Intelligent end-to-end resource virtualization using Service Oriented Architecture

E. Onur[1], E. Sfakianakis[2], C. Papagianni[3], G. Karagiannis[9], T. Kontos[5], I. Niemegeers[1], I.P. Chochliouros[2], S. Heemstra de Groot[6], P. Sjödin[7], M. Hidell[7], T. Cinkler[4], M. Maliosz[4], D. I. Kaklamani[3], J. Carapinha[8], M. Belesioti[2], E. Fytros[5]

[1]Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, the Netherlands
[2] Labs & New Technologies Division, Research Programs Section, Hellenic Telecommunications Organization S.A. (OTE), Athens, Greece
[3]School of Electrical and Computer Engineering, National Technical University of Athens, Greece
[4] Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Hungary
[5]Siemens IT Solutions and Services, Athens, Greece
[6] Twente Institute for Wireless and Mobile Communication, Enschede, the Netherlands
[7] School of Information and Communication Technology, Royal Institute of Technology, Stockholm, Sweden
[8]PT Inovação S.A., Portugal Telecom Group, Portugal
[9] Department of Electrical Engineering, Mathematics and Computer Science and CTIT University of Twente, Enschede, the Netherlands

***Abstract*— Service-oriented architecture can be considered as a philosophy or paradigm in organizing and utilizing services and capabilities that may be under the control of different ownership domains. Virtualization provides abstraction and isolation of lower level functionalities, enabling portability of higher level functions and sharing of physical resources. However, dynamics, environmental conditions and increasing complexity / heterogeneity of underlying resources call for adaptive resource handling. In this view an intelligent distributed architecture that enables dynamic user management and control on network-wide resource sharing by using the Service Oriented Architecture concept is presented. Additionally, the proposed architecture supports trading of resources that will enable the transformation of traditional business models.**

***Index Terms*— Service Oriented Architecture, cognition, resource virtualization**

I. Introduction

THE face of the Internet is continually changing, as new services and novel applications appear and become globally noteworthy at an increasing pace. With the advent of Web 2.0, various social resource sharing platforms arose which allowed users to easily share digital resources. Nowadays the locus of computation is changing, with functions migrating to remote data centers via Internet based communication. Computing and communication are being blended into new ways of using networked computing systems. The network retains its traditional role as a means of information exchange but, at the same time, it is also perceived as a means of sharing resources, where communication networks are used to share programs, information, processing capacity, communication, storage, and media. Next generation networks and service infrastructures should overcome the scalability, flexibility, resilience and security bottlenecks of current network and service architectures, in order to provide a large variety of services and opportunities, adoptable by business models capable of dynamic and seamless utilization of IT resources based on user-demand across a multiplicity of devices, networks, providers, service domains and social and business processes.

Envisioning the computing utility based on the service provisioning model, where resources are readily available on demand, has led to contemporary computing paradigms that have emerged in the last decade, exploiting technological advances in networked computing environments e.g. GRID computing, peer to peer computing and more recently cloud computing [1]. Virtualization technology makes it all possible. Nowadays infrastructure resources are decoupled from actual components (hardware etc) whereas companies roll out new services and applications without investing in and maintaining an expensive IT infrastructure. Virtualization provides abstraction and isolation of lower level functionalities and underlying hardware, enabling portability of higher level functions and sharing/aggregation of physical resources [2]. It exists in the entire context of IT resources, but current virtualization techniques suffer from being based on managed configurations with static properties. This makes management and service provisioning cumbersome and significantly limits the areas of applicability: In computing, virtualization is mainly used for the sharing of resources in restricted user communities such as Grid networks and emerging cloud platforms such as Amazon Elastic Compute Cloud etc, while in networking, it is primarily used for virtual private network services.

In this broader scope the Service Oriented Architecture

978-1-4244-5626-0/09/$26.00 ©2009 IEEE

(SOA) can be considered as a philosophy that dictates the way services and capabilities that may be under the control of different ownership domains are organized [3]. SOA can also be seen as a way of promoting reuse and interoperability by enabling users and organizations to get more value from capabilities, such as local as well as external resources.

