

An Intermediate Framework for Unifying and Automating Mobile Communication Systems

G. Koumoutsos, K. Lampropoulos, N. Efthymiopoulos, A. Christakidis,
S. Denazis, K. Thramboulidis

University of Patras

{koumouts, klamprop, nefthymiop, schristakidis, sdena, thrambo}@ee.upatras.gr

Abstract. In the last few years we are witnessing a rapid growth in the mobile use of the internet and services like telephony. Mobility offers a great deal of advantages to end users but has also restrictions due to heterogeneity of networks, protocols and mobile devices with limited capabilities. In that we can add the need for manual interaction with service providers which costs in time and flexibility. We propose a framework that solves major mobility issues like connection maintenance, roaming, automated service selection regardless the underlying devices, protocols, providers and services. We describe important steps that will automate procedures and become the bases for an autonomic mobile environment transparent to the end user.

Keywords: Mobile Router , SLA ,negotiation, matchmaking, ontologies, network mobility, addressing, roaming.

1 Introduction

The rapid growth of the internet resulted in the proliferation of services and technologies comprising a complex network mosaic of various access methods, protocols, frequency bands, data speeds, and core transport networks. Different administrative domains and ownership add to this complexity through a variety of authentication, billing schemes, domain services and applications creating a highly heterogeneous environment, which is called to sustain a “homogeneous” end-to-end path as perceived and experienced by end users.

In contrast, users’ ways of working and interacting have also drastically changed as they have become more mobile and dependent on the internet. Their primary, albeit elusive, requirement is ubiquitous connectivity through a wide range of devices and maintaining the same user environment/experience.

Unfortunately, users today are not in a position to recognize services or applications as distinct commodities where they can be purchased or negotiated from interacting with the network, as the only recognizable service today is mere connectivity with a

specific provider or providers. Even this type of rudimentary service requires users' intervention and time in order to negotiate and sign a contract with the provider which in turn restricts the user to function within the limitations of this specific provider. Moving to an area of non coverage entails new negotiations and contracts (WLAN access) and/or higher fees (roaming).

The aforementioned issues have been our motivation behind our vision described in this paper according to which, users are capable of engaging themselves in a higher form of "conversation" with the network shopping for services without even having the knowledge of who the provider(s) is. Services and a "language" for requesting them are the main entities that the user recognizes and can access anytime, anywhere and most importantly by any available means or device. To this end, the device and the technology take a backseat – they are just the means for provision, and services become the true commodity on offer, thus, giving real meaning to globalization of networking and computing.

Driven by this vision we have identified a number of steps and issues that require immediate attention and solutions.

2 Towards Autonomic Communications

Our vision is to create a service-centric, user-friendly environment where a mobile-end-user will have the ability to move through different networks, providers and acquire services transparently, letting machines deal with the advanced complexity issues. Until today a mobile user has to manually:

- choose provider and specific service
- buy a connection service like any other product
- configure his devices for connection
- monitor service performance and act accordingly

So he is restrained by:

- predefined and non-adjustable specific provider and service terms selection
- providers' network coverage
- complex and time consuming procedures (configuration, services purchase) which require expert's knowledge

In our scenario the same user is moving across town using a device (PDA, Laptop, mobile phone, etc ...) to connect to the internet. Using our intermediate framework he has the ability to:

- automatically search for accessible networks, providers and services
- choose and acquire the appropriate service based on his demands
- monitor the quality of the service
- dynamically change network, provider, service or user end-device and maintain connection
- make use of certain advice capabilities to help with essential knowledge needed

3 Architecture

Our framework is based on a mobile prototype device called Chameleon Router (CR). This device will stand at the frontiers of end-users world, adopting to environments and automating procedures (Figure 1). It is a device with all basic hardware characteristics and a number of slots that can accept different kinds of network cards (NIC), interconnecting the mobile users devices (personal area network -PAN) with wide area networks (WAN).

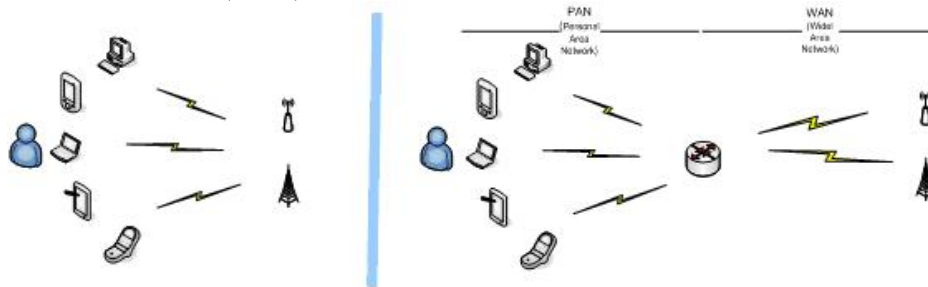


Fig. 1. Mobile connectivity without and with our intermediate framework.

On boot, CR performs a scan in multiple interfaces for all types of available networks in both sides: PAN and WAN. After making a list with the available choices, it begins the service selection procedure in order to select for his user the appropriate service. A negotiation scheme is launched depending on user's predefined preferences with all available service providers. Matchmaking takes as inputs user's requirements and available network-provider-service choices to select the best matching service. Then, an SLA (Service Level Agreement) is "signed" between two parties (user and provider) and the use of the service begins. From that point, a monitoring agent observes the consistency of the service based on the SLA, notifying user for any abnormal behavior.

