An IP-video Surveillance Indoor System Using Mesh Networks

Pablo Corral[1], Ricardo García[1], Antonio Molina[1], A. C de C. Lima[2]
Miguel Hernández University, Signal Theory and Communications, Elche, Spain[1]
Federal University of Bahia, Electrical Engineering Department, Salvador, Brazil[2]
pcorral@umh.es, r.garcia@umh.es, antonio.molina02@alu.umh.es

***Abstract*- IP-Video surveillance is a term used for a security system that allows the users to view, control and record video images. In its simplest way, it is composed of a camera and a computer, although most applications require more network devices. Video surveillance is being used in a lot of environments (banking, transportation, industry...) and each market has their own needs. In this paper, we describe a network using IP cameras with mesh networks 802.11a/b/g. To control the security system we develop an application in order to adapt the quality of image to the number of hops in our network.**

I. INTRODUCTION

A surveillance system is used to monitor the behaviour of people, objects or processes within systems for conformity to expected or desired norms in trusted systems for security or social control. There are many ways to monitor this behaviour, but most of them are very expensive, compared to IP solutions. Surveillance video systems exist since last twenty five years. At the beginning, they were completely analogical systems and since then, the process has turned to digital.

Nowadays, the market offers different network solutions, depending on if we are dealing with a new infrastructure or if we are migrating from an analogical system. The first one is the simplest one, but it is necessary to take into account the necessity of migration.

At present, the most appropriate solution to create a surveillance system is based on an IP camera. This device is composed of two main parts: a camera and a computer. On the one hand, the camera is just an optical sensor which converts the light into an electrical signal. On the other hand, the computer gets that electrical signal as an input and returns a digital and compressed image.

The operating system running on the computer, has several network applications installed, such as a web server, an ftp server or a mail client. In that way, it is possible to view the compressed images (real time video) remotely from a simple web browser.

Apart from web browsers, there are other ways to view and monitor the surveillance system. The advantage of developing new software is that it can be adapted to the requirements of the users. In this way, digital image processing and computer vision techniques can be applied. Alarms, viewing and network options can be configured as wished. The manufacturers software solution can be appropriate in most cases, but if a professional surveillance is desired, it is recommended to create a personalized software.

Surveillance systems have become essential nowadays, although they exist since some time ago. The evolution of these systems could be compared to the evolution of the telephone system.

At the beginning it was completely analogical. The cameras were analogical, the cables were analogical and the monitors (viewers) or the video recorders were analogical, too. This fully analogical system is known as a closed-circuit television (CCTV).

The first step was to solve the problem of storage. The analogical data is much more difficult to store than digital data. Because of that, a Digital Video Recorder (DVR) is used, which replaces video tapes by hard disks.

The next stage was the use of a video server. In this case, analogical cameras are still used, but immediately it is applied a digital conversion. Most video servers have several analogical inputs to connect more than one camera. This option can be a good solution in case of migrating from a previous analogical system.

The analogical cameras can be used and accessed remotely as if they were IP cameras.

The outline of this paper is as follows: in section II, we describe our wireless network 802.11a/b/g using different radio links depending on the type of traffic. Network topologies used in wireless LAN are introduced in section III. Section IV presents the characteristics of digital IP cameras. In Section V we show the surveillance application developed. In the last section the results of the complete system are presented, and conclusion of the performance is drawn on those results.

II. Wireless Networks

The interest in indoor wireless communications has grown rapidly because of the advantages over cable networks such as mobility of users, elimination of cabling or flexibility [1]. In our case, the wireless mesh implemented must be a low latency network that dedicates separate wireless bandwidth links to ingress an egress backhaul traffic (similar to a full duplex connection) and utilizes the highest available throughput automatically. It must be:

- Multi-radio link (the first one for wireless client traffic, the second one for uplink wireless backhaul traffic, and the last one for downlink wireless backhaul traffic).

- Multi-channel (using only three possible non-overlapping channels in 802.11b/g, in contrast 802.11a does not have this limitation, for this reason is used for the backhaul mesh infrastructure formation).

- Multi-RF (working in 5 GHz band with 802.11a or 2.4 GHz band with 802.11b/g).

III. Network Topologies

Historically, there are three types of network topologies in wireless systems (see Figure1):

- A distributed or ad-hoc mode: without an access point (AP) in order to centralize the communications.

- A centralized or infrastructure mode: in this case, we strip the intelligence from the AP and put it all on the switch; however, this approach introduces many undesirable effects (a single-point-of-failure, bandwidth bottleneck…) [1].

- A mesh mode, which consists of an intelligent network where network nodes do not need to be wired to a switch since they can connect wirelessly among themselves via 802.11 links. With a wireless mesh, each node in the network establishes the optimal path to its neighbour. This is the topology implemented in our network [2].

Deploying a wireless mesh network offers several advantages over other types of wireless deployments. These advantages are mainly focused on reducing the cost of key factors in any network (as installation or maintenance needs).

Even though, there are many factors to take into account when applying a mesh deployment. On one hand, the transmission environment (air) is shared by all the users. It makes these networks more vulnerable than others. On the other hand, the bandwidth needed to transmit multimedia traffic is going to be a handicap for actual wireless networks, where transmission rates are much lower than using wire networks (54-108 Mbps vs. Gbps).

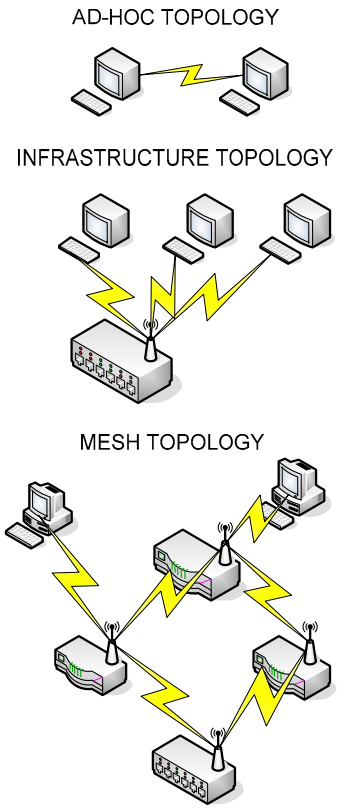


Fig. 1. Network topologies.

A. WLAN Security

The benefits from wireless networks also come with several security considerations. Security risks include risks of wired networks plus the new risks as a result of mobility. To reduce these risks and protect the users from eavesdropping, organizations have adopted several security mechanisms [3].

- WEP: As defined in [4], WEP is an optional encryption standard implemented in the MAC layers. The WEP algorithm prevents unauthorized access to the network relying on a RC4 secret key cryptographic algorithm. This key is shared between clients and access points. The WEP algorithm encrypts packets before they are transmitted and uses CRC-32 algorithm for data integrity.

- WPA: Wi-fi Protected Access is a system created to correct the failures from WEP. It includes better data encryption and wireless users can use passphrases from eight to 63 bytes.

- 802.1x: Although encryption is mandatory to provide security, it's not enough because it doesn't avoid intruders to access our wireless network. 802.1x offers a framework for authenticating and controlling user traffic, as well as dynamically changing encryption keys. When using 802.1x, it's necessary to choose an EAP (Extensible Authentication Protocol) type, such as TLS (Transport Layer Security) or TTLS (Tunneled Transport Layer Security), which define how the authentication takes place. A Radius server is essential.

B. Bandwidth degradation

Much research on network capacity has been done for mesh networks, where the traffic flows between pairs of nodes. In [5], the capacity is calculated adopting an asymptotic analysis with a large number of nodes.

The capacity for each node decreases as $1/\sqrt{n}$. In WMN, the traffic flows to or from a gateway, which act as a bottleneck. Due to the presence of this bottlenecks, the available capacity is reduced to $1/n$ where n is the number of nodes for one gateway.

In other words, each time the traffic hops from one node to another one, the throughput is almost cut in half. Clearly, the available throughput improves directly proportional to the number of gateways in the network [6].

C. Radio interference

One of the major problems facing wireless networks is the throughput degradation due to interference among simultaneous transmissions. In wireless mesh networks with multiple-radios the problem can be greatly alleviated, although it cannot be completely eliminated [7].

Understanding and characterizing such interference is important for a variety of purposes such as channel assignment, route selection, and fair scheduling. Although it is important, it is also infeasible due to the computational complexity involved.

Some protocols have relied on simplistic representations of interference with heuristic rules such as "everyone interferes with everyone else". Although simple to calculate, such heuristics can be far from the real interference.

There are other kinds of interferers which emit signals whose structure is very different from the desired signal. Bluetooth nodes, cordless telephones, security systems are examples of these interferers.

In addition, there are some electronic devices that can leak radio signals in the unlicensed band (for example, microwave ovens, computers, and mobile telephones) [8].

D. Latency

As a packet goes through the network form node to node, processing delays are naturally introduced. There are various causes of delay or latency that can affect the overall performance of the wireless network such as congestions, timeouts or retransmissions.

In a large wireless mesh network, also the capacity of self-tuning and self-healing comes to be a problem referring to latency. In [9], it is given a solution to reduce the time spent on establishing new routes. The solution lies in two points, accelerating the detection of fallen ports and reducing the time spent on recognizing the new route.

Two important real time applications are IP telephony and video conferencing. For these applications, the end to end delay of a packet should not exceed 150 msec.

If a packet exceeds this delay, it must be dropped. On the other hand, file transfers or web traffic have minimum time requirements. Providing priority to UDP traffic over TCP traffic delays can be drastically reduced [10].

In [11] is presented a formal study on the delay introduced by the gateway nodes using the M/D/1 queuing theory. The bottleneck delay is defined as the average delivery delay of data requests at the gateway nodes. The limited number of these nodes could be the bottleneck of the entire network.

IV. IP CAMERAS

Finally, if we are dealing with a new topology, we recommend using IP cameras (see Figure 2). The process is fully digital. The IP camera is connected to a network device in charge of routing the data to a PC with management software. The advantages of using IP cameras against others surveillance solutions are:

- High resolution cameras (mega pixel).
- Progressive scan against interlaced scan (improve movement detection).
- Remote accessibility.
- Cost-effective: IP systems are cheaper than CCTV systems.
- Power supply through the network cable.

Fig. 2. IP Cameras.

In this paper, we want to highlight the digital process advantage. The fact of dealing with digital video provides the opportunity to apply algorithms to analyse this video data. In older installations, security staff had to be continuously in front of the monitor to detect, for example, motion events. Actual systems can be configured to raise an alarm in case of motion detection.

V. Surveillance Application

Some IP cameras have an HTTP-based application programming interface (API) to encourage developers to create their own applications (see Figure 2). This API provides different functions to manage the camera. The surveillance application and the camera employ a client-server model, where both nodes create a bidirectional communication based on http requests and responses.

The real time video can be obtained in two ways. The first one deals with sampling the MJPG data stream and isolate the JPEGs. The second ones consists in requesting a JPEG periodically. In both cases, the data stream has to be analyzed, and that is the main goal of the surveillance application.

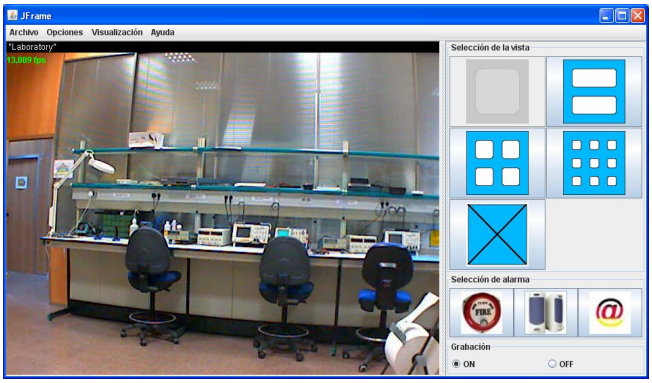


Fig. 3. Application's screenshot.

The standard JPEG indicates that the first two bytes of every JPEG stream are the Start of Image (SOI) marker values FFh and D8h. In the same way, the end of a JPEG image (EOF) is set by the bytes FFh and D9h. So, every time these values match the stream, a JPEG image is detected.

The JPEG image is going to be the most important element in our application. When buffering several images, real time video can be viewed or recorded. Our approach differs from traditional security systems in that it raises alarms when significant events are detected.

A. Digital Image Processing

In real time video applications, techniques for the manipulation of images have to be computed as fast as possible. It is going to be necessary to reach a compromise solution between the video quality offered and the image processing tasks. The following are the functions applied in our system:

- Image type conversions. It is not the same to process black and white images than RGB ones.
- Geometric operations, such as enlarging or shrinking the image.
- Extracting regions of interest. This technique stands out important regions of the image. Zooming is an example.
- Arithmetic and logical combination of images. The main application of image substraction is applied in change detection.
- Image correlation. Correlation is often used to measure the similarity between frames. Pattern recognition can be done with this method.

B. Alarms

Once the application carries out processing algorithms, some alarms can be configured when interesting events are detected. The most common event to detect is motion, although other kind of events can be programmed. For example, it is possible to set an alarm if a strange object appears in the scene and it doesn't move for a long time. Or another alarm could occur if light is switched on or off.

The alarms notification can be performed in several ways. The easiest one is that the surveillance application raises an audio alarm. In our application, it is programmed another kind of alarm, which sends an email with the reason of the alarm attached to it. The last alarm used depends on the IP camera. Some cameras have available output ports to connect external devices. In our case, an alarm bell is connected to the camera.

Fig. 4. Camera, bridge and alarm bell.

VI. System Surveillance

As it has been mentioned, using wireless networks provide a valuable solution for many users and for many reasons. In the same way, we have applied a mesh topology, because it provides simplicity to the user. Nodes have the ability to communicate with each other, so that the entire system becomes an intelligent network. Traffic is routed on optimal paths and it adapts to changing conditions. WDS[1] is the system that allows this operation. It can be referred to as a repeater mode because it appears to bridge and accept wireless clients.

Previously, it also has been explained some problems when using this kind of networks. Some of them, like security or interferences, can be lessened or even solved. Otherwise, bandwidth degradation is unavoidable in mesh wireless networks. Its value is proportional to the number of hops and each time the traffic hops from node to node, the throughput is reduced almost in half. As it was said in [chapter 3], the degradation factor comes to be 1/n, where n is the number of hops.

The bandwidth it is need for the multimedia traffic depends on some configuration factors of the IP cameras. These factors are:

- Image resolution, compression and Complexity.
- Frames per second.

The first three factors can be modified through our surveillance application, but the last one is referred to the compressing algorithm used by the JPEG standard. The complexity depends on the details of the image. The standard exploits that human eyes are more sensitive to slow changes of brightness and color than to rapid changes over a short distance.

In figures 4, 5 and 6, it is shown the bandwidth that an IP camera needs, depending on the configurable factors. About the image complexity, it is supposed that the camera is focusing a quiet place, such as a garage or a reception.

We can observe that the necessary bandwidth with the highest video quality is around 9 Mbps (figure 5). This value is too large for a wireless mesh network. One way to lessen this bandwidth is to use other image compressing standards, such as MPEG. But this matter is out of our scope. Anyway, if applying fewer frames per second or some compression, the bandwidth is reduced to 2 or less Mbps as it can be seen in the figure.

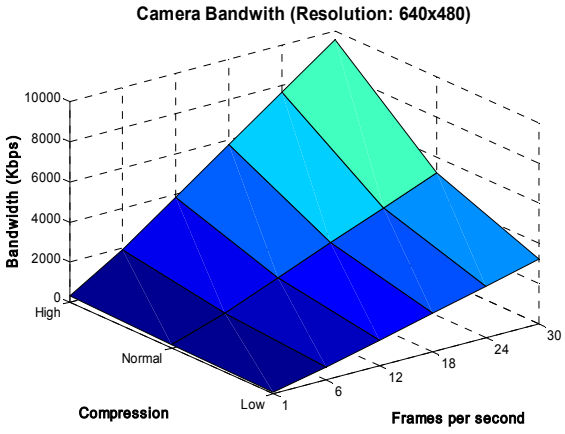


Fig. 5. Camera bandwidth with constant resolution.

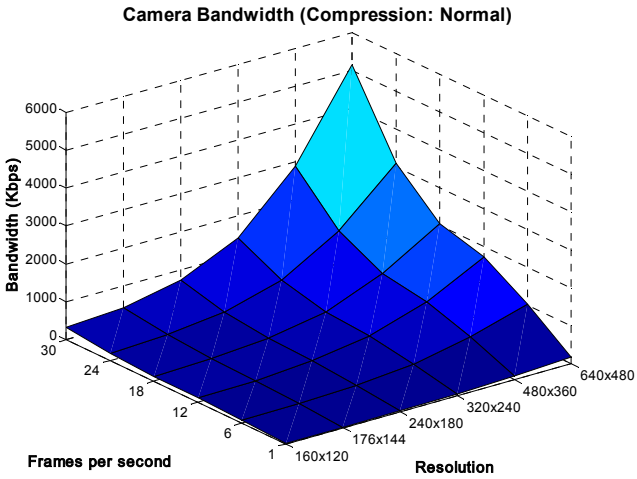


Fig. 6. Camera bandwidth with constant compression.

[1] Wireless Distributed System.

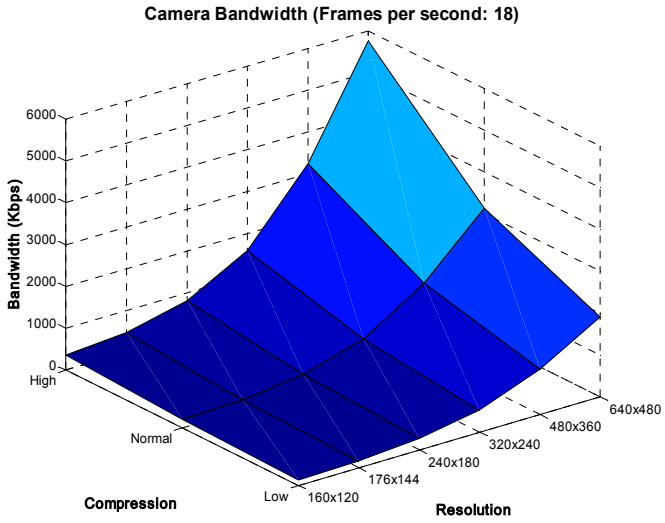


Fig. 7. Camera bandwidth with constant transfer rate.

Although the required bandwidth can be configured as wished, it is still very large in a wireless mesh deployment. In figure 7, it is shown the throughput degradation. It is important to note that it is not the same the throughput and the maximum data rate. The maximum data rate in 802.11a/g is 54 Mbps, although the throughput, which can be defined as the amount of information that can be sent under the stresses of a real communication, is much lower.

