Context Awareness in Network Selection for Dynamic Environments*

Daniel Díaz, Andrés Marín, Florina Almenárez,
Carlos García-Rubio, and Celeste Campo

Telematic Engineering Department, Carlos III University of Madrid
Avda. Universidad, 30, 28911 Leganés (Madrid), Spain
{dds, amarin, florina, cgr, celeste}@it.uc3m.es

Abstract. Mobile devices of new generation are able to connect to multiple networks and to constitute new infrastructureless networks. These dynamic environments require new security paradigms and automatic mechanisms to minimize user intervention. Our goal is the definition of a new concept of distance that considers the current domain constraints and the user preferences. This paper addresses some of the problems of these complex environments by using Multidimensional Scaling (MDS) techniques. We also propose collaborative mechanisms for automatic environment marking. Based on these ideas we have developed PervsIM, a decision mechanism that selects the most appropriate network or peer to interact with. Besides we have defined an embedded access control module which ensures that PervsIM decisions are followed by all applications. Furthermore, several simulation results and implementation details outline how these results can be incorporated in today's mobile devices.

Keywords: Context, network selection, trust, access control.

1 Introduction

Wireless network technologies are evolving providing more coverage, speed and quality of service. Moreover, the cost of the technology is decreasing so that it benefits the deployment. As a consequence, the number of mobile devices increases. Mobile devices are also enhancing their network support, being usually shipped by manufacturers with different network interfaces, like IrDA, bluetooth or WiFi. This enables them to connect to multiple networks and to constitute new infrastructureless networks.

In dynamic environments it is desirable that devices can be grouped defining domains. Grouping devices in domains makes it easier to determine, where we are, how closer the devices of a domain are and what we can do within a domain.

* This work has been partially supported by Everywhere (MCyT N2003-08995-C02-01) and by grant UC3M-TEC-05-056 of the Program to Support the Creation and Consolidation of Universidad Carlos III Research Groups.

P. Cuenca and L. Orozco-Barbosa (Eds.): PWC 2006, LNCS 4217, pp. 216–227, 2006.

In this paper we outline a mechanism to determine "where we are" by collecting context information and the unique IDs of access points and static devices. Currently, mobile devices require human input either from final users or providers to mark networks, access points or peers. The marking information helps the mobile device to select among the (growing) list of preferred networks. In this paper we propose an automatic mechanism of collaborative marking, which allows setting up marking information without user intervention for devices in a domain.

We also depict "what we can do" by defining policies. The mark given to domain devices, used together with the policies, parameterize the behavior. We also propose to embed the access control mechanisms in the operating system so even legacy applications can be controlled.

Moreover, when mobile devices switches on, or moves to other places it is necessary to select the appropriate network or peer to interact in order to satisfy user needs or, for example, if it is roaming, to reduce handoff delay (make-before-break). Deciding the peer, network, or entity to interact with, may be conditioned by multiple factors. Humans tend to consider multiple factors when deciding but, as a last resort, tend to simplify problems. Thus, why do not implement decision engines that simplifies such decisions?. This paper address the problem of selecting the most appropriate network or peer to interact with, defining a new concept of distance that considers the current domain constraints and the user preferences.

Section 2 introduces the problem domain, and the previous works are described in section 3. Section 4 outlines the prototype: domain definition and marking, policy manager and finally, the decision engine where it will be shown how multidimensional scaling, a psychometric algorithm, helps to alleviate decision problems. Section 5 gives implementation details, and finally section 6 present the conclusion and future work.

2 Motivation

Marc Weiser stated that "*the most profound technologies are those that dissapear*" [1], meaning that the user is not aware of them. Context awareness and processing is definitely needed to operate under the consciousness level of human. Moreover, intuitive ways of displaying context information, even mimicking human thinking are desirable capabilities for the technologies described by Weiser.

Satyanarayanan said that interactions in pervasive computing environments decay with the square of distance [2]. This statement applies to every interaction since the energy of signals decays in the same fashion. The goal of Satyanarayanan is to establishes a way to measure what can be called *interaction distance*. But, what about other metric or non metric attributes as trust, economic cost, type of service and any other defined by the user? Shall them be taken into account when selecting an appropriate peer to interact with? How can we assist the user in selecting the network with the least interaction distance, and do this *invisible* to the user?