The architecture proposed by the authors in [4] reuses the SOA and Open Grid Services Architecture (OGSA [16]) principles for IT infrastructure resource sharing through virtualization. The distributed architecture allows for dynamic user management and control of network-wide shared resources. However as the complexity and heterogeneity of the underlying resources escalates, the number of tunable parameters increases enormously and selecting of the adequate values for those parameters is becoming very difficult. Furthermore, the dynamics of resources and environmental conditions require an adaptive way for tuning the parameters. In this view, a cognitive control plane is introduced into the presented architecture that will adapt the tunable parameters through short-term reasoning and long-term learning. In addition the architecture is complemented with an enhanced resource trading brokerage mechanism that interacts with the cognitive control plane to provide services and applications with the best possible available resources while enabling various business models.

The remaining part of this paper is organized as follows: In Section II several concepts used are described in a more detailed way. Section III provides an overview of the resource sharing architecture [4] and describes the proposed cognitive control plane and resource trading module. Conclusions and possible future work activities are described in Section IV.

II. Relevant Concepts

A. Service-Oriented Architecture

The SOA concept has been receiving considerable attention in recent years. According to the Organization for the Advancement of Structured Information Standards (OASIS), SOA can be defined as "*a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with, and use capabilities to produce desired effects consistent with measurable preconditions and expectations*" [3]. Following the SOA model, a service provider publishes his service interface via a service registry (e.g. Universal Description Discovery and Integration) where a service request can find the service and, subsequently, may bind to the service provider. Service discovery, brokering, and reliability are important and services are usually designed to co-operate (i.e. to "composite" services). Services can be described and accessed using well defined service interfaces (e.g. by Web Service Description Language [5]) and via standard message-exchanging protocols (e.g. Simple Object Access Protocol [6]) and exercised by following a service contract with certain policies. This provides a loose coupling of services and ensures operational agility.

B. Resource sharing paradigms

Over the years new computing paradigms have been adopted to run on top of virtual machines across the global Internet, with the emergence of technological advances in networked computing environments, multi-core systems etc. These proposed resource-sharing frameworks include Grid computing, on-demand computing and utility computing. Grid computing [3][7] is a key enabling infrastructure for resource sharing and coordinated problem solving in dynamic multi-institutional virtual organizations by using a large amount of virtualized computing resources. On-demand computing and utility computing are related to making computing resources available across a network on a per-usage basis, with the goal to allow for cost-efficient utilization and sharing of resources. According to [2] cloud computing builds upon decades of research in virtualization, distributed computing, grid computing, utility computing, and networking, as well as web services. The tern implies a service oriented architecture, on demand services etc. Appropriate definitions for cloud computing may be also found in [1][8] and [9]. In these definitions that are still evolving, the services in the cloud are provided by namely-enterprises and accessed by others via the Internet. Evolving in parallel, Service Oriented Infrastructure involves the application of fundamental principles of service orientation to the IT infrastructure that enables all applications (either service oriented or not) to operate efficiently. According to the Open Group SOI Reference Model [10], "*a service oriented infrastructure can be regarded as a natural part of a service-oriented enterprise architecture involving the definition and provisioning of IT infrastructure in terms of service*s".

The aforementioned computing models address different aspects of resource sharing. A key issue for Grid systems is resource discovery to locate hardware and software resources for computational-oriented applications [11]. The purpose of on-demand/utility computing is to make computing resources available as reliable commodity services, by employing a more efficient use of resources through dynamic and intelligent reconfiguration on the fly. The service oriented infrastructure paradigm aims at delivering infrastructure as a service from a "pool" of shared virtualized resources, accommodating dynamic demand for IT services. The cloud computing concept is a combination of the above and depending on the case it may include provisioning of infrastructure, platform or software as a service. Many cloud computing deployments[update] depend on grids, employ autonomic characteristics, and bill like utilities. However, there also exist successful cloud architectures that have little or no centralized infrastructure or billing systems, such as peer-to-peer networks like Skype and SETI@home.

