeating the same actions again and again.

Acknowledgment

This work is partially funded by the French National Agency for Research (ANR) project under grant KISS (2011-INSe-005-03).

6. REFERENCES

[1] ABI research. `http://www.abiresearch.com/press/1466-In+2014+Monthly+Mobile+Data+Traffic+Will+Exceed+2008+Total`.

[2] Arduino site. `www.arduino.cc`.

[3] S. Abbar, M. Bouzeghoub, D. Kostadinov, S. Lopes, A. Aghasaryan, and S. Betge-Brezetz. A personalized access model: concepts and services for content delivery platforms. In *Proc. of the 10th Int. Conf. on Information Integration and Web-based Applications and Services.*, pages 41–47, New York, USA, 2008.

[4] A. Arasu, B. Babcock, S. Babu, M. Datar, K. Ito, I. Nishizawa, J. Rosenstein, and J. Widom. STREAM: the Stanford stream data manager (demo). In *Proc. of the ACM SIGMOD Int. Conf. on Management of data*, pages 665–665, New York, USA, 2003.

[5] A. Arasu, S. Babu, and J. Widom. The CQL continuous query language: semantic foundations and query execution. *The VLDB Journal*, 15(2):121–142, 2006.

[6] M. Baldauf, S. Dustdar, and F. Rosenberg. A survey on context-aware systems. *Int. J. Ad Hoc Ubiquitous Comput.*, 2(4):263–277, 2007.

[7] G. Biegel and V. Cahill. A framework for developing mobile, context-aware applications. In *Proc. of the 2nd IEEE Int. Conf on Perv. Comp. and Comm. (PerCom'04)*, pages 361–, Washington, USA, 2004.

[8] I. Botan, R. Derakhshan, N. Dindar, L. Haas, R. J. Miller, and N. Tatbul. SECRET: a model for analysis of the execution semantics of stream processing systems. *Proc. VLDB Endow.*, 3(1-2):232–243, Sept. 2010.

[9] M. Eckert, F. Bry, S. Brodt, O. Poppe, and S. Hausmann. A CEP Babelfish: Languages for Complex Event Processing and Querying Surveyed. In S. Helmer, A. Poulovassilis, and F. Xhafa, editors, *Reasoning in Event-Based Distributed Systems*, volume 347 of *Studies in Computational Intelligence*, pages 47–70. Springer Berlin / Heidelberg, 2011.

[10] L. Feng, P. P. M. Apers, and P. W. Jonker. Towards context-aware data management for ambient intelligence. In *15th Int. Conf. on Database and Expert Systems Applications*, pages 422–431, 2004.

[11] T. Hofer, W. Schwinger, M. Pichler, G. Leonhartsberger, J. Altmann, and W. Retschitzegger. Context-awareness on mobile devices - the hydrogen approach. In *Proc. of the 36th Annual Hawaii Int.Conf. on Syst. Sci. (HICSS'03)*, pages 292.1–, Washington, DC, USA, 2003.

[12] J. Indulska and P. Sutton. Location management in pervasive systems. In *Proc. of the Australasian inf. sec. workshop conf. on ACSW frontiers*, pages 143–151, Darlinghurst, Australia, Australia, 2003.

[13] N. Jain, S. Mishra, A. Srinivasan, J. Gehrke, J. Widom, H. Balakrishnan, U. Çetintemel, M. Cherniack, R. Tibbetts, and S. Zdonik. Towards a streaming SQL standard. *Proc. VLDB Endow.*, 1(2):1379–1390, Aug. 2008.

[14] J. Krämer and B. Seeger. Semantics and implementation of continuous sliding window queries over data streams. *ACM Trans. Database Syst.*, 34(1):4:1–4:49, 2009.

[15] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TinyDB: an acquisitional query processing system for sensor networks. *ACM Trans. Database Syst.*, 30(1):122–173, 2005.

[16] A. Margara and G. Cugola. Processing flows of information: from data stream to complex event processing. In *Proc. of the 5th ACM int. conf. on Distributed event-based system*, DEBS '11, pages 359–360, New York, NY, USA, 2011. ACM.

[17] N. W. Paton and O. Diaz. Active database systems. *ACM Comput. Surv.*, 31(1):63–103, 1999.

[18] F. Sadri. Ambient intelligence: A survey. *ACM Comput. Surv.*, 43(4):36:1–36:66, Oct. 2011.

[19] K. Stefanidis, G. Koutrika, and E. Pitoura. A survey on representation, composition and application of preferences in database systems. *ACM Trans. Database Syst.*, 36(3):19, 2011.

[20] T. Strang and C. L. Popien. A context modeling survey. In *UbiComp 1st Int. Workshop on Advanced Context Modelling, Reasoning and Management*, pages 31–41, September 2004.

[21] G. Wiederhold. Mediation in information systems. *ACM Comput. Surv.*, 27(2):265–267, 1995.

Context-Aware Routing Method for P2P File Sharing Systems over MANET

Taoufik Yeferny
Dept. of Computer Science
Faculty of Sciences of Tunis
MOSIC Research Group
Tunis, Tunisia
Taoufik.Yeferny@it-sudparis.eu

Khedija Arour
Dept. of Computer Science
National Institute of Applied Sciences and Technology of Tunis
URPAH Research Group
Tunis, Tunisia
Khedija.arour@issatm.rnu.tn

Amel Bouzeghoub
Dept. of Computer Science
Telecom SudParis
SAMOVAR CNRS LAB
Paris, France
Amel.Bouzeghoub@it-sudparis.eu

ABSTRACT

Mobile devices have achieved great progress. They allow user to store more audio, video, text and image data. These devices are also equipped with low radio range technology, like Bluetooth and Wi-Fi, etc. By means of the low radio range technology, they can communicate with each other without using communication infrastructure (e.g. Internet network) and form a mobile ad hoc network (MANET). The peers in the MANET are typically powered by batteries which have limited energy reservoir also they are free to move from their locations at anytime. Recently, P2P file sharing systems are deployed over MANET. A challenging problem in these systems is (i) the selection of best peers that share pertinent resources for user's queries and (ii) guarantee that the pertinent peers can be reached in such dynamic and energy-limited environment. To tackle this problem, we propose a context-aware integrated routing method for P2P file sharing systems over MANET. Our method selects the best peers based on the query content and the user's profile. Furthermore, it considers the energy efficiency, peer mobility and peer load factors into the query forwarding process to guarantee that the pertinent peers can be reached.

1. INTRODUCTION

In the last few years, peer-to-peer file sharing systems have emerged as platforms for users to search and share information over the Internet network. There are different kinds of P2P systems architectures that can be roughly classified into structured, unstructured and hybrid architectures [7]. Nowadays, mobile and wireless technology has achieved great progress. Cell phones, PDAs and other handheld devices have larger memory, higher processing capability and richer functionalities. They allow user to store more audio, video, text and image data with handheld devices. These devices are also equipped with low radio range technology, like Bluetooth [1] and Wi-Fi [2], etc. By means of the low radio range technology, they can communicate with each other without using communication infrastructure (e.g. Internet network) and form a mobile ad hoc network (MANET). Mobile peers that are in the transmission range of each other can communicate with their peers directly. To communicate with peers outside the transmission range, messages are propagated across multiple hops in the network. Hence, P2P file sharing systems can be also deployed over MANET. Due the nature of MANET, these systems suffer from tow principles constraints. Firstly, wireless medium is much more dynamic due to peer mobility and the frequent variations in channel quality due to interference and fading [4]. Secondly, mobile devices are battery operated and energy-limited. If a peer is frequently asked to provide or relay files, its battery would be quickly exhausted.

A challenging problem in these systems is (i) the selection of best peers that share pertinent resources for user's queries and (ii) guarantee that the best peers can be reached in such dynamic and energy-limited environment (query routing problem).

In the literature, several works proposed different techniques of query routing in P2P systems on wired scenarios [16]. However, they are not applicable to MANET, since they don't consider the constraints of this network; thus they cannot grantee that the pertinent peers can be reached in such dynamic and energy-limited environment. Hence, energy efficiency and peer mobility are uncompromising factors in the design of query routing P2P file sharing systems over MANET. Several routing methods have been proposed for P2P file sharing systems over MANET. Each of them has its own advantages and limits.

In this paper, we propose a context-aware integrated routing method for P2P file sharing systems over MANET. The key contributions of our proposal are:

- The selection of best peers that share pertinent resources is based on the query content and the user's profile. Indeed, each peer builds a profile of its neighbors. The profile contains the list of the most recent past queries and neighbor that supplied answers for. We defined a similarity function that computes the aggregate similarity of a peer to a given query.

- Our routing method takes into account the constraints of MANET environment to guarantee that the best peers can be reached. Hence, we defined a Link_stablity function that combines the peer mobility and battery energy factors to compute the stability of link between two peers. In addition, we defined a function to guarantee a load balancing and palliate the congestion problem.

The rest of the paper is organized as follows. In Section 2, we present a critical overview of query routing methods in P2P systems over MANET. Section 3 discusses our approach. Section 4 concludes with some proposed direction for further works.

2. RELATED WORK

In the literature there are several points of view of the routing problem in unstructured P2P systems over MANET. Bin Tang et al [15] classify the existing approaches for unstructured P2P systems over MANET into layered or integrated design approaches.

2.1 Layered design approach

The layered design decouples functionalities of the application layer and the network layer, which enables independent development of protocols at the two layers. In this design, routing protocol at application layer (for example, Gnutella) are operated on top of an existing MANET routing protocol at network layer. This design is similar to the approach in the Internet, which layers a P2P protocol on top of the existing IP infrastructure. The routing protocol at the application layer selects the overlay neighbor to forward the search query then it uses an existing routing protocol at the network layer (i.e. DSR [8], AODV [11], DSDV [12], etc) to localize this neighbor. However, due to peer mobility, these overlay neighbors may not reflect the current physical topology of the ad hoc network, and thus may need a multi-hop route to be reached. As a result, each such overlay hop required by Gnutella at the application layer could result in a costly flooding-based route discovery by the multi-hop routing protocol.

2.2 Integrated design approach

MANETs are a limited resource environment where the performance can be more important than portability and separation of functionalities. Hence, integrated design approach is proposed as alternative to layered design approach. In integrated design, routing protocol at the application layer is integrated with a MANET routing protocol at the network layer. In the literature, there are several integrated approaches.

A first idea consists to build an efficient unstructured P2P overlay over MANET. In this overlay connections between mobile peers are closely match the physical topology of the underlying MANET. To find relevant resources for a given search query, flooding technique is used. Andrew et al [9] propose a decentralized and dynamic topology control protocol called $TCP2P$. This protocol allows each peer in MANET to select a set of neighbors according to preference defined function that take into account the energy efficiency, fairness and incentive. After building the network topology, each peer routes the query to its neighbors regardless the query content. Although this protocol virtually controls the macroscopic usage of energy and establishes a stable link, in term of energy efficiency, fairness and incentive, between a source and destination peers. However, it does not compromise the satisfaction of user because queries are flooded regardless their content. Moreover, user's mobility is not considered. $E-UnP2P$ method [14] builds an efficient overlay avoiding redundant links and redundant transmissions while ensuring connectivity among the peers, it introduces a root-peer in the P2P network connecting all other peers. Each peer maintains connection with other closest peers such that it can reach the root-peer. Using the information of its directly connected and 2-hop away (logically) neighbor peers, each peer builds up a minimum-spanning tree to identify far away peers and builds up the overlay closer to the physical network. Thereafter, when a peer wants to retrieve a file, it sends the query to all of its neighbor peers.

A second idea consists to define a progressive search mechanism that allows to route the search queries to the best neighbors. In order to find content, a peer sends a query to its best neighbors, which, in turn, forward the query to their best neighbors and so on, until the query time-to-live (TTL) expires. To select the best neighbors a peer is based on some factors (i.e. Battery energy, signal power, neighbor velocity, neighbor's content, etc.). In Data Dissemination in Mobile P2P Networks [13] each peer maintains a global description of other peers' content (content synopses), and utilizes that synopses in order to route queries more efficiently. A peer that receives a query searches in its local collection. If it is not possible to answer this query, it calculates a score of peers from the global index then propagates the query to the peers, which have the greatest score. If there is no match between the query and the content synopses, the query is forwarded to a set of random neighbors. Content synopses must be updated whenever an object is added, deleted or its contents have changed, which generates a lot of message traffic and load charge of peers. Furthermore, this method does not consider the mobility and the energy factors. In enhancing peer-to-peer content discovery techniques over mobile ad hoc networks [4], the authors propose to improve the unstructured P2P over MANET using Gossiping [5] approach of MANET routing protocol. This is achieved by computing the forwarding probability of a link based on the network load. Indeed, if a peer want to send a query it computes the forwarding probability for a given neighbor based on it computational load (the queue utilization of the neighbor) then forwards the query to neighbors with lower load. Significantly, this probability allows sending more messages to neighbors with lower load, while less messages are sent to saturated peers. This method grantees a load balancing between peers but it floods the query regardless its content. Furthermore, it does not consider the mobility and the energy factors.

3. PROPOSED APPROACH

In this paper, we consider the pure peer-to-peer systems (Gnutella system) over MANET and we propose new techniques that are more efficient than the Gnutella search. Flooding is a fundamental file search operation in pure peer-to-peer (P2P) file sharing systems, in which a peer starts the file search procedure by broadcasting a query to a random set of its neighbors, who continue to propagate it with the same manner to their neighbors. This procedure repeats until a time-to-live (TTL) counter is decremented to 0. If a contacted peer has pertinent resources for the search query, it sends a query hit message to the source peer. The query hit message is routed back to the source peer through the reverse path of the query message. This solution generates a very large number of messages and it cannot quickly locate the request resource. Furthermore, query hits may not be received by the source peer due to the peer mobility and energy limitation. Indeed, peers in the reverse path of the query message may turn off or move out of the network.

In our approach, to find relevant resources for a specific user query a peer sends the query to its best neighbors, which, in turn, forward the query to their best neighbors and so on, until the query time-to-live (TTL) expires. Neighboring peers refer to those peers which are within the transmission range of the forwarding peer.

Assume that a peer p_i which has a set N of neighboring peers. Now the question is "How we determine the best k neighbors?", k is a user defined threshold and $k \leq N$. In the following, we present the different context features considered to select the best k neighbors for a given query q. Thereafter, we present our neighbors selection algorithm.

3.1 Context features

3.1.1 User's profile and query content

We consider the query content to help the querying peer to find the most relevant answers to its query quickly and efficiency. To achieve this, a peer estimates, for each query, which of its neighbors are more likely to reply to this query, and propagates the query message to those peers. To determine the pertinent neighbors, we compute the similarity between the query and each neighbors. Hence, each peer maintains a profile for each of its peers. The profile contains the list of the most recent past queries, that the specific peer that provided the answer for. Although logically we consider each profile to be a distinct list of queries, we use a single Queries table with (Query-peer) entries that keeps the most recent queries the peer has recorded.

For each query it receives, the receiver peer uses the profiles of its peers to find which ones are more likely to have documents that are relevant to the query. To compute the similarity, the receiver peer compares the query to previously seen queries and finds the most similar ones in the repository. To find the similarity between the queries, it uses the cosine similarity [10]. Thereafter, we compute an aggregate similarity of a peer to a given query. The aggregate similarity of peer n_j to query q that peer p_i computes is:

$$Psim_{p_i}(n_j, q) = \sum_{q_k \quad was \quad answered \quad by \quad n_j} Cosine(q_k, q) \tag{1}$$

3.1.2 Link stability

We defined a Link_stablity function that combines the peer mobility and battery energy factors to compute the stability of link between two peers. Before describing our function, we present two principle metrics. The first one takes into account the peer mobility factor to predict lifetime of a link between tow peers. The second one predicts the remaining battery energy of a given peer.

