SAGA: AN ADAPTIVE INFRASTRUCTURE FOR SECURE AVAILABLE GRID ON AD-HOC NETWORKS

Manel Abdelkader, Noureddine Boudriga and Mohammad S. Obaidat
University of November 7th at Carthage, Tunisia and Monmouth University, NJ, USA
Corresponding Author: Prof. M. S. Obaidat, Department of Computer Science,
Monmouth University, W. Long Branch, NJ 0764, USA

Keywords: System security, grid computing, ad hoc networks, adaptive infrastructures.

Abstract: Security management is a major issue in Grid computing systems. One approach to provide security is to implement techniques such as encryption and access control on all grid elements. The overhead generated by such an approach may however limit the advantages of Grid computing systems; particularly, when the network experiences different types of variations. This paper examines the integration of the notion of adaptive Grid service along with security management and accounting. It also provides a fault tolerance technique to build Grid systems that are intrusion tolerant.

1 INTRODUCTION

Recently, Grid computing has emerged as an attractive area characterized by large scale resource sharing, innovative applications and high-performance capabilities. The Grid problem has been defined as a flexible, secure, coordinated resource sharing among dynamic sets of individuals, institutions, and resources. In such settings, one can come across authentication, authorization, resource access, resource discovery, and quality of service (QoS) provision challenges.

Until recently, the main priority for grid developers has been to design working prototypes and demonstrate that applications can be developed over a grid environment with a restricted support to application-level fault tolerance in computational grids. Limited work also has addressed the integration of quality of service (QoS) and security while allocating resources. However, failure detection services were among the main supportive tools in developing grid environments; but, neither solution has considered the intrusion detection and intrusion tolerance (Casanova et al., 2003) (Song and Hwang, 2004), nor did they provide schemes for the management of GRID resources when variability in the network dynamics is experienced.

Several recent studies have investigated security and trust management while allocating Grid resources (Butt et al., 2002) (Azzedin and Maheswaran, 2002) (Czajkowski et al., 1999). Quite a few models have been proposed to quantify trust in Grid applications, including fuzzy model that was provided for e-commerce applications by (Gefen et al., 2003). Even though these studies have made interesting contributions, they did not address situations where multi-level trust is required, which is a natural assumption in environment where businesses are provided.

Wireless ad-hoc networks do not rely on a pre-existing network infrastructure and are characterized by a wireless multi-hop communication. Wireless ad-hoc networks are increasingly used in situations where a network must be deployed rapidly without an existing infrastructure. Unlike fixed wired network, wireless ad-hoc networks may have many operational limitations such as the transmission range, bandwidth, and energy. Additionally, wireless ad-hoc networks are vulnerable to more threats than those observed for the wired network, due to the dynamic nature of the routing infrastructure and the mobility of nodes. Applications of ad-hoc networks are emerging tremendously. New applications are nowadays getting more interest including target sensing, tactical battlefield and GRID computing.

Abdelkader M., Boudriga N. and S. Obaidat M. (2006).
SAGA: AN ADAPTIVE INFRASTRUCTURE FOR SECURE AVAILABLE GRID ON AD-HOC NETWORKS.
In *Proceedings of the International Conference on e-Business*, pages 22-30

Different features and challenges are introduced by the deployment of a GRID system over wireless ad-hoc networks. Among these features, one can consider the provision of protection to the whole GRID structure, the completion of the GRID execution, and the need for authentication and access control to resources, processes and messages involved in the GRID execution. In fact, to ensure distributed resources provision, different nodes in the wireless ad-hoc network should contribute, in the presence of the fact that each node of this structure does not have to know (or communicate directly with) all the other participants in the service provision.