3.1 Acquiring Services

In the direction of automated service discovery, negotiation, selection, SLA binding and monitoring multiple modern tools, protocols and technologies can aid. In order to automate these procedures, "machines" (mainly software intelligent agents) have to be given the means to interoperate, understand each other, translate the well described knowledge and make decisions in account of their owners. In our scenario we will examine closely:

1. Pre - configuration of our agent according to user preferences.
2. Service discovery on user's-application's request.
3. Negotiation and service selection.
4. Matchmaking and SLA forming.

4 **G. Koumoutsos, K. Lampropoulos, N. Efthymiopoulos, A. Christakidis, S. Denazis, K. Thramboulidis**

5. Monitoring and reaction in any case that can arise.

In order to achieve all the above our device will include significant processing and memory capacity, as well as wide area wireless access, so parts of application code, as well as infrastructure functions such as packet forwarding, routing, QoS support and service selection, will run on the device itself. It will have an interaction GUI that will use model-driven-architecture techniques user friendly in order to be preconfigured at the beginning and altered every time that is needed. Our approach doesn't use AI in service discovery, matchmaking and selection as in MIT's suggestion [1] where a system is educated to model users behaviour and make the most appropriate choices. Instead we will use mostly modern semantic tools to describe and make machines understand knowledge in order to discover it, standard negotiation schemes and protocols that will use flexible rules and ontologies that can easily be customised and adapted, parameterized matchmaking algorithms that will -with the pre-configuration of the user- make the best choice. We will use and extend all knowledge from recently very developed Semantic Web Services (SWS) world to achieve automation and transparency to the end user. SWS provide an approach for representing the functionality of Web services with the help of ontologies. Popular approaches for SWS include OWL-S[2], WSMO[3], FLOWS[4] and WSDL-S[5]. In service level we think of our framework as a special case of a web service as we have to contact a provider, understand the "service" he is offering, negotiate and choose based on our needs. We won't have to use all tools and protocols (e.g. SOAP) from web services but we can take advantage of semantic annotation with ontologies and rules, agent cooperation, interaction schemes and other. Our main goal is to adjust all this tools to the mobile agent as described with all limitations mobility and devices could put in our way. Moreover in that way a web services enabled agent will arise ready to absorb and take advantage of this fast evolving area. We are planning to bring as many capabilities as possible on client's side giving a powerful agent that will automate procedures to the end-user.

3.1.1 User Interaction

Our agent will be accessible through a simple user interface that will be adjustable to each user's device. It will be the tool to create a profile of the service the user will need in various cases. Through it he will have the ability to form rules that will create the desired service based on his knowledge. It will have multiple interaction schemes and description-aspects in order to let naïve and expert users configure their system just as well. In figure 2 we can see how the user will be able to create his desired service profile through ticks, clicks and wizards, at the left window, and how depending on his choice he will see the result in three levels of understanding. At the top level he will see his choices in a plain very simple text with explanations, choices effects on standard services and advises based on systems knowledge as we will explain later. The next two levels will include advanced features in order to let more experienced users configure their system more precisely. The user will have the ability to edit in both sides based on his experience, see his choices from multiple aspects and configure his system. All this aspects of the same description of a service

will derive from an ontology that will describe the service area and map between different level-views.

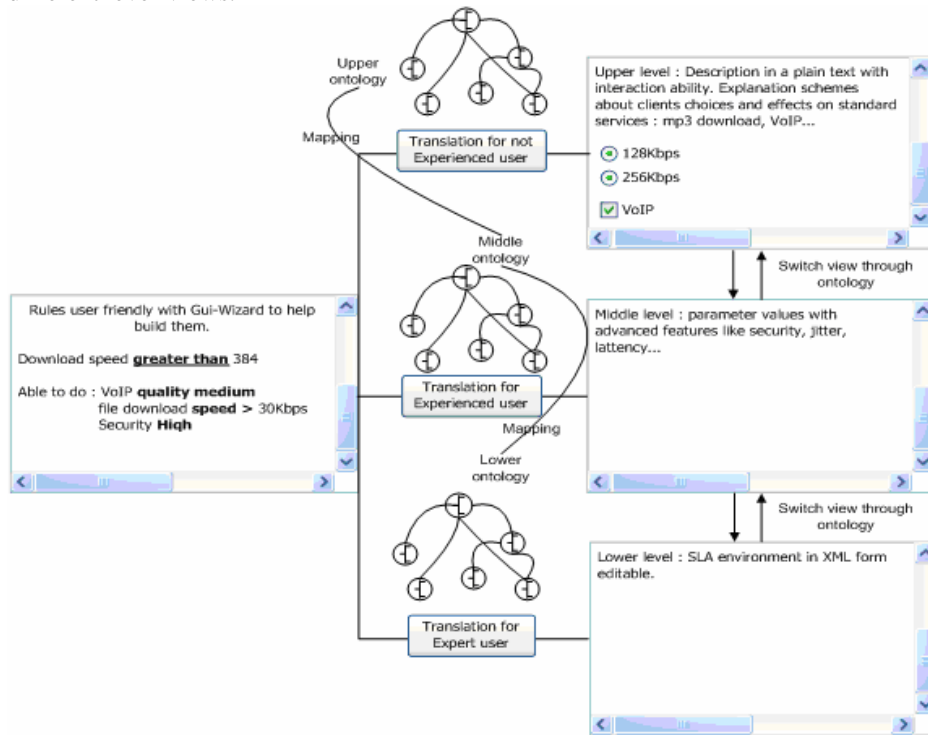


Fig. 2. Users profile creation ontology-based interface.