C. Network resource virtualization and control

Virtualization constitutes the enabling technology for interconnecting groups of network nodes into virtual private networks (VPNs) [12]. According to [10], in a virtualized network, virtual networks exist co- instantaneously on top of a shared substrate. Different virtual networks may use different protocols and packet formats and provide alternate packet delivery systems. For example over a MPLS/GMPLS (Multi-Protocol Label Switching/Generalized Multi-Protocol Label Switching) network, virtual private networks are provisioned as Label Switched Paths tunnels.

Current virtual networking access and control methods are based on statistical multiplexing of simple conceptual models such as virtual links (Layer 2 Tunneling Protocol, Virtual Private Wire Services), virtual switches (Virtual Private LAN Service) or virtual routers (L3-VPN). These simplistic models have the advantage of keeping the complexity and the details of the underlying infrastructure invisible (and inaccessible) to customers, and instead provide them with a uniform, simple abstraction as the interface. However, these approaches manage to offer little flexibility in the services provided to the end-user who is unable of controlling or monitoring network resources whose functionality is based on static properties. On this direction CANARIE has developed the User Controlled Light Path (UCLP) [13], which enables users to define their own network architectures including topology, routing, virtual routers, switches, etc., by using web services techniques. NFS's (National Science Foundation) GENI (Global Environment for Network Innovations) initiative [14] uses a more refined network concept where resources (links etc) are substrates, and virtualization stands as an abstraction level to hide the underlying physical resources that can be shared by multiple infrastructures.

D. *Intelligent resource management and control*

Future networks will consist of vast amount of heterogeneous devices and different kinds of user/operator networking technologies. Virtualization abstracts the technological implementation details and complexity of the underlying devices and networks from the user. It also helps all the underlying constituents to emerge as a single virtual device providing seamless services to the user and enables user to offer/provide services. In this approach, energy-efficiency, resource utilization, and dependability requirements can be addressed in a modular way and backward-compatibility (in the sense of supporting existing technologies) can be provided. Furthermore, future technologies can be integrated in the virtualized infrastructure without any change-requirements in the existing operational applications or services. In this light, virtualization reduces the complexity, increases manageability, increases usability, enables service provisioning by the end-users, supports application/service portability and compatibility and enhances security/dependability and enables evolutionary advancement of the technologies without impacting the users.

As the complexity and heterogeneity of the resources increases, the number of tunable parameters also increases enormously and selecting the adequate values for those parameters becomes very difficult. Furthermore, the dynamics of resources and the environmental conditions require an adaptive way for tuning the parameters. A cognitive control plane may adapt the tunable parameters through short-term reasoning and long-term learning which is currently performed by the humans, namely, the operations manager. However, for a person to be prompt to instantaneous changes in the operational conditions is nearly (and practically) impossible. Hence, a cognitive engine is required to perform of the duty of that person. A cognitive control plane therefore is needed to "produce decisions" in the presence of incomplete, inconsistent, maybe misleading or malicious information, conflicting or inconsistent high-level objectives, drastically/instantaneously changing operational conditions, high complexity, increased number of tunable parameters, and spatial and temporal dynamics

III. System Functional Description and Architecture

Resource sharing architectures nowadays sustain several limitations considering resource virtualization. On demand/utility computing dictates a server oriented resource usage model whereas GRID computing focuses on sharing geographically distributed computational resources. In addition basic management operations such as resource allocation are based on static configurations. Furthermore by considering all resource sharing paradigms including cloud computing, the resource/service provider retains its traditional role, maintaining control over resources. Therefore, existing models for resource sharing do not enable users to manage available (local / remote) resources in a transparent and efficient way, while they still retain the privacy of their efforts and results whereas they do suffer from several limitations on supporting dynamic trading of resources. Infrastructure services need to be highly reliable, scalable, and autonomic to support ubiquitous access, dynamic discovery and composability. In particular, consumers can determine the required service level through Quality of Service (QoS) parameters and Service Level Agreements (SLAs).

The presented architecture builds on the resource sharing architecture presented in [4] that reuses the SOA and OGSA principles ([15][16]). The architecture will enable provisioning of IT resources through virtualization into an intelligent, distributed architecture that allows for dynamic end-user management and control on network-wide resource sharing. However the added value in the presented architecture is the incorporation of an *intelligent resources control* plane and advanced *resource trading mechanisms* that allow for flexible business models where the roles of infrastructure service provider and consumer may coincide.