Typically, wireless ad-hoc networks raise additional challenges to the provision of intrusion tolerant GRID systems, for which the effectiveness of wired solutions can be limited. To access a GRID service, a node needs to be under the coverage of an access point. In addition, a node agreeing to participate in a GRID service needs to stay connected until the service has terminated; otherwise a procedure for its replacement should be implemented. This is induced by the fact that a GRID architecture may vary in terms of time, location, and even availability. Security mechanisms must be deployed in order to counter threats against GRID over wireless ad-hoc networks. While cryptography-based mechanisms provide protection against some types of attacks from external nodes (with respect to the GRID service), they cannot be able to protect against malicious internal nodes, which may already have legitimate cryptographic keys. Therefore, mechanisms are necessary to provide intrusion tolerance for GRID applications.

The work presented in this paper aims at developing a general framework for the implementation of Grid applications in ad-hoc networking. The framework provides a generic extension mechanism for integrating multi-level trust management, QoS, and intrusion tolerance functionalities into Grid applications and handle variations in topology and availability. It consists of three models: a) a resource management scheme, which is responsible for resource description, request handling, and service continuity; b) an intrusion tolerance scheme; which integrates a scheme for event passing, a model for event correlation, and an alert notification procedure; and c) an accounting scheme, which includes the definition of a third party role, payment authentication, and payment processing.

We have addressed, in (Abdelkader and Boudriga, 2005), the design of a GRID architecture on ad hoc networks, which includes service discovery, service request and service allocation. In addition, we have introduced the notion of real-time control and management of trust in ad hoc nodes. This work can be considered as a first step in the development of SAGA. In the present paper, we extend this architecture by addressing other issues for the GRID service provision, the availability of GRIID services and the tolerance to attacks and failure. Considering the high variability of ad hoc topology, we also discuss the role of rescue plans to ensure GRID service provision continuity.

The remaining of this paper is organized as follows. Section 2 provides a definition and architecture for the GRID system. Section 3 develops the main characteristics of SAGA service continuity and system flexibility to cope with ad-hoc variability and node autonomy. Section 4 defines an approach to integrate intrusion tolerance capabilities in Grid computing systems and the management of multi-level trust. Section 5 discusses an application of SAGA to micro-payment environment. Finally, section 6 concludes this paper.

2 ADAPTIVE INFRASTRUCTURE

Resource and connectivity protocols facilitate the sharing of individual resources in Grid systems. These protocols are designed so that they can be implemented on top of a large spectrum of resource types defined at a Grid layer, called Fabric layer (as depicted by Figure 1). They also can be used to construct a wide range of services and applications. Figure 1 depicts a layered Grid architecture for ad-hoc networks and its relationship to the Internet protocol architecture. Our architecture presents some similarities with the one discussed in (Abdelkader and Boudriga, 2005), but it builds a number of useful services for GRID continuity and protection.

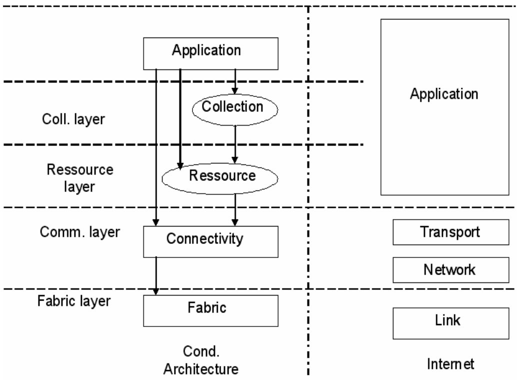

Figure 1: Layered Grid architecture.

The *Grid Fabric layer* provides the resources to which shared access is mediated by Grid protocols. A resource may be a logic entity, a storage resource, a network resource or a computational resource. The Fabric layer provides a resource-specific description of capabilities such as: (a) mechanisms for starting, monitoring and controlling of the resulting processes and controlling the resources allocated to these processes; (b) mechanisms for storing and retrieving files and (c) management mechanisms that allow control over resources allocated to processes and data transfers.