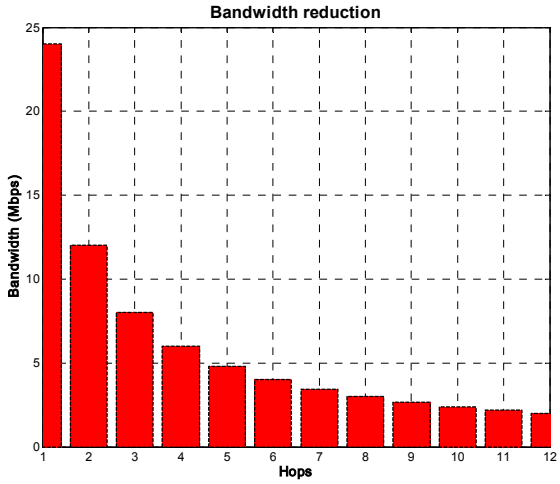


Fig. 8. Bandwidth degradation.

Comparing figures 5-7 with figure 8, we get to the conclusion that when configuring a camera with the highest quality, the maximum number of hops in the mesh network is two. On the other way, when using a normal configuration, the number of hops increases considerably. When applying the following configuration, much better results are obtained.

- Resolution: 320 x 240.
- Frames per second: 24.
- Compression: 50%.

In this case, the necessary bandwidth is about 1 Mbps. In this way, it can be developed for example a mesh surveillance system with 4 cameras, which can be located to a distance of 6 or less hops from the gateway.

A. Wireless bridge

A wireless bridge can be defined as an Ethernet-Wifi converter. This device provides with wireless capability another device connected through an Ethernet cable. In our system, we are using this device to connect it with the IP camera. In this way, we can get the video from the camera wirelessly.

Many actual cameras have already embedded wireless capabilities, but in our case, we are upgrading the surveillance system from a previous one. Using a wireless bridge is much more economical than getting new cameras.

As it has been mentioned, security precautions must be taken when connecting to a wireless network. The device must support different security levels and in our case, it is supported encryption and authentication. So, we are applying the most secure configuration, combining WPA and RADIUS (see Figure 8).

Fig. 9. Security configuration.

B. Radius server

Remote Authentication Dial In User Service is an AAA (authentication, authorization and accounting) protocol for applications such as network access. Our surveillance system pretends to be as secure as possible. For that reason, we are going to use a RADIUS server to provide access to the network. The operation is very simple, the server radius keeps a database with all the users which have permission to access the network and there is a bidirectional communication between the mesh nodes and the radius server.

RADIUS is extensible; therefore many vendors can implement their own hardware and software variants. In our system, we have used a free open source radius server called FreeRADIUS. It offers an alternative to other commercial RADIUS servers with high scalability. The system can run in a simple PC and manage thousands of users. It can perform authentications via the PAP, CHAP, MS-CHAP, EAP-MD5, EAP-GTC, EAP-TLS, EAP-TTLS, PEAPv0, LEAP, EAP-SIM, and Digest authentication protocols.

From the entire previous list, we have opted for the EAP-TLS protocol. Its main characteristic is that it uses certificates in both parts of the communication, in the server part and in the client part.

A certificate, contains information about the owner of the certificate, like e-mail address, owner's name, certificate usage, duration of validity, organization and the certificate ID. It contains also the public key and finally a hash to ensure that the certificate has not been tampered with. The certificates for the server and for the clients have been generated through OpenSSL. OpenSSL consists on a free software project that provides cryptographic functions to other packets.

The fact that in EAP-TLS it is needed to install a certificate in every client can be seen as something thorny, but then, if the client-side certificates are housed in smartcards, the only possibility to access the network is to steal the smartcard itself.

C. Mesh nodes

A fundamental aspect of the networks operation in mesh is that the communication between a node and any other can go beyond the rank of cover of any individual node. This is obtained making a routing multihop, where any pair of nodes that wishes to communicate will be able to use for it other intermediate wireless nodes that are in their way.

This is important if it is compared with the traditional networks WiFi, where the nodes must be within the rank of cover of a AP and they are only possible to be communicated with other nodes by means of the AP; these AP needs a twisted network as well to communicate to each other.

With the networks in mesh, it is not necessary to have AP, because all the nodes can communicate directly with the neighbors within their rank of wireless cover and with other distant nodes by means of the multihop routing already mentioned.

In this occasion we will focus to a well-known technology like Wi-Fi in mesh (Wi-Fi mesh networks) which defines in the IEEE specification 802.11s, which is in status of rough draft. Many manufacturers have begun to send products with proprietary technologies trying to seize as rapidly as possible of the market of radio networks, before the final specification of the standard is published.

The standard 802.11s is a proposal of the work group known like Wi-Mesh Alliance (www.wi-mesh.org). The rough draft of the standard 802.11s defines the physical layer and data link for networks in mesh.

This topology increases the cover of the network and allows to be always active even though one of the joining points fails. It is possible to add users and joining points to the network to gain capacity. In the same way, Internet also works in mesh. Adding nodes to the network is something very easy.

The standard offers required flexibility to satisfy the requirements with residential atmospheres, such as offices, campus, public security and military applications. The proposal focuses on multiple dimensions: the MAC sub-layer, routing, security and the interconnection.

The standard 802.11s defines only systems for atmospheres in interiors, but the main manufacturers of wireless equipment are also betting to him to outer atmosphere systems.

Finally, we can appreciate in figure 10, the whole surveillance system. In it, we can observe the three different subsystems:

- IP Camera and wireless bridge (Figure 4).
- Mesh network.
- Radius server.

VII. CONCLUSIONS

In this paper, we describe a network using IP cameras with mesh networks 802.11a/b/g. To control the security system we develop an application in order to adapt the quality of image to the number of hops in our network.

On the one hand, the system to provide access to the network carrying out the AAA protocol requirements is the RADIUS server, in our case an open source solution, called FreeRADIUS. The advantage is the use of non-commercial software but the disadvantage is the trouble configuring this server.

On the other hand, the maximum number of hops varies depending on the image quality from one camera with maximum quality (9 Mbps) which permits 2 hops, to 4 cameras with medium quality (1 Mbps per camera) which supports up to 6 hops.

Finally, we have described a wireless system which permits surveillance functionality without the limitations of the wireline systems.

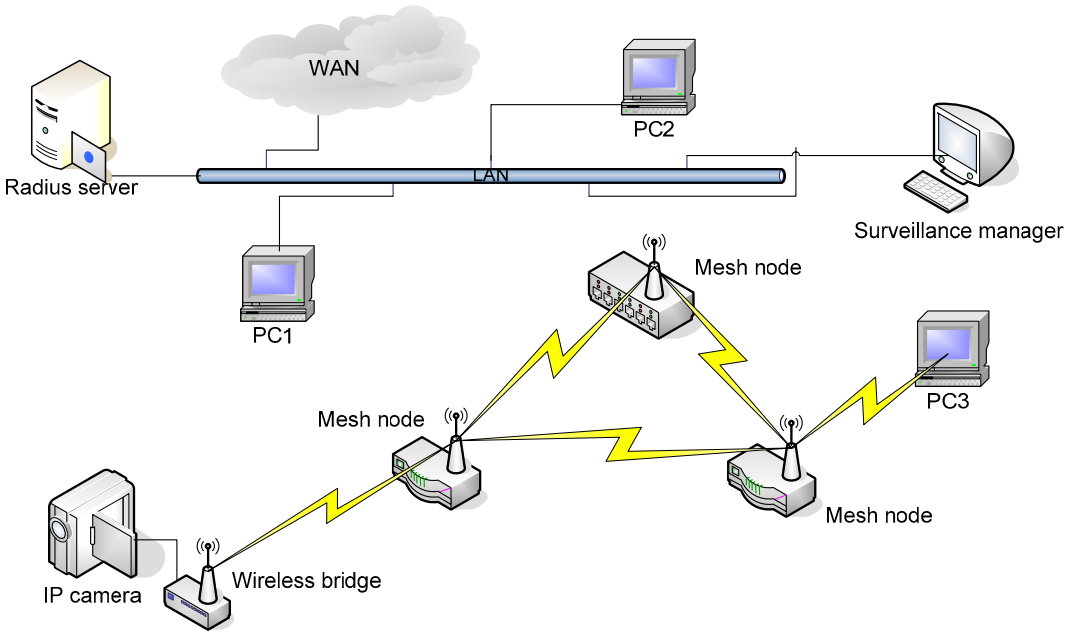


Fig. 10. Complete system surveillance.

ACKNOWLEDGEMENTS

The current project has been financed and supported by *Miguel Hernández University*.

REFERENCES

[1] P. Corral, M. Mompó "Diseño e implementación de una red inalámbrica basada en el estándar 802.11 de área extensa en la población de Montaverner", XX Simposium Nacional de la Unión Científica Internacional de Radio, Gandia (España), September 2005.

[2] Da-Ren Guo; Kuochen Wang; Lung-Sheng Lee, "Efficient Spatial Reuse in Multi-Radio, Multi-Hop Wireless Mesh Networks", Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th, 22-25 April 2007, pp. 1076 – 1080.

[3] G. Zeynep Gurkas, A.Halim Zaim, M. Ali Aydin, "Security mechanisms and their performance impacts on wireless local area networks", Computer Engineering Department, Engineering Faculty, Istanbul University, Avcilar, Istanbul - TURKEY, *zeynepg@istanbul.edu.tr.*

[4] Mrs. Megha Bone, "Wireless Security & Privacy", Cummings College of Engineering For Women, Karvenagar, Pune 41 152.

[5] P. Gupta and P. R. Kumar, "The capacity of wireless networks", *IEEE Trans. Inform. Theory*, vol. 46, pp. 388-404, Mar. 2000.

[6] J. Jun and M. Sichitiu, "The nominal capacity of wireless mesh networks", *IEEE Wireless Commun. Mag.*, vol. 42, no. 5, pp. 8-14, Oct. 2003.

[7] J. Jun and M. Sichitiu, "The nominal capacity of wireless mesh networks", *IEEE Wireless Commun. Mag.*, vol. 42, no. 5, pp. 8-14, Oct. 2003.

[8] M. Alicherry, R. Bhatia, L.E. Li, "Joint channel assignment and routing for throughput optimization in multiradio wireless mesh networks", Network Software Res. Dept., Lucent Technologies. Bell Labs., Murray Hill, NJ.

[9] Saumitra M. Das, Dimitrios Koutsonikolas, Y. Charlie Hu and Dimitiros Peroulis, "Characterizing multi-way interferente in gíreles mesh networks", *Proceedings of the 1st international workshop on wireless network testbeds, experimental evaluation & characterization*, Los Angeles, CA, USA, pp. 57-64, 2006.

[10] M. Portolés, J. L. Valenzuela, D. Pérez, O. Sallent, "Link recovery in IEEE 802.11 WLAN using WDS", Vehicular Technology Conference, 2004. VTC 2004-Spring. IEEE 59th, vol. 4, pp. 2239-2242, 2004.

[11] H. Shetiya, V. Sharma, "Algorithms for routing and centralized scheduling to provide QoS in IEEE 802.16 mesh networks", Dept. of Electrical Communication Engineering, Indian Institute of Science, Vol. 1, pp. 147 – 152, Sept. 2006.

[12] X. Wu, J. Liu, G. Chen, "Analysis of bottleneck delay and throughput in wireless mesh networks", Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference, pp. 765-770, Oct. 2006.

Performance Analysis of Total Cost on HMIP Parameters

Sapna Gambhir[1], M. N. Doja[2], Moin Uddin[3],

1. Deptt. Of Computer Engg. YMCAIE, Faridabad.sapnagambhir@rediffmail.com,

2. Deptt. Of Computer Engg., Jamia Milia Islamia, New Delhi

3. Dr. B. R. Ambedkar National Institute of Technology, Jalandhar

Abstract

Mobile-IP standard presents a fundamental solution for mobility management in the internet. MIPv6 allows a mobile node to transparently maintain connections while moving from one subnet to another. Hierarchical Mobile IPV6 (HMIPv6), an enhanced Mobile IPv6 protocol, provides a scheme for performing registrations locally in the foreign network, thereby reducing the number of signaling messages forwarded to the home network as well as lowering the signaling latency that occurs when a mobile node moves from one foreign network to another. In this paper, we analyze the performance of wireless/mobile networks with respect to signaling cost in multiple levels HMIPv6. Signaling cost consists of cost of location update and packet delivery cost. Location update cost is the cost for global binding update and local binding update. Packet delivery cost is comprised of the processing cost and transmission cost. We study the impact of various parameters viz. MAP domain Size, Cell Residency time and the mobility nature of MNs i.e. Static and Dynamic MNs on the total cost

Keywords-**HMIPv6, MAP, MN**

1. INTRODUCTION

The demand for mobile service has motivated research in updating existing high speed wire line (fixed) communication networks with wireless networks. In wireless/ mobile networks, users freely change their point of attachment while they are connected. In this environment, mobility management is an essential technology for keeping track of users' current location and for delivery of data. Thus mobility management [1] [2] supports mobile terminals and enables communication networks to:

1. Locate roaming terminals in order to deliver data packets.
2. Maintain connections with terminals moving in to new service area.

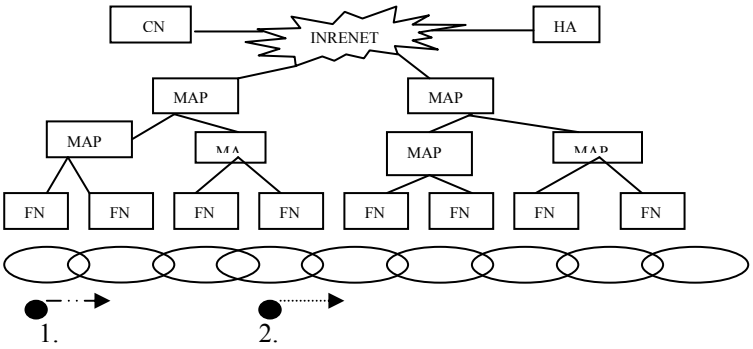
The Mobile IP working group with in the Internet Engineering Task Force (IETF) proposed a mobility management protocol, called Mobile IPv6 (MIPv6) [3] [4]. MIPv6 was an update of the Mobile IP standard designed to authenticate mobile devices using IPv6 addresses. MIPv6 allows a mobile node to transparently maintain connections while moving from one subnet to another. Each device is identified by its home address although it may be connecting to it through another network. When connecting through a foreign network, a mobile device sends its location information to a home agent which intercepts packets intended for the device and tunnels them to the current location. To do this the MN sends Binding Update (BU) messages to it are HA and all CNs every time it moves. This process contributes in high signaling cost when the mobile device moves very frequently. In order to reduce signaling cost, it is required to reduce the no. of messages sent over the air interface to all CNs and HA. Thus introducing a Mobility Anchor Point (MAP) allows MIPv6 to benefit from reduced mobility signaling with external network. Hierarchical Mobile IPV6 (HMIPv6), an enhanced Mobile IPv6 protocol uses this concept of reducing signaling cost in the network.

2. HIERARCHICAL MOBILE IP (HMIPv6)

Hierarchical Mobile IP (HMIPv6) [5] provides a scheme for performing registrations locally in the foreign network, thereby reducing the number of signaling messages forwarded to the home network as well as lowering the signaling latency that occurs when a mobile node moves from one foreign network to another. HMIPv6 introduces a conceptual entity, Mobility Anchor Point (MAP) [6], to separate local mobility from global mobility. MAP acts as a Mobile Node's (MN) Home Agent (HA) in the foreign network. MAP intercepts all packets destined for MN and tunnels them to its on-link CoA (LCoA). MN attaches itself to two Care of Addresses (CoA), a regional (or site) address and an on-link (or point of attachment) address. In HMIPv6, if we consider multiple levels of hierarchy, a Binding Update (BU) message is forwarded to the root MAP (RMAP) by way of one or more intermediate MAPs. When the BU message arrives at the first MAP, the MAP checks its mapping table to see whether the MN is already registered with it or not. If the MN is already registered in the mapping table, local binding update is completed at the MAP. Namely, in this case, the MAP generates a BU reply and sends the message to the next lower-level MAP in the hierarchy. However, if it is not, the MAP forwards the BU message to the next higher-level MAP. This process is repeated in each MAP in the hierarchy until a MAP having the MN in its mapping table can be found. Therefore, in the case of the first binding update in a foreign network, the BU message is forwarded up to the RMAP in the foreign

network and the HA. Fig. 1 shows a typical HMIPv6 multiple level network architecture.

When the multiple hierarchical levels are used, a lot of MAPs can be organized as a form of tree, so that it is possible to provide more scalable services and to support a larger number of MNs. Furthermore, although some nodes (i.e., intermediate MAPs) are failed, only sub-trees rooted from the failed nodes are affected from the failures. Therefore, the multi-level HMIPv6 can be a more reliable solution. However, the multilevel HMIPv6 results in a higher processing cost than the one level HMIPv6 when a packet is delivered to an MN. This is because the packet goes through more intermediate MAPs and the encapsulation/decapsulation procedures are repeated at each MAP.



1. Local binding update

2. Global binding update

Fig. 1. HMIPv6 Multiple Level Architecture.

In this paper, we analyze the performance of wireless/mobile networks with respect to signaling cost in multiple levels HMIPv6. Signaling cost consists of cost of location update and packet delivery cost [7].

$$C_T = C_{LC} + C_P$$

3. ANALYTICAL MODEL

In order to analyze location update cost, we consider hexagonal cellular architecture shown in Fig. 2. In addition, Each MAP domain is assumed to consist of many rings [8]. Each ring r (r>=0) is composed of 6r cells. Number of cell in ring 0 is 1 which is considering as centre point for the coverage area.

For ring 1: no. of cells = 6r = 6*1 = 6

For ring 2: no. of cells = 6r = 6*2 = 12 and so on.

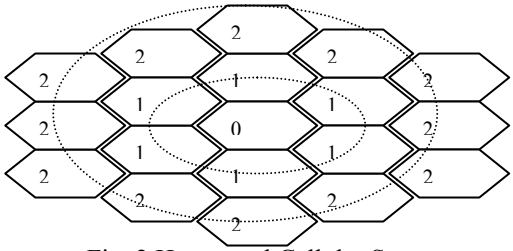


Fig. 2 Hexagonal Cellular Structure

Now, we consider a mobility model where the next position of an MN is equal to previous position plus a random variable whose value is drawn independently from an arbitrary distribution. Let p be the probability that MN remains in the current cell and the probability that MN moves to another area is (1-p). If MN is located in a cell of ring r (r>=0) then probability that a movement will result in an increase (q_{inc} (r)) or decrease (q_{dcr} (r)) in the distance from the centre point given by:

$$q_{inc}(r) = (1/3) + (1/(6 * r))$$
$$q_{dcr}(r) = (1/3) - (1/(6 * r))$$

Now, the transition probabilities $\alpha_{r,r+1}$ and $\beta_{r+1,r}$ are the probabilities of the distance of the MN from the centre cell increasing or decreasing are given by:

$$\alpha_{r,r+1} = \begin{cases} (1-p) & \text{if } r = 0 \\ (1-p)\left(\frac{1}{3}+\frac{1}{6r}\right) & \text{if } 1 <= r <= R \end{cases}$$

$$\beta_{r+1,r} = (1\text{-}p)\,(1/3\text{-}1/(6*r)) \quad \text{if } 1<=r$$

We define state r as the distance between the current cell of the MN and the center cell. This state is equivalent to the index of a ring where the MN is located. As a result, the MN is said to be in state r if it is currently residing in ring r. Let π_r be the state probability of state r with in a MAP domain. This is expressed as:

$$\pi_r = \pi_0 \prod_{i=0}^{r-1} \frac{\alpha_{i,i+1}}{\beta_{i+1,i}} \quad \text{for } 1<=r$$

With the requirement of $\sum_{r=0}^{1} \pi_r = 1$

$$\pi_0 = \frac{1}{1+\sum_{r=1}^{R}\prod_{i=0}^{r-1}\frac{\alpha_{i,i+1}}{\beta_{i+1,i}}}$$

3.1 LOCATION UPDATE COST FUNCTION

In HMIPv6, an MN performs two types of binding updates [9] i.e. global binding update and local binding update.

$$C_{LC} = C_g + C_l$$

The global binding update is a procedure in which an MN registers its RCoA with the CNs and HA. On the other hand, if an MN changes its current address within a local MAP domain, it only needs to register the new address with the MAP. A local binding update refers to this registration.