The implementation of the Intelligent Unified Resource Virtualization, Control and Support tier (see Figure 1) aims to allow for intelligent, uniform and location-independent access and control of virtualized resources, independently of the specific properties of the underlying resources.

The Unified Resource Virtualization, Control and Support component tier establishes secure connectivity to the underlying shared physical resources through the Virtual Link Layer; registers, manages and controls resources through the Virtualized Resources Layer; handles interactions across collections or pools of resources through the Coordinating Resources Layer; performs resource discovery and scheduling, as well as resource trading, through the Control Resources

Layer. Each layer is comprised of several sub-components that provide the functionality of the layer. Details about the sub-components of each layer are provided in [4]. The system layers interact with each other by means of top-down request and execution, and bottom-up activation.

Intelligence is required to distribute and schedule tasks/processes/resources in heterogeneous environments in an efficient, safe, dependable and cost-efficient way. For example, in case that lack of resources is anticipated, intelligent resource management may warn or automatically compensate for the expected shortage with the appropriate reservations. In addition when considering the variety of the underlying devices and networks and the multiplicity of existing applications and services there is a vast field of heterogeneity created. These must first be abstracted in terms of requirements and capabilities and then must be mapped accordingly to virtual resources to enable portability of applications and enhancement of the resources without affecting the already developed applications or deployed services. Through the cognitive control plane an intelligent resource mapping mechanism will provide enough information about the capabilities of the resources and the device/network capability information will be thus "harmonized" in the Unified Resource Virtualization, Control and Support component.

Additionally, the resource trading mechanism should be able to help stakeholders according to their level of expertise, from layman to expert, understand how to best offer or place their resources and reach a profitable agreement for them. For this reason the resource trading module is required to be able to support various business models / trading mechanisms.

A. Cognitive Control Plane

Present networking and resource utilization functions follow the layered protocol architecture. By design, the abstracted layers have little information regarding the other layers. Because of this unawareness, the network and resource utility maximization problems being solved today produce sub-optimal results. An integrated scheme that imitates the human approach may produce better results. The role of the cognitive control plane is therefore to replace the network and resource manager as much as possible. Within this scope, the objective of the cognitive control plane is to observe, act and learn to produce better results in real-time. This scheme is already proposed for cognitive radio, which, in principal, operates in the physical and data-link layer of the OSI (Open Systems interconnection) stack and is usually unaware of the needs of.

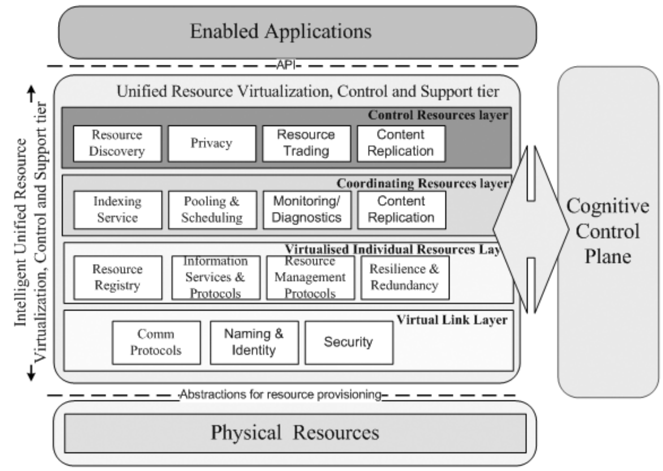

Fig. 1 Conceptual Architecture

the other layers and users. Beyond the network control functionality, the aim of introducing cognition in the resource life cycle is to understand and anticipate the user needs as well. To achieve this, context awareness is required to enable the decision making process and, consequently, this imposes the situatedness (being embedded in the environment and infrastructure to monitor the states) challenge on the cognitive control plane. In our approach, we basically assume that the cognitive engine has enough degrees of freedom in acting