The *Communication layer* defines the communication and authentication protocols required for Grid-specific transactions. While communication protocols enable the exchange of data between fabric layer resources, the authentication protocols build on communication services to provide security services, such as authentication and integrity of users and resources and tolerance to intrusions. The communication layer should provide mechanisms for delegation, integration of local security and trust management.

The *Resource layer* builds on top of the connectivity layer for the secure negotiation, initiation, monitoring, accounting and billing of sharing operations on individual resources. Therefore, resource layer protocols are concerned entirely with individual resources and ignore issues of global state and atomic actions. Examples of resource layer protocols include information protocols, which collect information about the structure and state of a resource, and management protocols, which are used to negotiate access to shared resources while specifying resource requirements and the operation to be performed.

The *Collection layer* contains protocols and services that are able to capture interactions across the collection of resources. Example of services include (but are not restricted to): (a) the directory services that allow Grid users to discover resources; (b) the brokering services that allow users to request the allocation of one or more resources and the scheduling of tasks related to these resources; (c) software discovery services that help discovering and selecting execution platforms (or nodes) based on user/application parameters and (d) collaboration services that support accounting GRID services.

In Grid systems with distributed resources and task ownership, it is important to consider quality of service and security while discovering, allocating, and using resources. The integration of QoS has been examined with resource management systems by different studies. However, little work has been done for the integration of security considerations. Most cases have assumed that security is implemented as a separate subsystem from the Grid and the resource management system.

In a previous work, (Abdelkader and Boudriga, 2005), we have developed a scheme to search and use resources and access a GRID application. In particular, we have demonstrated that after finding the resource responding to the node requirements on security and QoS, the requester delegates to this resource the rights to use other resources that may be needed during service provision. In this section, we recall the major features of this scheme and extend it to provide an adaptive behaviour that takes into consideration the variability of network topology, autonomy of nodes and security requirements.

Figure 2 depicts a Grid service setup. Three tasks are basically involved in this process:

1. A node requesting a Grid service discovers the ad-hoc nodes that are able to allocate tasks and resources to establish the desired service.
2. Upon receiving the request, a node willing to be involved in the Grid service answers the request by sending a response specifying the accepted tasks, the amount of resources it can allocate, the security level of the process (engaged in that node), the cost and whether the node will act as a service provider or service operator.
3. On receiving the responses, the requestor selects the set of nodes that will be engaged in the provision of the grid service. A negotiation may take place between the requestor and a respondent before completing the service established. The negotiation involves QoS parameters, resources parameters and security services.

Features of the aforementioned process include the following three items:

- A service operator is a node that is in charge of offering the service using its own resources and the resources it can request on behalf of the requester. Therefore, the original requester does not need to know the identity of nodes involved in that share. In this case, the service operator is called delegated requestor.
- A service provider designates a node that acts as a server. It allocates the resources needed to the contracted tasks. It can leave the Grid on a simple message informing the requestor of its leave. It also can be dropped from the established Grid for various reasons, including security needs or renegotiation.

- Nodes contributing to a grid service are autonomous in the sense that they act as ad-hoc network node. They can move out of radio coverage, power off, or be attacked.

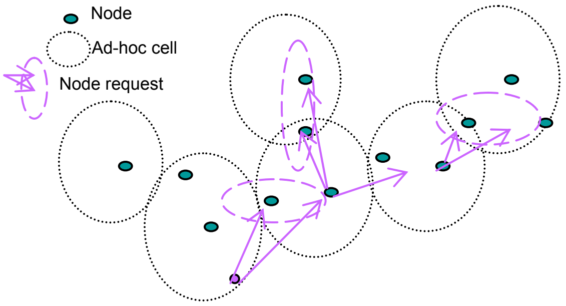

Figure 2: Grid application on Ad-hoc network.