In the mobile networks signaling cost is proportional to the distance between two network entities. Due to this signaling cost of global binding update is larger than local binding update. Global binding update and local binding update cost is given by:

$C_g = 2(\kappa + \rho(D_{MA} + D_{HM})) + 2N_{CN}(\kappa + \rho(D_{MA} + D_{CM})) + PC_H + N_{CN}PC_C + PC_M$

$C_l = 2(\kappa + \rho.D_{MA}) + PC_M$

Where

κ = Wired link delay D_{MA} = Distance between MAP and AR,

ρ = Wireless delay D_{HM} = Distance between HA and MAP,

N_{CN} = No. of CN nodes D_{CM} = Distance between CN and MAP

PC_H = Processing Cost at HA. PC_C = Processing Cost at CN.

PC_M = Processing Cost at MAP.

In conclusion, probability that an MN performs a global binding update is $\pi_R \alpha_{R, R+1}$ and total cost of location update is

$$LC = \frac{\pi_R \alpha_{R, R+1} C_g + (1 - \pi_R \alpha_{R, R+1}) C_l}{T}$$

3.2 PACKET DELIVERY COST FUNCTION

Packet delivery cost is comprised of the processing cost and transmission cost. Packets sent to a MN will always go to the MN's home agent. These are then encapsulated and sent to the MN's FN through tunneling. But in case of route optimization, When the MN receives the first packet that was sent through the HN; it responds directly to the CN and sends together with the acknowledgement the BU message. Thereafter the CN sends packets directly to the MN at its FN. Packet delivery cost with the route optimization concept is expressed as cost of sending initial packet ($Cp_{(i)}$) and cost of sending successive packets ($C_{p(s)}$).

$$C_p = Cp_{(i)} + C_{p(s)}$$

Where

$$C_{p(i)} = \beta_T(D_{CH} + D_{HM} + D_{MA} + \kappa) + \beta\,(PC_H + PC_M + PC_A)$$

$$C_{p(s)} = \beta.\, PC_M + (D_{CM} + D_{MA} + \kappa)\,\beta_T$$

PC_A= Processing cost at AR β = Packet arrival rate for each MN

β_T = Proportionality Constant D_{CH}= Distance CN and HA

4. CASE STUDY

In order to analyze the performance of HMIPv6 with respect to location update cost. We consider the scenario shown in Fig. 3. The values of various parameters corresponding to the scenario are shown in Table 1.

This study presents various analysis results based on the developed analytical method. For this, we consider two values of p i.e. 0.2, 0.8. In each case total location update is calculated using the equations discussed in the previous sections. In this, we calculate location update cost for different sizes of MAP domain (different values of R) and different values of average residency time (T). The resulted values are shown in Table 2.

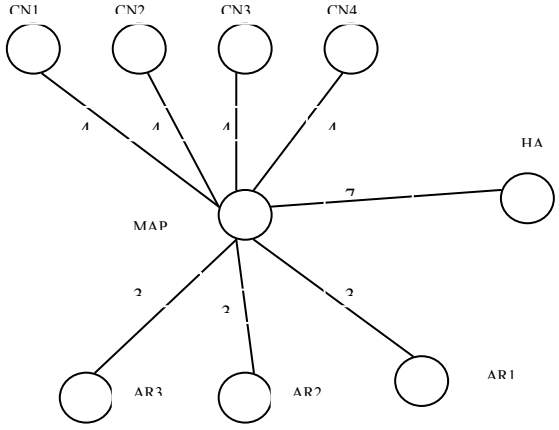


Fig. 3. Scenario for analysis

ρ	κ	$PC_$	$PC_$	$PC_$	N_{CN}	$D_$	D_{CM}	D_{MA}	PC A	D_C H	β	β_T
1	2	10	18	8	4	7	4	3	5	6	0.6	0.4

Table1. Values of various parameters for analysis

	R=1	R=2	R=3	R=4
P=0.2	66.63/T	48.63/T	40.58/T	36.05/T
P=0.8	31.66/T	27.16/T	25.15/T	21.34/T

Table 2. Location Update Cost for Various Values of R and p

The Location update cost with respect to the average residency time for different number of rings of each MAP domain (i.e. for R=1, 2, 3, 4) is shown in Fig.4. In this, we consider the probability of movement of mobile device is very high (p=0.2).During the high probability of movement of mobile device, increase in the MAP domain size decreases the location update cost .Increase in the MAP domain size reduces the chances of global binding update which plays a very important role in calculating location update cost.

The Location update cost with respect to the average residency time for different number of rings of each MAP domain (i.e. for R=1, 2, 3, 4) with the constant value of p=0.8 is shown in fig.4. In this, we consider the probability of movement of mobile device is very low (p=0.8). Further p is the probability that MN remains in the current cell. As the value of p increases, the probability of MN to stay in the current cell increases which decreases the MN movements and further

decreases the location update cost During the low probability of movement of mobile device, increase in the MAP domain size does not effect so much than that of previous case where p=0.2. Cell residence time is the period that an MN stays in a cell area. As the average cell residence time increases, the MN performs less movements and the location update cost per unit time decreases. Only change in the average residency time of a mobile device causes the change in location update cost. We can consider mobile device in this time as static.

Packet delivery cost consists of cost of processing and transmission cost. Packet delivery cost depends on the processing cost at MAP which further depends on the size of mapping table in order to select the current LCoA of the destination MN. Size of mapping table is proportional to the number of MNs in the MAP domain. More the number of MNs in the domain will result in the increase of mapping table size which further effects the processing time at MAP. Total number of MNs in the MAP domain is calculated as

$$n(MNs) = N(R) * \mu$$

where μ= Average number of MNs who located in the coverage of an AR .

$N(R)$ = Total no. of cells (ARs) up to ring R

$$N(R) = 3R(R+1)+1$$

Table 3 shows the total no of MNs for different values of R and μ. And Table 4 shows the impact of change of processing cost at MAP on the packet delivery cost

	μ=1	μ=2	μ=3	μ=4
R=1	7	14	21	28
R=2	19	38	57	76
R=3	37	74	111	148
R=4	61	122	183	244

Table 3. Number of MNs for different values of R

PC_M										
	10	12	14	16	18	20	22	24	26	28
C_p	48.6	54.1	59.6	65.2	70.7	76.2	81.72	87.2	92.8	98.3

Table4. Packet delivery cost for different values of PC_M

5. NUMERICAL RESULTS

In this section we analyze the impact of cell residency time, MAP domain size, Static and dynamic mobile device on the location update cost.

5.1 IMPACT OF CELL RESIDENCY TIME

Cell residence time is the period that an MN stays in a cell area. As the average cell residence time increases, the MN performs less movements and the location update cost per unit time decreases. As the probability of movement of mobile device increases, location update cost depends on cell residency time as well as MAP domain size as shown in Fig. 4

5.2 IMPACT OF STATIC AND DYNAMIC MNs

.

Movement of static mobile devices is very less than dynamic ones. So, Location update cost does not depends too much on the cell residency time and MAP domain size. Main consideration in case of Static MNs is packet delivery cost. In case of dynamic mobile devices where probability of movement of mobile device is very high, Location update cost depends on cell residency time and MAP domain size. Increase in MAP domain size increases local binding cost which is very less than global binding cost as shown in Fig.4. The overall saving of location update cost with the increase in average residence time is 38.4 % (from P=0.2 to p=0.8).

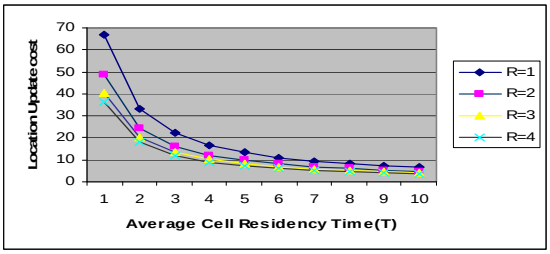


Fig 4. Relation between Location Update Cost and Average Residency Time for various values of R with the constant value of p(p=0.8)

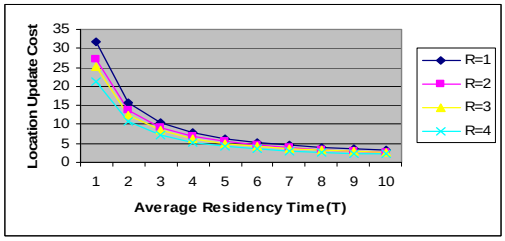


Fig 5. Relation between Location Update Cost and Average Residency Time for various values of R with the constant value of p(p=0.2)

5.3 IMPACT OF MAP DOMAIN SIZE

Increase in the MAP domain size causes decrease in global binding cost which further decreases location update cost expect the situation where the probability of movement of mobile device is low (p=0.8) . Fig 4 and Fig 5 shows the location update cost for static and dynamic MNs. Therefore in case of static MN, MAP domain size does not affect location update cost too much as compared to dynamic MNs. The overall saving of location update cost with the increase in

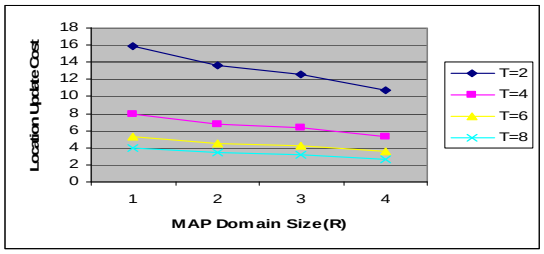
average residence time is 26.98 % (from R=1 to R=2), 16.5 % (from R=2 to R=3) and 11.2 % (from R=3 to R=4).
In this case, some optimal MAP domain size should be considered in order to minimize the location update cost of dynamic MNs (considering p=0.2). In Fig. 6, Change in Map domain size from R=3 to R=4 has a very small effect on the location update cost as compared to switching from R=1 to R=2 or R=2 to R=3.So, we consider optimal MAP domain size is 3.

5.4 IMPACT OF NUMBER OF MNs

Processing cost at MAP on the size of mapping table in order to select the current LCoA of the destination MN. Size of mapping table is proportional to the number of MNs in the MAP domain. More the number of MNs in the domain will result in the increase of mapping table size which further effects the processing time at MAP. In this paper, we define MAP domain size as the no. of rings. Large the size of the MAP domain causes large no. of MNs which results in the large size of the mapping table. Fig. 7. Shows the impact of Map domain size on the no. of MNs with different values of average number of MNs who located in the coverage of an AR
.

6. CONCLUSION

HMIPv6, an enhanced mobile IPv6 protocol, reduces location update cost and packet delivery cost by introducing MAP in the architecture. Location update cost depends on various factors viz. average cell residency time, MAP domain size and the probability of the MN (i.e. p) to stay in the current cell and packet delivery cost is effected by MAP domain size and average number of MNs who located in the coverage of an AR .The analytical results indicate that the MAP domain size is a critical performance factor to minimize the total cost in HMIPv6.For our case study we calculate optimal MAP domain size which is 3.



5:.Impact of Static MNs on Location Update Cost

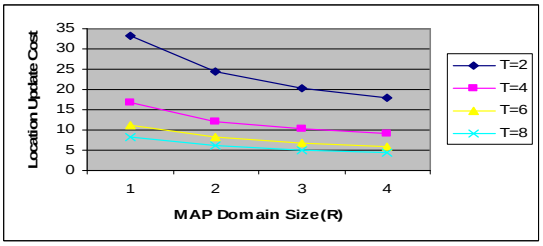


Fig: 6 Impact of Dynamic MNs on Location Update Cost

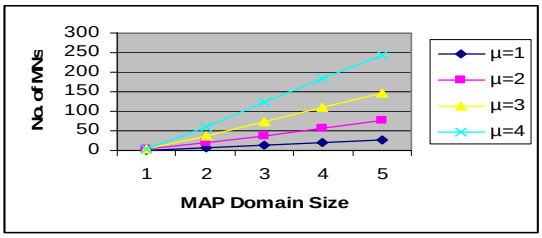


Fig. 7. Impact of MAP domain Size

7. REFERENCES

[1] Jun-Zhao Suna, Douglas Howiea, and Jaakko Sauvola , “Mobility management techniques for the next generation wireless networks”, Proceedings of SPIE, Wireless and Mobile Communications, Beijing, China, 4586 (2001), pp. 155-166.

[2] Ian F. Akyildiz Janise McNair Joseph Ho Huseyin Uzunalio_glu Wenye Wang, “Mobility Management in Next Generation Wireless Systems”, Proceedings of IEEE, 87 (1999), pp. 1347-1384.

[3] Maria Luisa Cristofano, Andrea G. Forte, Henning Schulzrinne, “Generic Models for Mobility Management in Next Generation Networks”, Columbia University Technical Reports CUCS-031-05, New York, 2005.http://mice.cs.columbia.edu/g e t Te c h r e p o r t . p h p ? t e c h r e p o r t ID=353&format=pdf&

[4] Jun-Zhao Sun and Jaakko Sauvola, “Mobility And Mobility Management: A Conceptual Framework”, Proceedings of 10th IEEE ICON 2002, Singapore, (2002) pp. 205-210.

[5] Esko Kupiainen, “Mobile IPv6 Mobility Management” http://www.cs.helsinki.fi/u/kraatika/Courses/MobInt/Essay1/EskoKupiainen.pdf

[6] H. Soliman, C. Castelluccia, K. El Malki and L. Bellier, “Hierarchical MIPv6 Mobility Management”, draft-soliman-mobileipmipv6-01.txt, IETF (2000). http://tools.ietf.org/html/draft-ietf-mobileiphmipv6-01

[7] Steve Mtika and Fambirar Takawira, “Mobile IPv6 Regional Mobility Management”, Proceedings of the 4th international symposium on Information and Communication Technologies, 92 (2005), pp.93-98.

[8] Sangheon Pack and Yanghee Choi, “A Study on Performance of Hierarchical Mobile IPv6 in IP-Based CelluLar Networks” , IEICE Trans. Communications, E87-B (2004), pp. 462-469

[9] Sangheon Pack and Yanghee Choi, “Performance Analysis of Hierarchical Mobile IPv6 in IP-Based Cellular Networks”, Proceedings of 14th IEEE PIMRC, 3 (2003), pp. 2818-2822.

An Architectural Self-Organization Model for Large-Scale Wireless Mesh Networks Based on Technology of Agents

Lucas Guardalben and João Bosco M. Sobral
Federal University of Santa Catarina
Computer Science Program,
Florianópolis-SC, Brazil
{guardalben,bosco}@inf.ufsc.br

***Abstract*— Wireless mesh networks is an emerging technology and it is quickly acquiring forces for their countless advantages in relation to the covering area and the low implementation cost. A main characteristic in wireless mesh networks is the self-organization capacity. The eminent advantage of a self-organized network is the reduction in the development complexity and also maintenance. In this work, we propose an alternative architecture for self-organization of wireless mesh networks based on the Optimized Link State Routing Protocol (OLSR) and technology of agents. As first results we argue that architectural self-organization model improve the delay, throughput and delivery of packets of the overall network, in comparison to the original OLSR protocol.**

I. Introduction

In the last few years, wireless mesh networks have attracted considerable attention of the business industry [1] as well as of the academic environment. They consist of client-nodes, which can support mobility, and mesh routers, which have wireless interfaces for communication among them, and can be organized in a fixed, autonomous or pre-determined way to form a backbone. The mesh routers can work as *gateways* or *bridges*, allowing the interconnection of different types of networks [2]. They are usually placed at high altitudes, in order to avoid physical interferences in their antennas area. It is an emerging technology which may have advantages in relation to a larger covering area and low implementation cost. There is a range of applications which can be implemented through those networks.

The wireless mesh networks consist in a great amount of nodes and are projected to have multiple hops. Consequently, there are two problems: heterogeneity and self-organization (form of organizing the network without manual or external intervention [3]). The self-organization capacity avoids the network to lose its autonomy, because there is not manual configuration of every mesh router. Another advantage of self-organized networks is the reduction of the installation cost and maintenance of the nodes involved in the network, once the client-nodes entrance in the network happens in a transparent way for the users. However, the current technology for wireless mesh networks only allows accomplish that aim partially [2].

Due to the complex nature of the wireless mesh networks, new paradigms are necessary to the design, manage and evolution of communication and computing. This article proposes an alternative architecture for self-organization, using agent technology [4]. The proposed architecture is based on the Optimized Link State Routing Protocol (OLSR) [5] and on the construction of modules for self-optimization, self-healing, self-protection and self-configuration of the routing protocol. For the acquisition of information concerning the overall network, we use the software agents that are responsible for the discovery of the network density and monitoring between mesh routers.

This article is organized in the following way: the section II contains an overview of the architecture for self-organization. The section III presents the experimental results and section IV shows the formal especification of the self-optimization process and finally the section V, which contains the conclusions and future works.

II. An Architecture for Self-Organization

The term self-organization had been first reported in the biology systems, later on it was adapted for different technical areas, such as engineering and computing.

For our work, we considered the following definition:

Self-organization in wireless computer networks, is defined when the wireless nodes are self-organized in order to efficiently perform the tasks required by the application, without human intervention and thereby can reduce the cost of installation[3]. In general, networks can be self-organized in two ways: application and network layers [6]. There are some topological forms, for example, flat, hierarchical and hybrid topology. We considered an hybrid topology and network layer.

The self-x capacities (optimization, healing, protection, configuration) [7] [8] appear jointly with the sub layers of the OLSR protocol. It is important to emphasize that the self-x capacities are not entirely independent. For example, self-configuration and self-optimization have the strong correlation, while they self-configures some parameters of the network, the other capacity try improve to the performance of these parameters. We follow the paradigm of design to exploit the

implicit coordination among the agents with the aim to build a design based on their local behavior to achieve some global properties.