It will be sub-ontology of the SLA-ontology which will have a lot more in order to fully describe the relationship between the two parties. Except from this aspect-oriented service ontology, rules are going to be used for multiple reasons:

1. Rules to extract information from provisional SLAs and calculate effects on standard services along with explaining what each parameter is to naïve users. This will be part of the advice capabilities of our framework that will help users as they build their best-suit service profile as well as during the selection process if needed.
2. Matchmaking decision rules with the aid of ontologies that will enable a much more accurate approach than the syntactic one available on today's XML based transactions.
3. Editable preference rules that will give the user the ability to customize his agent adding information to the existing ontology based profile. He will be in position to extend existing rules and create new that fit him better. This will be done through a user-friendly model-driven based interface described above.

Writing rules precisely is difficult even for logic experts. The rule editor let users write facts and rules as if they were writing natural sentences. The user simply selects

a predicate for a fact or rule as a constraint on a parameter and sees the result on the right side in one of the three described views.

After the user has described his provisional SLA the agent forms it according to the ontology so it can be sent and understood by ISP's intelligent agents.

3.1.2 Scan

One fundamental issue we have to deal with, in order to succeed a reliable communication base is first of all the good knowledge of the available surrounding networks. Chameleon design has a number of slots where many different network cards (NIC) can be adjusted dynamically. The end user can insert at any time a new network card in a free slot of chameleon. The activation and configuration of the new card will automatically be managed by hotplug. This means that it will be functional immediately without the need of a device reset. So with a synchronous "all interfaces scan" the system may acknowledge new detected networks or keep track of much information about already existing. Such information is the link level, the power transmit, etc. Afterwards, the measurements are processed and a decision is made, whether there is an anomaly in the connection and if a transition is considerable, to a new more stable network.

An additional feature of the proposed system is the ability to identify whether specific QoS protocols are available e.g. 802.11e and trigger relative features in the NIC.

3.1.3 Discovery and Negotiation

The discovery of services follows the initial scan for networks. At this point we can directly "ask" the providers that were scanned for services or a special repository such as UDDI for services can be questioned. In direct contact with the provider multiple negotiation schemes can be adapted from FIPA [6] (Foundation for Physical Agents) where formal definitions of several standard negotiation protocols are presented. Another approach is that of WS-Agreement [7] a standard proposed by Global Grid Forum [8]. Depending on our approach this choices can be described using semantic tools so it can be understood by both parties and dynamically adapted based on users preference or preconfigured strategies in the agent. For example our client after using his PDA during his transportation he boots his powerful laptop in a friends house. Our agent scans the change in its owner's environment and automatically –or after asking based on its pre-configuration- searches for a more appropriate service based on new needs that may arise. The new environment states a more stationary condition and more time of possible use of the service. This means that a more persistent search and a possible exhaustive auction is more appropriate instead of a simple request-replay that it used during the transportation and the PDA use. The description of this multiple negotiation schemes with semantic tools will make them easy to understand and be used. In more advanced scenarios this schemes will be easy, through model-driven techniques, to be altered by simple users and providers that want to design their own strategies in negotiations. In that way we will avoid hard-coding one negotiation protocol to our agent, which must be known in advanced by all parties. Instead we will design a basic interaction protocol capable of supporting all possible negotiation schemes that will be described. As we see in figure 3 such a protocol will

act as a platform capable of understanding descriptive rules or transaction ontologies and adapting. It will have only hard coded the information needed to form lower communication levels and understand the semantic guidance.

The idea of dynamically adapting multiple negotiation schemes has been proposed with multiple variations. Bartolini et al [9] defines an agent-based generalized interaction protocol that can be specialised with rules very similar to our approach. They provide a taxonomy of such rules for various negotiation mechanisms with good results for a general negotiation protocol that will be in position to deal with everything. Tamma et.al [10] also describes an approach to negotiation, where the negotiation protocol does not need to be hard-coded in agents, but it is represented by an ontology: an explicit and declarative representation of the negotiation protocol. In this approach, agents need very little prior knowledge of the protocol, and acquire this knowledge directly from the marketplace. The ontology is also used to tune agents' strategies to the specific protocol used.

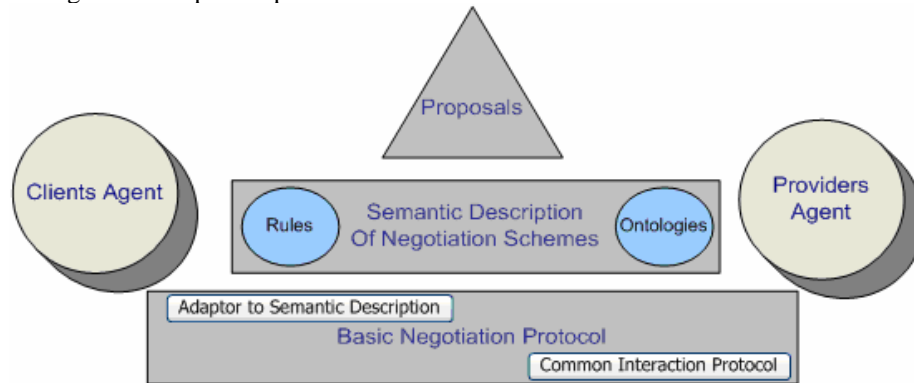


Fig. 3. Adaptable negotiation stack.

We intend to use ontologies to describe the concepts and relationships that are shared across major negotiation protocols in a higher level negotiation ontology. This will provide us with the basic vocabulary that the negotiation agents must share. Upon this and with the use of other domain specific ontologies like time and FSM (Finite State Machine) ontologies we can describe multiple negotiation schemes. An engine capable of understanding and using this basic ontologies is going to be implemented with the ability to adjust dynamically to described schemes. This will give us the ability to describe new strategies based on the same basic ontologies as described above.