Peer mobility

In MANET environment peers are free to move from their location at anytime. In our approach we consider this important factor, thus we predict the lifetime of a link between the forwarding peer and its neighbors. To predict the lifetime of a link $i - j$ between the peer p_i and its neighbor $n_j \in N$ we are based on the RABR protocol [3] functions. This protocol operates at network layer. It predicts the lifetime of a link $i-j$ using a metric called the "affinity" a_{ij} and it is a measure of the time taken by peer n_j to move out of the range of peer p_i. Peers exchange beacons periodically. Peer p_i periodically samples, for every $\triangle_t$ time units, the strength of the beacon signals received from peer n_j. The rate of change of signal strength is given as:

$$\Delta(S_{ij}) = \frac{S_{ij}(current) - S_{ij}(prev)}{\Delta_t} \tag{2}$$

The above quantity is then averaged over the last few samples to obtain $\Delta(S_{ij}(ave))$. Hence, based on this metric we define a link lifetime measure $Lifetime(i - j)$, which computes the time taken by peer n_j to move out of the range of peer p_i, as follows:

$$Lifetime(i-j) = \begin{cases} \Delta(S_{ij}(ave)) \quad if \quad \Delta(S_{ij}(ave)) \geq 0 \\ \frac{S_{thresh} - S_{ij}(current)}{\Delta(S_{ij}(ave))} \quad otherwise \end{cases} \tag{3}$$

Battery energy

The calculation of energy level is important to determine the battery level of every peer during active data transmission. We assume that the battery level of a wireless peer decreased when the peer initiated data transmission or when the peer forwards packets. A peer gets killed (disconnected) if the battery power finishes. To predict the remaining battery power we assume that the transmit power is fixed. As in [6], energy required for each operation like receive, transmit, broadcast, discard on a packet is given by:

$$E(packet) = b \times (packet\_size) + c \tag{4}$$

Coefficient b denotes the packet size dependent energy consumption whereas c is a fixed cost that accounts for acquiring the channel and for MAC layer control negotiation. Each peer has to maintain a table to record the remaining energy of its neighboring peer. This data is used by the peer to predict the remaining energy of the neighboring peer n_j. Assume the remaining energy, of a neighbor peer at time $t1$ and t2 are $rengy1(n_j)$ and $rengy2(n_j)$. The prediction of remaining energy of this peer at time t is given by

$$\begin{aligned} rengy(n_j) \quad &= rengy2(n_j) \\ &+[(rengy2(n_j) - rengy1(n_j))/(t2 - t1)] \times (t - t2) \end{aligned} \tag{5}$$

Every peer has to calculate the $rengy$ by itself and sends it to its neighbors.

We combine the lifetime and the remaining energy metrics to define our function Link_stability. This metric calculates the time taken by peer n_j to move out of the range of peer p_i or the battery power of n_j finishes. The $Link\_stability(i-j)$ of a link $i-j$ between the peer p_i and its neighbor n_j is computed as follows:

$$Link\_stability(i-j) = Min(rengy(n_j), Lifetime(i-j)) \tag{6}$$

Where, $rengy(n_j)$ is the remaining energy of the neighbor n_j.

3.1.3 Peer load

A vital part of the optimal network is the load balancing. For instance, job completion becomes complex, if huge load is given to the peers with less processing capabilities. There is a possibility of load imbalance due to that the computing/processing power of the systems are non-uniform few peers may be idle and few will be overloaded. A peer which has high processing power finishes its own work quickly and is estimated to have less or no load at all most of the time. However, if we send queries only to peers that have hight processing capabilities data packets will take routes that could introduce more delay hence increasing latency. With proper ways to transferring traffic load onto routes that are relatively less congested can result in overall better throughput and reduced latency. An important parameter indicates the line congestion is the queue utilization of the neighbor (i.e. Number of packets waiting in queue), a high count indicates line congestion. We define a Peer_Load function based on the CPU capabilities and the queue utilization of the neighbor. The Peer_Load of a neighbor n_j is calculated as follows:

$$Peer\_Load(n_j) = cpu \times (1-u) \tag{7}$$

where cpu is the processing power and $0 \leq u \leq 1$ is the queue utilization of the neighbor n_i. This function allows to send more messages to neighbors with lower load, while less messages are sent to saturated peers.

3.2 Neighbors selection algorithm

To select its K best neighbors, the forwarding peer p_i ranks its neighbors according to a Preference function that we define. Thereafter, it selects the first k neighbors, which have the greatest score. Our Preference function computes the score of each neighbor n_j for a given query q, as a weighted arithmetic sum of Link_stability, Peer_Load and Psim metrics:

$$\begin{aligned} Pref(n_j) &= \alpha 1 \times Link\_stability(i-j) + \alpha 2 \times Peer\_Load(n_j) \\ &+ \alpha 3 \times Psim_{P_i}(n_j, q) \end{aligned} \tag{8}$$

where $\alpha 1$, $\alpha 2$ and $\alpha 3$ represent the relative importance of these three metrics.

4. CONCLUSION AND FUTURE WORKS

We have presented a novel context-aware integrated routing method for P2P file sharing systems over MANET. Our method selects the best peers based on the query content and the user's profile. Furthermore, it considers the energy efficiency, peer mobility and peer load factor into the query forwarding process to guarantee that the pertinent peers can be reached. As the future work, we plan to implement the proposed method and evaluate its retrieval effectiveness and routing efficiency.

5. REFERENCES

[1] Bluetooth. http://simple.wikipedia.org/wiki/Bluetooth, March, 2012.

[2] Wifi. http://simple.wikipedia.org/wiki/Wifi, March, 2012.

[3] S. Agarwal, A. Ahuja, J. P. Singh, and R. Shorey. Route-lifetime assessment based routing (rabr) protocol for mobile ad-hoc networks. In *ICC (3)*, pages 1697–1701, 2000.

[4] D. N. da Hora, D. F. Macedo, L. B. Oliveira, I. G. Siqueira, A. A. F. Loureiro, J. M. Nogueira, and G. Pujolle. Enhancing peer-to-peer content discovery techniques over mobile ad hoc networks. *Comput. Commun.*, 32(13-14):1445–1459, Aug. 2009.

[5] M. Dietzfelbinger. Gossiping and broadcasting versus computing functions in networks. In *STACS 97*, volume 1200, pages 189–200. Springer Berlin / Heidelberg, 1997.

[6] N. Gupta, K. Bafna, and N. Gupta. 2011 international conference on information and network technology. In *2011 Seventh International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, volume 4. IACSIT Press, 2011.

[7] H. Jin, X. Ning, H. Chen, and Z. Yin. Efficient query routing for information retrieval in semantic overlays. In *Proceedings of the 21st Annual ACM Symposium on Applied Computing*, pages 23–27, Dijon, France, April 23–27 2006. ACM Press.

[8] D. B. Johnson, D. A. Maltz, and J. Broch. Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks. In *In Ad Hoc Networking, edited by Charles E. Perkins, Chapter 5*, pages 139–172. Addison-Wesley, 2001.

[9] A.-H. Leung and Y.-K. Kwok. On topology control of wireless peer-to-peer file sharing networks: energy efficiency, fairness and incentive. In *World of Wireless Mobile and Multimedia Networks, 2005. WoWMoM 2005. Sixth IEEE International Symposium on a*, pages 318 – 323, june 2005.

[10] J. Makhoul, F. Kubala, R. Schwartz, and R. Weischedel. Performance measures for information extraction. In *Proceedings of DARPA Broadcast News Workshop*, pages 249–252, Herndon, VA, February 1999.

[11] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (aodv) routing, 2003.

[12] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *Proceedings of the conference on Communications architectures, protocols and applications*, SIGCOMM '94, pages 234–244, New York, NY, USA, 1994. ACM.

[13] T. Repantis and V. Kalogeraki. Data dissemination in mobile peer-to-peer networks. In *Proceedings of the 6th international conference on Mobile data management*, MDM '05, pages 211–219, New York, NY, USA, 2005. ACM.

[14] N. Shah and D. Qian. An efficient unstructured p2p overlay over manet using underlying proactive routing. In *2011 Seventh International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, pages 248 –255, dec 2011.

[15] B. Tang, Z. Zhou, A. Kashyap, and T. cker Chiueh. An integrated approach for p2p file sharing on multi-hop wireless networks. In *WiMob (3)*, pages 268–274, 2005.

[16] T. Yeferny and K. Arour. Learningpeerselection: A query routing approach for information retrieval in p2p systems. In *International Conference on Internet and Web Applications and Services*, pages 235–241, Barcelona, Spain, May 09-15 2010. IEEE Computer Society.

Considering the High Level Critical Situations in Context-Aware Recommender Systems

Djallel Bouneffouf
Department of Computer Science,
Télécom SudParis, UMR CNRS
Samovar
91011 Evry Cedex, France
Djallel.Bouneffouf@it-sudparis.eu

Amel Bouzeghoub
Department of Computer Science,
Télécom SudParis, UMR CNRS
Samovar
91011 Evry Cedex, France
Amel.Bouzeghoub@it-sudparis.eu

Alda Lopes Gançarski
Department of Computer Science,
Télécom SudParis, UMR CNRS
Samovar
91011 Evry Cedex, France
Alda.Gancarski@it-sudparis.eu

ABSTRACT

Most existing approaches in Context-Aware Recommender Systems (CRS) focus on recommending relevant items to users taking into account contextual information, such as time, location, or social aspects. However, none of them have considered the problem of user's content dynamicity. This problem has been studied in the reinforcement learning community, but without paying much attention to the contextual aspect of the recommendation. We introduce in this paper an algorithm that tackles the user's content dynamicity by modeling the CRS as a contextual bandit algorithm. It is based on dynamic exploration/exploitation and it includes a metric to decide which user's situation is the most relevant to exploration or exploitation. Within a deliberately designed offline simulation framework, we conduct extensive evaluations with real online event log data. The experimental results and detailed analysis demonstrate that our algorithm outperforms surveyed algorithms.

1. INTRODUCTION

Mobile technologies have made access to a huge collection of information, anywhere and anytime. In particular, most professional mobile users acquire and maintain a large amount of content in their repository. Moreover, the content of such repository changes dynamically, undergoes frequent updates. In this sense, recommender systems must promptly identify the importance of new documents, while adapting to the fading value of old documents. In such a setting, it is crucial to identify and recommend interesting content for users.

A considerable amount of research has been done in recommending interesting content for mobile users. Earlier techniques in Context-Aware Recommender Systems (CRS) [3, 6, 12, 5, 22, 23] are based solely on the computational behavior of the user to model his interests regarding his surrounding environment like location, time and near people (the user's situation). The main limitation of such approaches is that they do not take into account the dynamicity of the user's content.

This gives rise to another category of recommendation techniques that try to tackle this limitation by using collaborative, content-based or hybrid filtering techniques. Collaborative filtering, by finding similarities through the users' history, gives an interesting recommendation only if the overlap between users' history is high and the user's content is static[18]. Content-based filtering, identify new documents which match with an existing user's profile, however, the recommended documents are always similar to the documents previously selected by the user [15]. Hybrid approaches have been developed by combining the two latest techniques; so that, the inability of collaborative filtering to recommend new documents is reduced by combining it with content-based filtering [13].

However, the user's content in mobile undergoes frequent changes. These issues make content-based and collaborative filtering approaches difficult to apply [8].

Few works found in the literature [13, 21] solve this problem by addressing it as a need for balancing exploration and exploitation studied in the "bandit algorithm" [20].

A bandit algorithm B exploits its past experience to select documents (arms) that appear more frequently. Besides, these seemingly optimal documents may in fact be suboptimal, because of the imprecision in B's knowledge. In order to avoid this undesired situation, B has to explore documents by choosing seemingly suboptimal documents so as to gather more information about them. Exploitation can decrease short-term user's satisfaction since some suboptimal documents may be chosen. However, obtaining information about the documents' average rewards (i.e., exploration) can refine B's estimate of the documents' rewards and in turn increases long-term user's satisfaction.

Clearly, neither a purely exploring nor a purely exploiting algorithm works well, and a good tradeoff is needed.

The authors on [13, 21] describe a smart way to balance exploration and exploitation in the field of recommender systems. However, none of them consider the user's situation during the recommendation.

In order to give CRS the capability to provide the mobile user's information matching his/her situation and adapted to the evolution of his/her content (good exr/exp tradeoff in the bandit algorithm), we propose an algorithm witch takes into account the user's situation for defining the (exr/exp) tradeoff, and then selects suitable situations for either exploration or exploitation.

The rest of the paper is organized as follows. Section 2 reviews some related works. Section 3 presents the user’s model of our CRS. Section 4 describes the algorithms involved in the proposed approach. The experimental evaluation is illustrated in Section 5. The last section concludes the paper and points out possible directions for future work.

2. RELATED WORKS

We review in the following recent relevant recommendation techniques that tackle the two issues mentioned above, namely: following the evolution of the user’s contents using bandit algorithm and considering the user’s situation on recommender system.

2.1 Bandit Algorithms Overview

The (exr/exp) tradeoff was firstly studied in reinforcement learning in 1980's, and later flourished in other fields of machine learning [16, 19]. Very frequently used in reinforcement learning to study the (exr/exp) tradeoff, the multi-armed bandit problem was originally described by Robbins [20].

The ε-greedy is the one of the most used strategy to solve the bandit problem and was first described in [14]. The ε-greedy strategy choose a random document with epsilon-frequency (ε), and choose otherwise the document with the highest estimated mean, the estimation is based on the rewards observed thus far. ε must be in the open interval [0, 1] and its choice is left to the user.

The first variant of the ε-greedy strategy is what [9, 14] refer to as the ε-beginning strategy. This strategy makes exploration all at once at the beginning. For a given number $I \in N$ of iterations, the documents are randomly pulled during the εI first iterations. During the remaining $(1-\varepsilon)I$ iterations, the document of highest estimated mean is pulled.

Another variant of the ε-greedy strategy is what Cesa-Bianchi and Fisher [14] call the ε-decreasing strategy. In this strategy, the document with the highest estimated mean is always pulled except when a random document is pulled instead with an ε_i frequency, where n is the index of the current round. The value of the decreasing ε_i is given by $\varepsilon_i = \{\varepsilon_0 / i\}$ where $\varepsilon_0 \in]0,1]$. Besides ε-decreasing, four other strategies are presented in [4]. Those strategies are not described here because the experiments done by [4] seem to show that, with carefully chosen parameters, ε-decreasing is always as good as the other strategies.

Compared to the standard multi-armed bandit problem with a fixed set of possible actions, in CRS, old documents may expire and new documents may frequently emerge. Therefore it may not be desirable to perform the exploration all at once at the beginning as in [9] or to decrease monotonically the effort on exploration as the decreasing strategy in [14].

Few research works are dedicated to study the contextual bandit problem on Recommender System, where they consider user’s behavior as the context of the bandit problem.

In [10], authors extend the ε-greedy strategy by updating the exploration value ε dynamically. At each iteration, they run a sampling procedure to select a new ε from a finite set of candidates. The probabilities associated to the candidates are uniformly initialized and updated with the Exponentiated Gradient (EG) [10]. This updating rule increases the probability of a candidate ε if it leads to a user’s click. Compared to both ε-beginning and decreasing strategy, this technique improves the results.

In [13], authors model the recommendation as a contextual bandit problem. They propose an approach in which a learning algorithm selects sequentially documents to serve users based on contextual information about the users and the documents. To maximize the total number of user’s clicks, this work proposes the LINUCB algorithm that is computationally efficient.

The authors in [4, 9, 13, 14, 21] describe a smart way to balance exploration and exploitation. However, none of them consider the user’s situation during the recommendation.

2.2 Managing the User’s Situation

Few research works are dedicated to manage the user’s situation on recommendation.

In [7, 17] the authors propose a method which consists of building a dynamic user’s profile based on time and user’s experience. The user’s preferences in the user’s profile are weighted according to the situation (time, location) and the user’s behavior. To model the evolution on the user’s preferences according to his temporal situation in different periods, (like workday or vacations), the weighted association for the concepts in the user’s profile is established for every new experience of the user. The user’s activity combined with the user's profile are used together to filter and recommend relevant content.

Another work [12] describes a CRS operating on three dimensions of context that complement each other to get highly targeted. First, the CRS analyzes information such as clients’ address books to estimate the level of social affinity among the users. Second, it combines social affinity with the spatiotemporal dimensions and the user’s history in order to improve the quality of the recommendations.

In [3], the authors present a technique to perform user-based collaborative filtering. Each user’s mobile device stores all explicit ratings made by its owner as well as ratings received from other users. Only users in spatiotemporal proximity are able to exchange ratings and they show how this provides a natural filtering based on social contexts.

Each work cited above tries to recommend interesting information to users on contextual situation; however they do not consider the evolution of the user’s content.

As shown in above, none of the mentioned works tackles both problems of the evolution user’s content and user’s situation consideration in the recommendation. This is precisely what we intend to do with our approach, by modeling the CRS as a contextual bandit algorithm, and considering the user’s situation when managing the (exr/exp)-tradeoff on recommendation.

The two features cited above are not considered in the surveyed approaches as far as we know.

In what follows, we define briefly the structure of the user’s model and the methods for inferring the recommendation situations. Then, we explain how to manage the exploration/exploitation strategy, according to the current situation.

3. USER AND CONTEXT MODELS

The user's model is structured as a case base, which is composed of a set of past situations with their corresponding user's preferences, denoted $PS = \{(S^i; UP^i)\}$, where S^i is a user's situation (Section 3.2.1) and UP^i its corresponding user's preferences (Section 3.1).