In the other words, we presented the architecture details of the implementation involving the agents and the interaction of the self-organization modules with OLSR protocol (see the Fig. 1). We proposed a layered architecture because of the guarantee of the simplicity and flexibility of the architecture due to the very well defined functions in case it needs to be changed.

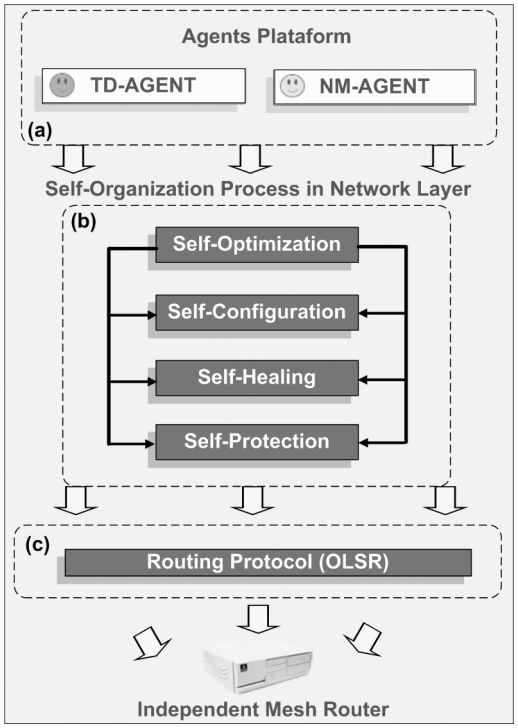


Fig. 1. Self-Organization Architecture for a Mesh Router

The agents that compose the superior layer are classified into two types, besides each agent plays a specific role in the self-organization architecture. The agents are:

- **(A)** ***TD-Agents*****:** they are mobile agents that are created in the mesh router and in each association of the client-node for a mesh router. The agent migrates for the client-node, this one has specific functions of the discovery of the network density, that vary of the small-scale, medium-scale and large-scale. A suit of *TD-Agents* form the base of **self-configuration** capacity of the OLSR protocol.
- **(A)** ***NM-Agents*****:** they are responsible for the monitoring of the network specific functions, as the **rate of loss of packets**, **delay**, **signal strength**, **throughput**, **latency**, **node active or not**, and also information about the state of the enlaces. They are static agents in the mesh routers, that serve as base for the **self-optimization** capacity of the OLSR protocol.

(B) Self-Optimization Process: this module is responsible for optimizing values of the parameters in the OSLR protocol, according to the network density.

(B) Self-Healing Process: this module is responsible for detecting, localizing and repairing failures on OLSR protocol in a transparent way.

(B) Self-Configuration Process: responsible for configuring the parameters of the OLSR protocol, according to the mesh routers and client-nodes that are registered in the network.

(B) Self-Protect Process: responsible for diagnosing and alerting the network administrator, regarding possible attacks against the OLSR protocol. In order to ensure one certain level of security in the network, we are doing an experience with the anonymous routing protocol. We consider the aspects and the definition of anonymity, presented in [9], Anonymity is defined in terms of unlinkability between items of interest, e.g., identity, location, transmissions and communication relationship. Related to the OLSR, some properties has been analyzed, such as: Source/destination identity anonymity, Intermediate identity anonymity, Location privacy and Route anonymity. **(C)** ***OLSR - Optimized Link State Routing Protocol*****:**

The information obtained through the agents of discovery (TD) and monitoring (NM), will be used to feedback parameters for configuration and optimization of the OLSR protocol. Among the parameters, it can be mentioned:

- **HELLO_INTERVAL:** the number of seconds among the sending of a message "hello" must be equal to 2 seconds by default. A HELLO message must be sent at least every HELLO_INTERVAL period, smaller than or equal to REFRESH_INTERVAL;
- **REFRESH_INTERVAL:** the Hello messages can be sent in the intervals in order to reduce routing overhead. In this case, each link and neighbor must be advertised at least once with REFRESH_INTERVAL, which is equal to 2 seconds by default.
- **TC_INTERVAL:** it orders to build the topology information, also each node, broadcasts Topology Control (TC) messages. TC messages are flooded to all nodes in the network. The information diffused in the network by these TC messages will help each node calculate its routing table. It is set to 5 seconds by default.
- **NEIGHB_HOLD_TIME:** it provides for how long one Hello message should be considered to be valid in the network. It is equal to 3*HELLO_INTERVAL by default.
- **TOP_HOLD_TIME:** it provides for how long one TC Message should be considered to be valid in the network. It is equal to 3*TC_INTERVAL by default.
- **WILLINGNESS:** it defines how willing the node is to be forward traffic. Its default value is 3.

Anothers parameters can influence OLSR performance. e.g (MPR_COVERAGE, which allows to define the amount of MRP's (MultiPoint Relay) that cover one node).

These parameters should be configured, in order to reduce messages overhead, as well as to support of the otimized routing paths creation. The factors that can influence the parameters of the OLSR protocol are: the size of the network, the

rate of nodes mobility, transmission distance, signal strength, overhead, jitter and delay. Meanwhile the major problem is commonly the defaults values, when they are configured in they do not provide the best performance in the overall network, in this case it is necessary an approach for self-configuration and self-optimization of parameters to reach the best overall network performance.

III. Experimental Results

To verify the improvement achieved by self-organization process, simulation are run in OPNET (OPtimized Network Engineering Tool) [10] Simulator. The mesh-routers are static positioning in the distributed campus network area in a 2000x2000m^2 with surround of the nodes clients mobiles, with diferent mobility (20, 5 m/s). The amount of nodes (density) in the network is configured in a dynamic way, and maintained by the TD-Agent. The traffic model used in this simulations is the HTTP. Every sender generates the 100 Kb/s HTTP traffic from a mesh-router. The time of simulation is 60s, and confidence interval is 95%. The agents are configured to send information from time to time, which do not avoid to accumulate monitoring messages. Each mesh-router was configured with Olsr protocol utilizing the default values and the values obtained by the software agents (self-optimized values). The first set of experiments we wish to evaluate the performance variation of the Olsr Original in comparison with Olsr self-optimized and self-configuration values in terms of the throughput, delay and dropped packets.

In Fig. 2. as it can be observed the comparison of the performance in average of throughput in bits/second between Olsr-Origin and Olsr-Self. We argue that the overall throughput of the network improve significantly due to the self-optmizing of the values in comparation to the default values of the OLSR protocol.

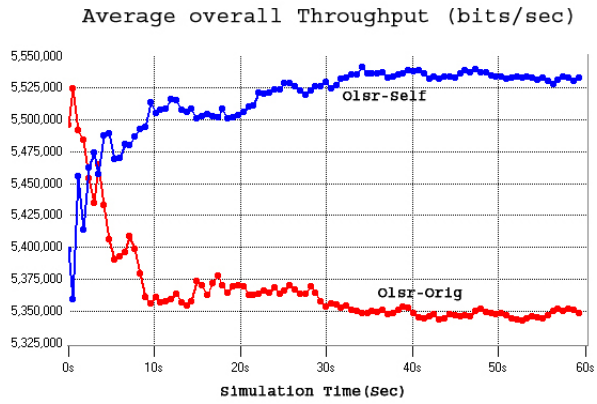


Fig. 2. Comparation the Throughput between Olsr-Original x Olsr-Self-Organized

In Fig. 3 the comparison the performance of the overall to delay in seconds, which utilizing the self-optmimized parameters, is smaller in comparison with the Olsr-Original.

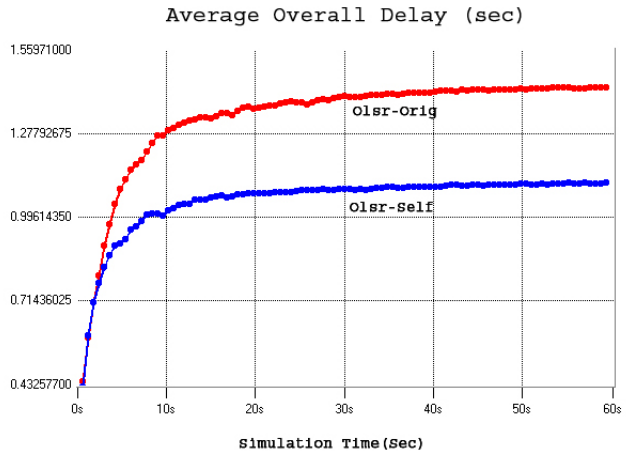


Fig. 3. Comparation the Delay of Local Discover Neighboring between Olsr-Original x Olsr-Self-Organized

In Fig. 4 as it can be observed it is compared the tax of dropped packets, which had been improved in the delivery packets utilizing the Olsr-Self.

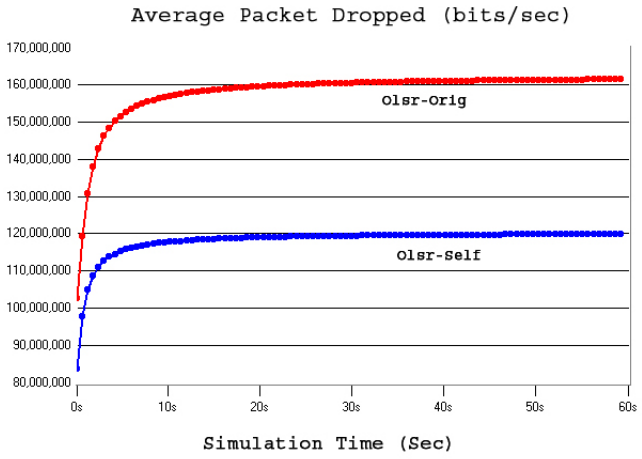


Fig. 4. Comparation the packets dropped between Olsr-Original x Olsr-Self-Organized

As observed in all of the experiments the values achieved by the agents and optimized by the process of sef-optmization, presented better result in comparison to the standard values of OLSR.

IV. Specifying the Architecture

In order to have a high level abstraction of the architecture, we used the formal language Object-$\mathbb{Z}$ [11] [12] to specify the self-organizing model based on the architecture defined. Our architecture is defined through the OLSR protocol and some of the software agents. These agents provide a feedback on the parameters, according to the changes in the full mesh network. Such changes are provided by the modules self-x. To show how we can build these modules self-x, we specify, for instance,

only the self-optimization module. The same procedure can be done in the other self-organizing modules: self-configuration, self-healing and self-protection. The first partial version for the Object-$\mathbb{Z}$ for the Self-Optimization on OLSR are described as follow.

Self_Optimization

Optimization values are classified in small_value_opt, medium_value_opt and large_value_opt.
The values of the OLSR protocol are changed.
The information of network size is comming the, TD_AGENT

OLSR − Parameters

HELLO_INTERVAL? : *NM_AGENT.HInterval*
REFRESH_INTERVAL? : *NM_AGENT.Rinterval*
DELAY? : *NM_AGENT.Delay*
TC_INTERVAL? : *NM_AGENT.Tc*
PACKET_JITTER? := *NM_AGENT.Jitter*
HELLO_JITTER? := *NM_AGENT.Jitter*
TC_JITTER? := *NM_AGENT.Jitter*
NEIGHBOR_HOLD_TIME? : *NM_AGENT.NHTime*
TOP_HOLD_TIME? : *NM_AGENT.THTime*
HELLO_ACQUIRE_COUNT? : *NM_AGENT.hacquire*
DUPLICATE_HOLD_TIME? : *NM_AGENT.duphold*
HOLDBACK_TIME? : *NM_AGENT.HTime*
NETWORK_SIZE? : *TD_AGENT.Size*

network_size? $<$ *small_scale*
Configure the OLSR Parameters conform network density
HELLO_INTERVAL$'$ = *small_value_HINT*
REFRESH_INTERVAL$'$ = *small_value_RINT*
DELAY$'$ = *small_value_DELAY*
NEIGHBOR_HOLD_TIME$'$ = *small_value_NHT*
.. ..
network_size? $>$ *medium_scale*
HELLO_INTERVAL$'$ = *medium_value_HINT*
REFRESH_INTERVAL$'$ = *medium_value_RINT*
.. ..
network_size? $>$ *large_scale*
HELLO_INTERVAL$'$ = *large_value_HINT*
REFRESH_INTERVAL$'$ = *large_value_RINT*
.. ..

V. Conclusions

This work proposed an architecture towards a self-organization of wireless mesh network. Our goal was using the OLSR in a self-organized way jointly with the agent technology aiming to help the self-organization modules. We followed the design paradigm to exploit the implicit coordination among the agents with the intetion building a design based on their local behavior in order to achieve some global properties.

We argued that by reviewing the specialized literature still exist few researches on self-organization for wireless mesh metworks. We can say that self-organizing networks can reduce the administrator's intervention for their configuration, management and maintenance, by only letting the administrator make decisions in critical situations.

In this work the motivation is not to invent a new wireless mesh protocol, but propose the improvements the for the routing protocol as its the case of the OLSR protocol. Our approach involved the layers definition of the architecture to support software agents helping in the self-organization process through discovery of network density, monitoring information acquirement of the mesh-routers and client-nodes.

As first experience, three improvements were given special importance in the original OLSR protocol, such as:

(1) the improvement of throughput of the overall network, and the;

(2) improvement of the delay of discovery local neighboring routes;

(3) improvement of the delivery of packets, due to smaller dropped packets.

Some practical aspects are visible to build the essence of a mesh network. Mesh routers can be built up through an alternative low cost with off-the-shelf components of the most known technologies and commercially available wireless devices (PDA, laptop, cell phones). At first we chose to work on the main self-organization capacities, by defining a more simplifyed self-organized model.

As future works, to validate the archicteture proposed, we intend to develop a prototype with all self-x capacities in a real environment. But before this we state the intention to simulate the other ones self-x capacities: self-healing and self-protection.

References

[1] Microsoft, "Self-Organizing Wireless Mesh Networks ," http://research.microsoft.com/mesh/ 2005.

[2] I. F. Akyldiz, X. Wang, and W. Wang, "Wireless Mesh Networks: a survey," *Computer Networks (Elsevier)*, vol. 47, p. 445 a 487, March 2005.

[3] S. Vasudevan, "Self-Organization in Large-Scale Wireless Networks," Ph.D. dissertation, University of Massachusetts - Dept Computer Science, September 2006.

[4] G. Weiss, *Multiagent System: A Modern Approach to Distributed Artificial Intelligence*. London: MIT Press, 1999.

[5] T. Clausen and P. Jacquet, *Optimized Link State Routing Protocol (OLSR)*, IETF RFC 3626, October 2003.

[6] H. Tang and H. Tianfield, "Self-Organizing Networks of Communications and Computing," in *International Transactions on Systems Science and Applications*, vol. 1, no. 4, 2006, pp. 421–431.

[7] O. Babaoglu, M. Jelasity, A. Montresor, C. Fetzer, S. Leonardi, A. van Moorsel, and M. van Steen, *SelfStar Properties in Complex Information Systems*. Springer, 2005.

[8] J. O. Kephart and D. M. Chess, "The Vision of Autonomic Computing," in *IEEE Computer Networks 36*, vol. 1, 2003, pp. 41–50.

[9] C. H. Tamashiro and J. B. M. Sobral, "An Analysis of Anonymous Routing Protocols for Ad Hoc Mobile Wireless Networks," Department of Computer Science - Federal University of Santa Catarina - UFSC, Tech. Rep., 2006.

[10] O. Technologies, "Inc," 2007, access 15 August. [Online]. Available: http://www.opnet.com/university_program/research_with_opnet/

[11] J. M. Spivey, *The Z Notation: a Reference Manual*, Prentice-Hall, Ed., 1989.

[12] R. Duke, P. King, G. Rose, and G. Smith, "The Object-Z Specification Language," Department of Computer Science - University of Queesland 4072 - Australia, Tech. Rep., 1991.

Online Charging to Sustain Sensor Networks

(Short Paper)

Philippe Moore* and Anil M. Shende†
*Tenable Network Security, Columbia, MD 21045. U.S.A. Email: pmoore@tenablesecurity.com
†Roanoke College, Salem, VA 24153. U.S.A. Email: shende@roanoke.edu

I. Introduction

Limited battery life severely hampers the deployment of sensor networks for long term use. At the same time, proposed uses for sensor networks assume significantly long lifespans, and deployment in geographically remote areas (see [1] for a survey of sensor networks). In the literature, researchers have proposed various mechanisms to maximise the lifespan of sensor networks by conserving energy. (See, for example, [2], [3], [4], [5].) These mechanisms usually compromise the communication efficiency, or computational quality of the networks.

In this paper we propose a new approach, that of *online charging*, to address this problem. Rather than try and conserve power, thus degrading the performance of the network, we propose to employ a roaming charger device to recharge, *online*, the individual motes in a functioning sensor network. Although such technology, to our knowledge, does not exist physically, we can easily imagine, for example, a device that has a repository of spare batteries, replaces the spent battery on a mote with a charged one, and then recharges the used battery for later use. Our methodology will allow applications designed for mobile adhoc networks, where the individual devices typically have larger batteries and/or are whose power can be readily replenished, to be easily ported to motes in sensor networks. (See the United States National Institute of Standards and Technology (NIST) website for a comprehensive survey of mobile adhoc networks and various applications [6].)

In this exploratory paper we formalise our proposed model and present two online algorithms for the charger device to schedule and recharge the motes in the sensor network so that, ideally, all the motes have power all the time (Section II). To the best of our knowledge this approach to extending the lifespan of sensor networks has not been reported in the literature. The model is arguably impractical at this time; we believe that careful study of this model and promising theoretical results about the model could spur technological research to make it physically viable. In closing (Section III) we indicate some of our future directions as we study this model further.

II. Proposed Model and a Solution

A. The Scenario and Problem

As mentioned above, our model comprises of a collection, $\mathcal{M}$, of motes distributed in a given region of two-dimensional space, along with a charger device, C. For convenience we will assume that the region is discretized as a square grid so that all the motes are at grid points. (See [7] for an algorithm to logically situate the devices of a wireless network on grid points.) The origin of the grid is the lower left corner of the grid. We will denote the euclidean distance in the grid by $\delta(\cdot,\cdot)$ and the graphical or rectilinear distance (L_1 metric) by $d(\cdot,\cdot)$. We will assume, without loss of generality, that the grid under consideration is a square $S \times S$; $D = 2S$ is the diameter of the grid.

The location l_i of each mote, m_i, then is a pair of non-negative integers. L_M will denote the set of locations of the motes. The transmitting range of each mote is r. Thus the resulting communication network is the graph $G_c = (V_c, E_c)$ where $V_c = \mathcal{M}$ and $(m_i, m_j) \in E_c$ iff $\delta(l_i, l_j) \leq r$. We assume that G_c is connected. We will count energy in "work units". Each mote has a battery with a maximum capacity of W work units. At each unit of time, each mote may perform some task consuming up to ω work units of power. We will denote the remaining power in the battery of mote m_i at time t by $\mathcal{P}(m_i, t)$. When a mote's power level drops below a threshold τ, it broadcasts a message requesting service.