In figure 4 we can see a negotiation scheme which describes a “small” auction that gives the opportunity to service providers to improve their suggestion only ones before the final decision is made. The first step for the end user is to create his service profiles with the aid of above described ontology based interface. We use “profiles” because the client will be able to design multiple service schemes in order to use it in different circumstances, with different end devices. Chameleon receives both profiles and providers list after scanning. It then produces and sends client’s goals-provisional SLA- to providers. The matchmaking process takes place after receiving provider’s

propositions two times before the final decision is made. The UML diagram will be formed in a transitional ontology based on the basic ontologies which our basic hard-coded engine and protocol will understand and enforce. We can think multiple strategies that can be developed based on available negotiation schemes and our vision is to unleash this potential with semantically described - easy adopted schemes.

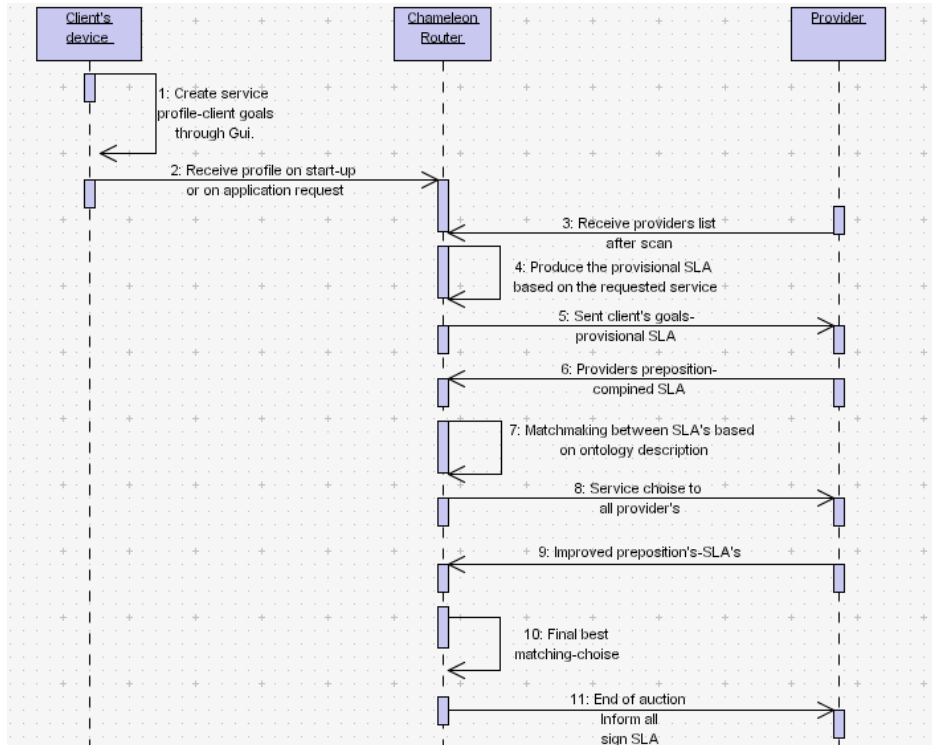


Fig. 4. A possible negotiation scheme in a transition diagram.

3.1.4 Matchmaking

Our model of software-agent communication is based on the assumption that two agents, who wish to converse, share a common Service ontology which ensures that they ascribe the same meaning to the terminology used. In an open environment, agents are designed around various ontologies (either implicit or explicit). For allowing their communication, explicit ontologies are however necessary, together with a standard mechanism to access and refer to them (such as an access protocol or a naming space). Without explicit ontologies, agents need to share intrinsically the same ontology to be able to communicate and this is a strong constraint in an open environment where agents, designed by different programmers or organizations, may enter into communication. Explicit ontologies can be considered as “a referring knowledge” and, as a consequence, can be outside the communicating agents. FIPA00006 specification deals with technologies enabling agents to manage explicit, declaratively represented ontologies. We intent to build upon this specification, using

modern techniques and tools. In figure 5 we can see a diagram showing the two sides –clients and providers- along with the basic function blocks of our framework. ISPs form their provisional SLAs based on multiple ontologies that can be easily located in a public repository. They can also create their own ontologies as long as they give a way for clients to interact with them. On the other side clients create their service profiles according to their multiple aspect ontology as described above in the user interface.

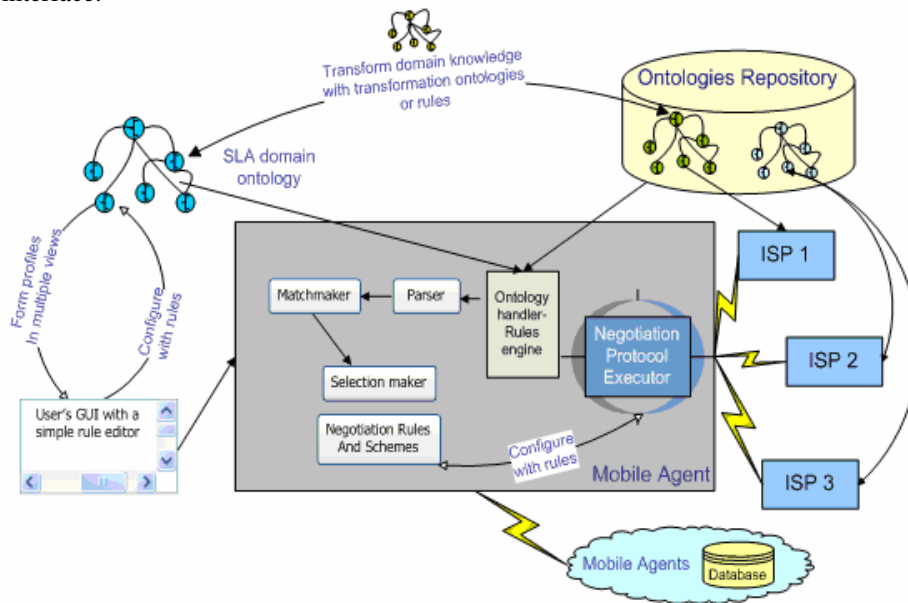


Fig. 5. Service acquiring architecture with basic components.