3.1 The User's Preferences

The user's preferences are contextual and might depend on many factors, like the location or the current task within an activity. Thus, they are associated to the user's situation and the user's activity. Preferences are deduced during the user's navigation activities. A navigation activity expresses the following sequence of events:

(i) the user's logs in the system and navigates across documents to get the desired information;

(ii) the user expresses his/her preferences about the visited documents. We assume that a visited document is relevant, and thus belongs to the user's preferences, if there are some observable user's behaviors through two types of preference:

- The direct preference: the user expresses his/her interest in the document by inserting a rate, like for example putting starts ("*") at the top of the document.

- The indirect preference: it is the information that we extract from the user's system interaction, for example the number of clicks on the visited documents or the time spent on a document.

Let *UP* be the preferences submitted by a specific user in the system at a given situation. Each document in *UP* is represented as a single vector $d=(c_1,...,c_n)$, where c_i $(i=1, .., n)$ is the value of a component characterizing the preferences of *d*. We consider the following components: the document's identifier, the total number of clicks on *d*, the total time spent reading *d*, the number of times *d* was recommended, and the direct preference rate on *d*.

3.2 Context Model

A user's context *C* is a multi-ontology representation where each ontology corresponds to a context dimension $C=(\boldsymbol{O}_{Location}, \boldsymbol{O}_{Time}, \boldsymbol{O}_{Social})$. Each dimension models and manages a context information type. We focus on these three dimensions since they cover all needed information. These ontologies are described in [1] and are not developed in this paper.

3.2.1 Situation Model

A situation is a projection on one or several user's context dimensions. In other words, we consider a situation as a triple $s = (\boldsymbol{O}_{Location}.x_i, \boldsymbol{O}_{Time}.x_j, \boldsymbol{O}_{Social}.x_k)$ where x_i, x_j and x_k are ontology concepts or instances. Suppose the following data are sensed from the user's mobile phone: the GPS shows the latitude and longitude of a point "48.8925349, 2.2367939"; the local time is "Mon *May* 3 12:10:00 2012" and the calendar states "meeting with Paul Gerard". The corresponding situation is:

S=($\boldsymbol{O}_{Location}$,"48.89,2.23",

$\boldsymbol{O}_{Time}$."Mon_May_3_12:10:00_2012", $\boldsymbol{O}_{Social}$. "Paul_Gerard").

To build a more abstracted situation, we interpret the user's behavior from this low-level multimodal sensor data using ontologies reasoning means. For example, from *S*, we obtain the following situation:

MeetingAtRestaurant=

($\boldsymbol{O}_{Location}$.Restaurant, $\boldsymbol{O}_{Time}$.Work_day, $\boldsymbol{O}_{Social}$.Financial_client).

For simplification reasons, we adopt in the rest of the paper the following notation:

$S = (x_i, x_j, x_k)$. The previous example situation became thus:

MeetingAtRestarant=(Restaurant, Work_day, Financial_client).

Among the set of captured situations, some of them are characterized as *high-level critical situations*.

3.2.2 High Level Critical Situations (HLCS)

A HLCS is a class of situations where the user needs the best information that can be recommended by the system, for instance, when the user is in a professional meeting. In such a situation, the system must exclusively perform exploitation rather than exploration-oriented learning. In the other case, for instance where the user is using his/her information system at home, on vacation with friends S = (*home, vacation, friends*). The system can make some exploration by recommending the user some information ignoring their interest. The HLCS situations are for the moment predefined by the domain expert. In our case we conduct the study with professional mobile users, which is described in detail in (section 5). As examples of HLCS, we can find *S1* = (*company, Monday morning, colleague*), *S2* = (*restaurant, midday, client*) or *S3*= (*company, morning, manager*).

4. THE PROPOSED RECOMMENDATION ALGORITHM

The problem of recommending documents can be naturally modeled as a multi-armed bandit problem with context information. In our case we consider the user's situation as the context information of the multi-armed bandit. Following previous work [11], we call it a contextual bandit. Formally, our contextual-bandit algorithm proceeds in trials $t = 1...T$. For each trial *t*, the algorithm performs the following tasks:

Task 1: Let S^t be the current user's situation, and *PS* be the case base containing the set of past situations and corresponding user's preferences. The system compares S^t with the situations in *PS* in order to choose the most similar S^p using the *RetrieveCase()* method (Section 4.2.1).

Task 2: Let *D* be the document collection and $D_p \in D$ the set of documents that were recommended in situation S^p. When the user read each document $d_i \in D_p$, the system observed his behavior and interpreted it as a reward. Based on the observed documents' rewards, the algorithm chooses the document $d_p \in D_p$ with the greater reward r_p; this is done using the *RecommendDocuments*() method (Section 4.2.2).

Task 3: The algorithm improves its document-selection strategy with the new current observation (d_p, r_t). The updating of the case base is done using the *Auto_improvement()* method (Section 4.2.3).

In tasks 1 to 3, the total T-trial reward for each document d_i in D is defined as $\sum_{t=1}^{T} r_{t,d_i}$ while the optimal expected T-trial reward is defined as $\mathrm{E}\left[\sum_{t\in T_i} r_{t,d_i^*}\right]$ where d_i^* is the document with maximum expected total reward, where T_i is the set of trials from T where d_i^* was recommended to the user. Our goal is to design the bandit algorithm so that the expected total reward is maximized.

In the field of document recommendation, when a document is presented to the user and this one selects it by a click, a reward of 1 is incurred; otherwise, the reward is 0. With this definition of reward, the expected reward of a document is precisely its Click Through Rate (CTR). The CTR is the average number of clicks on a recommended document, computed dividing the total number of clicks on it by the number of times it was recommended. It is important to know here that no reward is observed for non-recommended documents.

4.1 The ε-greedy() Algorithm

The *ε-greedy* algorithm recommends a predefined number of documents *N*, each one computed using the following equation:

$$d_i = \begin{cases} \arg\max_{UC}(\,getCTR(d)) & if\ q > \varepsilon \\ \\ Random(UC) & otherwise \end{cases} \qquad (1)$$

In Eq. 1, $i \in \{1,\ldots N\}$, $UC=\{d_1,\ldots,d_P\}$ is the set of documents corresponding to the user's preferences; *getCTR* is the function which estimates the CTR of a given document; *Random* is the function returning a random element from a given set, allowing to perform exploration; *q* is a random value uniformly distributed over [0, 1] which defines the exploration/exploitation tradeoff; *ε* is the probability of recommending a random exploratory document.

4.2 Contextual-ε-greedy()

To adapt the ε-greedy algorithm to a context aware environment, we propose to compute the similarity between the current situation and each one in the situation base; if there is a situation that can be reused, the algorithm retrieves it, and then applies the ε-greedy algorithm to the corresponding user preferences. Alg. 1 describes the proposed *Contextual-ε-greedy*() algorithm which involves the following three methods.

```
Algorithm 1   Context-ε-greedy()
Input:  $\varepsilon, N, PS, S^t, B$
Output: $D^t$
        // Retrieve the most similar case
           $(S^P, UP^p)$ = RetrieveCase($S^t$, PS);
        // Recommend documents
           $D^t$=RecommendDocuments($\varepsilon$,$UP^p$, $S^t$, $S^p$, N, B);
           Receive a feedback $UP^t$ from the user;
        // update user's profile
           Auto_improvement(PS, $UP^t$, $S^t$, $S^p$);
Endfor
```

4.2.1 *RetrieveCase()*

Given the current situation S^t, the *RetrieveCase()* method determines the expected user's preferences by comparing S^t with the situations in past cases *PS* in order to choose the most similar one S^p. The method returns, then, the corresponding case (S^p, UP^p). S^p is selected from *PS* by computing the following expression:

$$S^p = \underset{S^i \in PS}{\arg\max}\left(\sum_j \alpha_j \cdot sim_j\left(X_j^t, X_j^i\right)\right) \qquad (2)$$

In Eq.2, sim_j is the similarity metric related to dimension *j* between two concepts X^t and X^i. This similarity depends on how closely X^t and X^i are related in the corresponding ontology (location, time or social). α_j is the weight associated to dimension *j*, and it is set out by using an arithmetic mean as follows:

$$\alpha_j = \frac{1}{T}\sum_{k=1}^{T}\gamma_j^k \qquad (3)$$

In Eq. 3, $\gamma_j^i = sim_j\left(x_j^t, x_j^p\right)$ at trial $k \in \{1,\ldots,T\}$ from the *T* previous recommendations, where $x_j^p \in S^p$. The idea here is to augment the importance of a dimension with the corresponding previously computed similarity values, reflecting the impact of the dimension when computing the most similar situation in Eq. 2.

The similarity between two concepts of a dimension *j* in an ontological semantics depends on how closely they are related in the corresponding ontology (location, time or social). We use the same similarity measure as [24] defined by Eq. 4:

$$sim_j\left(x_j^t, x_j^c\right) = 2 * \frac{deph(LCS)}{(deph(x_j^t) + deph(x_j^c))} \qquad (4)$$

In Eq. 4, LCS is the Least Common Subsumer of x_j^t and x_j^c, and deph is the number of nodes in the path from the node to the ontology root.

4.2.2 *RecommendDocuments()*

In order to insure a better precision of the recommender results, the recommendation takes place only if the following condition is verified: $sim(S^t, S^p) \geq B$, where *B* the similarity threshold value and $sim(S^t, S^p) = \sum_j \alpha_j sim_j\left(x_j^t, x_j^p\right)$.

To improve the adaptation of the ε-greedy algorithm to HLCS situations, if $S^p \in HLCS$, we propose the system to not make exploration when choosing the document to recommend, as indicated in the following equation:

$$d_i = \begin{cases} \arg\max_{UC} (\, getCTR(d)) & if\ S^p \in HLCS \\ \varepsilon\text{-}greedy(\,) & otherwise \end{cases} \qquad (5)$$

In Eq. 5, if S^p is not *HLCS*, the system recommends documents using *ε-greedy* with an *ε* computed at an initialization step by testing different *ε* and selects the optimal one, this step is described below (Section 5.4).

4.2.3 *Auto_improvement ()*

This method is used to update the user's preferences w. r. t. the number of clicks and number of recommendations for each recommended document on which the user clicked at least one time. Depending on the similarity between the current situation S^t and its most similar situation S^p (computed with *Retrieve-Case()*, Section 4.2.1), being 3 the number of dimensions in the context, two scenarios are possible:

- $sim(S^t, S^p) \neq 3$: the current situation does not exist in the case base; the system adds to the case base the new case composed of the current situation S^t and the current user preferences UP^t.

- $sim(S^t, S^p)= 3$: the situation exists in the case base; the system updates the case having premise situation S^p with the current user preferences UP^t.

5. EXPERIMENTAL EVALUATION

In order to evaluate empirically the performance of our approach, and in the absence of a standard evaluation framework, we propose an evaluation framework based on a diary set of study entries. The main objectives of the experimental evaluation are:

(1) to find the optimal parameters of our algorithm.

(2) to evaluate the performance of the proposed algorithm w. r. t. the ε variation. In the following, we describe our experimental datasets and then present and discuss the obtained results.

5.1 Evaluation Framework

We have conducted a diary study with the collaboration of the French software company Nomalys[1]. This company provides a history application, which records time, current location, social and navigation information of its users during their application use. The diary study has taken 18 months and has generated 178369 diary situation entries.

Each diary situation entry represents the capture of contextual time, location and social information. For each entry, the captured data are replaced with more abstracted information using time, spatial and social ontologies. Table 1 illustrates three examples of such transformations.

IDS	Users	Time	Place	Client
1	Paul	Workday	Paris	Finance client
2	Fabrice	Workday	Roubaix	Social client
3	John	Holiday	Paris	Telecom client

Table 1: Semantic diary situation

From the diary study, we have obtained a total of 2759283 entries concerning the user's navigation, expressed with an average of 15.47 entries per situation. Table 2 illustrates examples of such diary navigation entries, where **Click** is the number of clicks on a document; **Time** is the time spent on reading a document, and **Interest** is the direct interest expressed by stars (the maximum number of stars is five).

IdDoc	IDS	Click	Time	Interest
1	1	2	2'	***
2	1	4	3'	*
3	2	1	5'	*

Table 2: Diary navigation entries

5.2 Finding the Optimal Parameters

In our experiments, we have firstly collected the 3000 situations (*HS)* with an occurrence greater than 100 to be statistically meaningful, and the 10000 documents (*HD*) that have been shown on any of these situations.

The testing step consists of evaluating the existing algorithms for a situation randomly selected from the sampling *HS*, taking into account the number of times that the situation was selected and the number occurrences of the situation^t in *HS*. The evaluation algorithm computes and displays the average CTR every 1000 iterations.

The average CTR for a particular iteration is the ratio between the total number of clicks and the total number of displays. The number of documents returned by the recommender system for each situation is 10 and we have run the simulation until the number of iterations reaches 10000.

5.2.1 *The threshold similarity value*

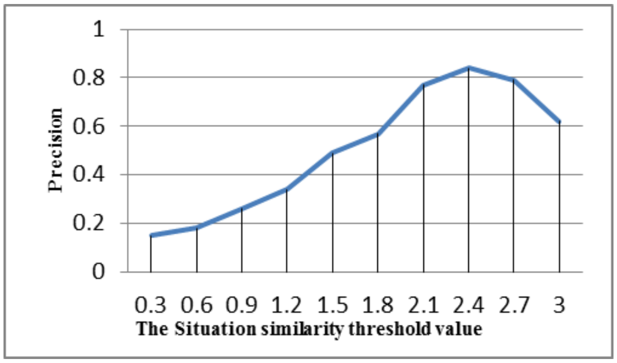


Figure 1. Effect of B threshold value on the similarity precision

[1] Nomalys is a company that provides a graphical application on Smartphones allowing users to access their company's data.

Figure 1 shows the effect of varying the threshold situation similarity parameter B (Section 2.2) in the interval [0, 3] on the overall precision. The results show that the best performance is obtained when B has the value 2.4 achieving a precision of 0.849.

So, we use the identified optimal threshold value (B = 2.4) of the situation similarity measure for testing our CRS.

5.3 Experimental Results

In our experiments, we have firstly collected the 3000 situations (HS) with an occurrence greater than 100 to be statistically meaningful, and the 10000 documents (HD) that have been shown on any of these situations.

The testing step consists of evaluating the existing algorithms for a situation randomly selected from the sampling HS, taking into account the number of times that the situation was selected and the number occurrences of the situation[t] in HS. The evaluation algorithm computes and displays the average CTR every 1000 iterations.

The average CTR for a particular iteration is the ratio between the total number of clicks and the total number of displays. The number of documents returned by the recommender system for each situation is 10 and we have run the simulation until the number of iterations reaches 10000.

5.4 Results for ε Variation

In order to evaluate only the impact of considering the user's situation in our bandit algorithm, we have replaced in *RecommendDocuments()*, the equation 5 by the equation 1, we call the new algorithm *Contextual-ε-greedy without HLCS*. Then we have compared this algorithm to the *ε-greedy* (Section 4.1).

Each of the competing algorithms requires a single parameter ε. Figure 2 shows how the average CTR varies for each algorithm with the respective ε.

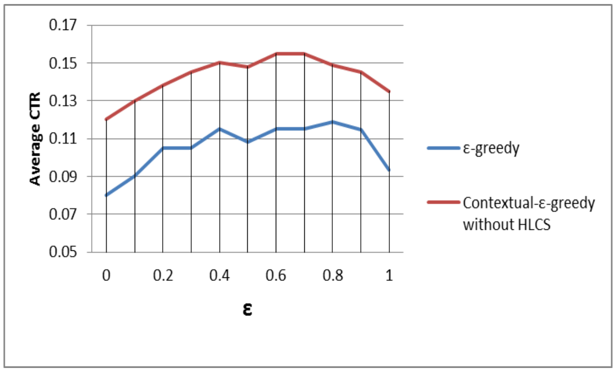


Figure 2. Variation ε tradeoff

Figure 2 shows that, when the ε is too small, there is an insufficient exploration; consequently the algorithms have failed to identify interesting documents, and have got a smaller number of clicks (average CTR). Moreover, when the parameter is too large, the algorithms seem to over-explore and thus lose a lot of opportunities to increase the number of clicks.

We can conclude from the evaluation that considering the user's situation is indeed helpful for *Context-ε-greedy* to find a better match between the user's interest and the evolution of his content (documents).

5.5 Evaluation The Impact of The HLCS

In order to evaluate the impact of the HLCS situations in the recommender system, we have compared *Contextual-ε-greedy without HLCS* and the original version of *Contextual-ε-greedy.* Figure 3 shows how the average CTR varies for each algorithm with the respective ε.