Initially, C starts at the origin, and, in our idealized scenario, we assume that C has an unlimited supply of power. We will denote the location of C at time unit t by l_t. C moves around in the region along grid lines, and during each unit of time, can move one grid unit. Thus, starting at time unit t, C takes as many units of time to arrive at m_i as $d(l_t, l_i)$. We assume that once C is at the location of a mote, the process of charging is instantaneous. We will denote the complete graph representing the possible motion of the charger by $G_m = (V_m, E_m)$ where $V_m = L_M \cup \{(0,0)\}$ and for each p_i, p_j in V_m, $(p_i, p_j) \in E_m$.

We will assume that each mote and the charger are equipped with some form of location service, e.g., GPS. When the sensor network is deployed, all the motes first broadcast their locations, go through a discovery phase and build their routing tables for their neighbours; the charger builds a network "map".

For a sensor network deployed from time unit 0 until time unit T, a schedule for the charger is a function $\sigma : \mathcal{N} \rightarrow \mathcal{M} \times \{0, \ldots, T\}$; $\sigma(k) = (m_i, t)$ means that the k-th recharge done by C is at time t for m_i. A schedule is said to be *ideal* if for all $t, 0 \leq t \leq T$, for all $m_i \in \mathcal{M}$, $\mathcal{P}(m_i, t) > 0$.

Our problem has two parts: For a given sensor network with a charger running until time T, (1) if the work done by each mote during each time unit is known a-priori, is there an *offline*

algorithm to compute an ideal σ?, and (2) is there an *online* algorithm to compute an ideal σ?

B. Problem Complexity

The first part of our problem above is an extended variant of the problem called *Deadline-TSP* that has been shown to be NP-complete. In the literature, several researchers have presented approximation algorithms for this problem (see, for example, [8]). Even if our network sizes are small enough for the NP-completeness of the problem to *not* manifest itself, our problem has an additional complexity. In the Deadline-TSP problem, each vertex needs to be visited once; in our case, after a mote is recharged, it continues working, and thus is in need of a recharge several times during the lifetime of the sensor network. A solution to the Deadline-TSP may schedule a vertex to be visited, i.e., a mote to be recharged, even if the mote has not done substantial work. As a consequence, it may need to be recharged a second time *before* some of the other motes have received their first recharge. This violates the constraints placed on solutions to the Deadline-TSP problem. Our problem is better approached as an online algorithm [9] – we are concerned with maximising the lifetime of all the motes in response to requests for service.

C. Online Algorithms and Results

Clearly, the performance of any algorithm to schedule the charger will depend on the problem parameters that are dictated by the application running on the sensor network. Nonetheless, using *any* Hamiltonian cycle in $G_m \backslash \{0,0\}$it can be easily shown that:

Theorem 1: Suppose $W \geq \omega \cdot |\mathcal{M}| \cdot D$. Then, there exists an ideal schedule for this problem instance. ■

The sufficient condition in the above result is fairly idealistic, though. Assuming a uniform distribution of nodes in the region, and a uniform distribution of power used by each mote per unit of time, we see that the expected distance between two motes is $S^2/|\mathcal{M}|$ and the expected workload per unit of time per mote is $\omega/2$. Thus, we have:

Theorem 2: Suppose $W \geq (\omega/2) \cdot |\mathcal{M}| \cdot (S^2/|\mathcal{M}|) = (\omega \cdot S^2)/2$. Then, a hamiltonian cycle witnesses, with high probability, an ideal schedule for this problem instance. ■

We used simulations of instances to study the performance of two algorithms, a naive algorithm and an adaptive algorithm, for computing a schedule. Our naive algorithm computes a Hamiltonian cycle and uses that as a schedule. Our adaptive algorithm has the charger constantly moving towards some mote to recharge that mote. The charger maintains a queue of requests for charging from motes. This queue starts as being empty. Initially, the next mote chosen to be recharged is the one closest to the origin. After that point, at any time unit t, the charger is either on its way to a mote, or has arrived at a mote. Figure 1 shows the algorithm the charger uses at each time unit t.

Although our naive algorithm can easily be made to fail with suitably chosen parameter values, our interest in studying simulations of this algorithm is two-fold: (1) to determine

```
if charger has arrived at a mote m_i,
    if the request queue is empty,
        Let the next mote be the mote that has been
            least recently recharged.
    else
        Let the next mote be the closest mote that has
            a request on the queue.
    endif
else
    if there is a new request,
        add the request to end of the request queue.
```

Fig. 1. Adaptive algorithm

ranges, if any, of parameters for which this algorithm, statistically, *does* compute an ideal schedule, and (2) to compare this algorithm with our adaptive algorithm.

We ran our simulations on a grid with parameter values: $S = 15$, $|\mathcal{M}| = 30$, $r = 5$, $T = 5000$, and $\omega = 20$. The mote locations were generated uniformly randomly, and we ensured that the resulting communication graph was connected. The work units used by each mote per time unit came from a uniform distribution as well. Simulations on this scenario with $W = 2000$ verify Theorem 2 – the naive strategy for the charger maintains power for all the motes, and so does the adaptive strategy. To get more meaningful results, we set $W = 500$. Our adaptive algorithm behaves exactly as the naive algorithm when the threshold τ is set to 0, since in this case, motes never send requests to the charger. To experiment with the adaptive algorithm, we varied the threshold τ between 0 and 500. We present a small sample of our simulation results for the adaptive algorithm. We use the two attributes, average time for which each mote had power, and the number of times that motes "died" (lost all power). The complete set of our results will be presented in the full paper. Figures 2 and 3 show the change in values of these two attributes as we varied the value of τ using our adaptive algorithm. With the naive algorithm ($\tau = 0$), each mote was alive (on an average) for 2500 units of time, and the simulation witnessed about 1300 deaths. With the adaptive strategy, the average time for which motes were alive peaked at about 3700 when the threshold value was 200, and the number of deaths steadily decreased as the threshold value was increased.

Fig. 2. Simulation Results – Motes' time being "alive"

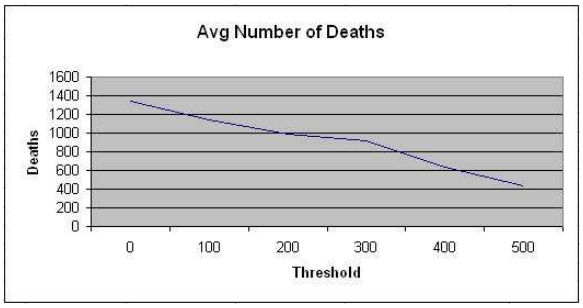


Fig. 3. Simulation Results – Number of mote "deaths"

III. Conclusions

In conclusion, we have proposed a new model – online charging – for addressing the energy considerations in a sensor network. We have shown via simulations the feasibility of converging on an online algorithm for computing nearly ideal schedules for the charger. Future directions include a formal competitiveness analysis, a statistical analysis of the problem, including using different distributions to generate the work load for the motes, extensive simulations to verify the theoretical results, working on dropping some of the idealized assumptions in our model, and studying the problem when there are $k > 1$ chargers.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, 2002.

[2] A. Michail and A. Ephremides, "Energy-efficient routing for connection-oriented traffic in wireless ad-hoc networks," *Mobile Networks and Applications.*, vol. 8, no. 5, pp. 517–533, 2003.

[3] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *HICSS*, 2000. [Online]. Available: citeseer.ist.psu.edu/rabinerheinzelman00energyefficient.html

[4] Y. Xu, J. S. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Mobile Computing and Networking*, 2001, pp. 70–84. [Online]. Available: citeseer.ist.psu.edu/xu01geographyinformed.html

[5] A. Sinha, A. Wang, and A. Chandrakasan, "Energy scalable system design," *IEEE Transaction on Very Large Scale Integration Systems*, vol. 10, no. 2, 2002.

[6] United States National Institute of Standards and Technology, "Wireless Ad Hoc Networks," http://w3.antd.nist.gov/wahn_home.shtml, (Accessed on September 26, 2007).

[7] V. Gupta, G. Mathur, and A. M. Shende, "Wireless adhoc lattice computers (WAdL)," *Journal of Parallel and Distributed Computing*, no. 66, pp. 531–541, 2006.

[8] N. Bansal, A. Blum, S. Chawla, and A. Meyerson, "Approximation algorithms for deadline-tsp and vehicle routing with time-windows," in *36th ACM STOC*, 2004.

[9] A. Fiat and G. J. Woeginger, Eds., *Online Algorithms: The State of the Art*. Springer-Verlag, 1998.

Distance Estimation System based on ZigBee

Pablo Corral[1], Miguel Fayos[1], Ricardo Garcia[1], V. Almenar[2], A. C de C. Lima[3]
Miguel Hernández University, Signal Theory and Communications, Elche, Spain[1]
Polytechnic University of Valencia, Communications Department, Valencia, Spain[2]
Federal University of Bahia, Electrical Engineering Department, Salvador, Brazil[3]
pcorral@umh.es

Abstract- **Many systems exist (TOA, AOA, RSSI) that can be used with diverse technologies (ultrasounds, infrared, Bluetooth, 802,11...) for indoor location. In this article we choose to apply TOA on a ZigBee network, to be able to add useful functionalityies for other applications in an existing network and, in addition, achieve minimum consumption and low cost. The results that we find are promising, because we obtain a precision within a meter in 90% of the cases.**

I. Introduction

An indoor system of range estimation allows you know the position of a mobile terminal with respect to a fixed node in closed surroundings. Appart from adding functionalities to existing networks or, as the search of devices or the presence detection, the range estimation is the base to create a system of indoor positioning, that mitigates the logic limitations of GPS location in interiors.

For that reason, there is a lot of bibliography on the range estimation and location in this type of surroundings. Using ultrasounds technology we can obtain better results like [1], that achieves precision of 15 cm. The problem of such systems is that are dedicated surroundings: their nodes are only useful for the position estimation. In the present systems, nevertheless, one tends to integrate all the possible functionalities in a single system.

With this philosophy, systems of range estimation in interiors have been developed on already existing networks, like Bluetooth in [2] or WLAN with standard 802.11 in [3]. In this type of networks, the achieved precisions usually are within a meter.

Our election has been to integrate a system of range estimation on ZigBee, a commercial specification based on the standard IEEE 802.15.4. This standard fulfills our requirements exactly. In first place, we can use this network as a home automation system. In that way, it is common to have several nodes in same environment. In addition, its philosophy is based on minimum consumption and low cost, and the system that we have implemented adds a minimum hardware that does not increase in price the final node. So, our system will try to reach precisions similar to which obtains other non-dedicated networks, but being added to the advantage of the minimum cost and low power.

The article is divided as it follows: in section II, we described the different existing techniques for the range estimation. In section III we make a small summary of the standard 802.15.4 and its described superior layers in ZigBee. In IV we explained the implemented system, whose results detail in V. Finally, we draw the conclusions on which our future investigations will begin in section VI.

II. Methods For Estimation Of Distances

Diverse techniques used for the indoor positioning of objects and people in surroundings exist. Most outstanding they are:

- AOA (Angle of Arrival): this method uses multiArray antennas located in the base stations to determine the angle of the incident signal. If a terminal that transmits a signal is in the direct line of sight (LOS), the multiarray antenna can determine from what direction comes the signal. In order to know the position of the terminal, at least, one second estimation coming from another base station is necessary with the same technology that the first one. The second base station will locate to the terminal and will compare its data with those of the first station later to calculate the position of the user by means of trigonometry.

- TOA (Time of Arrival): this technique is based on the measurement of the time of arrival of a signal transmitted by a mobile terminal to different base stations. In order to carry out the calculation, a possibility is to measure the time of roundtrip of the signal (RTT, Round Trip Time). This way, the whole range travelled by the signal is calculated as the product of the time used in arriving at the base stations and the speed of light. This is the technique that we will use.

- RSSI (Received Signal Strength Indication): well-known the power whereupon we received the signal in our node, we can consider the distance. The disadvantage of this technique is that it requires a good training, in addition to which its precision is seen remarkably affected by changes in the surroundings.

III. Wireles Personal Area Networks: ZigBee

The personal area networks (PAN) are networks for interconnection of devices near a person that habitually have smaller reaches of 10 meters. Between the radio networks of this type, we have the well-known Bluetooth, with a medium transmission rate, and UltraWideBand, whose standard still is unfinished, and that it allows high rates of transmission. For our objective, we choose to use the standard IEEE 802.15.4 [4], that describes layers physical and MAC of a radio network of low rate of transmission (LR-WPAN) and therefore of low cost. Between its basic characteristics:

- Maximum rate of data transfer of 250 Kbps.
- 16 channels in the frequency band of 2,4 GHz.
- Access to channel CSMA/CA.
- QPSK Modulation, DSSS.
- AES Encryption.

Three types of nodes in a network 802.15.4 are distinguished:

- ZigBee coordinator (ZC) or FFD (Full Function Device): The most capable device, the coordinator forms the root of the network tree and might bridge to other networks. There is exactly one ZigBee coordinator in each network since it is the device that started the network originally. It is capable of storing information about the network.

- ZigBee Router (ZR): As well as running an application function a router can act as an intermediate router, passing data from other devices.

- ZigBee End Device (ZED) or RFD (Reduced Function Device): Contains just enough functionality to talk to its parent node (either the coordinator or a router); it cannot relay data from other devices. This relationship allows the node to be asleep a significant amount of the time thereby giving you the much quoted long battery life.

On standard 802.15.4, a group of 25 well-known companies, like Philips and Motorola, created the ZigBee specification [5] that contributes the superior levels (network and application) of the WPAN, with functions like the routing or the extreme security to end. Between the solutions available in the market, our project leaves from contributed by Microchip with its kit of evaluation of ZigBee PICDEM Z.

IV. Implemented System

In order to implement the decided system, a system of range estimation integrated in a network 802.15.4, the basic components are a microcontroller, a transceiver and an antenna; in addition to the devices that we will use to measure the time.

This way, part of the evaluation kit PICDEM Z, that provides two ZigBee nodes, one working like FFD and another one like RFD. Each one of these nodes consists of two plates: the main one, with a microcontroller PIC18LF4620, and secondary with transceiver CC2420 of Texas Instruments and an integrated antenna.

The last one is a transceiver designed specifically for applications of low power and minimum voltage, in 2,4 GHz. We will emphasize next some of its outstanding characteristics:

TABLE I
CC2420 Characteristics

Voltage	$2,1-3,6\ V$	Consumption in reception	$18,8\ mA$
Range of frequencies	$2,4-2,4835\ GHz$	Consumption in transmission	$17,7\ mA$
Bit Rate	$250\ Kbps$	Power transmission	$-25\ dBm$ $0\ dBm$
Sensitivity	$-95\ dBm$	Frequency of the oscillator	$16\ MHz$

The configuration of the transceiver and its handling are carried out from the microcontroller. The communication microcontroller-transceiver makes use of the communication protocol SPI (Serial Communication Interface). It is a serial synchronous communication, in which the microcontroller acts like master, sending the clock signal and indicating when the communication by means of the pin is made Select Chip.

In addition to these two lines, other two are only required, for outgoing data and incoming data.

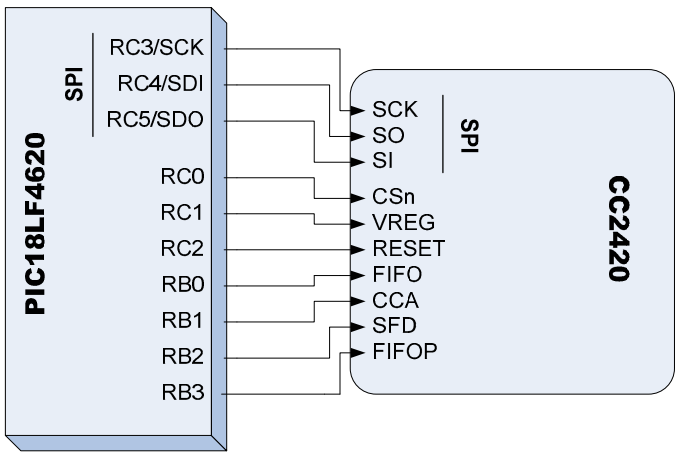


Fig. 1. Microcontroller-Transceiver Connections.

In each node PICDEM Z, the secondary card (daughter card) where the transceiver is, we also have a printed antenna of similar characteristics to the recommended one by Chipcon in [6] and that is omnidirectional in the horizontal plane.

When we use TOA for the range estimation, we need to implement a low cost system to measure the time that passes since we sent a MAC frame until we received answer.

Once decided the hardware to use, we see how we can use the measurement of RTT time to consider the distance between the nodes.

The basic procedure consists of sending a frame used by the standard and waiting for an answer. If we know the time that have taken in arriving the answer and the speed to which the signal travels and we avoided the time of processing of the frame, we can obtain the distance that separates both nodes:

$$d = \frac{c \cdot t}{2} \tag{1}$$

where d is the distance that separate both nodes, c the speed of the light and t the time that takes the signal in crossing the space that separates the nodes in roundtrip (for that reason it is divided by 2).

Let us apply the idea to our WPAN, and count on the fact that already the answer does not take place of instantaneous form, but that requires a time of processing in the other node. Let us observe the following diagram:

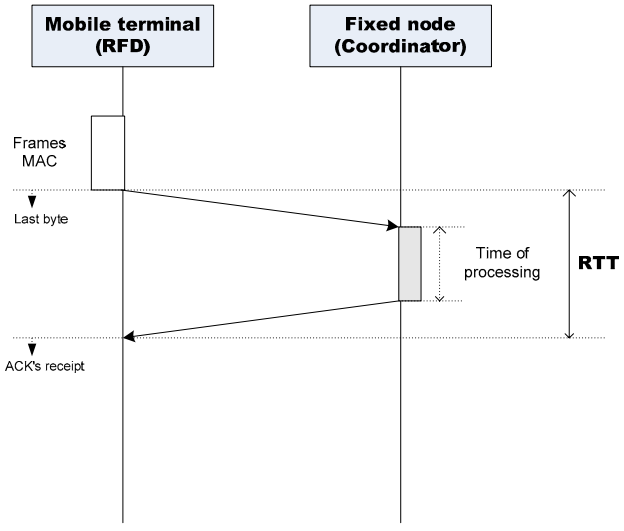


Fig. 2. Round Trip Time.

As we see, the RTT is the time passed from the shipment of the last byte of the MAC frame, to the reception of the answer. Modifying the previous equation to adapt it to our case, theoretically we could obtain the distance between two nodes of the following form:

$$d = \frac{c \cdot \left(RTT - t_{procesamiento} \right)}{2} \tag{2}$$

In order to avoid having another hardware of count in the fixed node, due to processing times of non-constant frames, we used AUTOACK function available with transceiver CC2420. When activating it, according to [7], the transceiver gives back an ACK with the direction of the node that has sent it the original frame, exactly 12 symbols (1 symbol = 4 bits) after receiving it.