The way to make the two worlds understand each other and interoperate is to translate service domain knowledge either with transformation ontologies or with rules. Both seem to be very powerful and promising tools in semantic annotation and reasoning, and currently there is ongoing work on combining both. Along with their power, questions regarding the need for such rich languages due to several complexity and computability barriers have aroused.

Our goal is to examine how this powerful but expensive, in terms of complexity, tools can fit in the more limited mobile world as part of our agent. Our internet service specific domain environment could be possibly solved with easier static approaches but in that way we would not create a mobile agent ready for the full semantically automated tomorrows web.

Our service selection mechanism need to be based not only on semantic service description given from the perspective of providers but also consider pragmatics which builds on but is deeper than semantics [11]. For that we intend to give our agent the means to cooperate with other agents on evaluating service providers and contacting special repositories (databases) that keep useful information for the selection process.

3.1.5 SLAs

An SLA specifies agreements between a service provider and a customer and the measurements to be taken in case of deviation and failure. The domain of SLAs is recently very developed as many companies and ISPs are developing tools to represent the relationship between two cooperating parties. Again in web services area we have WSDL (Web Services Description Language) which despite the name does not provide a means to express services capabilities; therefore such standards as WS-Policy [12] WSLA [13] and WS-Agreement specification exist to allow for the expression of additional attributes in order to discover, match and use a web service correctly. All the above standards will be our guide in forming an appropriate SLA ontology that will include service ontology, signatory and supporting parties as well as obligations and actions to be taken [14].

In a recent works [15] semantic tools (owl, ontologies, rules) are used able to reason over ontologies and retrieve data via ontology queries with much less effort giving more accurate matches. This work focuses on matching Service Level Objectives in the WS-Agreement specification with reasoning over qualifying conditions and business values. It combines OWL ontologies of the specific domain, QoS, agreements and ARL rules to provide the matcher with detailed knowledge. Our work is based on the same principles and technologies but deals with the more specific area of internet service in a mobile agent a restriction that will not let us use already developed tools as in a stable powerful machine. All described procedures have to adapt in the mobile environment keeping close all basic functionality and putting on remote sites everything that can be remotely invoked. On a web services ready agent remote invoking will not be a problem and that is the reason we designed our framework based on SWS.

3.1.6 Monitoring SLAs

Monitoring and management of SLAs is an important aspect of the relationship between the two parties and it is described in the SLA ontology. There we will have a complete description of the obligations of two signatory parties which will specify service parameters to be monitored with metrics and functions. It will allow other parties involve in the monitoring phase with details on their exact role as well as describe actions to be taken in any case that can arise. It will be like any other contract between companies, in the domain of service, properly designed to include all the above in order to automate monitoring and reaction. IBM's WSLA gives a description of such an SLA in an static XML description. We intend to build upon it a dynamic ontology model with all advantages this will mean for our framework.

3.2 Network Mobility

One major issue to be solved is the continuous connection to the Internet. A mobile user wants to move between networks like 802.11, 3G, etc and communicate with hosts using the internet, or by creating ad-hoc networks. He also wants to have the ability to choose his favorite provider, and have continuous connection from any

network to any network. But is the mobility problem a standalone problem or is it only a part of a huger issue, which is arising nowadays. Mobility is maybe the first negative sign of the tension of different services and networks to work under one communication system.

For example, Wi-Fi networks either using access points or ad-hoc schemes provide wireless connections to wireless users, and can be gateways to the internet. The addressing scheme serves only the mobility within one access point or, if TAP DANCE [16] is available, between access points of the same provider. To be more specific, moving to a different provider presumes the change of the end user's IP, something that causes all active connections to fail.

GSM and 3G mobile networks today provide internet services through mobile telephones, via WAP, GPRS e.g. The transition, though, from one provider to another is not yet available and the end user's choices are limited to the services and the network availability of his initial selection. Still it is obvious that even if one can use different providers, the capability for a smooth transition is impossible due to addressing problems.

Telephony as well is another major technology that is adjusting to the internet the later years. VoIP is already in use, as it is designed based on the internet infrastructure. On the other hand as far as it concerns the traditional telephony networks there is an effort from an IETF Working Group for E.164 [17] numbers to be expressed as a Fully Qualified Domain Name in a specific Internet Infrastructure domain defined for this purpose (e164.arpa).

Yet numerous other kinds of networks exist, like peer to peer, vpn's, etc. All these use specific protocols, addressing and routing schemes and operate over the internet using existing technologies only as the medium.

It is obvious that the combination of many different types of networks increases not only the complexity of the mobility problem, but generally creates many communication problems due to different addressing schemes, address translations and routing operations. Accordingly, we argue that, a global and scalable mechanism that unifies addressing in a way that is independent of networks, and addressing formats is critical for achieving ubiquitous and seamless connectivity.