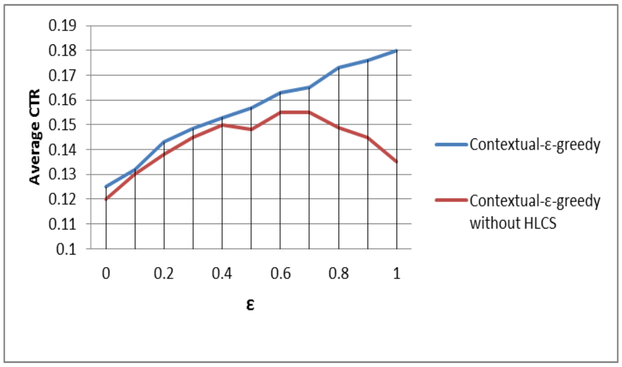


Figure 3. Variation ε tradeoff

As seen in the Figure 3, on one hand, when the ε is too small, there is an insufficient exploration; consequently the impact of the HLCS is low; on the other hand, when the parameter is too large, the *Contextual-ε-greedy* takes full advantage of exploration without wasting opportunities to establish good CTR (the impact of the HLCS is more important).

We can conclude from the evaluation that considering HLCS situations in recommender system allows a better precision on recommendation.

6. CONCLUSION

In this paper, we have studied the problem of exploitation and exploration in context-aware recommender systems and propose a new approach that balances adaptively exr/exp regarding the user's situation.

We have presented an evaluation protocol based on real mobile navigation contexts obtained from a diary study conducted with collaboration with the Nomalys French company. We have evaluated our approach according to the proposed evaluation protocol and show that it is effective.

In order to evaluate the performance of the proposed algorithm, we compare it with standard exr/exp strategy. The experimental results demonstrate that our algorithm performs better on average CTR in various configurations. Moreover, this study yields to the conclusion that considering the situation on the exploration/exploitation strategy significantly increases the performance of the system on following the user's contents evolution.

In the future, we plan to extend our situation with more context dimension, and we plan to evaluate our approach using an online framework.

References

[1] D. Bouneffouf, A. Bouzeghoub & A. L. Gançarski, Following the User's Interests in Mobile Context-Aware recommender systems. AINA Workshops, 657-662, 2012.

[2] G. Adomavicius, B. Mobasher, F. Ricci, Alexander Tuzhilin. Context-Aware Recommender Systems. AI Magazine. 32(3): 67-80, 2011.

[3] S. Alexandre, C. Moira Norrie, M. Grossniklaus and B.Signer, "Spatio-Temporal Proximity as a Basis for Collaborative Filtering in Mobile Environments". Workshop on Ubiquitous Mobile Information and Collaboration Systems, CAiSE, 2006.

[4] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite Time Analysis of the Multiarmed Bandit Problem. Machine Learning, 2, 235–256, 2002.

[5] R. Bader , E. Neufeld , W. Woerndl , V. Prinz, Context-aware POI recommendations in an automotive scenario using multi-criteria decision making methods, Proceedings of the 2011 Workshop on Context-awareness in Retrieval and Recommendation, p.23-30, 2011.

[6] L. Baltrunas, B. Ludwig, S. Peer, and F. Ricci. Context relevance assessment and exploitation in mobile recommender systems. Personal and Ubiquitous Computing, 2011.

[7] V. Bellotti, B. Begole, E.H. Chi, N. Ducheneaut, J. Fang, E. Isaacs, Activity-Based Serendipitous Recommendations with the Magitti Mobile Leisure Guide. Proceedings On the Move, 2008.

[8] W. Chu and S. Park. Personalized recommendation on dynamic content using predictive bilinear models. In Proc. of the 18th International Conf. on World Wide Web, pages 691–700, 2009.

[9] E. Even-Dar, S. Mannor, and Y. Mansour. PAC Bounds for Multi-Armed Bandit and Markov Decision Processes. In Fifteenth Annual Conference on Computational Learning Theory, 255–270, 2002.

[10] J. Kivinen and K. Manfred, Warmuth. Exponentiated gradient versus gradient descent for linear predictors. Information and Computation, 132-163, 1997.

[11] J. Langford and T. Zhang. The epoch-greedy algorithm for contextual multi-armed bandits. In Advances in Neural Information Processing Systems 20, 2008.

[12] R. Lakshmish, P. Deepak, P. Ramana, G. Kutila, D. Garg, V. Karthik, K. Shivkumar , Tenth International Conference on Mobile Data Management: Systems, Services and Middleware CAESAR: A Mobile context-aware, Social Recommender System for Low-End Mobile Devices, 2009.

[13] Li. Lihong, C. Wei, J. Langford, E. Schapire. A Contextual-Bandit Approach to Personalized News Document Recommendation. CoRR, Presented at the Nineteenth International Conference on World Wide Web, Raleigh, Vol. abs/1002.4058, 2010.

[14] S. Mannor and J. N. Tsitsiklis. The Sample Complexity of Exploration in the Multi-Armed Bandit Problem. In Sixteenth Annual Conference on Computational Learning Theory, 2003.

[15] D. Mladenic. Text-learning and related intelligent agents: A survey. IEEE Intelligent Agents, pages 44–54, 1999.

[16] H. Robbins. Some aspects of the sequential design of experiments. Bulletin of the American Mathematical Society, 1952.

[17] G. Samaras, C. Panayiotou. Personalized portals for the wireless user based on mobile agents. Proc. 2nd Int'l workshop on Mobile Commerce, pp. 70-74, 2002.

[18] J. B. Schafer, J. Konstan, and J. Riedi. Recommender systems in e-commerce. In Proc. of the 1st ACM Conf. on Electronic Commerce, 1999.

[19] R. Sutton and A. Barto. Reinforcement Learning: An Introduction. MIT Press, 1998.

[20] C. J. C. H. Watkins, Learning from Delayed Rewards. Ph.D. thesis. Cambridge University, 1989.

[21] Li. Wei, X. Wang, R. Zhang, Y. Cui, J. Mao, R. Jin. Exploitation and Exploration in a Performance based Contextual Advertising System: KDD'10, pp. 133-138. Proceedings of the 16th SIGKDD International Conference on Knowledge discovery and data mining, 2010.

[22] W. Woerndl, Florian Schulze: Capturing, Analyzing and Utilizing Context-Based Information About User Activities on Smartphones. Activity Context Representation, 2011.

[23] W. Woerndl, J. Huebner, R. Bader, D. Gallego-Vico: A model for proactivity in mobile, context-aware recommender systems. RecSys , 273-276, 2011

[24] Z. Wu and M. Palmer. Verb Semantics and Lexical Selection. In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, 133-138, 1994.

Privacy Preservation for Location-Based Services Based on Attribute Visibility

Masanori Mano[*]
Graduate School of Information Science
Nagoya University
mano@db.itc.nagoya-u.ac.jp

Xi Guo
Graduate School of Information Science
Nagoya University
guoxi@db.itc.nagoya-u.ac.jp

Tingting Dong
Graduate School of Information Science
Nagoya University
dongtt@db.itc.nagoya-u.ac.jp

Yoshiharu Ishikawa
Information Technology Center / Graduate School of Information Science
Nagoya University
y-ishikawa@nagoya-u.jp

ABSTRACT

To provide a high-quality mobile service in a safe way, many techniques for *location anonymity* have been proposed in recent years. Advanced location-based services such as mobile advertisement services may use not only users' locations but also users' attributes. However, the existing location anonymization methods do not consider attribute information and may result in low-quality privacy protection. In this paper, we propose the notion of *visibility*, which describes the degree that an adversary can infer the identity of the user by an observation. Then we present an anonymization method which considers not only location information but also users' attributes. We show several strategies for the anonymization process and evaluate them based on the experiments.

1. INTRODUCTION

1.1 Background

In recent years, *location anonymization* has become one of the important topics in location-based services and mobile computing [6]. The issue concerned is that a user should send her location information to receive a high-quality service in general. However, if the service provider is an adversary, the detailed location information may be used for non-intended purposes. In an extreme case, the user's identity may be estimated by combining the location information with additional information sources. The use of location anonymization would solve the problem in some sense, but it may result in the degradation of service quality; an appropriate anonymization method is required.

1.2 Location-based services that use attribute information

For a typical location-based service which only utilizes location information, the conventional notion of location anonymity is effective for privacy protection. However, advanced location-based services may use additional *attribute information* such as user's age, sex, and occupation. For illustrating our motivation, let us consider an example of a *mobile advertisement service*.

In this service, we assume that a mobile user issues a request for an advertisement and it is delivered to an appropriate advertiser. Then the advertiser sends corresponding advertisements to the user. In this sense, the advertisement service is a pull-based service. The matching service (called the *matchmaker*) plays the role of a mediator between users and advertisers, and uses users' attribute information for selecting appropriate advertisers. Since the success of an advertisement is charged by the matchmaker, advertisers would like to perform effective advertisements with low investments. If an advertiser can specify the type of the target users (e.g., young women), then the effectiveness of the advisement would increase.

Figure 1 illustrates the overview of a mobile advertisement service assumed in this paper. The *matchmaker* between mobile users and advertisers is a trusted third party and manages each user's information as her *profile.* As described later, the matchmaker is responsible for anonymization. When a mobile user issues a request for a service (i.e., an advertisement), the matchmaker anonymizes the location and profile of the user and sends them to the advertisers. Then appropriate advertisers send corresponding advertisements to the user via the matchmaker. By the obtained

[*] Current Affiliation: NTT DOCOMO Inc.

advertisement, the user can receive benefits such as coupons and discounts. In this paper, we focus on the anonymization part in this scenario.

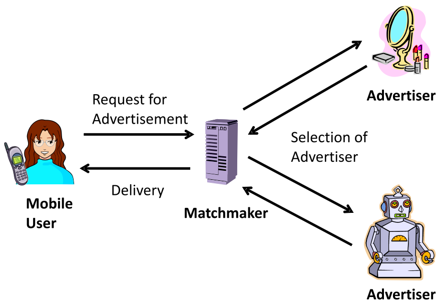
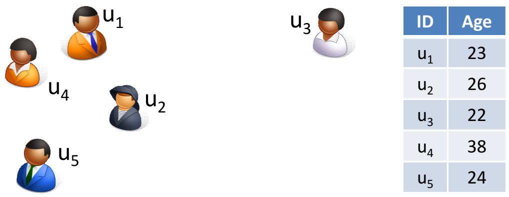


Figure 1: Location-based anonymization framework

1.3 Privacy issues

In the system architecture, we should note that an advertiser is not necessarily reliable and it may be an *adversary*. If the exact location is notified to an adversary, there is a risk that the advertiser identifies the user by watching the location. For this problem, we may be able to apply a conventional location-based anonymization method, but the following problem happens if we consider users' attributes.

Assume that users in Fig. 2 issue requests of advertisements with the order $u_1, u_2, \ldots, u_5$. Their profile information is also shown in the figure. The matchmaker needs to consider tradeoffs between requirements of users, who want to preserve privacy, and advertisers, who want to know the details of user information to improve the service quality. One idea is to apply the k-*anonymization* technique; it groups k users based on proximity. For example, given $k = 3$, we can perform anonymization as $\{u_1, u_2, u_4\}$ as an example. If the matchmaker provides the users' profiles, the received advertiser would know three persons with ages 23, 26, and 38 are requesting advertisements. The problem is that the advertiser easily identifies user with age 38 by watching the target area.



ID	Age
u_1	23
u_2	26
u_3	22
u_4	38
u_5	24

Figure 2: Users and their profiles

If the matchmaker considers not only user proximity but also user profiles, we have another option. Consider the grouping $\{u_1, u_2, u_5\}$. In this case, it is not easy to determine who corresponds to each profile entry. Therefore, this anonymization is better than the former one.

1.4 Research objectives

In this paper, we propose a location-based anonymization method that also considers users' attributes. For this purpose, an important point is whether we can guess each user attribute with an observation. To represent this idea, we incorporate a new criterion called *observability*. In addition, since different users may have different privacy policies, we provide an anonymization method which considers users' preferences.

The preliminary version of the paper was appeared in [7]. In this paper, we revised the problem setting and the method proposed is a totally novel one.

2. RELATED WORK

2.1 Anonymization for location-based services

There have been many proposals on privacy preservation in location-based services. A popular approach in this field is *spatial cloaking*, in which an anonymizer constructs a *cloaked region* which contains target users. For example, [4] uses the notion of k-*anonimity* [9], which is often used in database publishing. The notion of k-anonymity is used in many proposals and there are variations such as the use of graph structure [3] and cell decompositions [1, 8]. In this paper, we extend the idea for our context.

Most of the anonymization methods for location-based services do not consider users' properties. One exception is [10], in which an attribute vector is constructed for each user based on her attribute values. In the anonymization process, a user group is constructed based on the proximity of vectors. The problem of the approach is that it does not consider difference of attributes in terms of observability so that attribute values tend to be over-generalized and results in low-quality services.

2.2 Classification of attributes

In traditional privacy-preservation methods based on k-anonymity [9], user attributes are classified into the following three categories:

- *Sensitive attribute*: It represents privacy information such as disease names.
- *Identifier*: It is used for uniquely identifying individuals such as names and addresses.
- *Quasi-identifier*: Like age and sex attributes, it does not identify individuals directly, but their combinations with other attributes may reveal the identity.

In contrast to the assumption of traditional data publishing, an adversary in our context is not easy to identify individuals using quasi-identifiers and external information (e.g., telephone directory) because it is difficult to determine the candidate users who appear in the target location for the given time. In contrast, visual observation is more problematic in our context. If an adversary watches the target area, he may be able to identify the person who requested the service.

For this problem, we need to enhance the traditional treatment of attributes. In the context of privacy protection in social networks, [5] considered two properties of attributes:

- *Sensitivity*: It describes how the attribute is related to privacy violation. For example, "address" is more sensitive than "birthplace" because the latter is not so useful for identifying people. [5] assumes that sensitivity of each attribute does not depend on a specific user and takes a constant value in the system.

- *Visibility*: It is used as a criterion of how much a user can disclose a detailed value for the attribute. Visibility preference depends on each user and each attribute. For example, different users may have different disclosure policies for “Birthdate”.

The notion of visibility cannot be applied to our context. In a location-based service, an adversary can observe some of the user properties even if the user does not want that—it means that visibility is not controllable. In contrast, *observability* of an attribute, which means how much we can estimate the actual value of the attribute from the observation, is more important. We describe the notion in detail later.

2.3 Personalized anonymization

For our context, a personalized privacy-protection mechanism is required because the exposure of user profiles depends on each user’s preference. However, most of the existing data anonymization techniques do not consider personalization. [11] proposed a personalized privacy preservation method for a static database. In this method, a hierarchical *taxonomy* is constructed for each attribute. Every user can specify the level of detail in the hierarchy for each attribute and then she can represent her preference. In this paper, we extend the idea considering our context.

3. OVERVIEW OF THE APPROACH

3.1 Objectives of anonymization

We employ the following policies to take trade-off between privacy preservation and service quality.

- *Identification probability*: The probability represents how a user is related with a profile. A user prefers a low identification probability, but an advertiser would expect to high identification probability for the good service. Thus, we assume that each user can specify the *threshold* of the identification probability in her profile. In our approach, the identification probability of an anonymization result should be as large as possible with the constraint that the probability should be smaller than the threshold.

- *Attribute generalization*: Attribute generalization is a fundamental method for protecting privacy. However, excessive generalization results in low service quality, and preference on attribute generalization depends on each user. Therefore, we consider that each user can specify a preferred *disclosure level* for each attribute; the anonymization algorithm should not violate this restriction and tries to group users with similar attribute values.

- *Area size*: A cloaked region with a large size results in a poor service quality. We assume that the system sets the maximum area size for a cloaked region.

3.2 Taxonomy for attribute domain

The taxonomy for an attribute domain is used in the process of generalization. We assume that there exists a hierarchical taxonomy for each attribute domain. Figure 3 shows an example for “age” domain. The root node *any* at level 0 represents all the domain values and the leaf nodes correspond to the most detailed information. Note that Fig. 3 only shows only the descendants of node [20-39] for simplicity. We assume that taxonomies are available for other domains (e.g., ZIP code).

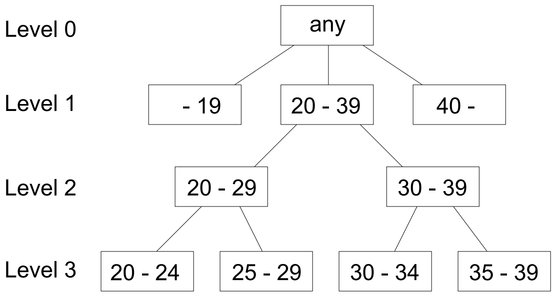


Figure 3: Taxonomy for “Age” domain

We also assume that each user can specify a *disclosure level* for each attribute. For example, consider a user with age 23. The user can specify node [20-29] as her disclosure level for the age domain. If the selected node is near the leaf level, the user can receive more personalized advertisements, but the privacy may not be well protected.