There are other works that focus on network services, providing security and service continuation for wireless communications [3], and [4], but do not take into account the network selection problem. MDS data analysis techniques have been used for several problems with good results. [5] shows how MDS can be used to determine the distance among elements of sensor networks that takes $O(n^3)$ time to find a solution. A mechanism based in MDS is described in [6] to classify music, browse it and generate playlists.

Limited devices, specially personal devices, are very rich in context information. They can hold information on user location, and user personal information like the agenda, or the contacts list. This work presents a solution for assisting users to select the best network according to their preferences.

From the point of view of applications using the network, the selection process is part of the access control protecting the resource *network access*. In this paper we are focusing on network access as the resource to be protected: we want to ensure that applications use the most appropriate network available at each moment. We will introduce the context information available for personal devices into the access control.

For the selection process, we will take into account valuable context information including location, trust, and cost, process it according to the user preferences, and take the decision or alternatively present the context information to the user in a comprehensive way using MDS.

3 Previous Work

3.1 Pervasive Trust Manager

Pervasive Trust Manager (PTM) allows to manage ad-hoc relationships with other peers in a secure way (see [7]). This manager has been designed for personal devices that act as autonomous peers, belonging to different trust domains. These autonomous peers protect their own resources and communicate securely with each others.

PTM benefits from the common knowledge in the environment. Such knowledge is obtained from close peers, which recommend other known peers. This information is exchanged using a Pervasive Recommendation Protocol (PRP). Devices derive their own opinion about third peers from the recommendations. Such opinions are expressed using fuzzy logic and are calculated taking into account both recommendation data and the trust data about the recommenders. PTM keeps trust data about third peers, which are identified by their public key. It stores both trust and distrust information. After the formation of an initial opinion, PTM takes into account the behaviour of entities to vary the trust data and consequently the opinion.

3.2 Multidimensional Scaling

Multidimensional Scaling [8], MDS, is a set of techniques widely used in behavioral, psychologic and econometric sciences to analyze similarities of entities.

From a pairwise dissimilarities matrix, usually m-dimensional Euclidean distances [5], MDS can be used to represent the data relations faithfully providing a geometrical representation of these relations. MDS is used to reduce the dimensionality of a problem to a small value.

MDS can consider not only Euclidean distances but also any other evaluation of the dissimilarities of the entities. Dissimilarities can be classified according to whether the data is qualitative or quantitative. The dissimilarities from attributes of data can be weighted (weighted MDS), thus, assigning a different weight to each attribute allows to obtain more particular results depending on the problem. So, a complex m-dimensional problem can be simplified preserving the essential information using MDS.

There exists a multitude of variants of MDS with slightly different cost functions and optimization algorithms. The first MDS for metric data was developed in the 1930s and later generalized for analyzing nonmetric data [9].

In classical scaling the proximities are treated as distances, however, any (di)similarity can be derived from data attributes in order to obtain a metric, but it is necessary to hold the properties of non-degeneracy (diagonal elements should be zero, $d_{i,i} = 0$) and triangular inequality that states that $d_{i,j} + d_{i,k} \geq d_{j,k}$ for every i, j, k. The distance between two points i and j in a m-dimensional Euclidean space is defined as follows:

$$d_{i,j} = [\sum_{a=1}^{m} (x_{i,a} - x_{j,a})^2]^{\frac{1}{2}} \tag{1}$$

For Euclidean distances, distances $d_{i,j}$ are related to the observed proximities $p_{i,j}$ by an appropriate transformation $d_{i,j} = f(p_{i,j})$, depending on the measurement characteristics. A linear transformation, $d_{i,j} = a + bp_{i,j}$, can be assumed for unique distances with $b < 0$ for similarities and $b > 0$ for dissimilarities.

If the solution is derived using least-squares, a linear transformation of proximities $I(P)$ can be defined as $I(P) = D + E$, with D the distances matrix (that is a function of the coordinates) and E the residual error. The solution obtained is the X such the sum of squares of E is minimized. The double centered matrix of scalar products, B, can be defined as $B = XX^T$ where X is the coordinate matrix. The value of B is:

$$B = -\frac{1}{2}[I - \frac{1}{n}ii^T]D^2[I - \frac{1}{n}ii^T] \tag{2}$$

where n is the number entities, I an $n \times n$ identity matrix and i a unity vector of length n. Decomposing the matrix B into its singular values, $B = VAV^T$, the coordinate matrix X can be calculated as $X = VA^{\frac{1}{2}}$.