For the creation of the code for our application, we will leave from stack of Microchip, modifying and adding the necessary thing; and with precaution of not interfering in the on-speed operation of a network ZigBee, since we looked for a system non-dedicated exclusively to the location. Basically, it consists of a set of archives .c, archives of head .h and other files, in which the inferior levels of the standard the 802.15.4 and basic ones of ZigBee are implemented. From this way, the end user obtains a support to create its ZigBee applications without having to worry about basic functions of ZigBee such as to program the form in that the coordinator looks for a free channel to form a network, or the series of necessary messages so that a RFD can adhere to the same one.

When not having control on the processing of the messages of the application layer of ZigBee to measure the RTT, we decided to use messages of level MAC directly. The following consideration is that the coordinator is always wide-awake, with which can receive messages at any time; it does not happen the same with the RFD.

For this reason, it is logical to think that the one that receives the order, when pressing must be the RFD a button or to receive a message of the computer or the own coordinator, to send a MAC frame and to remain wide-awake waiting for the answer. Returning to Figure 1, the one will be the RFD that acts like mobile terminal, whereas the coordinator is the fixed node.

V. RESULTS

Theoretically, known the time that the answer takes to arrive at the frame, we will be able to compute the range to which they are. Obviously, a single transmission will not be sufficient to obtain a trustworthy calculation. In the first place, the own resolution of the oscillator which we use with the binary account marks an error that will be able to be reduced when making many transmissions. But, in addition,

there are other possible causes of error that could affect to the estimation [3]:

- The **discreet quantification of the time** when using a signal of an oscillator for the measurement of the same one. In principle, the clock of 50 MHz would allow a resolution of 20 ns, which could lead to errors about 3 ms. Nevertheless, the clock that marks to the maximum resolution is the one of the transceiver: 125 ns, which takes us to 18 meters of precision if we only made a transmission.

- **Delays due to the own hardware**. In any other application, they would not be important, whereas, for example, a logical door introduces a typical delay of "only" 15 ns. However, we remember that in our application a delay of 7 ns introduces an error of one meter in the measurement. From this way, the important thing of this type of delays is not as much that they are small, because they are constant. Parameters as the temperature of the circuit could affect these times.

- **Drift of the clocks**. The stability of the crystal of the transceiver is about ±40 ppm (parts by million); whereas the one of our crystal of 50 MHz is about ±100 ppm. We divided the pulse which we measured in the SFD (about 360 µs) between the 20 ns of oscillator period, and obtain that, in each measurement; they will pass approximately 18000 cycles of clock. With the commented stability, statistically it would be very probable an error about two cycles by transmission.

- **The characteristics of the indoor surroundings.** Let us take a break in this point.

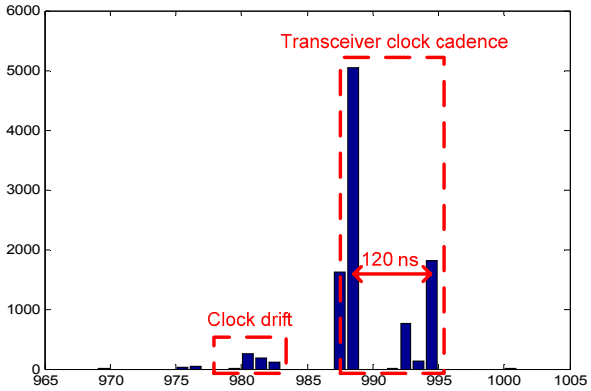


Fig. 3. First test to 50 MHz.

To 2,4 GHz, many elements indoor can affect the transmission of the signal. (To 2,4 GHz, the wavelength is λ = 12,5 cm). Wood furniture, walls, doors or the presence of people can cause effects of fading of the signal or scattering. Other standards as 802.11 working to 2,4 GHz can cause interferences and that the channel is occupied constantly, affecting to the access means; and certain elements as microwave ovens or alarms can remarkably affect the reception [8].

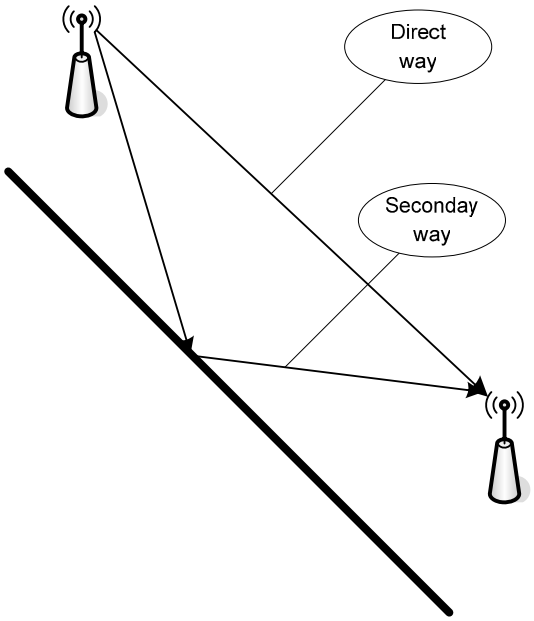


Fig. 4. The multipath effect.

In [9] remembers that although the surroundings as much outdoor as the indoor can be model, in the case of these last ones introduce many more random components and more heterogenous elements, taking in frequency and spatial channels that vary constantly with time. In addition, phenomena as the reflection and the dispersion cause to the multipath effect, which will make difficult the estimation of the distance.

Tests will be made. The nodes will be separated one meter, and 20.000 transmissions will be made. This procedure will be repeated for different distances in different environments. We are looking for:

- The pattern between time and distance in each environment. The least squares method will be used for this matter.

- The number of transmission needed to obtain an adequate estimation. We will use the batch means method.

-Least Squares Method

The procedure for fitting some experimental results to a function is called *linear regression*. We can obtain also its error. This function can be either linear or curved. We will try to fit the data to a linear one, represented by the function:

$f(x) = ax + b$, which will allow us to predict future values.

Having some experimental data, the objective is to minimize the error between that data and the values that the final function would obtain. Minimizing the sum of the squares of the residual:

$$s = \sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} \left(y_i - f(x_i) \right)^2 = \sum_{i=1}^{N} \left(y_i - (ax_i + b) \right)^2 \quad (3)$$

where s is the residual, N is the total number of samples that we have, y_i is the i-th value for x_i, and the values we want to obtain are those that determinate our linear function: the coefficients a and b.

There is also a way to know if the data fits well to the obtained function: we can use the "correlation coefficient", r, which can be obtained from:

$$r = \frac{N\sum_{i=1}^{N} x_i y_i - \sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i}{\sqrt{N\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i \right)^2} \sqrt{N\sum_{i=1}^{N} y_i^2 - \left(\sum_{i=1}^{N} y_i \right)^2}} \quad (4)$$

This coefficient will return a value between 0 and 1. The more r approximates to 1, the more the data fits the calculated function, and therefore, our approach will be more trustworthy.

-<u>The batch means method</u>.

The batch means method, also called method of subsamples, is a stopping criteria recommended in [10], which can be used in our case to determine the number of samples needed to get a nice estimation of the distance. As we have said, 20.000 transmissions are done for each distance on our tests; but surely we will not need so many transmissions to calculate the distance.

The main idea of the method consists on, given a long run of N samples, where N = 20.000, we divide them into m = [N/n] batches of n samples each. We calculate the mean of each one, and the variance of those means; and then we repeat the process with bigger batches. If the system works well, as we increase the number of samples on each batch, the variance of the means will be reduced. Then, we start with a small value of n, for example, n = 1, and proceed as follows:

1. Compute means for each batch:

$$\bar{x}_i = \frac{1}{n} \sum_{j=1}^{n} x_{ij} \quad (5)$$

with i=1,2…m.

2. Compute an overall mean:

$$\bar{\bar{x}} = \frac{1}{m} \sum_{i=1}^{m} \bar{x}_i \quad (6)$$

3. Calculate the variance of batch means:

$$Var(\bar{x}) = \frac{1}{m-1} \sum_{i=1}^{m} \left(\bar{x}_i - \bar{\bar{x}} \right)^2 \quad (7)$$

-<u>Results obtained</u>:

Some samples of the results obtained in two real indoor environments are shown. The first one is a corridor, (2 meters of width, 30 meters long).

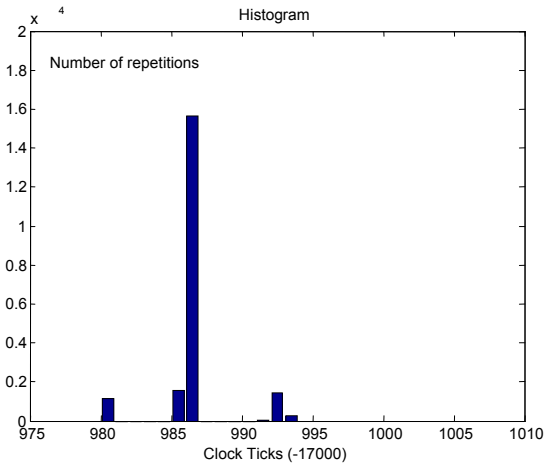


Fig. 5. 4 meters.

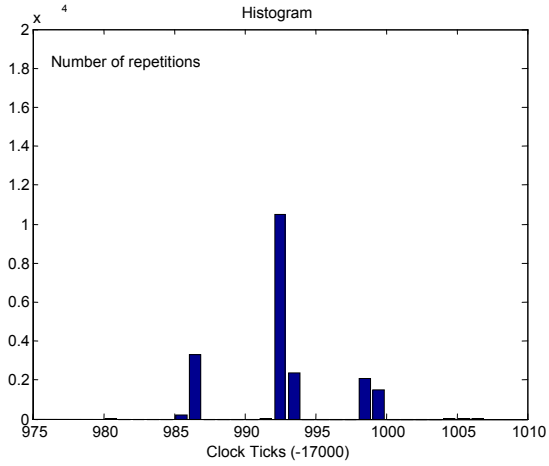


Fig. 6. 14 meters.

As can be seen, the results are qualitatively consistent because they shift to the right as the actual distance increases, with approximately constant separation between consecutive distributions. We could say that, with an specific training, the precision obtained with the system could be near one meter. With these results, we have tried some estimators to choose the one with the greatest correlation coefficient. Table II shows the performance of each one:

TABLE II
ESTIMATOR IN SPECIFIC STAY

Estimator	correlation coefficient
η	0.9088
$\eta-\sigma$	0,8136
$\eta-2\sigma$	0,8503
$\eta-3\sigma$	0,7532

The average η is the best one, as we can see on Fig. 7:

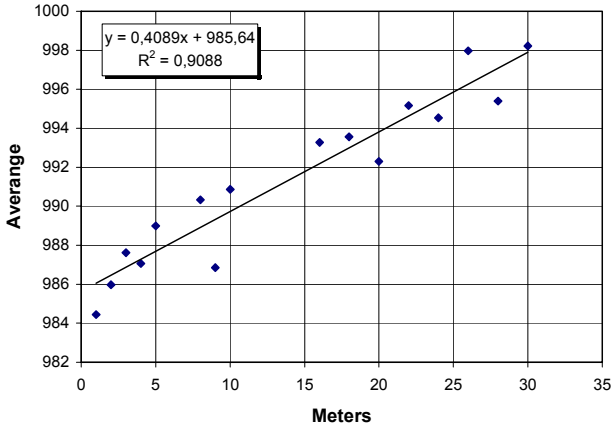


Fig. 7. Average RTT/ distance in specific stay.

The second environment was a hall.

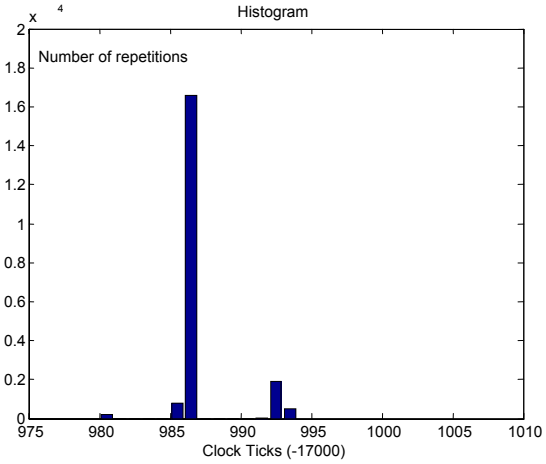


Fig. 8. 2 meters.

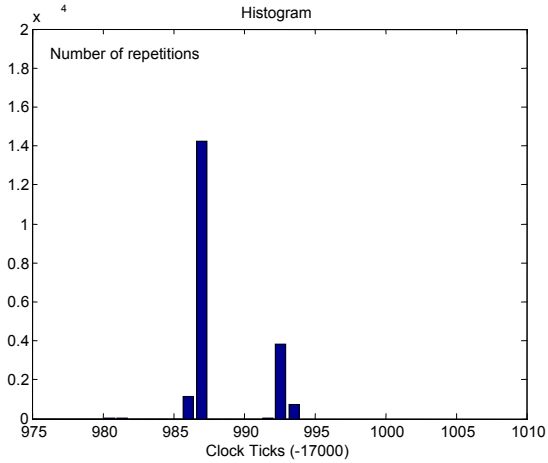


Fig. 9. 6 meters.

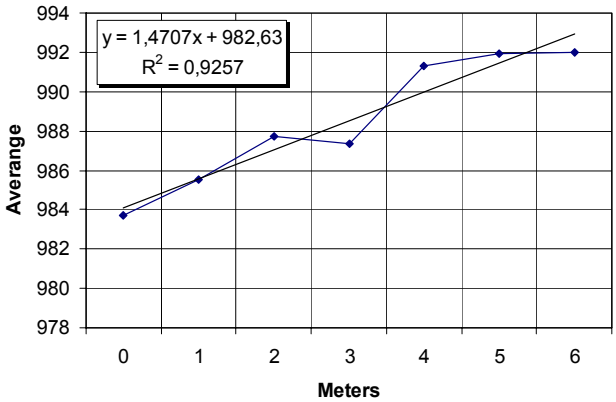


Fig. 10. Average RTT/ distance in specific stay.

The results obtained in the hall, as we can see on the graphs above, are even more promising. The batch means method has been performed for each test, concluding that the most efficient number of transmissions varies between 100 and 1000. On the next figures the percentage of successes with errors of +/- 1 meters using 100, 500 and 1000 transmissions is showed.

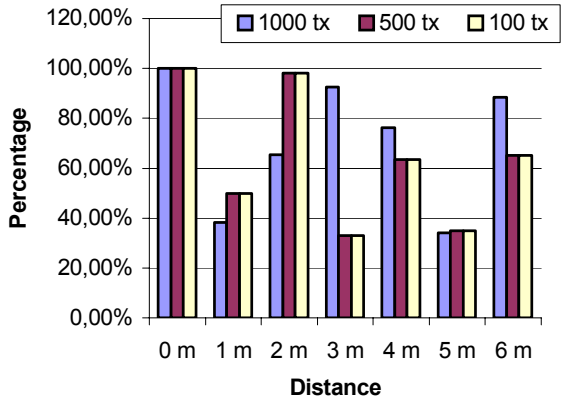


Fig. 11. Percentage of success in model Hall.

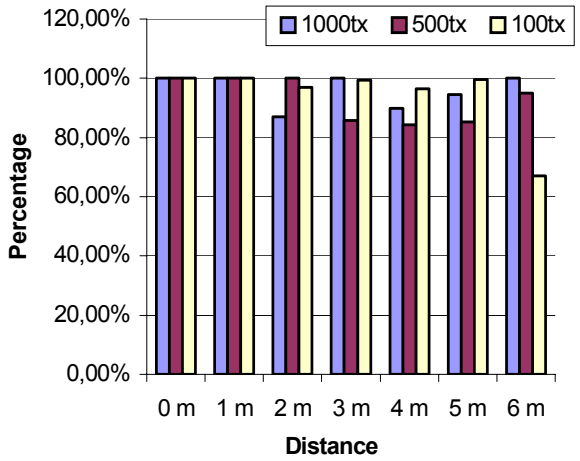


Fig. 12. Percentage of success with an error of 1 meter**.**

VI. CONCLUSIONS AND FUTURE WORK

In this study, a ranging system implemented on a ZigBee network has been performed. The results prove that applying the measure of RTT to this kind of network can achieve accurate ranging capabilities: in 90% of the cases, a maximum error of +/- 1 meter is obtained. It is also demonstrated that 100 transmissions of MAC data frames are enough to achieve good precisions; with the advantage of obtaining it wasting less time. In fact, only 3 seconds are needed to estimate the distance using 100 transmissions.

Thus, a similar precision to the ones obtained by other commercial systems and studies based on wireless networks has been achieved. Moreover, this precision has been accomplished with a cheaper system.

The results of this research can be applied to indoor location. Our future work will consist on applying multilateration techniques on a complete network, to be able to determine the position of a mobile node.

ACKNOWLEDGEMENTS

The current project has been financed and supported by *Miguel Hernández University*.

REFERENCES

[1] Sherratt, R.S. Makino, S. *"Numerical precision requirements on the multiband ultra-wideband system for practical consumer electronic devices"* IEEE Transactions on Consumer Electronics, Vol. 51, Issue 2, pp: 386- 392, May 2005.

[2] Yasuhiro Fukuju et al., "*DOLPHIN: An Autonomous Indoor Positioning System in Ubiquitous Computing Environment*", IEEE Workshop on Software Technologies For Future Embedded Systems, 2003.

[3] Miguel Rodríguez et al., "*Blueps: Sistema de localización en interiores utilizando Bluetooth*", URSI 2005.

[4] M. Ciurana et al., "*A ranging system with IEEE 802.11 data frames*", Radio and Wireless Symposium, IEEE, january 2007.

[5] IEEE 802.15.4, "*Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*", February 2003.

[6] ZigBee Alliance, *"ZigBee Specification"*, documento 053474r13, December 2006.

[7] *CC2420DBK User's Manual: Demonstration Board Kit*, rev. 1.3, Chipcon Products from Texas Instruments, 2004

[8] P. Corral, M. Mompó "*Diseño e implementación de una red inalámbrica basada en el estándar 802.11 de área extensa en la población de Montaverner*", XX Simposium Nacional de la Unión Científica Internacional de Radio, Gandia (España), September 2005..

[9] Tadeusz A.Wysocki y Hans-Jürgen Zepernick, "Characterization of the indoor radio propagation channel at 2.4 GHz", Journal of Telecommunications and Information Technology, 2000.

[10] H. Hashemi, *"The indoor radio propagation channel"*, Proceedings of the IEEE, Vol. 81, No.7, pp. 943-968, july 1993.

[11] Raj Jain, *"Art of Computer Systems Performance Analysis Techniques for Experimental Design Measurements Simulation and Modeling"*, Wiley Computer Publishing, John Wiley & Sons, Inc., January 1991, pp. 442-449.