In essence, two approaches can be employed to introduce intelligent control in the architecture, i.e.: centralized cognitive control plane or distributed middleware that runs on each device as a virtualization layer. In the centralized implementation, there is a central entity in the network to "run" the cognitive functionalities. In the distributed middleware approach, each entity in the network "runs" their own cognitive functionalities with or without communicating to each other. If the second approach is employed, defining only the individual cognitive control plane is not enough to have a complete view of the system. The emergent (property of a collection of subsystems due to the interactions among subsystems) behavior of the cognitive planes must also be analyzed. For example, the cognitive control plane implemented on a device may exhibit some selfish behavior. The overall impact of this selfishness may be "balanced" by adequate selection of the strategies in the community of the devices.

The detailed functionalities of the cognitive control plane are shown in Figure 2. Since the cognitive engine in the cognitive control plane is there to solve multi-objective optimization problems, the solutions are Pareto-optimal [17]. Based on the gathered state and context information about the physical resources, user objectives and user patterns, the elements of the cognitive engine runs the planning and decision procedures and provides the physical control layer the execution plan regarding the action to be taken. The elements in the cognitive engine can be neural networks, swarm intelligence, evolutionary programming, immune or fuzzy systems, etc. The cognitive engine adapts itself continuously

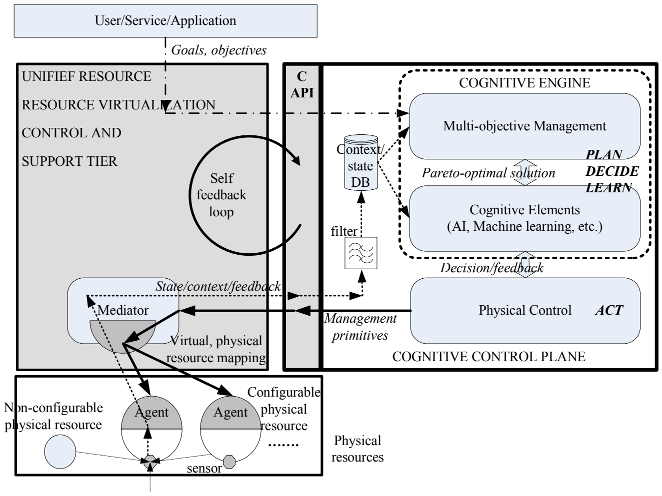

Fig. 2 Cognitive Control Plane functions

through using the context database, and a priori actions and their feedbacks. Also the virtualization tier provides feedbacks to the cognitive control plane. The cognitive application programming interface (CAPI) provides the required interface between the virtualization tier and the cognitive control plane.

The interactions of the cognitive control plane with the virtualization tier are depicted in Figure 2. The interaction between the cognitive control plane and the user/application layer has two basic purposes: (i) collecting information from the user/application through direct/indirect observations or service level negotiations to learn/anticipate the user needs and (ii) allowing the user to be in the control loop of the cognitive decision process. Although, for the sake of autonomous operation the second functionality should be minimal, from time to time user interventions may be required.

The cognitive control plane requires input through observations of the states of the virtual or physical resources via sensors; e.g., state of the congestion levels in the network, to which the multimedia trunks are assigned, residual battery levels, queue sizes of the services, memory utilization of the applications or user behaviors. In order to learn the behavior of the virtual infrastructure, context information must be gathered and stored in a database. Instead of using the vast amount of unprocessed data, the input can be filtered to extract the useful information that can be considered as feature extraction. Accuracy, timeliness and the price of the ignorance (extraction of the irrelevant data) are the key measures to be analyzed [17].