Illustrative example: An application of Grid can be demonstrated by a micro-payment system for packet service delivery similar to the one provided in (Tewari and O'Mahony, 2003) and which has the following properties:

- The application is only employed for paying packets forwarding: a node desiring to send a set of packets to a given set of destinations can request the delivery of these packets to other nodes that are known to be on the routes to destinations (neighbours or close cluster heads).
- Each node involved in the delivery has a prior knowledge of the routes that the packets it sends should follow.
- Each node should be able to know the costs of all paths leading to destination and details of costs for each intermediate nodes (or at least the costs to pay to the next nodes involved in the delivery).

3 ADAPTIVE GRID BEHAVIOUR

To fulfill its objectives, a Grid service has to cope with the ad-hoc topology variability, node autonomy, and security intrusiont. The continuity of Grid service and the guarantee of the offered quality of service should be maintained as long as it is needed; otherwise correcting measures should be taken to correct any deviation. Measures fall into two classes: predictive and reactive.

A. Predictive Infrastructure Modification

When a server *S* (service provider or service operator) realizes that it cannot continue acting within a Grid to a requestor *R*, it starts searching for a replacement resource that can replace it and provide the remaining part of its agreement with *R*. This can be done as follows:

Resource reservation: To perform the replacement task, *S* uses the delegation firstly given by *R*. It starts by negotiating with the servers it has involved in its offers to *R*. Let S_m, $1\leq m\leq n$ be these servers; S checks with resources S_m, $1\leq m\leq n$, whether one of them can fulfil the QoS agreement of the Grid service provided to *R*. If there is a server, say S_m, *S* presents to S_m the tasks required by the client *R*, the state of their execution, the remaining jobs and the related QoS information including the period of time *t* after which *S* will make the transfer to S_m , and S_m reserves the necessary resources to be used after *t*. Then, S gives S_m the list of the other resources participating to the execution of the job J_R requested by *R*.

Announcement: After that, *S* informs the resources involved with it about their new manager S_m to which they should send the results related to J_R after *t*. On the other hand, *S* informs its cluster head CH_S and the cluster head of *R*, CH_R, about this modification. CH_S liberates the connection between S and *R*. However, CH_R starts building a new route between *R* and the new server S_m. Further, the cluster head of S_m, $CH_{S(m)}$, starts to establish new routes between S_m and the other elementary resources. At the end of this phase, one can say that the 3-tuple $(R, S, \{S_m, 1\leq m\leq n\})$ is replaced by the 3-tuple $(R, S_m, \{ S_i, i\neq m \text{ and } 1\leq i\leq n\})$ in the Grid service.

Job transfer: Before making the transfer to S_m, *R* generates a new delegation credential to S_m to be used after time *t*. In this credential, *R* allows S_m to execute job J_R on its behalf and search for the required resources. When *t* expires, *S* transfers to S_m all the information and the results related to J_R. If S_m finds that it needs more resources, it uses the delegation credential given by *R* to search them. Thus, the job transfer is done from *S* to S_m without disturbing the execution of J_R and with respect to the QoS required by *R*.

B. Reactive Infrastructure Modification

The interruption of the GRID service provision may occur suddenly. Neither the resource, nor the requester has prior knowledge about this interruption. This case is critical and happens after a node crash or a damaging attack on a node involved in the Grid service provision. To handle this situation, appropriate mechanisms need to be made available (at the cluster heads) to discover and handle any disconnection at any step of Grid service provision period.

To react to sudden modifications, the cluster observing a disconnection of a server S should inform the sources requesting a Grid service to S that S became unavailable. This can be performed because a cluster head acts as a router in ad-hoc networks. It also can maintain the routes established by nodes belonging to its cluster. However, since it cannot distinguish between the users and the data transmitted on the different routes, we propose the introduction of a new field in the header of IP packets, called *Grid-index field*, to distinguish between routes used in GRID applications. In fact, each application would be characterized by a unique number A_i that should be maintained on all the routes related to the same application. This field is written by the first node initiating a GRID application and is maintained on all the routes to GRID servers.