Performance analysis of the association of the routing protocols AODV and DSR with the *Gossip* algorithm and the *Quorum* system

Renata Lopes Rosa, José Roberto de A. Amazonas

***Abstract*— Nowadays, the utilization of ad-hoc networks is increasing and due to the high probability of node damage, fault tolerance has become an important reliability issue. In this work we analyze the impact of the association of semi-probabilistic routing algorithms to contents replication strategies on the faultless network to verify if such techniques are viable candidates to implement fault tolerance.**

***Index Terms*— fault tolerance, *Gossip*, *Quorum*, ad-hoc networks, routing protocols.**

I. Introduction

AD-HOC networks are designated as non-structured networks and their use is increasing in a steady pace. In these networks, nodes can be included or excluded without the need of a previous configuration. Due to the nodes' high damage sensitivity, great mobility and absence of a central node, it is necessary to devise ways of increasing the fault tolerance, either by means of routing protocols or their association with other protocols and systems that may help the network survival.

This work has the objective of analyzing fault tolerant networks and to study means to minimize the performance degradation due to the nodes' mobility by means of improved routing protocols and data replication systems. In this specific article, the impact of semi-probabilistic routing algorithms, known as *Gossip*, and data replication systems, known as *Quorum*, on the networks performance in a faultless environment, is studied. As it will be seen, the decrease of both the overhead load and the total consumed energy, make these strategies viable candidates to implement fault tolerance.

Among the several ad-hoc networks routing protocols, we selected the AODV *(Ad hoc On-Demand Distance Vector)* e DSR *(Dynamic Source Routing)*, which are classified as on-demand protocols, because of their large acceptance and the existence of many studies that serve as reference for comparison [1], [2] and [3].

As previously mentioned, besides the routing protocols, this work also studied:

- the semi-probabilistic routing information relay scheme named *Gossip*;
- the data replication system named *Quorum*.

Besides the *Gossip*, there are other kinds of algorithms reported in the literature, as [9], [10] and [11]. However, the *Gossip* has been chosen because of the decrease of the number of AODV messages sent between nodes in order to find a path from a source node and a destination node, reducing the overhead load. According to [6], a 35% less overhead messages has been obtained than the flooding (information spreading among all network nodes) technique reported in [4] and [5]. Due to the *Gossip's* degree of redundancy of information distribution, it has the the characteristics of fault tolerance and load sharing among the network nodes.

The results reported in [6] were based only on the AODV protocol. In this work, we have also analyzed the *Gossip* impact on the DSR protocol.

Additionally, the association of the data replication system *Quorum* with the AODV and DSR protocols is studied, and its association with *Gossip* is evaluated in terms of the network throughput.

This being so, this work extends the results reported in [6] by the study of the DSR and the *Quorum*.

After this introduction, Section II presents a theoretical revision of ad-hoc networks routing protocols (with emphasis on AODV and DSR), the *Gossip* algorithm and *Quorum* system; Section III introduces the simulation scenario; Section IV discusses the experimental results; and Section V is a summary of our conclusions and indication of future work.

II. Theoretical Revision

The routing protocols are classified in the categories shown in Fig. 1 and detailed in the following:

- Based on routing information update mechanisms:
  - Proactive (or table-driven): each node keeps up-to-date information about the network topology, i. e., all nodes have complete knowledge about the paths to all other nodes by means of frequent exchange of routing information. These information are usually distributed by means of flooding through all the network. Whenever a node needs a path to a destination, an appropriate algorithm is used to find a route based on the topology or best route information kept by the node itself;
  - Reactive (or on-demand): the nodes don't periodically exchange routing information, but they find the path, whenever it is necessary, by means of a route discovery process between the interested nodes;
  - Hybrid: combination of the before mentioned categories - table-driven and on-demand.
- Based on the use of temporal information:
  - Using past temporal information: they use information of the past state of the link at the time of taking routing decisions;

  - Using future temporal information: they have a future expectation about the state of the link and take approximate routing decisions.
- Based on topology information organization:
  - Flat routing: they use an addressing scheme similar to that used in IEEE 802.3 networks;
  - Hierarchical routing: they make use of a logical hierarchy and an associated addressing scheme.
- Based on utilization of specific resources:
  - Power-aware: they aim at minimizing the energy consumption of a node or of the whole network;
  - According to geographic information: they tend to improve the routing performance and to reduce the control overhead using geographic information of the network.

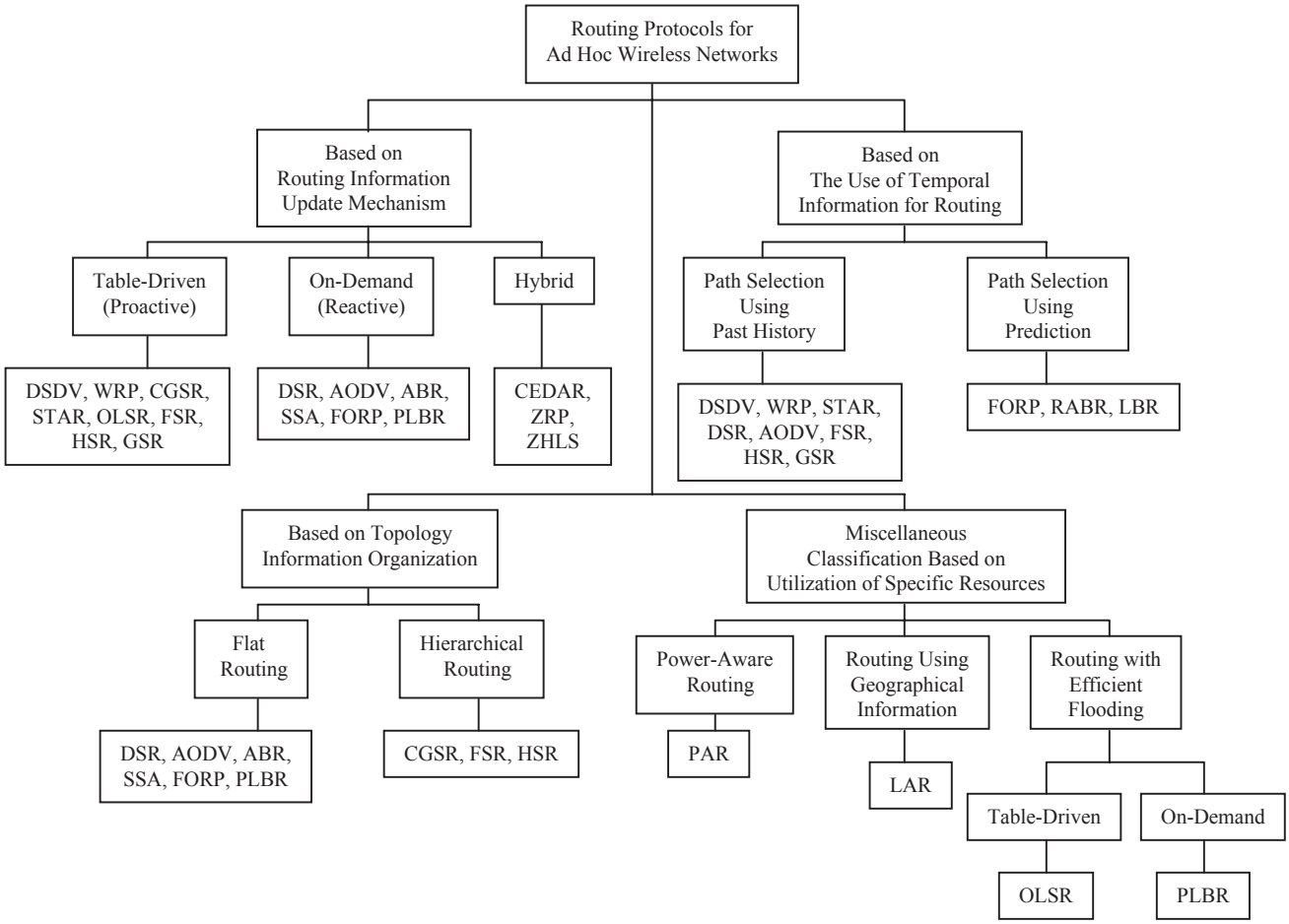


Fig. 1. Routing protocols category [14]

A. AODV Protocol

This protocol use the following variables:

- Source sequence number (SrcSeqNum): sequence number generated by the source node;
- Destination sequence number (DestSeqNum): ever increasing sequence number, used as a means to verify how recent a route is, because a DestSeqNum with a larger value will represent a more up-to-date route;
- Source identifier (SrcID): it specifies a source node that intends to send packets to a determined network node;
- Destination identifier (DestID): it specifies a destination node that shall receive packets from a determined network node;
- Broadcast identifier (BcastID): when several broadcasts are sent into the network, each of them receives an identifier associated to a given RREQ message enabling its identification. It is incremented when any new route request is sent;
- Messsages' time to live (TTL): a counter that represents the maximum time that can elapse before a message is discarded;
- Hops count : number of necessary hops to reach a given node from a specific source node.

And the following messages:

- Hello: these messages are periodically sent to confirm the connectivity between neighbors;
- RouteRequest (RREQ): route request messages are sent to the entire network to find a destination node, when the

route is not known anymore;
- RouteReply (RREP): this is the answer to a route request specifying how to reach a destination node;
- Route Error (RERR): it is a link failure warning. When a link between two nodes comes down a RERR is sent;

When a node wants to send a packet to another node and does not know a route yet, the source node sends a RouteRequest (with SrcID, DestID, SrcSeqNum, DestSeqNum, BcastID and TTL) to all its neighbors by diffusion. If a neighbor is not the destination or if it does know a valid route to the destination, it forwards the RREQ to all its neighbors. This flooding process is repeated, till the requisition reaches either the destination or a node that knows a route to the destination.

A destination node answers to a RREQ by sending a RREP message, through the reverse path, that contains the source and destination addresses, the destination sequence number, the hops count that is incremented by one at each hop, and its TTL. Before sending a RREP, the destination node updates its sequence number as the maximum between the present value and the value received in the RREQ message. At each node traversed by the RREP message on the reverse path, the next hop to reach the destination is stored in an entry concerning the destination, i. e., the neighbor's address from which the answer has been received and the destination's sequence number.

The movement of an active node may make a link that was being used to come down. In this situation an error message (RERR), warning about the link that is down, is sent to all affected nodes. Thus, each node that receives a RERR must decide to forward or not this message, besides to update its routing table, registering that the destinations specified in the message are unreachable and updating their sequence numbers.

Fig. 2 illustrates the AODV's RREQ and RREP messages in the network. The RREQ broadcast is initiated by node 1 that send a RREQ to nodes 2, 3 and 4. By their turn, they verify if they know a route to node 10 and they forward the RREQ to their directly connected neighbors, i. e., nodes 5, 6 and 7. As the node 7 knows a route to the destination node 10, it then sends a RREP to the source node. If any other node also knew a route to node 10, it would also answer with a RREP to the source node, as for example, node 8.

B. DSR Protocol

The DSR protocol uses the same messages as the AODV protocol, working in a similar way, with the difference that it employs a route cache that stores all possible information extracted from the source route of a data packet. In the AODV only the most recent route is stored, the other ones are discarded.

It has been designed to consume less bandwidth, due to the suppression of the periodic messages about the network state used to update the routing tables. This is an unique characteristic of this protocol, when compared with other on-demand routing protocols, because the active nodes do not need to transmit periodic Hello messages to their neighbors to notify they are alive.

The DSR works with a maximum number of 200 nodes and from 5 to 10 hops, while the AODV works with more than 200 nodes. Other differences between theses protocols are illustrated in Table I.

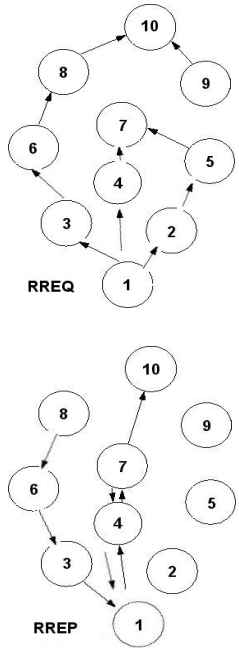


Fig. 2. AODV's RREQ and RREP messages

TABLE I
DIFFERENCES BETWEEN THE AODV AND DSR

	Support to multiple paths	Send periodic messages	Intermediate nodes need intelligence to find a route	Routing information are exchanged with neighbors at different levels
AODV	Yes	Yes	Yes	No
DSR	No	No	No	Yes

C. Gossip algorithm

The *Gossip* algorithm is a probabilistic multicast protocol in which a node forwards information only to a given number of neighbor nodes, that is established by means of a probability setup in the algorithm. In this way, the network flooding is avoided, unless the probability is set to 1, meaning that 100% of the neighbors should receive messages.

Each node in this kind of *Gossip* algorithm keeps only a partial view of the entire system, that is employed to guide the selection of nodes that will participate in the information exchange.

The fact that the senders randomly choose to whom they send messages, makes the *Gossip* fault tolerant and completely distributed.

There are different kinds of *Gossip* algorithms, and the most common are;

- GOSSIP1 (p, k): it sends a broadcast to the first k hops, allowing the initial messages to be forwarded with a

probability p = 1 (100%). After receiving a message for the first time, the node forwards it with p < 1.

- GOSSIP2 (p1, k, p2, n): the information forwarding of a given node depends on its relative position to the source node and on the number of neighbor nodes (n).
- GOSSIP3 (p1, k, m): the information forwarding of a given node is made according to a sufficient number of retransmissions (m). If a node decides not to forward a message and after that it receives less than m copies of messages, it sends a broadcast message at once.

This work employs the GOSSIP2 algorithm, whose parameters are described hereafter:

- p1 is the probability of a node forwarding the *gossip* if the number of its neighbors is greater than n.
- k is the number of initial hops for which the *gossip* is forwarded with propbability equal to 1.
- p2 (p2>p1) is the probability of a node forwarding the *gossip* if the number of its neighbors is less than n.
- n is the number of neighbors threshold.
- Fanout: is the selected number of nodes (t) to receive the message sent by the emitter. High values of fanout ensure a higher level of fault tolerance at the expenses of a higher degree of redundant traffic generation in the network.

When a node receives a message for the first time, it stores the message and forwards it to a given number of nodes (fanout). These nodes, by their turn, select some of their neighbors and propagate the message. If a node receives a message that it already knows, it ignores it.

The bottom line of the *Gossip* algorithm is to make all nodes participate in the same manner of the information dissemination. Thus, if a node wants to send a *broadcast* message, it randomly selects t nodes (its fanout) and sends the message to them.

Fig. 3 is a step-by-step illustration of how the information is disseminated through the network. The black nodes are information disseminators and receivers. The gray nodes are only receivers. The black nodes select an average number of 2 neighbors (initially only the gray nodes and then gray and black nodes). This number is given by the information forwarding probability, 40-50%. The path followed by the information is represented by the lines in the figure. The nodes that receive the information, also select two other neighbors and forward the information to them. The fact that some of the neighbors have already received the information (black nodes) does not prevent them to receive the same information again. This characteristics makes the *Gossip* a redundant algorithm. Eventually, all nodes receive the information (black nodes of the step 6).

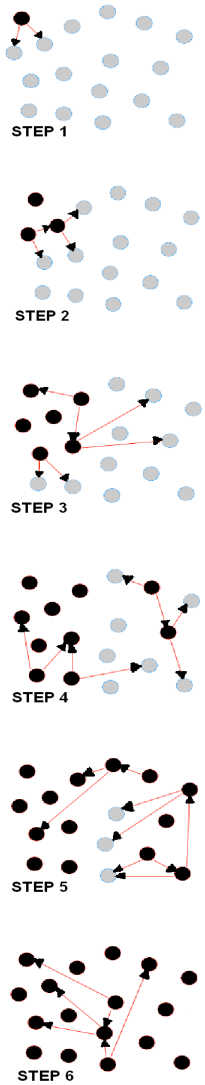


Fig. 3. A step-by-step illustration of the *Gossip's* information dissemination

D. The Quorum System

The *Quorum* system is a set of tools used to implement high availability of distributed shared memory. Consider a set of servers that are divided in a certain number of subsets named *Quorums*. The intersection between any pair of *Quorums* is always non-empty. As a principle, a *Quorum* system says that if a shared data source is stored in a set of servers, then read and write operations need to be implemented in only one *Quorum*. The intersection property of *Quorums* ensures that each reader has access to the most recent data. Any practical *Quorum* system must take into account that some of the servers may fail and, even in this situation, they must exchange data among them and continue to work.

This work presents a system made of an arbitrary number of clients and a set S of a fixed number of servers.

There is a *Quorum* system Q if:

$$Q = Qr + Qw \tag{1}$$

where Qr is a set of quorums used for read operations (Read Quorum) and Qw is a set of quorums used for write operations (Write Quorum). Any Read Quorum and Write Quorum pair have a non-empty intersection.

The *Quorum* system used in this work is based on [12] and its functioning is as follows:

- To write data on a *Quorum's* server, the client checks the Read Quorum's servers to choose a timestamp larger than

any existing timestamp for any data already saved in the servers and send the data to the Write Quorum's servers associated to the largest chosen timestamp.
- To read data from a *Quorum's* server, the client checks the Read Quorum's servers and looks for the most recent data, i. e., those that have the largest timestamp, and return the data to the user.

E. Performance Parameters

The performance parameters chosen to evaluate a fault tolerant system were:

- RREQ: messages sent by a node to a set of nodes with the objective of find a route. The total number of RREQs has been measured.
- Throughput (bps): parameter that indicates the effective data transmitted in bits and measured in an time interval.
- Latency (s): represents the elapsed time for a packet to go from a source node to a destination node. It is given by the sum of sending and receiving delays of CBR (Constant Bit Rate) flows.
- Packet loss (%): represents the difference of the number of sent packets and the number of received packets over the number of sent packets, given in percentages.
- Energy consumption (mWhr): is given by the sum of energy consumption employed on receiving (RX) and transmitting (TX) information. As an example, the energy consumed by data transmission is computed by 2:

$$\text{Consumption} = [(\text{TxPowerCoef} \times \text{txPower}) + \text{TxPowerOffset}] \times \text{txDuration} \quad (2)$$

where TxPowerCoef and TxPowerOffset are statistically defined by the WaveLAN specifications [13] as 16/s and 900 mW, respectively. txPower is proportional to the distance the signal has to travel.

III. Simulation Scenario

Two network simulators have been analyzed, namely: NS-2 [15] and Glomosim 2.03 [16] educational version. Both of them are free software and have been extensively used in research providing many research references for comparison.

The Glomosim 2.03 is a scalable discrete events simulator developed by the UCLA and, besides the forementioned arguments, it has been chosen because it has been used in [3] that is an important reference for this work.