We propose a new mechanism that unifies network addressing and works on the existing internet infrastructure based on peer to peer network techniques aiming at finding any host on the internet.

Similar effort towards the solution in a unified network can be found in "Internet Indirection Infrastructure" (i3) [18]. The idea of i3 contains the nodes and a hardware infrastructure of a system based on CHORD's [19] network model. Every node registers its id in the infrastructure. So when someone is trying to communicate with that node, searches in the overlay for the intermediate point where the node is registered, and forwards the packets to it. The intermediate point is then responsible to deliver the packets to the final destination.

In contrast we suggest a scheme for a fully distributed overlay naming and binding system, based on CAN P2P network model [20]. The idea is to create a distributed overlay network on a virtual space, and bind a fixed point with a variable value like the physical address. In that space, all nodes will use a unique and by that, each one exists only in a specific point. Anyone who wants to communicate with a node, all he has to

do is to look in the overlay for the specific id, and ask the node for its current IP address. So instead of having intermediate nodes forwarding our packets, we have nodes which discover each other only once, and then communicate based on the existing IP technology. We could compare the procedure with a distributed DNS model which is continuously updated for every IP of its nodes. This way, our system can be scalable and have better performance because data travel shorter distances towards their destination. QoS is not affected and the recovery of a loss node is not invidious for our system because the time to find the node once is much very small. CAN is a structured P2P network which creates a d-dimensional virtual space. At every node that participates in the overlay is assigned a portion of that space (which is called zone). When a node wants to enter the overlay it hashes its IP address. The product of that hash function is some coordinates that correspond to a point in the virtual space. The node, then, contacts the node that is responsible for that point in the overlay, splits that node's zone in a given dimension and takes the one of the two new zones that were produced.

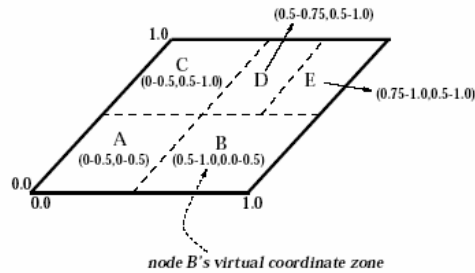


Fig. 6. CAN space distribution [20]

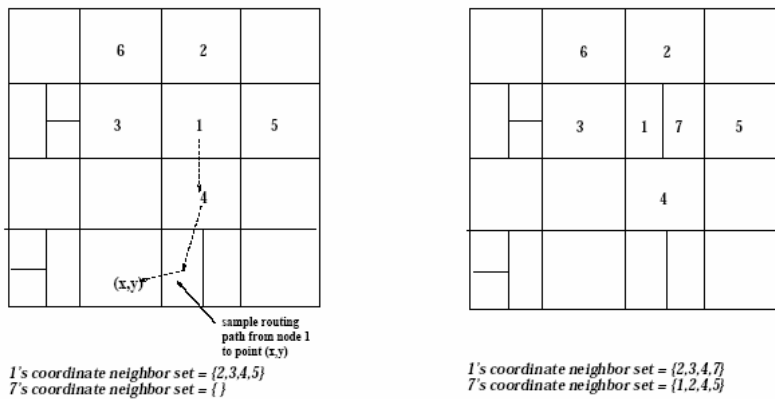


Fig. 7. CAN node insertion [20]

In figure 6 we can see an example for a 2-d space with 5 nodes and the space distribution. Moreover in figure 7 it is shown an example with the insertion of a new

node (node 7) and the re-distribution of the space in the overlay. Every node maintains a structure, called routing table, with the nodes (called neighbors) that have zones adjacent to its zone. This structure has the IP addresses of its neighbors and the coordinates of their zones. With the help of that structure any node can contact every other node in $O((d/2)(n^{1/d}))$ hops (where d is the dimension of the overlay and n the number of the nodes) by forwarding a message in every hop closer to its destination.

Our system is based in CAN but differs in two ways. First, nodes enter the overlay not according to the hash product of their network address but according to the hash product of a unique id, $H(id)$, in exactly the same way as $i3$. The unique id could for example be a URL. So, now the point in which a node is mapped in the overlay is the same every time the node participates in the overlay and independent of its network address. Secondly, when a node N enters the overlay it contacts the node who is responsible for the point $H(id_N)$ and splits its zone not in the middle but in a way such as each node i has in its zone the point $H(id_i)$.

When a node wants to contact another node with $id=x$ it routes through the overlay to the point $H(x)$. If the node which is responsible for that point has $id=x$ it sends back its network address and the addresses of its neighbors, otherwise it responds that the node with $id=x$ is not in the system. In that way any node can discover if any other node is in the system, if it is can learn its network address and can establish a direct connection with it.

In the case of a mobile user which changes his network address the only thing that needs to be done is that the mobile user must inform its neighbors about its new network address and nothing else (his place in the overlay remains the same as that is independent of its network address). If that happens during a transaction with another node N , that node detects a timeout and contacts the neighbors of the mobile user, which their addresses has previously cached, to obtain its new network address. If the cached addresses are stale, which means that those nodes have also changed their addresses, node N uses again the overlay to obtain the new address of the mobile user. Our system inherits from the CAN overlay its stability, scalability and fault-tolerance even if it is comprised of unreliable components. It is fully distributed and doesn't need any special infrastructure as in $i3$. Every node that participates in the system can discover any other node, if it knows its id, in $O((d/2)(n^{1/d}))$ hops and, most importantly, can make a direct connection with it. There is no notion of redirection as in $i3$ and mobile IP architectures. The recovery process, when a node changes its address during a connection, is very fast and with high probability $O(1)$.