Using this field, one can select an access to a GRID application by the IP address of the source, the IP address of the destination and the value A_i contained in the new field. Therefore, every cluster head can select and manage different groups of routes where each group is attached to a GRID application and contribute to the reaction to disconnections as follows:

Announcement: When server *S* stops suddenly to offer a GRID service to a requestor *R*, the first node that can notice the interruption would be the cluster head of *S*, CH_S. In addition, CH_S can distinguish the different GRID applications to which *S* is involved by using its routing table. After CH_S has realized the unavailability of *S*, it extracts from its routing table the different addresses of the nodes present in routes related to job J_R. Then, CH_S informs all the actors involved with S in the Grid service provision that S (let us call these servers again S_m, $1 \le m \le n$) is no longer available so that they can they can suspend the execution of tasks related to J_R during a period τ and free their resources to use them for other purposes during a period of time of length τ. This allows resources exploitation ever when S is unavailable. Finally, CH_S informs the requester R about the abrupt interruption.

Information collection: In this phase, CH_S requests from all servers S_m, $1 \le m \le n$ reports on the usage related to the Grid service involving *S*. A report should contain the nature of the task, the state of the execution, the intermediate results and the remaining tasks scheduled, if needed. Each report should refer to the server responsible of the execution of job J_R. *S* then collects all the reports and sends them to the immediate requester *R*.

Resource discovery: When receiving the intermediate reports, *R* should start a new request for new nodes (and resources) that are able to replace S and continue J_R execution. *R* begins by communicating with servers S_m, $1 \le m \le n$. It starts with the resource that may provide a better QoS in a secure manner (Abdelkader and Boudriga, 2005). If one among servers S_m accepts the replacement, *R* sends it the set of collected reports execution reports. In addition, *R* generates a new delegation credential allowing S_m to start service recovery. If no node in $\{S_m, 1 \le m \le n\}$ is able to accept the request, *R* can get back to the selection phase of *S* and asks whether one node among those competing with S can still handle the replacement, otherwise, it restarts the discovery process with the initial request published during the selection phase of S.

In the case where an agreement is concluded, R presents to the new server the different intermediate reports related to J_R, if a server S_m is selected, or the initial request, if no S_m is willing the replacement.

Service recovery: Three cases should be considered:

1. If one server S_m is selected for the replacement of *S*, the recovery process described in the preventive case is applied. In this case, the 3-tuple $(R,S,\{S_m, 1 \le m \le n\})$ is replaced by the 3-tuple $(R, S_m,\{S_i, i \ne m \text{ and } 1 \le i \le n\})$ in the Grid service, and new routes are built.
2. If a server, say *S'*, that has competed with S is selected, the procedure used in the first case is involved, provided that *S'* plays the role of S_m. In this case, the 3-tuple $(R,S,\{S_m, 1 \le m \le n\})$ is replaced by the 3-tuple $(R, S',\{S_i, 1 \le i \le n\})$

3. If a new server T, servers S_m, $1\le m\le n$, are dropped and the 3-tuple $(R,S,\{S_m, 1\le m\le n\})$ is replaced by the 3-tuple $(R, T,\{T_i, 1\le i\le t\})$ and new routes are built appropriately.

By receiving the identity of the new job manager, the servers involved in cases a) and b) give the priority to suspended tasks and continue their. In case 3, the server delete all computation made for the Grid service. We should note, however, that service recovery is only possible before the expiration of τ and the release of resources immediately after τ has expired, if no resume is engaged.

4 SECURE INFRASTRUCTURE

In this section, we show what makes the presented infrastructure secure. In fact, security mechanisms integrated in our solution protect GRID applications from different types of attacks such as those related to integrity violation, denial of service, and node defacement. However, the intrusion tolerance provided with SAGA guarantees robustness against a set of attacks that should be maintained.