The research challenges of introducing intelligent control over the resources in a virtualized infrastructure can be summarized as follows:

- Developing a cognitive control plane to learn and control the network elements, aggregations and relationships among network elements, to keep track of transport entities, and their endpoints, to monitor alarms, to configure resources.
- Decisions produced by the cognitive control plane are required in real-time. Therefore, the elements of the cognitive control plane are to be designed to support real-time resource management.
- Specification of the cognitive network application programming interface (CAPI). Definition of the objectives, the management primitives, feedback loops, resource mappings are part of the CAPI.
- Developing an intelligent resource mapping mechanism. When the heterogeneity of the underlying devices and networks are considered, there will be a large number of resources. These resources must be mapped to virtual resources to enable portability of applications and enhancement of the resources without affecting the already developed applications. Therefore, one of the research challenges is mapping of the resources to corresponding virtual resources. This sort of "mapping" must provide enough information about the capabilities of the resources, while the device/network capability information must be harmonized in the virtualization tier.
- The price of control is also to be analyzed [17]. Introducing the cognitive plane and the required messaging to bring in intelligence in the virtualized infrastructure may use in-band communication resources; therefore, the trade-off between the overhead and benefits must be further analyzed.
- Developing mechanisms to make the virtualization layer as smart as possible to organize and heal itself to reduce the number of user interventions. From the viewpoint of the user, this reduces the complexity and increases the manageability.
- The devices and users are mobile. Therefore, the opportunistic and dynamic nature of the underlying wireless communication technologies must also be exploited in the virtualization layer. This should be done handle the dynamic nature that can be expected from the underlying networks.
- Along with the four logical layers of network management defined by the International Telecommunication Union (business, service, network and element), the user management must be incorporated into the cognitive control plane.

B. *Resource Trading*

A resource trading module will be utilized enabling indirect ccess to available physical resources in accordance to the various business models supported by various end-user applications. So far the concept of cost functions and/or reputation schemes has been proposed in mesh networks in order to regulate demand for resources by discouraging 'free riding' ([18][19]). This being known in the case of ad hoc wireless networks, is also an option for a heterogeneous (essentially ad hoc) network such as the proposed infrastructure. Application business models enabled by the architecture are the following:

- "Best-effort": Best-effort infrastructure services are supplied similar to best-effort internet services that provide no guaranteed QoS with low or zero cost. Since there are no guarantees, resources may disappear without prior notice. The only assurance is end-user's set priority on current resource usage compared to the other competing users.

- “Peer-to-peer trading”: The end-user publishes unused resources to the public. No centralized management or control is applied. The resources could be charged or provided as a best-effort virtualization service.
- “Direct trading”: This particular business model assumes that the resource provider performs trading, directly to end-users/enterprise. In this case resources are either resource provider’s assets or “out-sourced” end-user resources. The resource provider controls, configures and manages diametrically the shared resources.
- “Trading broker”: This last business model specifies a separate business entity that does not own the resources but supports advertising, allocation, accounting and billing of the resources.

The architecture design of such a component will be based on the implementation of a resource trading broker. Interacting with the various modules of the Unified Resource Virtualization, Control and Support tier, the broker will identify and choose the proper negotiation mechanism between the user and the owner of the resource. The user is bound to participate in the trading process only in the case that particular quality guaranties for the resource are needed, or the resource is not available on a best-effort basis. In addition the resource trading module interacts with the cognitive control plane in order to provide services and applications with the best possible available resources

IV. Conclusion

In previous publication we have presented an architecture that can be used to support resource virtualization for Web 2.0 applications, by using the SOA concept. In the presented study, this resource sharing framework is enhanced with a cognitive engine that is set to produce solutions on resource utility optimization problems. Specifically the adoption of a cognitive control plane is described, its functionalities and required interactions with the remaining resource sharing architecture, resources and applications. Moreover we have identified and presented the research challenges of introducing intelligent control over the resources in a virtualized infrastructure. In addition, the resource trading module of the architecture was analyzed in terms of the various business models that will be enabled by the architecture. Efficient resource trading brings forth a new service architecture where resource ownership and administrative rights can be decoupled and traded between users. Future activities will focus on the implementation and evaluation of the resource sharing architecture, as well as in developing additional resource trading mechanisms to meet upcoming business and social needs. Finally, new virtualization methods may need to be developed in order to complement the aforementioned functionalities.

Acknowledgment

The authors wish to express their gratitude to N. Tselikas and L. Westberg for valuable discussions. This work is partially funded by the Dutch Ministry of Economic Affairs in the “Future home network” project of the IOP-GenCom programme.