To reduce the complexity of a m-dimensional problem, we can choose $l < m$ eigenvalues and eigenvectors. Taking only the largest l eigenvalues and eigenvectors the problem is simplified to a l-dimensional problem.

However, in case of ordinal data, another procedure has to be followed than the use of singular value decomposition since we want to recover the order of the proximities and not the proximities or a linear transformation of the proximities.

A solution to this problem was given by Shepard [10] and refined by Kruskal [11]. These solution iteratively minimize a fit measure called *Stress* by an iterative algorithm, which is suitable for processing.

We have used an algorithm called ALSCAL [12], which uses alternate least-squares, combined with weighted (di)similarities, for simulation and implementation. ALSCAL finds a local minimum and can be used for both metric and nonmetric analysis. Furthermore, the ALSCAL algorithm can also deal with spare proximity matrixes so it is suitable for simplify problems in the absence of some data.

4 Pervasive Interaction Manager

The Pervasive Interaction Manager (PervsIM) is the solution we propose to address the aforementioned problems. PervsIM is composed by four modules: the domain definition module, the collaborative domain marking module, the policy manager and decision engine.

The prototype is described through this section. A brief description of some concepts may help the reader to understand better what is addressed in this section. **Devices** are grouped together in **domains**. The closest set of devices that surround us is considered the current **domain**. Devices within a domain are divided in static devices, called **anchors** and moveable devices called **peers**.

4.1 Domain Definition Module

This module is in charge of determining the current environment and grouping devices together in domains. The major constraint of interaction is the physical distance [2] since the energy of signals decays with the square of its value. So, the nearest set of devices define the current domain. The module uses the mentioned relative localization and neighbor information to define an domain.

Given a domain, the static wireless devices within that domain, for instance, network access points, printers and screens can be uniquely defined by their MAC address or other cryptographical identifier and considered as **anchors** or reference points. The anchors of a domain help the mobile device to recognize the domain as known.

For every element of the domain, the module finds out the attributes that will be used to compute the *interaction distance*. The attributes represent context information (quantitative, ordinal or category membership information) that depend on the user preferences (see section 4.4). The type and number of attributes are user-defined, but at least, two should be considered: physical distance and trust value. Besides, other attributes like service information from discovery protocols [13] (if applicable), required credentials, or economic cost, can be considered.

Physical distance is derived using received signal strength measures. The module takes values for the received signal strength from each network access point or anchor. Once out of bound anchors are deprecated, the signal strength is

scaled by a factor, that depends on the network interface technology, in order to provide a normalized value within 0 and 1.

Furthermore, localization techniques using signal strength provide good privacy and are inexpensive: radio hardware is used not only to establish communications, but also to determine the relative position. The accuracy of signal strength localization techniques is limited and decrease even more in indoor environments [14] [15], however, network interfaces are enough to uniquely determine the current domain by using unique identifiers and to determine if the mobile node is approaching or moving away from that domain.

The trust value for each element of the domain is handled by PTM (section 3.1), for ad-hoc elements, and by the collaborative domain marking module (section 4.2) for anchor elements.

Obviously, the domain borders are rouge but, combining all the attributes, a useful measure of *interaction distance* can be derived and used to take decisions (section 4.4). Finally, the aforementioned attributes are stored as XML elements. These elements contains, at least, the necessary information to identify that domain (anchors) and a time-to-live value.

4.2 Collaborative Domain Marking Module

The aim of this module is to automatically give marks to domain anchors, instead of asking the user for that information, other peers are asked for opinion. The anchors and attributes that define a domain can be different even for the nearest peers. So that, when two peers exchange information they only consider what they have in common. In general, several attributes can be exchanged among peers to compute a mark, but currently, information exchange is restricted to trust values but the model is opened.