A. Security Provision

Since the integration of digital credentials such as the X.509 public key certificates ensures mutual authentication between ad-hoc nodes, provides efficient digital signature, and protect the message exchange related to Grid service provision, SAGA assumes the nodes contributing to Grid services have digital credential allowing them to authenticate each other, provide confidentiality and protect ad-hoc nodes. Therefore, integrity of transmitted messages is ensured. Furthermore, a delegation service is guaranteed through specific credential definition. A delegation credential specifies identities of delegating and delegated nodes, delegated rights, and delegation validity. It may also address information related to accounting. Protection of delegation credentials is ensured by a digital signature. This allows the verification of authenticity, integrity and non repudiation of the delegated rights.

On the other hand, Intrusion tolerance in SAGA is guaranteed through the use of:

1. Preventive mechanisms that are based on the notion of trust level, which classify applications according to their requirements on security and behaviour. The management of trust guarantees that requesting nodes (to access a Grid service) are authorized only when they present a level of trust higher or equal to the level required to access the service
2. Detection mechanisms that allow the employment of cooperative local intrusion detection systems (IDS), which detect appropriate events and exchange security alerts
3. Recovery mechanisms allow dynamic trust management which is responsible for maintaining the trust level of node according to their behaviour and the attack they are subject to.

The basic idea behind the trust management assumes that the trust level of a node (or application) seen by a second node is initially defined by the digital credential of the node and can be decreased with the reporting alerts.

B. Protection Against Denial of Service

The deployment of an adaptive GRID on ad hoc networks introduces new types of attacks. Those attacks are related to the nature of information and messages exchanged between nodes participating in GRID service provision. Among these attacks, one can mention the denial of service attack (DoS) that can target the adaptive GRID. The following special attacks are important to protect SAGA against:

Erroneous-Alert attacks: These attacks are launched to force reactive infrastructure modification. Such an attack operates as follows:

- A hacker interrupts all the messages transmitted between the server *S* and its cluster head CH_S during a period of time lasting sufficiently long.
- After a certain period of time, CH_S realizes that *S* is no longer reachable and begins the reactive procedure which induces the abortion of the GRID jobs handled by *S* and the waist of resources on the ad-hoc network.

This attack induces the augmentation of the execution delay of a GRID service. When repeated on different servers, these attacks might generate a distributed denial of service and endanger the system availability. Different protections can be used against these attacks. Protections may include; but are not limited to: (a) the duplication of the role of CH_S; (b) an effective control of the reachability of *S* (CH_S asks the remaining nodes of its cluster to try to reach *S*, for example) and (c) keeping the role and communication made by *S* confidential. In fact, when the reachibility of *S* is under investigation,

CH_S should not generate the alerts related the needed reaction, it saves the reports and messages to deliver to S until investigation completion and delivers them appropriately.

Alert-Absence attacks: The attacks target the reactive infrastructure. They operate as follows:

- When a server *S* involved in a Grid service is no longer available, its cluster head CH_S generates an alert to inform elementary resources to suspend the execution of jobs managed by *S*.
- A hacker interrupts the transmission of the alert. The resources involved with *S* (*S* is a service operator) continue to be reserved for *S* although *S* is no longer available.

These attacks may induce several damages including: (a) useless resource locking and other jobs are delayed; (b) loss of intermediate reports that continue to be sent to CH_S where they are deleted or ignored and (c) unacceptable overhead for the network and all the nodes contributing to the Grid service routing.

A protection against these attacks aims at imposing that every resource involved with server *S* should receive an acknowledgment from *S* after sending an intermediate report. Those acknowledgments should be protected and should refer to the report they are acknowledging.

Service Interruption Attacks: These attacks can target the predictive and reactive infrastructure modification. They aim at interrupting the initialization of a Grid service, blocking the communications between a service operator and the resources (or nodes) $\{S_i\}_{1\leq i\leq n}$ involved with it, replaying sensitive messages, or modifying the value of the new Grid-index field that we have added to the packet header (see Section 2). Regarding the latter attack, one can perform it as follows:

- A hacker can distinguish the application that has a Grid-index A_i. Then it can interrupt all the communications related to this application by modifying the Grid-index field content.
- It also can copy all messages, reports and transported results related to the application associated with index A_i.
- The hacker also can change the index A_i and so neither S nor the nodes $\{S_i\}_{1\leq i\leq n}$ can reach each other. This induces the starvation of the GRID application and the loss of network resources.