References

[1] Buyya R, "Market-Oriented Cloud Computing: Vision, Hype, and Reality of Delivering Computing as the 5th Utility," 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, pp. 1 2009

[2] Vouk M., “Cloud computing - Issues, research and implementations.”, Journal of computing and information technology, vol.4 (16), pp. 235-246, 2008

[3] MacKenzie, M., Laskey, K., McCable, F., Brown, P. F., Metz, R., “OASIS – Reference Model for Service Oriented Architecture 1.0”, http://www.oasis-open.org/, 2006

[4] Papagianni C., Karagiannis G., Tselikas N. D., Sfakianakis E., Chochliouros I.P., Kabilafkas D., Cinkler T., Westberg L., Sjodin P.,. Hidell M, He emstra de Groot S., Kontos T., Katsigiannis C., Pappas C., Venieris I. S., "Supporting End-to-End Resource Virtualization for Web 2.0 Applications using Service-Oriented Architecture," 2nd IEEE Workshop on Enabling the Future Service-Oriented Internet, IEEE GLOBECOM 2008, New Orleans, 2008.

[5] ECMA-348, “Web Services Description Language (WSDL) for CSTA Phase III”, 2nd ed., June 2004, http://www.ecma-international.org/publications/standards/Ecma-348.htm

[6] SOAP/WSDL, WSDL, http://www.w3.org/2002/ws/

[7] Foster, I., C. Kesselman, and S. Tuecke, The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of High Performance Computing Applications, 2001. 15(3): p. 200-222

[8] Voas J., Zhang J., “Cloud Computing: New Wine or Just a New Bottle?”. IT Professional Volume 11 (2), pp. 15-17, 2009

[9] Mei L., Chan W.K., Tse T.H., “A Tale of Clouds: Paradigm Comparisons and Some Thoughts on Research Issues” , Asia-Pacific Services Computing Conference, IEEE, pp 464– 469, 2008

[10] The *Open Group*, “Service Oriented Infrastructure Reference Framework”,http://www.opengroup.org/projects/soa-soi/uploads/40/19218/soi-V1-5-P1.pdf , 2009

[11] Talia, D., Trunfio, P., “Peer-to-Peer Protocols and Grid Services for Resource Discovery on Grids”. In: L. Grandinetti (Ed.), Grid Computing: The New Frontier of High Performance Computing, Advances in ParallelComputing, vol. 14, Elsevier Science, 2005

[12] Khanvilkar, S., Khokhar, A., "Virtual Private Networks: An Overview with Performance Evaluation", IEEE Communication magazine, vol. 42 (10), pp. 146-154, October 2004

[13] UCLP project, http://www.uclpv2.ca/

[14] GENI—Global Environment for Network Innovations, http://www.geni.net

[15] Bhatia K., ed., “Peer-to-Peer Requirements on The Open Grid Services Architecture Framework”, Informational Document, Global Grid Forum (GGF), OGSAP2P Research Group, July 12, 2005, located at: http://www.ogf.org/documents/GFD.49.pdf

[16] Foster, I,. Kishimoto, H., Savva, A., Berry, D., Djaoui, A., Grimshaw, A., Horn, B., Maciel, F., Siebenlist, F., Subramaniam, R., Treadwell, J., Von Reich, J., “The Open Grid Services Architecture, Version 1.5.. Informational Document, Global Grid Forum (GGF), July 24, 2006, located at: http://www.ogf.org/documents/GFD.80.pdf

[17] Ryan W. Thomas; Daniel H. Friend; Luiz A. Dasilva; Allen B. Mackenzie, "Cognitive networks: adaptation and learning to achieve end-to-end performance objectives," *IEEE Communications Magazine*, vol.44 (12), pp.51-57, 2006

[18] Andrade N., Mowbray M., Cirne W., Brasileiro F., “When Can an Autonomous Reputation Scheme Discourage Free-riding in a Peer-to-Peer System?”, 2004 IEEE International Symposium on Cluster Computing and the Grid, IEEE Computer Society

[19] Marti S., Garcia-Molina H, "Taxonomy of trust: Categorizing P2P reputation systems", Computer Networks Volume 50, Issue 4, 15 March 2006, Pages 472-484 Management in Peer-to-Peer Systems.