The process is simple, trust values are exchanged securely among peers, and scaled by a factor that depends on the trust value assigned by PTM to the recommender peer. The peer i uses the received information from peer k to compute a value, $\beta_{i,j}$, which is the trust value that peer i has for an anchor j. The peer i quantify its trust to another peer k with a value among 0 and 1, $\alpha_{i,k}$, and it only accepts recommendations from peers with a trust value higher than α_{min}. The trust value $\beta_{i,j}$ increment for the n^{th} recommendation is calculated using the following expression:

$$\Delta\beta_{i,j} = \frac{\alpha_{min}}{n \log n}(\beta_{k,j} - \beta_{i,j})\alpha_{i,k} \quad \forall \quad (\alpha_{min} < \alpha_{i,k}) \tag{3}$$

$$\Delta\beta_{i,j} = 0 \quad \forall \quad (\alpha_{min} > \alpha_{i,k}) \tag{4}$$

The marking module uses a scale factor that permits an initial fast increment of the trust value for an anchor, but avoid collaborative attacks since its value decreases with the number of recommendations. This scale factor can be customized by the user. Fig. 1 shows the evolution of the trust value for an anchor using a scale factor of $\frac{\alpha_{min}}{n \log n}$.

As can be seen in Fig. 1, the results are conditioned to the value of α_{min}. This is a very conservative approach like used in reputation systems, which

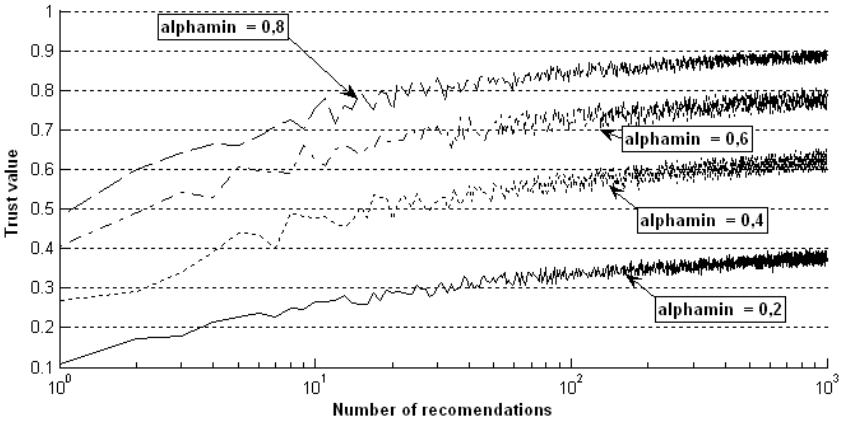

Fig. 1. Anchor trust value evolution from 0 vs number of recommendations. Recommended value 1.0.

tend to protect against malicious recommendations. The higher the value of α_{min} is, the higher trust value can be reached but the less recommendations are taken into account ($\alpha_{i,k}$ should be grater that α_{min}). Besides this model, others have been considered. A less conservative approach will be using $\frac{\alpha_{i,k}}{n \log n}$, so that recommendations coming from high trusted recommenders influence more our final trust value.

This mechanism allows to automatically derive a trust value for new environments that helps the mobile device to identify trusted or distrusted environments and behave in consequence as depicted in section 4.3.

4.3 Policy Manager Module

Limited devices host resources subject of protection, in this way, we use a policy manager to made access decisions based on policies. Access control policies allow defining a dynamic and semi-automatic mechanism of protection, in order to adapt our applications to the context and to minimize the user intervention.

A generic access control system has been previously defined in [16], so in this work, we include a specific application for controlling the access to the network interfaces. Such system is based on the XACML standard [17] to define the policies and the exchange of information.

XACML defines an architecture for access control in web systems comprising PCs and servers. It is a flexible approach which allows to specify different policies and rules which can be later evaluated by the Policy Decision Point (PDP) to permit or deny access to resources. Requests to resources should be trapped by the Policy Enforcement Point (PEP), to avoid malicious entities from bypassing

the access control. The collaboration among PEP and PDP ensures the access control is performed. Regarding the PEP there are two main approaches: either the PEP is included in the applications, or the applications access the PEP via an API to ensure correct access control. Nevertheless, non-cooperating applications or even malicious like virus, trojan horses, and the like, could circumvent the PEP and access the resources directly. One possible solution we propose here is to implement the PEP at the operating system (kernel) level, making unauthorized access more difficult to such kind of applications. Besides, it ensures that the applications shipped by the manufacturer also comply with the access control.

We benefit of the flexibility of XACML, extending the attributes to include trust data, and external context information. So, the decisions are made based on the trust assigned to other peers and available context information such as location, user preferences, or even cost.