To protect against these attacks, various mechanisms can be added including the protection of the integrity of the Grid-index content, the insertion of protected nonce (a sequence number or time stamp) and IPSec.

5 SAGA APPLICATION

In this section we illustrate the use of the scheme presented in the previous sections. We consider an interesting domain of applications that can take place on an ad hoc network. This is a distributed application, referred to as micro-payment on ad-hoc environment.

In (Tewari and O'Mahony, 2003), authors propose a protocol employing micro-payment techniques. The protocol allows each ad hoc node involved in packets relaying to be paid by the sender as it provides the service. It allows paying all nodes in the path to a given destination without the requirement to contact a third trust party or a bank to issue a new payment contract. The main steps of this protocol are described as follows:

1. A user buys prepaid tokens through his/her terminal from a broker whose main purpose is to aggregate micro-payments between entities. The user starts by generating an unbalanced one-way binary tree and sends the set of *N* defined anchors to the broker.
2. The broker generates a set of *N* secret endorsement values; one for each anchor value that was sent by the user. A broker endorsement consists of an anchor value, a random number corresponding to the endorsement value, the length of the hash chain, the value of a hash in the chain, the identity of the user that purchased the chain and the expiration date of the chain.

All of the above fields are signed with the private key of the broker. The Grid service associated with this application assumes the following:

- When it is desired to set up a call to a remote destination (or asset of call to remote destinations). The user should have knowledge of the total costs involved in forwarding the related packet through the ad hoc network. Each node in the path to destination(s) must indicate its charge for packet forwarding.
    - With every packet or message sent by the user, the user should attach the cost of transmission. Every node in the established route extracts the value of the cost required. The unit defined to pay the different nodes is a single hash token. Thus, every node presents the number of tokens it wants for the forwarding.

The integration of this application in a Grid infrastructure should introduce some modifications for the sake of flexibility. In fact, we propose the integration of this application to pay the use of different resources used in a Grid application (i.e., Grid accounting). Modifications assume that the requester *R* knows only the set of immediate servers $\{S_i\}_{1\leq i\leq n}$ and it is not supposed to identify the set of the other nodes contributing to the execution of the Grid service. In addition, a Grid application can offer other services than packets forwarding, which means that every node should determine dynamically the cost that it requires to contribute to the Grid service.

To manage the new assumptions, we assume that a server S_i (e.g.;, service provider) is responsible for paying for the resources,$\{S_{i,j}\}_{1\leq j\leq ni}$, that it can get involved in. In fact, after determining the set of resources with which S_i will collaborate, S_i asks for a total cost including the use of all resources needed to perform the tasks it is assigned. S_i collects all the costs and adds them to its local costs. The total cost is then sent to the applicant *R*. Based on the delegation credentials initially generated and the trust levels of *R* and S_i. *R* presents an appropriate set of tokens to S_i. These tokens are encrypted with the public key of S_i. Then, S_i manages these tokens to cover its own cost and the costs of $S_{i,j}$, for $1\leq j\leq n_i$. When paying a contributing $S_{i,j}$, S_i should encrypt the required tokens by the public key of $S_{i,j}$. In addition tokens should no longer go together with each packet.

We assume that after collecting the tokens corresponding to a service provision, a server presents these tokens to the broker which is responsible for concluding the payment. Before making the transfer, the broker should wait for a period of time. This period is fixed by the administrator allowing the reception of any objection. In fact, if a node takes the tokens without achieving the tasks for which it was paid, the payer could protest against concluding the payment. If this period of time expires without receiving any objection, the transfer is concluded and the server is paid.