We decided to use the CAN overlay instead of any other structured DHT system, like Chord or Pastry [21], for its consistency mechanism. Unlike the other DHT systems, in CAN every node knows which nodes will be affected by a change in its network address and informs them instantly. In Chord, for example a change in the network address of a node will result in stale entries in the routing tables of other nodes with no easy way to be updated instantly. Additionally, CAN is flexible by the perspective of the number of dimensions used to define the virtual space. The number of dimension is a tradeoff between the number of nodes, desirable consistency level and bandwidth overhead for maintenance.

4. Related Work

Agent negotiation research has focused on the specification of protocols, often using conversations [22] specified as finite state machines. FIPA and WS-Agreement has defined various interaction protocols, including English and Dutch auctions [23]. Pitt [24] defines a semantic framework around FIPA ACL to allow the easier specification of multi-party interactions by adding structured conversation identifiers and a richer representation of protocol states. Multiple other efforts have been made like server-based auction rules and contract templates [25], the more general [26] which defines a formal approach for electronic agents interaction, [27] which covered the auction design space classifying mechanisms according to several parameters. Our approach takes the above as inputs and goes beyond defining a library of protocols, to a dynamically adaptable basic negotiation protocol that can be parameterized through semantic annotation.

In the domain of Service Level Agreement (SLA) major work in negotiating and matching is purely syntactic. Yang [14] developed a methodology for matching Web Service Level Agreements (WSLA). There SLAs are syntactically matched by parsing them into syntax trees and comparing them node by node. Heterogeneous SLAs are handled by referencing a table containing instructions to convert them into the same format. Such syntactic approaches must take a more exhaustive and laborious approach to matchmaking and are challenged by less obvious matches. In Paschke [28] a rule based SLA language (RBSLA) is used to express Service Level Agreements. The rules are based on the logic components of Derivation, Event Condition, Event Calculus, Courteous Logic, Deontic Logic, and Description Logic. Rule based SLAs can be written and modified using the management tool (RBSLM) which also enables the management, maintenance and monitoring of contract rules. Uszok [29] has developed KAOS using Semantic Web technologies for the specification, management, analysis, and enforcement of policies. The policy is represented using concepts from an OWL ontology. Our work also builds upon semantic tools and adjusts them in the mobile environment.

Until today many solutions for the mobility problem have been presented in relative projects with mobile routers, with the majority of them based on the Mobile IP [30] protocol. All of them try to approach the mobility problem using existing protocols and technologies and in most cases introduce solutions for specific network models, adopting the Mobile IP scheme for the rest of them. The IOTA (Integration Of Two Access technologies) [31] project with a network element called *IOTA gateway* it provides seamless connectivity for a user with AAA authentication services in different networks, but solves the mobility issue by implementing Mobile-IP agents. MAR (Mobile Access Router) [32] project introduces a mobile router for public transportation which provides to the end user aggregated bandwidth of multiple different connections, and shift load poor quality better quality channels. But it doesn't provide mobility between the different interfaces, in case one of them is disconnected. MPA (Mobile People Architecture) [33] project is focused on person-to-person reachability with a connection between two end users and not between a user and the internet. Personal Router [1], a project from MIT is focused only in developing an intelligent system that takes decisions about the choice of the available

networks, according to the usual preferences of the mobile user, and selects a network depending on the accounting and QoS preferences of the user.

Only MobileNAT and Mobile IP proposals introduce schemes for mobility, but the first requires hardware Anchor Nodes in essential spots in the internet, and the second requires software Agents to redirect the traffic. It is clear that scalability and additional data load is analogical to the number of mobile users.

5. Future Work

We proposed a complete framework that will automate procedures and adjust useful tools in the mobility area. In order to make our scenario work we have to deal with constraints mobility puts in our way. In service selection a lot of work has been done in all involving procedures (negotiation, matchmaking, SLA formation, monitoring and management). The problem is that these suggestions are not based in a mobility environment and thus most of our work is to experiment on how these are going to be adjusted and how remote invocation can lighten our mobile agent. As far as it concerns the mobility proposed scheme, future work will focus on simulations and measurements to check if and how much the balance of the system is affected by the way nodes enter the overlay, and check potential bottlenecks due to the heterogeneity of the participating nodes and networks.

Acknowledgments

This work is funded by the Greek General Secretariat for Research and Technology in the context of PENED 2003 03ED723 project, (75% EC, 25% Greek Republic, according to 8.3, 3rd Framework programme).

References

- [1] D. Clark and J. Wroclawski, "The Personal Router Whitepaper", Available online: http://www.ana.lcs.mit.edu/papers/PDF/PR_whitepaper_v2.pdf
- [2] OWL-S, OWL-based Web Service Ontology web site, <http://www.daml.org/services/owl-s/>
- [3] Web Services Modeling Ontology web site, <http://www.wsmo.org>
- [4] SWSF web site, <http://www.w3.org/Submission/SWSF/>
- [5] WSDL-S web site, <http://www.w3.org/Submission/WSDL-S/>
- [6] The Foundation for Physical Intelligent Agents web site, <http://www.fipa.org>
- [7] Andrieux, A., Czajkowski, C., Dan, A., Keahey, K., Ludwig, H., Pruyne, J., Rofrano, J., Tuecke, S., Xu, M., "WebServices Agreement Specification (WS-Agreement)", Available online: <http://www.cct.lsu.edu/personal/maclaren/GridPrimer/Sources/WS-AgreementSpecification.pdf>
- [8] Global Grid Forum web site, <http://www.ggf.org>
- [9] Bartolini C., Preist C., Jennings N.R.: Architecting for Reuse: A Software Framework for Automated Negotiation, in Giunchiglia, F., Odell, J., Weiss G. (Eds.): Agent-Oriented Software Engineering III (2002), Springer-Verlag LNCS 2585/2003.
- [10] Valentina Tamma, Steve Phelps, Ian Dickinson, Michael Wooldridge, "Ontologies for supporting negotiation in e-commerce", Engineering Applications of Artificial Intelligence, 18 (2005) pp. 223–236, Elsevier