4.4 Decision Engine Module

Multidimensional scaling techniques (section 3.2) are used in this module to find an ordered sequence of peers (including access points) to interact with, depending on the user preferences. The problem of deciding which is the best network or peer in complex environments is addressed by using techniques that allows the mobile device to *simplify problems as humans do*. Thus, a simple measure of what can be called *interaction distance* is derived for every peer using all the available information.

Consider an environment with many anchors and peers (elements). (Di) Similarities between pairs of elements can be derived as follows:

$$\delta_{i,j,\alpha} = \frac{|u_{i,\alpha} - u_{j,\alpha}|}{max(u_\alpha) - min(u_\alpha)} \quad for\ quantitaive\ data \tag{5}$$

$$\delta_{i,j,\alpha} = \frac{|rank(u_{i,\alpha}) - rank(u_{j,\alpha})|}{max(rank(u_\alpha)) - 1} \quad for\ ordinal\ data \tag{6}$$

$$\delta_{i,j,\alpha} = \begin{cases} 0 : u_{i,\alpha} = u_{j,\alpha} \\ 1 : otherwise \end{cases} \quad for\ category\ mebership\ data \tag{7}$$

where $u_{i,\alpha}$ is the α^{th} attribute value of the peer i. We consider data of different nature: quantitative data is used, to describe trust relations (section 3.1) and distances [5]; ordinal data for QoS classes, and to distinguish among different services; membership data help to classify elements, for example, ad-hoc peer or infrastructure network access point.

Once the (di)similarities are calculated they are weighted with the user preferences in order to obtain an unique weighted (di)similarities matrix. These weighted (di)similarities are defined for a set of n objets with q attributes as follows:

$$\delta_{i,j} = \left(\frac{\sum_{\alpha=1}^{q} w_{i,j,\alpha} w_\alpha \delta_{i,j,\alpha}^{\lambda}}{\sum_{\alpha=1}^{q} w_{i,j,\alpha} w_\alpha}\right)^{\frac{1}{\lambda}} \tag{8}$$

where $w_{i,j,\alpha}$ takes value 0 if objects i and j can not be compared on the α^{th} attribute and 1 otherwise, w_α is the weight given by the user to attribute α and $\delta_{i,j,\alpha}$ is the (di)similarity between objects i and j on the α^{th} attribute.

Although the model can include any other context relevant information, Table 1 shows a possible scenario for a user that measure the *interaction distance* in terms of trust (a value between 0 and 1), distance (derived from received signal strength) and economic cost. The first element represents the ideal element that will be used to measure the *interaction distance*: it has a trust value of 1, is very close to the device (distance 0) and interactions are free. Using the MDS ALSCAL algorithm, solving for one dimension and setting $\lambda = 2$ to handle attributes as distances, it is possible to derive a value for the *interaction distance* between the ideal element and the others, and also classify the elements. In this table we show the attribute values $u_{i,\alpha}$ for every element.

Table 1. Attribute values in a possible decision scenario

	Ideal(1)	2	3	4	5	6	7	8	9	10	11
Trust	1.0000	0.9429	0.8430	0.9573	0.8344	0.0206	0.0464	0.0075	0.0597	0.0191	0.0935
Distance	0	0.5259	0.5048	0.4633	0.5270	0.4757	0.5635	0.2540	0.2587	0.2509	0.2670
Cost	0	0.2054	0.2738	0.8636	0.8931	0.8461	0.8513	0.8424	0.8416	0.0	0.0

In the example we consider two situations: for the first one, the policy establishes the weights vector $Trust, Distance, Cost = 0.8, 0.1, 0.1$. The decision engine provides an ordered list of elements that meet this criteria and the distance to the ideal element 1. In Fig.2 there is a pair of representations of this decision for one and two dimensions. The axis of the figure do not represent any criteria, the figure just represent how closer elements are from each others. The result of this decision is $1, 4, 2, 5, 3, 11, 9, 7, 6, 10, 8$. Examining the results it can be seen that peers can be divided in two groups, the peers of the first group $(4, 2, 5, 3)$, since are close to the ideal element 1, are eligible. The others, are grouped together far from the ideal element, so are not eligible peers.