Finally, new security features should be considered according to the SAGA model. They include:

- In the case of predictive modification, before a server *S* withdraws, it should pay all the contributing resources for the jobs they are involved with *S*. Furthermore, *S* should return the unused tokens to requester *R*. The latter will give other tokens to the new server *S'*. The operation of giving new tokens can be kept as it was defined in (Tewari and O'Mahony, 2003).
- In the case of reactive modifications, the contributing resources $\{S_i\}_{1\leq i\leq n}$ to a withdrawing server *S* should inform R about the last payment they obtained. *R* informs the broker to revoke the remaining tokens and use new tokens to continue GRID service provision. In this case, we believe that there is a need to tolerate very limited token losses since the applicant can not define precisely when *S* was gone away. Two reasons can justify our tolerance. This assumption can be made since micro-payment is a field where such assumptions are accepted, when they are limited. In fact, this kind of payment reduces the possible losses since it divides the payment amount into small values. In addition, the trust management provided in SAGA can require that a node, which does not conclude its agreements without signalling it, will see a decrease in its trust level. This will impact its further works.

6 CONCLUSION

In this paper, we addressed the issue of adaptive behaviours in a Grid service provision. Our approach builds systems called SAGA that are able to define a framework for the design of Grid application that are secure, tolerant to intrusion and can cope with the variable nature of ad-hoc networks.

The framework provides a generic extension mechanism for integrating multi-level trust management, QoS, and intrusion tolerance functionalities into Grid applications and handle variations in topology and availability.

REFERENCES

Casanova, W. Cirne, H. Dail, M. Faerman, S. Figueira, J. Hayes, G. Obertelli, J. Schopf, G. Shao, S. Smallen, N. Spring, A. Su, and D. Zagorodnov, *Adaptive computing on the Grid using AppleS*, IEEE Transactions on parallel and distributed systems, Vol. 14, No. 4, pp. 369- 382, Apr. 2003

S. Song and K. Hwang, *Dynamic Grid security with trust integration and optimized resource allocation*, Int. Symp. On High-performance distributed computing, Honolulu, June 4-6, 2004.

A. Butt, S. Adabala, N. Kapadia, R. Figueiredo and J Fortes, *Fine-grain access control for securing shared resources in computational Grids*, 2002 Proc. Int.

Parallel and distributed processing symposium (IPDPS'02), pp. 159-165, April 2002,

F. Azzedin and M. Maheswaran, *Towards trusted-aware resource management in Grid computing systems,* Proc. 2nd IEEE/ACM Int. Symp. on cluster computing and the Grid, 2002.

K. Czajkowski, I. Foster, and C. Kesselman,, "*Resource co-allocation in computational Grids"*, Proc. 8th IEEE Int. Symp. on high-performance distributed computing (HPDC-8), pp. 219-228, 1999.

D. Gefen, V.S. Rao, and N. Tractinsky, "*The conceptualization of trust, risk, and their relationships in e-commerce"*, Proc. 36th Hawaii Int. Conf. on System science (HICSS'03), 2003.

I. Foster, C. Kesselman, and S. Tuecke, "*The anatomy of the Grid: Enabling scalable virtual organizations"*, Int J. Supercomputer applications, 2001

Manel Abdelkader, Noureddine Boudriga. "Intrusion Tolerant GRID in Ad-Hoc Networks", 12th IEEE International Conference on Electronics, Circuits and Systems, ICECS 2005, Tunis, Dec. 11-14, 2005.

Rui Liu, and Errol L. Lloyd, "*A Distributed Protocol For Adaptive Link Scheduling in Ad-hoc Networks*", Proc. of IASTED Int. Conf. on Wireless and Optical Comm. (WOC2001), 2001.

Hitesh Tewari, Donal O'Mahony, "Multiparty Micropayments for Ad Hoc Networks", IEEE Wireless Communications and Networking Conference, WCNC 2003.