16 **G. Koumoutsos, K. Lampropoulos, N. Efthymiopoulos, A. Christakidis, S. Denazis, K. Thramboulidis**

- [11] R. Sreenath, M. Singh, "Agent-Based Service Selection", Journal of Web Semantics (2004)
- [12] The Web Service Policy Framework, Available on line: <http://www-106.ibm.com/developerworkds/library/ws-polfram>
- [13] The WSLA Specification Available online, <http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf>.
- [14] W. Yang, H. Ludwig, A. Dan, "Compatibility Analysis of WSLA Service Level Objectives", Workshop on the Design of Self-Managing Systems, RC22800 (W0305-082) Computer Science, 2003
- [15] N. Oldham, K. Verma, A. Sheth, F. Hakimpour, "Semantic WS-Agreements Partner Selection" WWW 2006, May 23–26, 2006, Edinburgh, Scotland. ACM 1-59593-323-9/06/0005.
- [16] Tap-dance available online: http://www.atmel.com/dyn/corporate/view_detail.asp?ref=&FileName=Tap-Dance.html&SEC_NAME=Product
- [17] Telephone Number Mapping (enum) available online: <http://www.ietf.org/html.charters/enum-charter.html>
- [18] Ion Stoica, Daniel Adkins, Shelley Zhuang, Scott Shenker, Sonesh Surana : "Internet Indirection Infrastructure", Proceedings of *SIGCOMM '02* Pittsburgh, Pennsylvania USA
- [19] Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek, and Hari Balakrishnan : "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications", Proceedings of the 2001 ACM SIGCOMM Conference
- [20] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, Scott Shenker, "A Scalable Content-Addressable Network" , *SIGCOMM'01*, 2001, San Diego, California, USA
- [21] Antony Rowstron and Peter Druschel : Pastry: "Scalable, decentralized object location and routing for large-scale peer-to-peer systems", in Proc. of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001). Heidelberg, Germany, November 2001
- [22] Parsons, S., Sierra, C., Jennings, N.R.: Agents that reason and negotiate by arguing. Journal of Logic and Computation (1998), 8(3), 261–292.
- [23] Foundation for Physical Agents. FIPA Interaction Protocol Library Specification, 2000. Available at <http://www.fipa.org>
- [24] Pitt, J., Guerin, F. and Stergiou, C.: Protocols and Intentional Specifications of Multi-Party Agent Conversations for Brokerage and Auctions. In Proc. Fourth International Conference on Autonomous Agents, ACM Press (2000), 269-276
- [25] Reeves, D., Wellman, M. and Grosz, B. Automated Negotiation from Declarative Contract Descriptions. In Proc. Fifth International Conference on Autonomous Agents, (2001)
- [26] Esteva, M., Rodriguez, J. A., Sierra, C., Garcia, P., Arcos, J. L.,: On the formal specifications of electronic institutions, In Dignum F. Sierra, C. (eds.) Agent-mediated Electronic commerce (The European AgentLink Perspective), Springer LNAI. (2000)
- [27] Wurman, P., Wellman, M. and Walsh W.: A Parameterization of the Auction Design Space, in Games and Economic Behavior, 35 Vol. 1/2 (2001), 271-303
- [28] Paschke, A., Dietrich, J., Kuhla, K. A Logic Based SLA Management Framework. Proc of the Semantic Web and Policy Workshop, November, 2005.
- [29] Uszok, A., Bradshaw, J.M., Jeffers, R., Johnson, M., Tate, A., Dalton, J., Aitken, S. Policy and Contract Managementn for Semantic Web Services, Proc of the AAAI Spring Symposium on Semantic Web Services, 2004
- [30] [RFC 3344] IP Mobility Support for IPv4 available online: <http://www.rfc-archive.org/getrfc.php?rfc=3344>
- [31] M. Buddhikot, G. Chandranmenon, S. Han, Y. W. Lee, S. Miller, L. Salgarelli: "Integration of 802.11 and third-Generation Wireless Data Networks", Proc. IEEE INFOCOM 2003

- [32] Rajiv Chakravorty, Ian Pratt, Pablo Rodriguez : “Exploiting Network Diversity in MARS. A Mobile Access Router System”, available online:
<http://www.cl.cam.ac.uk/~rc277/hotnets.pdf>
- [33] Mema Roussopoulos, Petros Maniatis, Edward Swierk, Kevin Lai, Guido Appenzeller, Mary Baker : “Person-level Routing in the Mobile People Architecture”, In Proceedings of the USENIX Symposium on Internet Technologies and Systems, October 1999
- [34] Milind Buddhikot, Adiseshu Hari, Kundan Singh, Scott Miller : “MobileNAT : A new technique for mobility across heterogeneous address spaces”, available online:
<http://www.bell-labs.com/user/mbuddhikot/psdocs/mobilenat-MonetIssue-Buddhikot.pdf>