3.3 Profile

Each mobile user constructs a *profile* to represent her preferences on service quality and privacy levels. The trusted matchmaker maintains profiles. An example of user profiles is shown in Fig. 4.

ID	Age		Sex		Threshold Prob.
u_1	23	[20-29]	M	[Any]	0.4
u_2	26	[20-39]	M	[M]	0.5
u_3	22	[20-24]	F	[F]	0.6
u_4	38	[30-39]	F	[Any]	0.5
u_5	24	[20-24]	M	[M]	0.5

Figure 4: Example of profiles

The contents of profiles are as follows:

- *Attribute value*: It represents the attribute value of the user (e.g., Age = 23 for user u_1)

- *Attribute disclosure level*: The level is given by specifying a taxonomy node (e.g., [20-29] for user u_1’s Age attribute)

- *Threshold for identification probability*: The user requests that her identification probability should be smaller than this value.

3.4 Attribute observability

Now we introduce a new criterion called *observability.*

Definition 1 (Observability). *Attribute* observability *is a measure of how we can guess its actual value by visually observing the user.* ■

For example, “Sex” is easy to guess, but “Birthplace” is difficult to estimate by an observation. In this case, the observability of “Sex” is higher than “Birthplace”. In this paper, we assume that the observability of an attribute domain (e.g., age) is represented by a probability and takes a system-wide constant value.

We take the following approach for other two properties on attribute privacy.

- A user can specify the disclosure level of each attribute to reflect her preference on *sensitivity*. For example, if a user considers that her age is very sensitive, she can specify “any” node in Fig. 3. Note that a user cannot fully control her sensitivity because an adversary may watch the user directly.
- A user can control *visibility* by specifying the disclosure level of each attribute. If we select the leaf-level node, the visibility is the highest, but it depends on the attribute domain whether the attribute is actually observable.

3.5 Matching degree

To use the notion of observability in an anonymization algorithm, we need to introduce a method to measure the observability of an attribute. We take the following approach: we measure the degree considering taxonomy nodes. For example, consider attribute “Age”. The attribute value $age = 21$ is highly related with node [20-24], but has little relationship with node [30-34]. We call the degree that user u_i and taxonomy node n_k match their *matching degree* and define it as follows:

$$match(u_i \rightarrow n_k) = \Pr(n_k \mid u_i). \tag{1}$$

When there are K nodes in a level of the taxonomy, the aggregated matching degree is defined as follows:

$$\sum_{k=1}^{K} match(u_i \rightarrow n_k) = \sum_{k=1}^{K} \Pr(n_k \mid u_i). \tag{2}$$

In this paper, we assume that the matchmaker holds the predefined matching degrees between all the combination of attribute values and taxonomy nodes. Figure 5 shows an example. Due to the limited space, we omit the level 0 node [any] and only show some representative nodes.

ID	$l = 1$	$l = 2$		$l = 3$			
	[20-39]	[20-29]	[30-39]	[20-24]	[25-29]	[30-34]	[35-39]
u_1	0.88	0.88	0.00	0.54	0.34	0.00	0.00
u_2	1.00	0.90	0.10	0.38	0.52	0.10	0.00
u_3	0.79	0.79	0.00	0.56	0.23	0.00	0.00
u_4	0.64	0.00	0.64	0.00	0.00	0.11	0.53
u_5	0.97	0.95	0.02	0.51	0.44	0.02	0.00

Figure 5: Matching degrees

In this paper, we assume that each attribute in a profile is independent. Therefore, the total matching degree can be calculated by multiplying attribute-wise matching degrees.

3.6 Identification probability

An *identification probability* is a probability that a user is identified by watching the users in the target area with the anonymized profiles. If the identification probability is lower than the threshold probability specified by the user, we can say that the requirement of the user is satisfied. As described below, an identification probability is calculated using matching degrees.

3.6.1 Computing identification probability for two users

We first consider a simpler case when there are two users (u_1, u_2) and their anonymized profiles are given as Fig. 6. Note that an adversary does not know which user corresponds to which of the profile entries. Therefore, the adversary should consider two cases $(u_1 : p_1, u_2 : p_2)$ and $(u_1 : p_2, u_2 : p_1)$. Clearly, the following equation holds:

$$\Pr(u_1 : p_1, u_2 : p_2) + \Pr(u_1 : p_2, u_2 : p_1) = 1. \tag{3}$$

pid	Taxonomy Node
p_1	[20-24]
p_2	[25-29]

Figure 6: Anonymized profiles

For computing the probability, we consider the following idea. We play a dice for each user u_i. A dice has a face corresponding to each taxonomy node and its occurrence probability obeys the matching degree. In this example, we play two dices for u_1, u_2 at the same time and there are four patterns of the results: $(u_1 : p_1, u_2 : p_1)$, $(u_1 : p_1, u_2 : p_2)$, $(u_1 : p_2, u_2 : p_1)$, and $(u_1 : p_2, u_2 : p_2)$. The occurrence probability of $(u_1 : p_1, u_2 : p_2)$ is calculated as

$$\Pr(p_1|u_1) \times \Pr(p_2|u_2) = 0.54 \times 0.52 = 0.281, \tag{4}$$

and the probability of $(u_1 : p_2, u_2 : p_1)$ is given as

$$\Pr(p_2|u_1) \times \Pr(p_1|u_2) = 0.34 \times 0.38 = 0.129. \tag{5}$$

Since $(u_1 : p_1, u_2 : p_1)$ and $(u_1 : p_2, u_2 : p_2)$ are prohibited patterns (one profile entry does not correspond to multiple users), we omit when these patterns occur. Thus, the identification probabilities are given as

$$\begin{aligned} \Pr(u_1 : p_1, u_2 : p_2) &= \frac{0.281}{0.281 + 0.129} = 0.69 & (6) \\ \Pr(u_1 : p_2, u_2 : p_1) &= \frac{0.129}{0.281 + 0.129} = 0.31. & (7) \end{aligned}$$

3.6.2 Computing identification probability for general case

The basic idea is similar to the former case. For example, if the number of users is three, we should consider six combination patterns.

For the anonymization, we need to consider an identification probability of each user. Consider users u_1, u_2, u_3 and profiles p_1, p_2, p_3 are given. User u_1 is only interested in her identification probability is lower than the specified threshold and does not care the identification probabilities of u_2 and u_3. As an example, the probability that user u_1 and profile p_1 is related with is calculated as

$$\begin{aligned} \Pr(u_1 : p_1) &= \Pr(u_1 : p_1, u_2 : p_2, u_3 : p_3) \\ &\quad + \Pr(u_1 : p_1, u_2 : p_3, u_3 : p_2). \end{aligned} \tag{8}$$

In the following, we use the term *identification probability* in this sense.

4. ANONYMIZATION ALGORITHM

Table 1 shows the symbols used for describing the algorithm. The algorithm consists of two components: profile generalization and user group construction.

Table 1: Symbols and their definitions

Symbol	Definition
u_i	Mobile user
p_j	Profile
n_k	Taxonomy node
u_q	User who requested an advertisement
$u_q.t$	The time when u_q issued a request
$u_q.e_t$	Request duration time for u_q
$u_q.th$	Threshold probability of u_q
U_R	Set of users in a cloaked region
$\mathcal{U}_C$	Candidate set for U_R
H_U	Priority heap of users who requested advertisements
P_R	Profiles for users in U_R

4.1 Generalization of profiles

For lowering the identification probability for each user, we perform *generalization* of user profiles in a target cloaked region. A profile is, as described above, a set of taxonomy nodes. Since we assume that attributes are independent, the process results in generalization of each attribute in the corresponding taxonomy. Note that the minimum identification probability obtained by generalization is $1/N$ when N users are in the candidate cloaked region.

Algorithm 1 shows the generalization algorithm when N users exist in the cloaked region. $\mathrm{LUB}(n_1, n_2, ..., n_N)$ returns the least upper bound of taxonomy nodes $n_1, \ldots, n_N$ for the target attribute. In Fig. 3 for example, we get

$$\begin{aligned}\mathrm{LUB}([\text{20-25}], [\text{25-29}]) &= [\text{20-29}]\\ \mathrm{LUB}([\text{20-25}], [\text{30-39}], [\text{40-}]) &= [\mathit{any}]\\ \mathrm{LUB}([\text{20-29}], [\text{20-25}]) &= [\text{20-29}].\end{aligned}$$

GENERALIZE is a function which generalizes n_i to the specified level. Given the least upper bound node and the disclosure level specified by the user, it employs the highest one for the generalization.

Algorithm 1 Taxonomy Node Generalization

```
procedure GENERALIZENODE
    $\tilde{n} \leftarrow \mathrm{LUB}(n_1, n_2, ..., n_N)$
    for all $i$ such that $1 \leq i \leq N$ do
        $n'_i \leftarrow \mathrm{GENERALIZE}(n_i, \max(u_i.\mathit{discl\_level}, \tilde{n}.\mathit{level}))$
    end for
    return $\{n'_1, n'_2, ..., n'_N\}$
end procedure
```

4.2 User group construction

Algorithm 2 shows the outline of the anonymization process when a user requests a service. At line 2, we insert the user id into priority heap H_U. H_U is ordered by the expiration time, which is the sum of the service request time and the duration time. At line 5, we check whether the bounding box for the grouped users is larger than the maximum limit size. GENERALIZEPROFILE at line 6 performs generalization of profiles. It uses the aforementioned GENERALIZENODE function for node generalization. From line 7 to 12, we check whether the identification probability is lower than the threshold. If it is successful, we remove all S's (the sets that contain the finished users) from the candidate set $\mathcal{U}_C$. Function CHECKEXPIRATION from line 17 is for checking and managing the expireation of user requests.

Algorithm 2 Anonymization

```
 1: procedure ANONYMIZE($u_q$)
 2:     Add user id into $H_U$
 3:             ▷ heap entries are ordered by $\{u_q, u_q.t + u_q.e_t\}$
 4:     for all $U_R$ such that $U_R \in \mathcal{U}_C$ do
 5:         $U_R \leftarrow U_R \cup u_q$
 6:         if GETMBRSIZE($U_R$) $\leq$ MAX_RECT_SIZE then
 7:             $P_R \leftarrow$ GENERALIZEPROFILE($U_R$)
 8:             if $\forall u_i \in U_R, \forall p_j \in P_R, \Pr(u_i : p_j) \leq u_i.th$ then
 9:                 $\forall S \in U_R$, remove $S$ from $\mathcal{U}_C$
10:                 return $\{U_R, P_R\}$
11:             else
12:                 $\mathcal{U}_C \leftarrow \mathcal{U}_C \cup U_R$
13:             end if
14:         end if
15:     end for
16: end procedure

17: procedure CHECKEXPIRATION
18:     while true do
19:         $\{u, \mathit{deadline}\} \leftarrow$ POP($H_U$)
20:         if $\mathit{deadline} > \mathit{now}$ then
21:             Remove all the sets that contain $u$ from $\mathcal{U}_C$
22:         else
23:             break
24:         end if
25:     end while
26: end procedure
```

We illustrate how the algorithm works using Fig. 2. Assume that the requests are issued with the order u_1, u_2, u_3, u_4, u_5. The process of candidate maintenance in the matchmaker is shown in Fig. 7, where "Ev" represents "Event". We can see that the candidates of cloaked regions increase during the process until the output of the user group $\{u_1, u_2, u_5\}$, which corresponds to a cloaked region. Note that each candidate of cloaked region consists of users, their profiles, and their identification probabilities.

Ev	Candidate Groups
init	$g_0 = \emptyset$
u_1	$g_1 = g_0 \cup \{\{u_1[\text{20-24}] : 1.0\}\}$
u_2	$g_2 = g_1 \cup \{\{u_2[\text{25-29}] : 1.0\},$ $\{u_1[\text{20-29}] : 0.5, u_2[\text{20-29}] : 0.5\}\}$
u_3	$g_3 = g_2 \cup \{\{u_3[\text{20-24}] : 1.0\}\}$
u_4	$g_4 = g_3 \cup \{\{u_4[\text{30-34}] : 1.0\},$ $\{u_1[\text{20-29}] : 1.0, u_4[\text{30-39}] : 1.0\},$ $\{u_2[\text{20-39}] : 0.91, u_4[\text{30-39}] : 0.91\},$ $\{u_1[\text{20-29}] : 0.55, u_2[\text{20-39}] : 0.5, u_4[\text{30-39}] : 0.95\}\}$
u_5	$g_5 = g_4 \cup \{\{u_5[\text{20-24}] : 1.0\},$ $\{u_1[\text{20-24}] : 0.5, u_5[\text{20-24}] : 0.5\},$ $\{u_2[\text{20-29}] : 0.56, u_5[\text{20-24}] : 0.56\},$ $\{u_1[\text{20-29}] : 0.4, u_2[\text{20-29}] : 0.37, u_5[\text{20-24}] : 0.34\}\}$
out	$\{u_1, u_2, u_5\}$ is output. After the output, candidate groups are $g_6 = \{\emptyset, \{u_3[\text{20-24}] : 1.0\}, \{u_4[\text{30-34}] : 1.0\}\}$.

Figure 7: Management of candidates

At the initial state, the candidate set is empty: $\mathcal{U}_C = \emptyset$. As requests arrive, the number of candidates increases, and the algorithm performs profile generalization and identification probability calculation. For example, since the threshold probability of u_1 is 0.4 in Fig. 4, if the calculated identification probability for u_1 is less than 0.4, the anonymization is considered successful for u_1. Note that the maximum size of MBR is defined by the system parameter. Therefore, user u_3, which is far away from u_1 and u_2, is not grouped with them.

In the example of Fig. 2, we cannot get a satisfactory grouping until u_4 arrives. When u_5 requests a service, we can get an anonymization group $\{u_1, u_2, u_5\}$, which satisfies the constraints of identification probabilities. The matchmaker sends the constructed group to an appropriate advertiser and then removes the candidates which include u_1, u_2, and u_5 from $\mathcal{U}_C$. The remaining users u_3 and u_4 should wait the forthcoming user requests.

4.3 Processing strategies and evaluation criteria

The algorithm shown in Subsection 4.2 was the baseline (naive) algorithm. It outputs an anonymized group when a group of users that satisfies the constraints can be constructed. We can consider other option such that we wait the decision for a better grouping until the earliest deadline of users is reached. For selecting an appropriate strategy, it is important how to evaluate an anonymization result. We employ the following evaluation criteria:

- *Throughput*: It is the ratio how many users can be anonymized among all the requested users. A large throughput is preferrable.

- *Quality (Detailedness)*: From the perspective of an advertiser, detailed information is better. For evaluating the detailedness, we use the average level of taxonomy nodes after the anonymization process. For example, assume that we only have "Age" attribute and there are two generaliation results: $r_1 = \{[20\text{-}24], [20\text{-}24], [25\text{-}29]\}$ and $r_2 = \{[20\text{-}24], [20\text{-}29], [20\text{-}29]\}$. Since the levels of [20-24] and [25-29] are three and the level of [20-29] is two, the average levels of r_1 and r_2 are 3 and 2.33, respectively. We can deduce that r_1 is better than r_2 in quality.

5. EXPERIMENTAL EVALUATION

5.1 Setting of experiments

We evaluate the performance of different strategies using synthetic data and simulation-based data. The synthetic data is generated by multiple two-dimensional Gaussians with different centers and variances. The simulation-based data is obtained from the road network of Oldenburg city used in Brinkhoff's moving objects generator [2]. Although the generator generates moving histories of moving objects, we only use their first appearance places since we do not consider movement of users.

The basic settings of simulation parameters are shown in Table 2. In the default setting, we assume that requests are issued based on a Poisson arrival and a new user requests a service in every 1/100 second with the probability parameter $\lambda = 0.1$ (if two users issue requests at the same time, one of the users should wait other one's process). Once a user issues a request, she does not issue another request later. In the simulation, we assume that there is only "Age" attribute in the profiles. The range of age is from 20 to 39, and the matching degrees are set based on Fig. 5 (the lacked entries in the figure are filled). We extend the taxonomy shown in Fig. 3 and selects disclosure levels from 1 (node [20-39]) to 3 (leaf nodes).

Table 2: Basic parameters and their settings

Name	Value
Number of users	1000
Unit time of advertisement request	1/100 s
Advertisement request frequencies	10 times/ s
Used attribute	Age
Range of user age	$[20, 39]$
Disclosure level	$1, 2, 3$
Threshold probability	$0.3, 0.4, 0.5$
Expiration duration	$10\,\text{s} \pm 10\%$
Maximum area of a cloaked region	1000×1000

5.2 Strategies for anonymization

Based on the idea shown in Subsection 4.3, we consider the following seven strategies:

- *Naive*: This is the algorithm in Algorithm 2. We process each user based on the arrival order and then output a group immediately when we can construct it.