In the second situation, (Fig. 3) the policy establishes the weights vector $Trust, Distance, Cost = 0.1, 0.8, 0.1$. The result, $1, 10, 11, 8, 9, 6, 4, 3, 2, 5, 7$, shows that the distance between the ideal element 1 and the closest group 10,11,8,9 is very high so the mobile device may decide not to interact.

Weights vectors for the example have been exaggerated for a better understanding. In general, other more reasonable criteria can be easily considered.

The simulations we have performed show that the model suits the data. ALSCAL minimize a parameter called S-STRESS that is used to stop the iterations when its value is lesser than a minimum. The average of S-STRESS obtained in the simulations (varying the number of elements from 2 to 60) is 0.2728 and the results seem to be useful. Perhaps, stopping the iterations for this S-STRESS value is not suitable for other data analysis problems that need more accuracy, but it is enough for the network selection problem and less resources are consumed. Moreover, the quadratic correlation between the (di)similarities

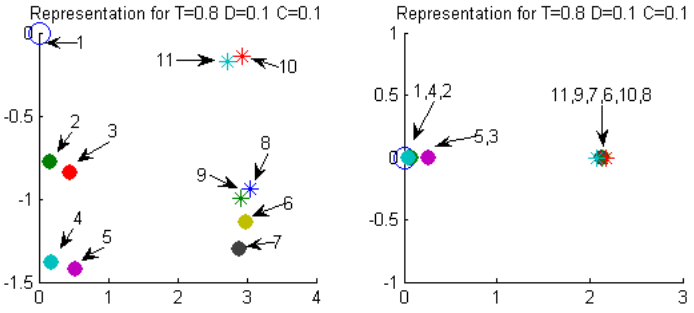

Fig. 2. Access point (anchor) selection favoring trust (Trust 0.8, Distance 0.1, Cost 0.1)

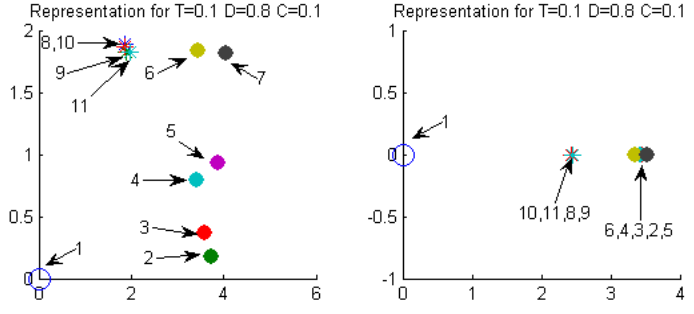

Fig. 3. Access point (anchor) selection favoring distance (Trust 0.1, Distance 0.8, Cost 0.1)

and the distances (RSQ), is a parameter that gives and idea of the goodness of the fit, 1 for a perfect fit and 0 for the worst fit. The model provided values for RSQ between 1 and 0.8. The complexity of the algorithm is $O(n^{2.65})$ where n is the number of elements.

5 Implementation Details

To validate our design, we have developed a prototype for Windows Mobile operating system. Windows Mobile, a Microsoft operating system, derives from Windows CE.

The implementation have been done in C++ under Windows Mobile. To gather information about anchors we have used the results of the Herecast project [18], a set of libraries that interact with the Network Device Interface System (NDIS), present in every Windows based operating system, that provides localization-based WiFi services.

We have implemented two Policy Enforcement Points (PEP) for handling legacy applications interactions. One of the PEPs controls the network traffic: the Network PEP (NPEP). The other controls the use that secure protocols, as SSL or TLS, make of the available credentials: the Secure PEP (SPEP).

When either an outgoing or an incoming connection takes place it is detected through the NPEP. The NPEP analyzes the destination, origin and protocol. Then, the NPEP provides that information to the Policy Decision Point (PDP).

The NPEP is a NDIS intermediate driver that is placed on the top of the NDIS miniport drivers but behind the NDISUIO driver. The PEP have bindings to all the network interface drivers below it so it can sniff the incoming and outgoing traffic and provide this information to the PDP. Thus, the PDP can allow or deny a particular interaction depending on the domain even for legacy applications.

The PDP not only decide when it is triggered by an application request, but also it can take decisions depending on the context changes. To select among the different network interfaces the PDP uses the NDISUIO driver [19] that is a connection-less, NDIS 5.1 compliant protocol driver. Using this intermediate driver, the PDP module can establish and tear-down bindings to network adapters.