- The following two strategies share the same idea. We do not output a constructed group immediately and wait the appearance of a better group.
  - *Deadline-based*: This strategy maintains the candidate groups until the earliest deadline of the current users approaches. If a new user arrives, we try to add this user into the existing candidate groups. If the existing groups cannot merge the user, we try to construct new groups with the existing non-grouped users based on Algorithm 2.
  - *Lazy*: This is similar to *deadline-based*. When we add a new user, *deadline-based* checks the existing groups which satisfy the threshold probabilities first. In contrast, this strategy checks the groups which do not satisfy the threshold probabilities first. The lazy strategy can be said as a variation of *naive* which waits the deadline and cares users who are not in the current candidate groups.

- The following two strategies are also based on the same idea. They maintain all the candidate groups that satisfy threshold probabilities. When the earliest deadline of users approaches, they select one group from the existing candidates. The groups selected and output are different as follows:
  - *Many-first*: The group which has the largest number of users among the groups that contain the user.
  - *Next-deadline-based*: The group which contains the user with the next-earlier deadline. The intuition is that we care the user whose deadline approaches near future.

- *Avg-deadline-based*: The group with the earliest average deadline.
- *Threshold-based*: The group which contains the lowest threshold probability.

5.3 Experiment 1: Users' request frequencies

In this experiment, we change the frequencies of user requests and we check the number of users whose anonymization processes are successful. Increase of request frequency results in a large number of users in the target area, and we can estimate that many groups will be generated. We consider four cases of request frequencies: 5, 10, 50, and 100 times per second. This experiment is done using the synthetic data and we use the parameter settings shown in Table 2. The experimental result is shown in Fig. 8. Three methods *naive*, *deadline-based*, and *lazy* have good throughputs as the increase of request frequency. In contrast, *many-first*, *next-deadline-based*, *avg-deadline-based*, and *threshold-based* have bad performance especially for 50 / 100 times per second. The reason is that the four methods maintain all the candidate groups so that their number rapidly increases as the increase of users.

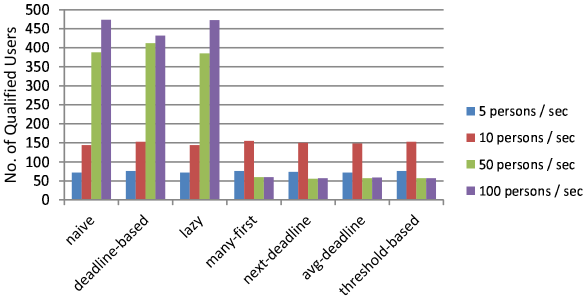


Figure 8: Request frequencies and number of qualified users

Figure 9 shows the number of users of two types: 1) whose process is delayed more than 0.1 seconds due to the foregoing users' processes do not finish, and 2) whose process is expired since the wait time reaches the deadline. We consider four strategies *naive*, *deadline-based*, *lazy*, and *many-first*. We can see that delays happen in *deadline-based* and especially in *many-first*. Note that *next-deadline-based*, *avg-deadline-based*, and *threshold-based* have almost the same result with *many-first*. Since *many-first*, *next-deadline-based*, *avg-deadline-based*, and *threshold-based* contain all the groups which satisfy the threshold probabilities, the increase of the number of candidates results in delays for the requests.

5.4 Experiment 2: Changing maximum area size

We perform experiments by changing the maximum area size of a cloaked region (MAX_RECT_SIZE in Algorithm 2) from 500 × 500 to 2000 × 2000.

Figure 10 shows the number of qualified users for the synthetic data and the uniform attribute distribution. When the maximum size is 2000 × 2000, delays happen only for *avg-deadline-based* and results in the low the number of qualified users. The number of qualified users are large for *many-first*, *deadline-based*, and *threshold-based*. Figure 11 shows how user attributes are generalized. In this figure, *naive* and *lazy* provide reults with good quality in which moderate generalization is performed.

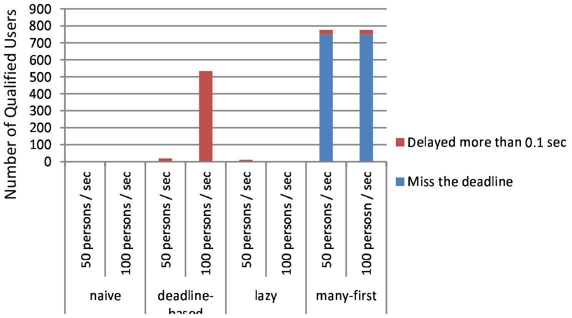


Figure 9: Request frequencies and delays

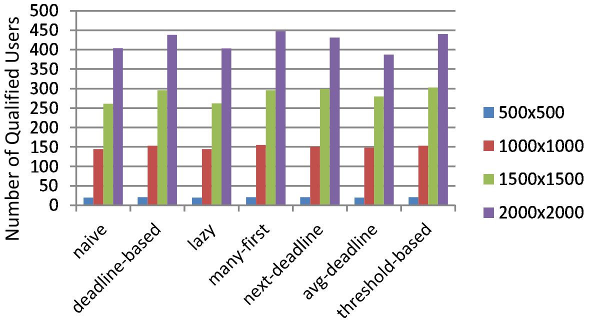


Figure 10: Maximum area sizes and number of qualified users

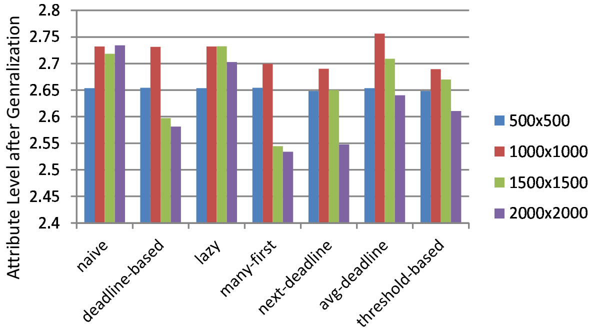


Figure 11: Averaged attribute generalization levels

Additionally, we performed similar experiments using the simulation-based dataset and the correlated distributions, but the trends were similar.

5.5 Experiment 3: Changing user conditions

In this experiment, we observe the behaviors when we change deadline and identification parameters in Table 2 using the synthetic data. First, we change the deadline to 10 ± 50%. Figure 12 shows the qualified users for each deadline setting. We anticipate that *next-deadline-based* and *avg-deadline-based* have good results, but the results are different—*deadline-based* and *many-first*, which do not care deadlines, perform well. Detailed analysis reveals that deadline-based strategies could output users with nearly expiring, but failed to output groups which contain many users.

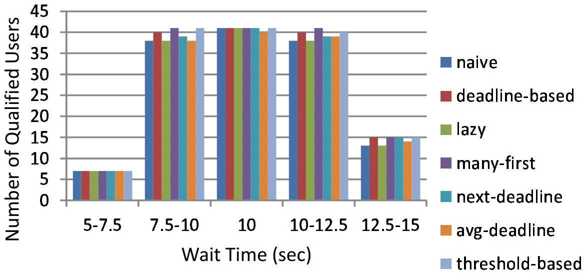


Figure 12: Number of qualified users for each deadline

Next, we change the deadline setting to the original one (10 ± 10%), but add 0.2 to threshold probabilities. Figure 13 shows the number of suceeded users for each threshold probability setting. In contrast to the case above, *threshold-based*, which tries to output low threshold ones, shows a good result for the threshold setting of 0.3. However, it is worse than *deadline-based* and *many-first*, which do not care thresholds and try to output groups with many users. All the strategies could not make a group for users with threshold settings lower than 0.2.

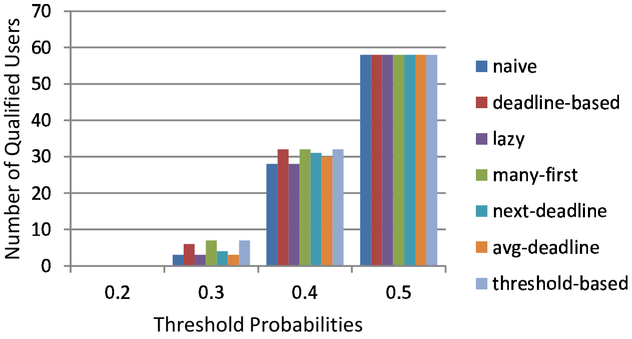


Figure 13: Number of qualified users for each threshold probability setting

5.6 Discussion

In terms of throughputs, *many-first* showed good performance. Compared to the strategies that considers deadline and threshold (*avg-deadline-based*, *next-deadline-based*, and *threshold-based*), the quality of the generated groups were better. However, these four strategies have a common problem when request frequency is high due to the increase of the number of candidate groups. For such a heavy-traffic case, the *naive* strategy might be a better choice since it can achieve high successful rate with low cost. It may be possible to change strategies considering the traffic.

In terms of the availability of cloaked regions, *lazy* was good. In this strategy, since generalization is not performed agressively, the quality of the results was generally good. This is a good property for advertisers. In addition, the strategy can support many users without serious delays.

6. CONCLUSIONS

In this paper, we have proposed a new anonymization method for location-based services. The feature is that we consider not only location information but also user attributes. For that purpose, we defined a new criteria called observability and introduced the notion of a matching degree. We proposed several variations of strategies and evaluated their performance based on the experiments.

Future work includes the development of robust and high-throuput method and a new algorithm which can anonymize users with low threshold settings.

7. ACKNOWLEDGMENTS

This research was partly supported by the Funding Program for World-Leading Innovative R&D on Science and Technology (First Program).

8. REFERENCES

[1] B. Bamba, L. Liu, P. Pesti, and T. Wang. Supporting anonymous location queries in mobile environments with PrivacyGrid. In *Proc. of WWW*, pages 237–246, 2008.

[2] T. Brinkhoff. A framework for generating network-based moving objects. *GeoInformatica*, 6:153–180, 2002.

[3] B. Gedik and L. Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing*, 7(1):1–18, 2008.

[4] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proc. MobiSys*, pages 31–42, 2003.

[5] K. Liu and E. Terzi. A framework for computing the privacy scores of users in online social networks. In *Proc. ICDM*, pages 288–297, 2009.

[6] L. Liu. Privacy and location anonymization in location-based services. *SIGSPATIAL Special*, 1(2):15–22, 2009.

[7] M. Mano and Y. Ishikawa. Anonymizing user location and profile information for privacy-aware mobile services. In *Proc. the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks (LBSN '10)*. pages 68–75, 2010.

[8] M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The New Casper: Query processing for location services without compromising privacy. In *Proc. VLDB*, pages 763–774, 2006.

[9] P. Samarati. Protecting respondents' identities in microdata release. *IEEE TKDE*, 13(6):1010–1027, 2001.

[10] H. Shin, V. Atluri, and J. Vaidya. A profile anonymization model for privacy in a personalized location based service environment. In *Proc. MDM*, pages 73–80, 2008.

[11] X. Xiao and Y. Tao. Personalized privacy preservation. In *Proc. ACM SIGMOD*, pages 229–240, 2006.

Applying Pervasive and Flexible Access Control to Distributed Multimedia Retrieval

Dana Al-Kukhun, Dana Codreanu, Ana-Maria Manzat, Florence Sedes
Université de Toulouse – IRIT – UMR 5505
118 Route de Narbonne, 31062 Toulouse, France
{kukhun, codreanu, manzat, sedes}@irit.fr

ABSTRACT

The distribution of data sources has formed a classical challenge for data management. The LINDO framework is an open system that manages the indexing, storage and retrieval of multimedia contents that are distributed in different remote servers and generated in a real time basis. The main objective of this framework is to provide efficient information retrieval with minimal processing costs. This was achieved through the proposal of an efficient decentralized content indexing mechanism. When considering the pervasive and mobile access to the managed content, the need of an access control becomes essential. In this paper, we apply an access control layer on top of the LINDO architecture that manages access based on the RBAC model and realizes decision making using the XACML standard. We explore the challenges that face the system in processing access requests showing how an access denial could influence the system's usability especially when returned to a user facing an important situation. Thus, we propose to apply flexible decision making that searches for alternative resources. This operation is performed using PSQRS, a query rewriting system that aims to provide users with pervasive accessibility where they could access any needed multimedia source at anytime, anywhere and anyhow.

1. INTRODUCTION

The necessity of handling a huge quantity of multimedia content created by multiple sources in a distributed environment emerges and raises new challenges concerning the indexing and access to the multimedia content, such as: distributed storage and decentralized processing, choice of the indexing algorithms, real time information retrieval and location-aware retrieval. On top of that we have to consider also that the users are more and more mobile and they need to access the system from anywhere. In such mobile and pervasive contexts, privacy and security management is a central issue.

In this paper, we present a new layer on top of the architecture proposed by the LINDO project[1] in order to tackle the above-mentioned challenges. The objective of the LINDO project was to build a distributed system for multimedia content management, and to ensure effective indexing and storage of data acquired in real time. The project didn't address the issues linked to data privacy and security.

Knowing that ensuring the protection of multimedia content is a key issue in certain application domains (e.g., video surveillance, medical domain, etc.), the access control management should be taken into consideration at the different levels of data processing and should take into account the user's mobility. Meanwhile, these security constraints should not affect the user's accessibility needs especially in important situations.

Our objective is to include the access control within the query processing and enrich it within the LINDO framework in order to attain a pervasive accessibility that enables the user to access multimedia sources at anytime, anywhere and anyhow. To achieve this goal, we have employed PSQRS – Pervasive Situation-aware Query Rewriting System – that offers adaptive context and situation-aware access solutions. The decision making within the system is based on the RBAC model [10] and employs the XACML standard [16]. These technologies are adapted to the distributed access management needs within the LINDO framework.

The solution overcomes the access denials taking place in real time mobile situations by modifying the query processing mechanism of the LINDO framework and by providing adaptive solutions that can bypass the access control constraints.

Next in section 2, we introduce a state of the art covering the different systems managing distributed multimedia content in 2.1, the basic standards for distributed access control management in 2.2 and some research about multimedia access control in 2.3. The LINDO approach for efficient multimedia distributed content management is described in Section 3 through its architecture, as well as its indexing and querying mechanisms. In section 4, we apply an access control layer on top of the LINDO architecture. In section 5, the adaptive access control solution is illustrated through a video surveillance use case. Finally, conclusions and future work directions are provided in section 6.

2. STATE OF THE ART:

2.1 Distributed Multimedia Systems

The constant growing dimension of the multimedia collections that are generated every day brings to the light problems of efficient indexing and retrieval. The solution to these issues passes through the generation and management of the metadata associated to the multimedia content.

These metadata are obtained through the application of indexing algorithms, which have different performances, purposes and constraints. Besides, a great heterogeneity of indexing algorithms has been defined in the state of the art (e.g., [4] for texts, [13] for images, [18] for audios, [20] for the videos). In a multimedia information system it is not desirable to execute all available

[1] *http://www.lindo-itea.eu*

indexing algorithms on all multimedia contents; because these will (i) overload the system and (ii) produce metadata that might never be used.

In the following, we present some distributed systems that manage multimedia contents by emphasizing the architectural choice and the adopted solution for multimedia indexing.

A distributed management of the multimedia is used by many projects due to the mobile acquisition context of these contents. An advantage of this kind of systems is that they benefit from the distributed storage and processing of the multimedia content and thus, the performances of the system can be improved.

The distributed systems that handle multimedia contents employ peer-to-peer or service-oriented architectures. The major problem that these systems encounter is the heterogeneity of indexing algorithms and of the generated metadata. The following projects addressed this problem in different manners.

The SAPIR (Search on Audio-visual content using Peer-to-peer Information Retrieval) project [2], [15] proposes a hybrid peer-to-peer architecture for the management of multimedia contents. It employs three specialized indexing servers, where each peer sends its ingested contents in order to be indexed. The resulted metadata is sent back to the peer that ingested the multimedia content in order to store it.

The DISCO (Distributed Indexing and Search by Content) project[2] has chosen a structured peer-to-peer architecture for the management of multimedia contents [5]. The indexing is accomplished at each peer, at the contents acquisition time. Each peer sends a summary of its index that is concatenated to a global index which is sent to all the other peers.

The CANDELA (Content Analysis and Network DELivery Architectures) project[3] is focused on the video content analysis and retrieval into a Service Oriented Architecture, where the content is stored and indexed on the distributed servers. The proposed solution was implemented for several use cases: personal mobile multimedia management [17], video surveillance [14], [12].

The WebLab project[4] proposes an integration infrastructure that enables the management of indexing algorithms as Web Services in order to be used in the development of multimedia processing applications [11]. These indexing services are handled manually through a graphical interface.

The VITALAS (Video & image Indexing and retrieval in the Large Scale) project[5] capitalizes the WebLab infrastructure in a distributed multimedia environment [22]. The architecture enables the integration of partner's indexing modules as web services. The multimedia content is indexed off-line, at acquisition time.

The MODEST (Multimedia Object Descriptors Extraction from Surveillance Tapes) project[6] proposes a multi-agent system for the video surveillance of motorways, in which they detect strange events, identify objects (persons, cars, trucks) and track the objects in the videos acquired by different cameras [1]. The video content is indexed by a segmentation agent on the same server where it is stored. The obtained segmentation is employed by other collaborative agents in order to detect anomalies, which are displayed to the user as summaries.

A comparative study of these systems shows that no matter what the architectural choice is, the content indexing is usually done on dedicated servers (the content and the associated resulting metadata are transferred over the network) using a pre-defined set of indexing algorithms. These algorithms are executed on all ingested multimedia. Thus, the resource consumption is not optimal. This important consumption problem was addressed by the LINDO project, which proposes a distributed architecture for the management of multimedia contents, which is favoring reduced resource consumption, in terms of data transfers over the network, storage and CPU utilization.

2.2 Distributed Access Control

Access control and privacy protection are key issues nowadays, especially in the context of distributed systems. In this section, we present two main standards that are widely employed for managing access control within distributed environments: the RBAC model and the XACML standard.

2.2.1 The RBAC Model

The principal motivation behind the proposal of the RBAC (Role Based Access Control) model [10] was to enable easy specification and enforcement for enterprise specific security policies in a way that maps naturally to an organization's structure. The RBAC model has simplified the administration and modification (updates) of access privileges especially in the case of assigning permissions for a large number of users accessing distributed resources.

The main concept of the RBAC model was to group users within roles that reflect their organizational positions then, simply distribute permissions to these roles instead of repeating the process for each individual.

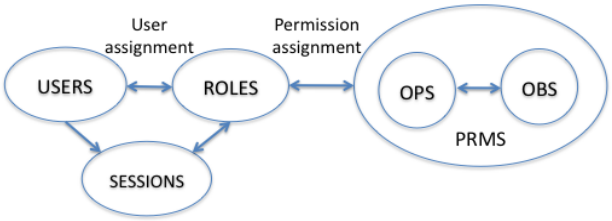


Figure 1: The RBAC Model

As illustrated in Figure 1, the role is placed at the heart of the RBAC model and is seen as an intermediary element that connects between the users and permissions as it attributes a set of privileges to those users based on their roles. These permissions (PRMS) allow the users to perform operations (OPS) on system sources expressed as objects (OBS).

2.2.2 XACML

The RBAC model managed to solve the challenge of administrating access permissions to distributed data sources by

[2] http://www.lamsade.dauphine.fr/disco/index

[3] http://www.hitech-projects.com/euprojects/candela

[4] http://weblab-project.org/

[5] http://vitalas.ercim.org

[6] http://www.tele.ucl.ac.be/PROJECTS/MODEST/index.html

providing centralized management for permissions through roles. With the evolution of service-oriented architectures and web services, new challenges has arisen and the problem of managing access becomes more complicated as the access control policies are also being distributed and more dynamic since they're managed by different administrating authorities. To resolve this problem, the XACML standard was introduced by [16].

XACML (extensible Access Control Markup Language) is an XML based policy language that describes access control policies to allow the attribution of user privileges on system sources. The standard provides a system for authentication and authorization taking into account various factors related to the user's context.

XACML provides an expressive security policy for data exchange within dynamic environments, which enables a flexible way to express and enforce access control policies.

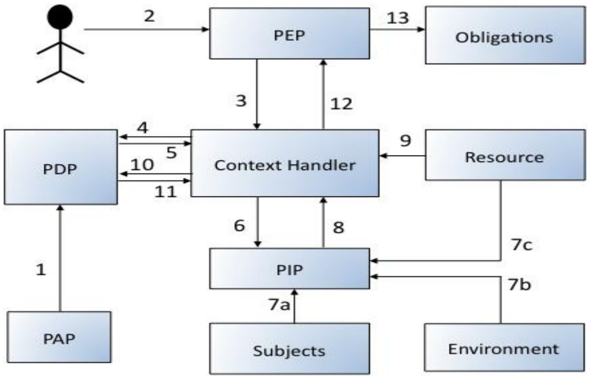


Figure 2: The XACML dataflow

As shown in Figure 2, as a client makes a resource request upon a server; a PEP (*Policy Enforcement Point)* interferes to ensure a secure and authorized access. In order to enforce a security policy, PEP will formalize attributes describing the requester (these attributes can be extracted from the user profile) to the PIP (*Policy Information Point)* and delegate the authorization decision to the PDP (*Policy Decision Point)*. Applicable policies are located in a policy store PAP (*Policy Administration Point)* and evaluated at the PDP, which then returns the authorization decision. Using this information, the PEP can deliver the appropriate response to the client and ensures that only authorized resources are accessed.

2.3 MULTIMEDIA ACCESS CONTROL

The projects mentioned in Section 2.1 were focused on the indexing and retrieval of multimedia contents, but none of them took into consideration problems related to the privacy and access control management of the contents and systems resources.

Meanwhile, many solutions have been proposed in order to secure the access to multimedia databases and systems. While some authors were interested in the security of the connection to the systems and on the distribution of the contents [19], others were focused on the content-based multimedia access control with fine-grained restrictions at a specific level of the multimedia data [9].

[8] proposes a framework that addresses multi-level multimedia access control by adopting RBAC, XML, and Object-Relational Databases. The authors associated roles to users, IP addresses, objects and time periods. All multimedia contents handled by their system have to be segmented. Only the objects which have roles associated to are extracted from the multimedia contents. The system stores several versions of the multimedia contents, the original one and one for each user-based restriction.

[21] Studied the confidentiality and privacy issues in the context of a video surveillance system. They also defined access rights to different hierarchical objects that can be extracted from the video contents. They focused on the detection of suspicious events.

3. THE LINDO APPROACH

3.1 System Architecture

The main goal of the LINDO project (Large scale distributed INDexation of multimedia Objects) is to define a distributed system for multimedia content management, while focusing on the efficient use of the resources in the indexing and query processes. Thus, not only the multimedia contents storage is distributed but also the indexing process. The originality of this solution is that: (a) the content is not moved to indexing servers, but indexing algorithms are deployed on the servers where the content is acquired; (b) the indexing process is accomplished in two steps: a generic indexing at ingest time (i.e., implicit indexing) and a more detailed one at query time (i.e., explicit indexing). The Figure 3 illustrates an example of the distributed architecture proposed within LINDO project. A more detailed presentation of the LINDO architecture can be found in [6].

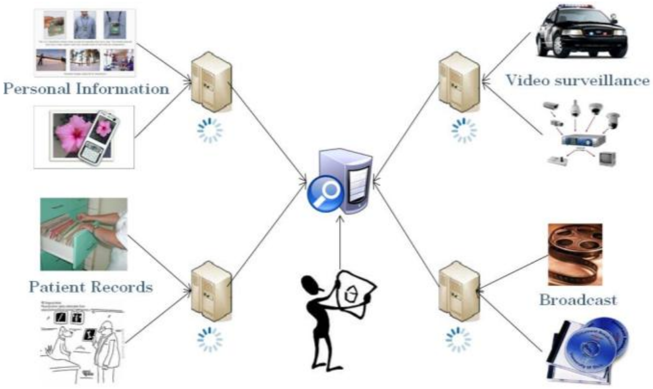


Figure 3: Example of LINDO architecture

Thus, the adopted distributed architecture enables to bypass problems that are specific to centralized systems like:

(1) The query processing slowness: executing the query on all metadata existing in the system might overload the central server, especially when processing complex queries and when several queries are executed simultaneously.

(2) The network bandwidth overload: in a classical approach all contents and associated metadata are transferred to central server or to dedicated servers.

(3) The system centralization: this could rise problems like fault resistance, if the central server is no longer available the metadata collection needs to be recomputed.

(4) The violation of access rights concerning the contents: some metadata shouldn't be stored on the central server for privacy reasons.

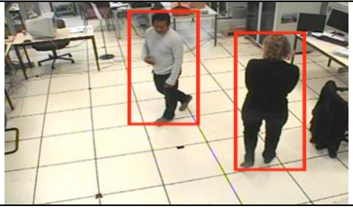
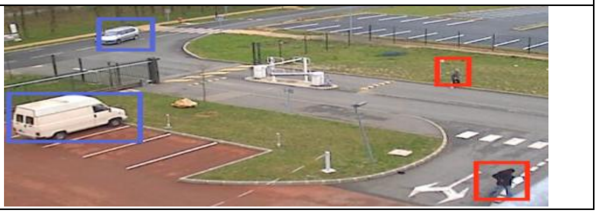
	Indoor	Outdoor
Intrusion	- Presence of people	- Presence of people & vehicles
Counting	- Number of people - Main color of the upper part of the people	- Number of people, number of vehicles - Main color of the people upper part. - Main color of vehicles
		

Figure 4: Examples of Metadata attained by applying Implicit Indexing Algorithms

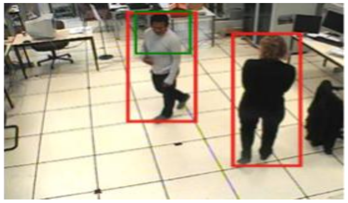
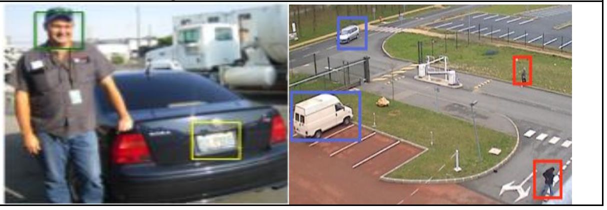
	Indoor	Outdoor
Intrusion	- Presence of people	- Presence of people & vehicles
Counting	- Number of people - Main color of the upper part of the people - Face recognition - voice recognition & speech-to-text	- Number of people, number of vehicles - Main color of the people upper part. - Main color of vehicles - Car plate number - Face recognition
		

Figure 5: Examples of Metadata attained by applying Explicit Indexing Algorithms

3.2 System Functionality

The functionality adopted within the previously presented system architecture goes as follows: the content is acquired and stored on the remote servers, and the collection of indexing algorithms is stored and managed on the central server. This collection is variable; at any moment we can integrate new algorithms with different functionalities, execution constraints and performances.

3.2.1 Indexing Mechanism

In order to reduce resource consumption, the architecture allows the indexing of multimedia contents to be accomplished at acquisition time (i.e., implicit indexing) with some generic algorithms (e.g., person detection, dominant color detection) and on demand (i.e., explicit indexing) with some algorithms that will analyze the contents more in detail (e.g., person recognition, register plate detection). This avoids executing all the indexing algorithms at once and producing metadata that might never be used but raises access rights issues concerning the explicit indexing. The Figure 4 and Figure 5 offer some indexing algorithms examples that illustrate the difference of the level of detail attained by the implicit and explicit indexing. These algorithms differentiate between two types of context acquisition (indoor and outdoor).

3.2.2 Query Processing Mechanism

The query processing (illustrated in Figure 6) begins with the query specification on the central server. First, the query is processed and executed on the metadata collection on the central server (which is a summary of the metadata collections from remote servers) in order to select the remote servers that could provide answers to the query and it is sent for execution to the selected servers. Among the servers that were not selected at the first step, there could be some servers that contain relevant information that has not been indexed with the right algorithms. For this reason, the LINDO solution detects such supplementary algorithms [7] and starts their execution (i.e., explicit indexing) on a sub-collection of multimedia contents. All the results obtained from the remote servers are sent to the central server, where they are combined and displayed to the user.

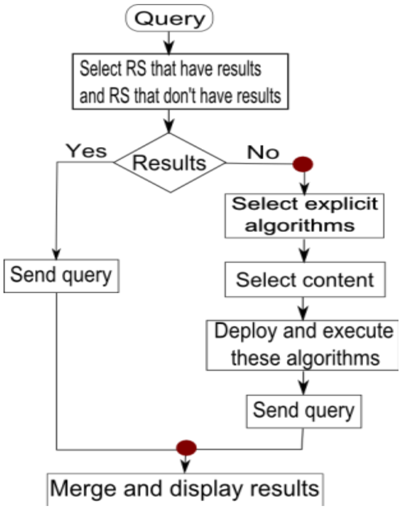

Query
Select RS that have results and RS that don't have results
Yes
Results
No
Select explicit algorithms
Send query
Select content
Deploy and execute these algorithms
Send query
Merge and display results

Figure 6: Query Processing Flow Chart

4. ADDING AN ACCESS CONTROL LAYER TO THE LINDO ARCHITECTURE

The sensitivity of the multimedia content and the privacy protection law that imposes anonymity constraints justify the need of applying an access control scheme on top of the LINDO architecture. The proposed layer customizes access based on the user's role (RBAC model) and is responsible for managing:

1. The access rights granted to users or services demanding access to the multimedia sources (e.g., video surveillance, medical domain, etc.) that vary not only according to their role but also in terms of their context (time, location, etc.).

2. The access rights for executing queries that employ the explicit indexing algorithms: the risk of disclosing personal or confidential information arises with the level of detail sought and provided by the indexing algorithm increases.

We highlight that in the context of adding this access control layer, the lack of responses returned to a user's query might not only be due to the lack of results existing within the system but also due to access restrictions imposed by the security layer.

4.1 A Pervasive Vision for LINDO

Our goal is to apply the access control layer and to balance between the security constraints and the user needs to find solutions that can ensure seamless accessibility to the requested resources at any time, from anywhere and anyhow.

The pervasive accessibility that we aim to provide matches with the pervasive characteristics of the LINDO system, which are:

- The distribution of multimedia sources.
- The variation of the entities managing these resources.
- The evolutive nature of these resources (generated and indexed in real time).
- The sensitivity and confidentiality of their content.
- The diversity of contextual information.
- The distribution of the indexing process performed by a variety of indexing algorithms.
- The execution of access requests in real time.
- The importance level of obtaining reactive solutions in important consultations or critical situations.

4.2 Confronting Accessibility Challenges with Adaptive Access Control

Managing access requests becomes more challenging within pervasive environments due to the dynamicity of contextual and situational information. Our objective is to ensure an efficient information retrieval process despite the security challenges. In order to achieve this objective, we employ PSQRS (Pervasive Situation-aware Query Rewriting System) - an adaptive decision-making system that confronts access denials taking place in real-time consulting situations by rewriting access requests in order to offer alternative-based access solutions.

The access control relaxation that we propose to carry out respects the access rights defined to protect the multimedia content and applies the adaptive decision-making at two functionalities:

1. The choice of using the explicit indexing algorithms (located on remote servers).
2. The presentation of the video contents (the identity of filmed persons in a video surveillance system is protected by privacy laws that assure their anonymity).

Next, we introduce the detailed functionality of the PSQRS architecture.

4.3 The PSQRS Architecture

As illustrated in Figure 7, the PSQRS (Pervasive Situation-aware Query Rewriting System) architecture contains several components and the sequence of its functionality starts from the user, who enters the system through an *authentication portal* (step 1) and launches an access request to a certain element (step 2). This request will be interpreted by our *Query Interpreter* that will translate the request into an XACML request and send it to the *Query Analyzer* (step3). The request (R) will be analyzed in consideration with the user's profile - automatically extracted at the sign in process - and according to his context (XACML flow chart, Figure 2). As the analysis finishes, the *Query Analyzer* would send the result directly to the user if it's a Permit (step 4a) or back to the *Query Interpreter*, if it's a deny (step 4b).

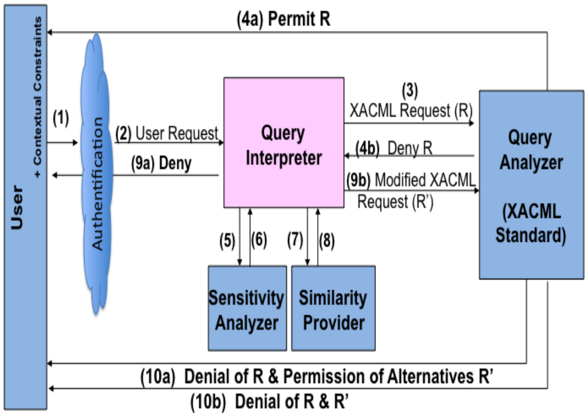


Figure 7: The PSQRS Architecture

In a deny situation the adaptive situation-aware query rewriting mechanism will take place and function as follows: the *Query Interpreter* will check the sensitivity of the consulting situation with the help of the *Sensitivity Analyzer* component (steps 5, 6) and according to the importance level of the situation, the *Query Interpreter* will search for similar or alternative resources through the *Similarity Provider* component (steps 7, 8) and employ them to rewrite the XACML request (R') and send it again to the *Query Analyzer* that will analyze the request and transfer the result back to the user (steps 10a,10b).