Thus, depending on the domain, some network interactions can be allowed or not, i.e. if the mobile device is in a distrusted domain the policy module can either tear-down all the bindings, to deny connections, or set filters for some protocols for incoming and outgoing traffic. The PDP uses an XACML engine.

6 Conclusions and Future Work

The solution depicted in this paper provides mechanisms that allow a mobile device to take decisions based in the environment. The decisions are driven by policies that consider both user preferences and environment information. We have focused on attributes as trust and distance but we have shown also that many others can be considered.

We have demonstrated also how multidimensional scaling algorithms, that helps to think as humans, are useful to simplify decision problems with a complexity of $O(n^{2.65})$. Other algorithms that minimize different cost functions than ALSCAL will be tested to improve performance.

We are now facing the validation phase of the work. Our next step is to test the solution in different environments to measure the load and the resource consumption. We are planning also to move the solution to Symbian mobile phones.

References

1. Weiser, M.: The computer for the 21st century (1991)
2. Satyanarayanan, M.: Pervasive computing: Vision and challenges. IEEE Personal Communications **8** (2001) 10–17 citeseer.nj.nec.com/gennaro99robust.html.

3. Dutta, A., Zhang, T., Madhani, S., Taniuchi, K., Fujimoto, K., Katsube, Y., Ohba, Y., Schulzrinne, H.: Secure universal mobility for wireless internet. In: WMASH. (2004) 71–80
4. Chaouchi, H., Pujolle, G., Armuelles, I., Siebert, M., Carlos Bader, F., Ganchev, I., ODroma, M., Houssos, N.: Policy based networking in the integration effort of 4g networks and services. In: Proceedings of IEEE Semiannual Vehicular Technology Conference (VTC2004-Spring), Milan, Italy (2004) 5
5. Shang, Y., Ruml, W., Zhang, Y., Fromherz, M.P.J.: Localization from mere connectivity. In: MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing, New York, NY, USA, ACM Press (2003) 201–212
6. Platt, J.C.: Fast embedding of sparse music similarity. In: Advances in Neural Information Processing Systems vol. 16. (2004)
7. Almenárez, F., Marín, A., Campo, C., García, C.: PTM: A Pervasive Trust Management Model for Dynamic Open Environments. In: First Workshop on Pervasive Security, Privacy and Trust PSPT'04 in conjuntion with Mobiquitous 2004. (2004)
8. Borg, I., Groenen, P.: Modern multidimensional scaling, theory and applications. In: IEEE SECON 2004, New York, NY, USA, Springer-Verlag (1997)
9. Deun, K.V., Delbeke, L.: Multidimensional scaling (2000) http://www.mathpsyc.uni-bonn.de/index.htm.
10. Shepard, R.N.: The analysis of proximities: multidimensional scaling with unknown distance function part i. In: Psychometrika 27. (1962)
11. Kruskal, J.B.: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. In: Psychometrika 29. (1964)
12. Takane, Y., Young, F.W., de Leeuw, J.: Nonmetric individual differences multidimensional scaling: an alternating least squares method with optimal scaling features. In: Psychometrika 42. (1977)
13. Campo, C., García-Rubio, C., Marín, A., F.Almenárez: PDP: A lightweight discovery protocol for local-scope interactions in wireless ad hoc networks. Computer Networks Journal. Elsevier (2006) Pending to be published.
14. Elnahraway, E., Li, X., Martin, R.P.: The limits of localization using rss. In: SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems, New York, NY, USA, ACM Press (2004) 283–284
15. Elnahraway, E., Li, X., Martin, R.P.: The limits of localization using signal strength: a comparative study. In: IEEE SECON 2004. (2004) 406–414
16. Almenárez, F., Marín, A., Campo, C., García, C.: TrustAC: Trust-based access control for pervasive devices. In: 2nd International Conference Security in Pervasive Computing (SPC'05). (2005)
17. OASIS: eXtensible Access Control Markup Language (XACML) (2003) http://www.oasis-open.org/apps/org/workgroup/xacml/.
18. Paciga, M.: An open infrastructure for location-based services using wifi (2005) http://www.herecast.com.
19. Microsoft: Ndisuio: Ndis user mode i/o (2005) http://www.ndis.com/pcakb/KB01010301.htm.