5. VIDEO SURVEILLANCE USECASE

In this section, we present an example where the implementation of our proposal is used to overcome the lack of answers provided by the system. As we will illustrate next, the system will modify the query processing and will adapt access decisions according to the level of importance of the querying situation.

Scenario: Taking the metro from « Trocadéro » station to « Place d'Italie » station at 14:15, Helen has forgotten her red bag on a bench at the waiting line. As soon as she realized, she went out to report the problem at the information counter.

A typical treatment of such situations goes through the customer service agent who would open a lost object file, take the descriptions and transmit them to the security officer on site. The security agent will follow different steps in order to find the object; he will check if the object has already been found or returned to the lost and found office by someone. Otherwise, he will try to see the video surveillance system to check if the object is still in the same location.

5.1 Typical LINDO Query Processing

Figure 8 shows the typical interpretation performed by the information retrieval system provided by LINDO. The launched request will be processed and parsed to extract the main keywords that are then reformulated in the form of an XML user query.

Query: Find all videos containing a *red bag*, forgotten in *Trocadéro, Paris* metro station, on the *2nd of February*, between *2:00pm and now (3:00pm)*.

```
<UserQuery>
        <QueryInText> find all videos containing a red bag, forgotten in
 Trocadéro, Paris metro station, on 2 February, between 2:00pm and 3:00pm.
         </QueryInText>
        <MediaLocation>metro station, Paris, Trocadéro </MediaLocation>
        <MediaFormat>Video</MediaFormat>
        <TimeSpan>
              <From>2012-02-02T14:00:00</From>
              <To> 2012-02-02T15:00:00</To>
         </TimeSpan>
</UserQuery>
```

Figure 8: Request represented in XML

The distributive nature of resource management and query processing in the LINDO system justifies the use of a filtering-based retrieval mechanism. The objective is to find the results that strictly meet the expressed needs in the application and minimize the subset of metadata that the system has to scan in real-time while processing the request.

After keyword extraction [6], the query processing proceeds by locating the servers responsible for managing the data streams captured by the cameras located in the Trocadéro station waiting line. Next, a filtering step is performed to restrict the search within the segments captured between 14:00 and 15:00.

The system will then, determine a list of indexing algorithms that would meet the needs, properties and context expressed within the query. This step will retrieve a subset of metadata describing the segments corresponding to the query.

In this scenario, the requested information are generic thus, the query processing will perform the search on the metadata generated by the implicit indexing algorithms and placed at the central server. The system will continue the search to find a red object in the retrieved list of metadata describing the chosen segments.

A filtering process is applied to take into account access control rules. Analyzing the access rights assigned to the security agent, we find that he is not authorized to access the videos containing passenger faces nor to use the personalized search options that employ the explicit indexing algorithms existing at remote servers. Therefore, considering these access restrictions, the system will perform another filtering step to eliminate the segments that contain people faces and finally return to the user the list of segments that contain a red object (if available).

5.2 Employing PSQRS for Adaptive and Alternative based Query Processing

The search results returned to the security agent in this case might be insufficient especially that the red bag might be present in the unauthorized segments containing passenger faces. Our proposal can take place at this level as a step towards ensuring a better quality of service by offering a wider subset of resources to the user while respecting the access rights defined on the consultation of the video surveillance data sources.

Through the usage of our proposed PS-RBAC model, the system would be able to offer more accessibility and adapt the permissions assigned to the security agent according to his contextual attributes and to the importance level of the situation of the consultation.

This adaptive solution can be employed when the system identifies access challenges related to the user's context or at an important situation. In this scenario, the « lost object » situation identification can be obtained from the file number.

The implementation of the adaptive solutions is performed by the PSQRS that adapts decision-making by rewriting the XACML queries. The solution proves its effectiveness due to its ability to achieve decision making to access video surveillance sources that are distributed and administrated by different authorities.

```
<Request xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:context:schema:os
http://docs.oasisopen.org/xacml/access_control-xacml-2.0-context-schema-os.xsd">
   <Subject>
      <Attribute AttributeId="urn:oasis:names:tc:xacml:2.0:subject:subject-id"
         DataType="http://www.w3.org/2001/XMLSchema#string">
         <AttributeValue>John Smith</AttributeValue> </Attribute>
      <Attribute AttributeId="urn:oasis:names:tc:xacml:2.0:subject:role"
         DataType="http://www.w3.org/2001/XMLSchema#anyURI">
         <AttributeValue>Security Agent</AttributeValue> </Attribute>
      <Attribute     AttributeId="urn:oasis:names:tc:xacml:
          2.0:example:attribute:securityAgent-id"
         DataType="http://www.w3.org/2001/XMLSchema#string" >
         <AttributeValue>sa2023</AttributeValue>   </Attribute>  </Subject>
   <Resource>
      <ResourceContent>
         <UserQuery>   <QueryInText> find all videos containing a red bag,
                 forgotten in Trocadéro, Paris metro station, on Thursday,
                 2 Febuary, between 2:00pm and 3:00pm).</QueryInText>
            <MediaLocation>metro station, Paris, Trocadéro </MediaLocation>
            <MediaFormat>Video</MediaFormat>
               <TimeSpan>
                  <From>2012-02-02T14:00:00</From>
                  <To> 2012-02-02T15:00:00</To>
               </TimeSpan>   </UserQuery>   </ResourceContent>  </Resource>
   <Action>
      <Attribute AttributeId="urn:oasis:names:tc:xacml:2.0:action:action-id"
         DataType="http://www.w3.org/2001/XMLSchema#string">
         <AttributeValue>Read</AttributeValue> </Attribute>   </Action>
   <Environment>
      <Attribute
        AttributeId="urn:oasis:names:tc:xacml:2.0:environment:environment-id"
        DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>Situation</AttributeValue> </Attribute>
      <Attribute
         AttributeId="urn:oasis:names:tc:xacml:2.0:environment:situation-id"
         DataType="http://www.w3.org/2001/XMLSchema#string">
         <AttributeValue>Forgotten Object</AttributeValue> </Attribute>
       <Attribute
        AttributeId="urn:oasis:names:tc:xacml:2.0:environment:sitLevel-id"
         DataType="http://www.w3.org/2001/XMLSchema#string">
         <AttributeValue>1</AttributeValue> </Attribute>   </Environment>
</Request>
```

Figure 9: XACML request embedding the user's query

As shown in Figure 8, the richness of the elements that we can embed within an XACML query enables it to describe the contextual attributes characterizing: (i) the requested source in the « resource » tag, (ii) the user launching the request in the « subject » tag and (iii) the situation at which the user has launched the access request in the « environment » tag.

The importance level of the situation will determine the level of adaptation to be realized. The activation of the adaptive search mode will be communicated from the XACML response in the form of an « obligation » that accompanies the resulting access decision, see Figure 10.

```
<Response>
   <Result>
      <Decision>Deny</Decision>
      <Status>
         <StatusCode Value="urn:oasis:names:tc:xacml:2.0:status:ok"/>
      </Status>
      <Obligation FulfillOn="Deny"
                     ObligationId="ApplyAdaptiveQueryingMode">
          <AttributeAssignment AttributeId="AQM"
            DataType="http://www.w3.org/2001/XMLSchema#string">
                 On
          </AttributeAssignment>
      </Obligation>
   </Result>
</Response>
```

Figure 10: XACML response containing the obligation

As the adaptive querying mode is triggered, the query processing mechanism will change to ensure the success of the search by providing a variety of adaptive solutions in correspondence with the situation's sensitivity level.

This adaptive search solution is realized by the PSQRS that detects the situation sensitivity through the *Situation Analyzer* component and turns to the *Similarity Provider* component to find similar resources that will guide the query rewriting process (see Figure 7).

In the case where the search didn't retrieve satisfactory results to the user and the consultation is taking place in a normal situation (Sit_Lvl = 0), the system will perform the adaptive query rewriting step through semantic similarity. The keywords of the user query will be reformulated using similar words or more generic concepts offered by the *Similarity Provider*. Similar works have been introduced in [3], the objective is to maximize accessibility chances without crossing the security boundaries.

The semantic reformulation options can be achieved with the help of a standard lexical dictionary such as WordNet. For example, the word "bag" can be replaced by various synonyms {backpack, luggage, purse, etc.}.

At the other hand, the adaptation process in the mentioned scenario will follow another scheme since the lost object situation is judged to be of higher importance (Sit_Lvl = 1). Hence, the *Similarity Provider* component will be replaced by an *Adaptive Solutions Provider*. This component will provide some predefined solutions that could bypass the access control challenge or would assist the user in adapting and reformulating his query by pointing out the access challenge and offering him adaptive solutions that would suit his context, the solutions are often saved in a predefined database. Table 1 shows examples of the solutions that the system can offer.

Table 1: The adaptive solutions that our adaptive query processing can employ

Problem	The adaptive solution
The privacy law imposing the protection of anonymity of audiovisual contents	
Passenger faces are not authorized	Display the content after the execution of an algorithm that applies a blur face function.
Voices are not-authorized	Use an algorithm for speech-to-text transcription
Volume of the video	
Lack of storage capacity on the user's machine	Use a compression algorithm in order to obtain a smaller file
Format not supported by the user's machine	Use a conversion algorithm into a compatible format.
Download problems due to a low bandwidth	Use a summarization algorithm in order to obtain a concise version of the content.

New solutions can also be inserted to the adaptive solutions database through a learning mechanism that detects the solutions that users employ when encountered with access challenges in real time.

The success of the adaptive solutions suggested by the users would eventually be more efficient if they knew the reason behind the access denial. The error messages that often accompany the returned access denial responses can serve as indicators to help the users in finding alternative solutions.

Therefore, the adaptive solution for this example will modify the treatment process and will: (i) neglect the filtering step responsible for imposing the access control constraints and (ii) replace it with an adaptive step-related to the presentation of resources with unauthorized content.

By applying this process to the scenario described above, the system will return the video segments taken from the Trocadéro station between 14:00 and 15:00 and containing a red object.

These results will be filtered in order to detect the unauthorized segments (containing passenger faces). This is where the system will apply the adaptation process that would filter the display to conform with the access restrictions imposed by the system.

The adaptation will be performed through a face detection step and the use of an algorithm that applies a "blur function" to protect the privacy of passengers appearing in these segments in order to return to the user a list of pertinent results that respect the access rules.

6. CONCLUSION

In this paper, we have presented an adaptive approach for access control management within multimedia distributed systems. Our solution overcomes the access denials that take place in real time access demands by modifying the query processing mechanism and by providing adaptive solutions to bypass the access control constraints. The proposed solution has been validated within the LINDO framework in the context of a video surveillance use case. We applied and validated the same access control approach for other use cases, such as Healthcare Systems [3].

The adaptive and alternative based situation-aware solution can increase the complexity of processing the request, but if we consider the usefulness of the results provided in real time and the fact they do not violate the access rights defined by the privacy law, this complexity seems quite acceptable.

In future works, we aim to extend our proposal by taking into account different contextual elements that might also influence the accessibility to multimedia content (e.g., hardware, network bandwidth, etc.) and to apply the adaptive process not only at the presentation level but also at the choice of the explicit indexing algorithms that are protected by RBAC constraints.

7. ACKNOWLEDGMENTS

This work has been supported by the EUREKA project LINDO (ITEA2-06011).

8. REFERENCES

[1] Abreu, B., Botelho, L., Cavallaro, A., Douxchamps, D., Ebrahimi, T., Figueiredo, P., Macq, B., Mory, B., Nunes, L., Orri, J., Trigueiros, M. J., and Violante, A. Video-Based Multi-Agent Traffic Surveillance System. In *Proc. of the IEEE Intelligent Vehicles Symposium.* 2000, 457-462

[2] Agosti, M., Buccio, E. D., Nunzio, G. M. D., Ferro, N., Melucci, M., Miotto, R., and Orio, N. Distributed information retrieval and automatic identification of music works in SAPIR. In *Proc. of the 15th Italian Symposium on Advanced Database Systems (SEBD'07)*, 2007, 479-482.

[3] Al Kukhun, D. and Sedes, F., Adaptive Solutions for Access Control within Pervasive Healthcare Systems. *In Proc. of International Conference On Smart homes and health Telematics (ICOST 2008)*, 2008, 42-53.

[4] Berry, M. W. and Castellanos, M., Survey of Text Mining II: Clustering, Classification, and Retrieval, Springer, 2008.

[5] Boisson, F., Crucianu, M., and Vodislav, D. Publication Framework for Content-Based Search in Heterogeneous Distributed Multimedia Databases. *Scientific Rapport CEDRIC No 1585*, 2008. 18 pages.

[6] Brut, M., Codreanu, D., Dumitrescu, S., Manzat, A.-M., Sedes, F. A distributed architecture for flexible multimedia management and retrieval. *In Proc. of Database and Expert Systems Applications (DEXA, 2011),2011*, 249-263

[7] Brut, M., Codreanu, D., Manzat, A.-M., and Sèdes, F. Adapting Indexation to the Content, Context and Queries Characteristics in Distributed Multimedia Systems. In *Proc. of International Conference on Signal-Image Technology & Internet-Based Systems (SITIS 2011)*, 2011, 118-125.

[8] Chen,S.-C., Shyu, M.-L., and Zhao, N. SMARXO: towards secured multimedia applications by adopting RBAC, XML and object-relational database. In *Proc. of the 12th annual ACM international conf. on Multimedia*, 2004, 432-435.

[9] El-Khoury, V. A Multi-level Access Control Scheme for Multimedia Database. *In 9th Workshop on Multimedia Metadata (WMM'09)*, 2009.

[10] Ferraiolo, D. F., and Richard Kuhn, D. Role-Based Access Controls. *In Proc. of the 15th National Computer Security Conference*, 1992, 554-563.

[11] Giroux, P., Brunessaux, S., Brunessaux, S., Doucy, J., Dupont, G., Grilheres, B., Mombrun, Y.,and Saval, A. Weblab : An integration infrastructure to ease the development of multimedia processing applications, In *the 21st Conference on Software and Systems Engineering and their Applications*, 2008

[12] Jaspers, E.G.T., Wijnhoven, R.G.J., Albers, A.H.R., Desurmont, X., Barais, M., Hamaide, J.,and Lienard B. Candela-Storage, Analysis and Retrieval of Video Content in Distributed Systems: Real-Time Video Surveillance and Retrieval. In *Proc. of the IEEE International Conference on Multimedia and Expo* , 2005, 1553-1556.

[13] Kosch, H. and Maier, P. Content based image retrieval systems – reviewing and benchmarking, *In Proc. of the 9th Workshop on Multimedia Metadata*, 2009, 1-21.

[14] Merkus, P., Desurmont, X., Jaspers, E., Wijnhoven, R., Caignart, O., Delaigle, J.-F.,and Favoreel, W. Candela - integrated storage, analysis and distribution of video content for intelligent information systems. *In European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology (EWIMT'04)*, 2004

[15] Michal, B., Fabrizio, F., Claudio, L., David, N., Raffaele, P., Fausto, R., Jan, S.,and Pavel, Z. Building a web-scale image similarity search system. In *Multimedia Tools and Applications.* 47, 3(May 2010), 599-629.

[16] OASIS, A brief Introduction to XACML, http://www.oasis-open.org/committees/download.php/ 2713/Brief_Introduction_to_XACML.html, 14 mars 2003

[17] Pietarila, P., Westermann, U., Jarvinen, S., Korva, J., Lahti, J., and Lothman, H. Candela-storage, analysis, and retrieval of video content in distributed systems: Personal mobile multimedia management. *In Proc. of the IEEE International Conference on Multimedia and Expo (ICME'05)*, 2005, 1557-1560.

[18] Pinquier,J., André-Obrecht, R. Audio Indexing: Primary Components Retrieval - Robust Classification in Audio Documents. *In Multimedia Tools and Applications*, 30,3 (September 2006), 313-330.

[19] Sánchez, M., López, G., Cánovas, O., Sánchez, J.-A., and Gómez-Skarmeta, A. F. An access control system for multimedia content distribution. In *Proc. of the Third European conference on Public Key Infrastructure: theory and Practice* (EuroPKI 2006), 2006, 169-183.

[20] Snoek, C. G., Worring, M. Multimodal video indexing: A review of the state of the art. In *Multimedia Tools and Applications*, 25, 1(January 2005), 5- 35.

[21] Thuraisingham, B., Lavee, G., Bertino, E., Fan, J., and Khan. L. Access control, confidentiality and privacy for video surveillance databases. In *Proc. of the eleventh ACM symposium on Access control models and technologies* (SACMAT '06), 2006, 1-10.

[22] Viaud, M.-L., Thièvre, J., Goëau, H., Saulnier, A., and Buisson, O. Interactive components for visual exploration of multimedia archives. In *Proc. of the International Conference on Image and Video Retrieval*, 2008, 609-616