The simulation scenario has been defined as:

- 1000 m x 1000 m area;
- simulation time: 5 minutes;
- number of nodes: 80 - uniformly distributed over a grid of degree equal to 11,7 (average number of immediate neighbors);
- traffic: a set of 30 CBR flows between of randomly chosen node pairs. The transmission rate is 2 packets per second; each packet has 512 bytes;
- mobility: five mobility situations have been defined by their pause duration - 50, 100, 150, 200 and 250 seconds. A no mobility scenario has also been simulated;
- protocols: the following combination of protocols have been simulated:
  - AODV;
  - DSR;
  - AODV + *Gossip*
  - DSR + *Gossip*
  - AODV + *Quorum*

IV. Experimental Results and Discussion

In this work, the number of Route Requests (RREQ), the total throughput in bps, the total network's latency, the total energy consumption in mWhr and the percentage of lost packets have been measured for the different simulation scenarios.

According to Fig. 4 we realize that the AODV generates a greater number of RREQs than the others because it employs a system of periodical signaling by means of its Hello messages. The Hello messages ensure that the AODV protocol always has up-to-date routes. Associating the AODV with the *Gossip*, the number of RREQs decreases as a direct consequence of the reduced number of messages employed by the *Gossip*, due to the probabilistic factor used for deciding about the messages forwarding that avoids the flooding behavior. On the other hand, the DSR presents a low value of the number RREQs due to its characteristic of scarce signaling that can be used because it employs a cache system instead of looking for the most up-to-date routes. Associating the DSR with the *Gossip*, there has been a slight increase on the number of RREQs because the *Gossip's* redundancy of information forwarding.

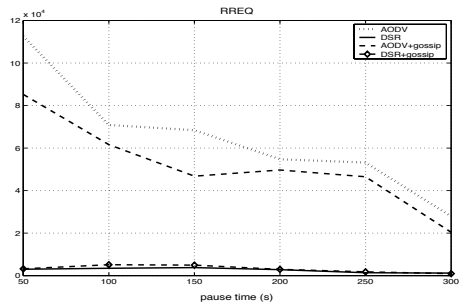


Fig. 4. RREQs messages

According to Fig. 5 it can be seen that the AODV provides a higher throughput than the DSR in the mobility scenario. This happens because the AODV employs a scheme of more up-to-date routes and, consequently, existing routes between nodes. Without mobility, the DSR shows a better performance than the AODV because of its low signaling overhead. In the mobility scenario, the probability that the information stored in the DSR's caches is invalid increases, decreasing the total throughput. The association of the *Gossip* with both protocols has little influence on their performance. However, the slight decrease of throughput observed can be explained by the fact that the *Gossip* sacrifices the packet delivery to keep the overhead load low.

According to Fig. 6 the DSR's performance in terms of latency is better than the AODV's. This can be easily explained

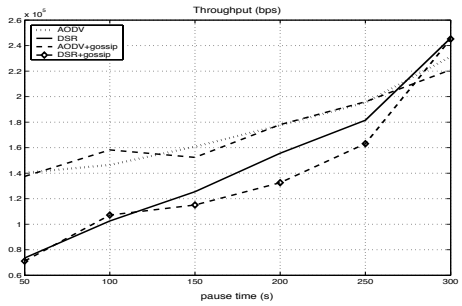


Fig. 5. Throughput (bps)

because the DSR employs a cache system and the AODV can be said to be more reactive, in the sense that it has to find new routes by sending RREQs over the network. Thus, the DSR does not have to evaluate new routes every time and the AODV protocol incurs in a delay to update the routing tables. Associating the *Gossip* to both protocols increases the latency because there is a delay to compute the neighbors to which the information should be forwarded.

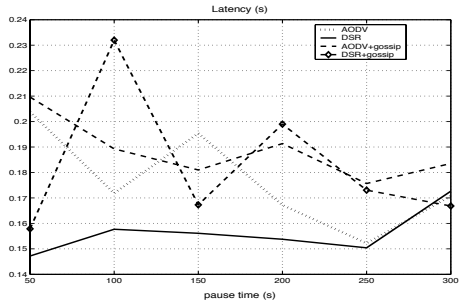


Fig. 6. Latency (s)

In summary, despite the fact that the *Gossip* algorithm slightly sacrifices the latency and throughput performance characteristics, it decreases the number of overhead RREQs messages that represent an overload and a factor of energy consumption. According to Fig. 7, it can be seen that the standard GOSSIP2 algorithm provides some energy consumption reduction. The observed reduction has not been very important but it indicates that making the *Gossip* power-aware may be an interesting research approach.

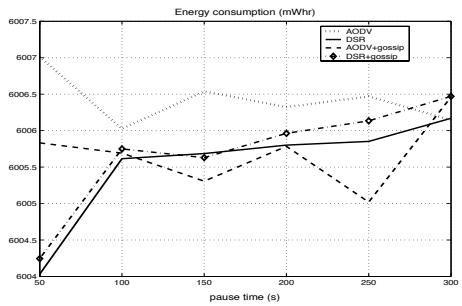


Fig. 7. Energy consumption (mWhr)

This work has also verified that the *Quorum* system in a fault-free scenario does not impact the performance results of any parameter.

V. Conclusions and Future Work

This work revised the most used routing protocols in ad-hoc networks, AODV and DSR, a semi-probalibilistic routing algorithm, *Gossip*, and a data replication scheme, *Quorum*.

For a fault-free simulation scenario, considering no mobility and five different mobility situations, the protocols' performances have been evaluated in terms of the number of RREQs messages, throughput, latency and energy consumption. It has been verified that the *Gossip* algorithm reduces the number of overload RREQs messages and slightly sacrifices the throughput and latency values. It has also been observed a minor energy consumption reduction. The *Quorum* system does not impact the performance for the fault-free scenario.

This being so, the *Gossip* algorithm and the *Quorum* system are interesting candidates to implement fault tolerance in ad-hoc networks.

As future work, we propose to analyze the performance degradation against fault tolerance for all the situations already analyzed in this work. Additionally, we intend to develop a power-aware version of the *Gossip* algorithm to improve the energy consumption reduction, that can be based on introducing a sleep mode state for the nodes that are not receiving any information in a determined instant of time. The new scheme will be compared with other solutions reported in the literature.

References

[1] S. Junk, N. Hundewale, A. Zelikovsky, *Node caching enhancement of reactive ad hoc routing protocols [MANET]*,Wireless Communications and Networking Conference, 2005 IEEE, Vol. 4, 13-17 March 2005, pp 1970 - 1975.

[2] N. Moghim, F. Hendessi, N. Movehhedinia, *An improvement on ad-hoc wireless network routing based on AODV*, Communication Systems, 2002, The 8th International Conference, Vol. 2, pp 1068 - 1070.

[3] Z. Xiaofeng, M. Shunliang, W. Youzheng, W. Jing , *Stable enhancement for AODV routing protocol*,Personal, Indoor and Mobile Radio Communications, 2003, 14th IEEE Proceedings on Vol. 1, pp 201 - 205.

[4] T. Korkmaz, M. Krunz, *Hybrid flooding and tree-based broadcasting for reliable and efficient link-state dissemination*, Global Telecommunications Conference, IEEE Vol. 3, 17-21 Nov. 2002, pp 2400 - 2404.

[5] C. Jaihyung, J. Breen, *A flood routing method for data networks Information* , Communications and Signal Processing, 1997, Proceedings of 1997 International Conference on Vol. 3, pp 1418 - 1422.

[6] H. Ling, D. Mosse, T. Znati,*Coverage-based probabilistic forwarding in ad hoc routing*, Computer Communications and Networks, 2005, Proceedings. 14th International Conference on 17-19 Oct. 2005, pp 13 - 18.

[7] L. Jun, P.T. Eugster, J.P. Hubaux, *Pilot: probabilistic lightweight group communication system for ad hoc networks*, Mobile Computing, IEEE Transactions on Vol. 3, Issue 2, April-June 2004, pp 164 - 179.

[8] Z.J. Haas,J.Y. Halpern, L. Li , *Gossip-based ad hoc routing Networking*, IEEE/ACM Transactions on Volume 14, Issue 3, June 2006, pp 479 - 491.

[9] J. Tingyao, L. Qinghua, *A self-stabilizing distributed multicast algorithm for mobile ad-hoc networks*, Computer and Information Technology, 2004. CIT 04. The Fourth International Conference on 14-16 Sept. 2004, pp 499 - 502.

[10] S. Mueller, D. Ghosal, *Analysis of a distributed algorithm to determine multiple routes with path diversity in ad hoc networks*, WIOPT 2005. Third International Symposium on 3-7 April 2005, pp 277 - 285.

[11] Z. Chun-Xiao ,W. Guang-Xing , *Routing protocol based on fuzzy regression for MANET Machine Learning and Cybernetics* , 2004. Proceedings of 2004 International Conference on Volume 3, 26-29 Aug. 2004, pp 1811 - 1815.

[12] Martin, J.-P.; Alvisi, L.; Dahlin, M.,*Small byzantine quorum systems*, Proceedings. International Conference on 23-26 June 2002, pp 374 - 383.

[13] NCR WaveLAN PC-AT Installation and Operations manual, part number ST-2119-09, revision number 008-0127167 Rev. B, copyright 1990,1991 by NCR Corporation.
[14] C. S. R. Murthy, B. S. Ghosal, *Ad Hoc Wireless Networks*, Prentice Hall.
[15] *NS-2 Network Simulator*, http://www.isi.edu/nsnam/ns/.
[16] *Glomosim - Global Mobile Information System Simulation Library*, http://pcl.cs.ucla.edu/projects/glomosim/.

José Roberto Amazonas graduated in electrical engineering from the Escola Politécnica of the University of São Paulo (Epusp), Brazil, in 1979. He received the MSc, PhD and postdoctoral degrees from Epusp in 1983, 1988 and 1996, respectively.

He is associate professor of the Telecommunications and Control Engineering Department at Epusp, where he is in charge of optical communications and high-speed communications networks education and research. He held various positions in universities in Brazil and Europe. He has also led research in partnership with several brazilian, european and north-american companies.

His research interests are in the area of optical communications, wired and wireless networks, quality of service (QoS) and remote learning.

Renata Lopes Rosa graduated in computer science from the UNIFEI, Brazil, in 2000. She is MSc student of the Telecommunications and Control Engineering Department of the University of São Paulo (Epusp).

Intelligent Medium Access for MANETs with Principle of Circularity

Prasanna S J
Hughes Systique Corporation
Prasanna.sj@hsc.com

Anjini Shukla
University of California, Santa Barbara
anjinishukla@umail.ucsb.edu

***Abstract* — MANET is characterized by highly changing network topologies and connectivities. Due to such highly dynamic scenarios, the conventional IEEE 802.11 MAC protocol has various shortcomings with regard to MANETs. The RTS/CTS access scheme, designed to reduce the number of collisions in an IEEE 802.11 network, is known to exhibit problems due to Deaf nodes, the imbalance between the interference range and the communication range of the nodes, and scenarios in which nodes are unnecessarily silenced, thus preventing parallel transmissions to take place. We present an approach for enhancing the performance of the IEEE 802.11 MAC protocol by introducing a new paradigm of Circularity for selectively discarding, delaying or extending the circularity satisfied RTS, CTS or DIFS to allow certain parallel transmissions to proceed and obviate some ACK/DATA collisions which is one of the major issue due to the formation of Deaf nodes to enable MANETs. We implemented the circularity approach in ns-2 simulator. Through a series of experiments, we show that the circularity approach provides a significant improvement in the throughput and contributes to a reduction of the number of collisions in most scenarios.**

***Index Terms* — MANETs, MAC layer, Deaf/Masked nodes, Throughput, RTS/CTS**

I. INTRODUCTION

A Mobile Adhoc Network (MANET) is a network architecture that can be rapidly deployed without relying on pre-existing fixed network infrastructure. The nodes in a MANET can dynamically join and leave the network, frequently, often without warning, and possibly without disruption to other node's communication. Finally, the nodes in the network can be highly mobile, thus rapidly changing the node constellation and the presence or absence of links.

Since node communication in a dense network happens at the same frequency band, the problem of packet loss due to collision becomes an area of focus. The IEEE 802.11 standard [1] has been primarily designed for wireless LANs and is responsible for scheduling medium access for multiple stations that are contending for the common channel. It uses a medium access scheme based on the Carrier Sense Multiple Access (CSMA) [2] protocol, where a node transmits only if it finds the medium to be idle for a pre-defined Inter Frame Space (IFS). It also uses a Request-To-Send (RTS) and Clear- To-Send (CTS) control packets to coordinate channel access [3] and minimize costly packet collisions in hidden node scenarios. In multi-hop networks, some nodes may not hear control packets from other nodes within the network. This leads to an RTS/CTS exchange with reduced chances of success and increased possibility of packet collisions. The problem is further complicated with packet loss arising due to some transmissions being masked by other on-going transmissions in their neighborhood [4]. The masked node problem is an example of a shortcoming of the IEEE 802.11 when it is used in MANETs.

In [5] the authors propose the selective disengagement of the RTS-CTS handshake in IEEE 802.11. They point out that the 802.11 standard was developed keeping in mind that the carrier sensing range is equal to the transmission range whereas it is 1.78 times of the latter [6]. This means that any node can actually hear transmissions going on two hops away thereby resolving the hidden node problem to some extent. This however leads to another problem: due to the basic CSMA/CA scheme, a sender will transmit neither control nor a data packet if it senses the channel is busy. By disengaging the RTS-CTS handshake based on the number of "CTS timeouts", they enable greater fairness and higher network throughput both due to lesser control overhead and enabling parallel transmissions to take place.

In this paper, we use a similar technique of turning off the RTS-CTS handshake for particular instances along with delaying the transmission of the CTS packet and extending the DIFS to enable parallel transmissions to be completed by obviating costly ACK/DATA collisions. These changes are made to the IEEE 802.11 MAC protocol and simulations are carried out to compare performance differences. However, revamping the entire MAC standard or the transport layer protocol for use in MANETs is absolutely infeasible due to wide adoption of both. Some modifications need to be incorporated to make the proposed and the legacy protocols inter-operable with each other to ensure fairness and optimal resource utilization. Effective changes should be incorporated in a "non-invasive" manner, i.e. the changes should be based on the underlying mechanisms of the current standard, with as few major changes as possible. The modifications proposed in this paper are novice but effective enhancements to the current standard keeping the underlying principles and the workings of the IEEE 802.11 standard almost untouched. Section II of this paper presents Related work with regard to MANETs. In Section III a Novel approach for effective bandwidth utilization is presented, while the Simulation study, and conclusions are made in sections IV, V.

II. RELATED WORK

The IEEE 802.11 standard uses the Distributed Coordination Function (DCF) which includes Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) as the fundamental access technique. There are two primary access methods in IEEE 802.11: the basic access and the RTS/CTS access method. The basic access scheme involves only a reliable transfer of the data packets from the source to the destination by using ACK

packets. In the RTS/CTS access scheme, the RTS and CTS control packets are first exchanged and the channel is reserved exclusively between the source and destination, which is followed by the DATA/ACK packet transmission. This RTS/CTS dialog helps in the implementation of the virtual carrier sensing mechanism which is also accompanied by physical carrier sensing in IEEE 802.11 DCF. The RTS/CTS frames contain a duration field which is used by the neighbors to set a specific Network Allocation Vector (NAV) during which the nodes are sent to ”silence” state during which the packet exchanges are being carried out between the sender and receiver nodes. On the other hand, physical carrier sensing is implemented using Interframe spaces. After a channel is sensed idle for a DCF Interframe Space (DIFS) time interval, the back-off procedure is invoked by the station which has to send the data. A Short Interframe Space (SIFS) is used to separate transmissions belonging to a single session (CTS, DATA and ACK packets). Extensive work has been carried out on the IEEE 802.11 DCF and the usage of the RTS/CTS mechanism [7],[8].

In [6], the authors present mathematical proof that the interference range is typically 1.78 times the communication range. Even though a node may not be within the transmission range to successfully receive a CTS packet, it may still be the cause of interference at the sender. As a simple solution to the above problem, it is suggested that a node should only reply with a CTS when the received RTS is above a certain receiving power threshold, i.e., it is sufficiently close to the transmitter and hence avoid perceptible interference from other nodes. [9] Shows that an optimal carrier sensing range along with an appropriate transmission range and an interference model significantly increases network throughput. Another related work [5] tunes the RTS/CTS exchange by selectively disengaging it when there are occurrences of the CTS not being returned. In [10], the author states with increasing network complexity and/or node mobility, a node which has not heard of a RTS or CTS packet may migrate into the footprint of a receiver and destroy a DATA packet by initiating its own transmission, oblivious to its surroundings. [4] Points out another type of nodes in the same class as that of hidden nodes which are termed”masked” nodes. The authors show that the RTS/CTS exchange is not enough under perfect operating conditions since neighbor nodes are masked by other on-going transmissions nearby. Masked nodes cannot decode the RTS/CTS packets correctly and may end up causing Data/ACK packet collisions later on.

Other widely used approaches to resolving medium access include splitting the available channel into separate control and data subchannels. [11] Proposes two schemes that attempt to pipeline contention resolution with data transmission to reduce the idle waiting time and decrease overall delay. For wireless environments, it also proposes a partial pipeline approach to overcome the shortcomings of the total pipelining scheme. The authors show that with proper channel division, a net throughput increase can be obtained. The authors of [12] propose, the Bi-directional Multi-Channel MAC protocol, where the bandwidth is divided into one control channel and several data channels. It is bi-directional because the receiver may also send his own data packet (if any) to the sender using any of the other available channels thereby eliminating the need of another RTS/CTS handshake.

III. A NOVEL APPROACH FOR EFFECTIVE BANDWIDTH UTILIZATION

A. *Motivation*

The 802.11 protocol was primarily designed with WLANs in mind and in the case of MANETs the existing protocol has simply been ported. Hence it is of utmost need that the protocol is fine-tuned keeping in mind the infrastructure less MANETs which are highly mobile. Due to this node mobility, the various problems (hidden nodes, exposed nodes, deaf nodes etc) crop up and with their presence the total number of packet collisions within the network increase manifold. These packet collisions decrease the net throughput of the network, increase the end-to-end delay between nodes and ultimately lead to inefficient bandwidth utilization. Packet collisions may be categorized as follows:

- RTS packets colliding with other RTS packets when two stations start transmission simultaneously.
- DATA packets colliding with RTS/CTS packets from masked/deaf nodes.
- ACK packets colliding with RTS/CTS packets from masked/deaf nodes.

In this work, we define a channel-access scheme for better channel utilization while encouraging parallel transmissions from other non-interfering nodes.

B. *Circularity*

Circularity is defined as a number which enables the identification of specific groups of control packets sent from each node. The total number of packets in each group is equal to its circularity value and the last packet in the group is termed as the circularity satisfied packet. These circularity value pairs are used to selectively discord RTS packets, delay CTS packets and extending DIFS before transmission. For example, if the circularity value is defined as four, then we divide the RTS/CTS packets being created in each node and DIFS into groups of four and the first such packet in a group is the circularity-satisfied packet. Hence, every fourth packet being created by the node and every fourth DIFS time interval (i.e. every multiple of four) is circularity-satisfied. Mathematically, a packet is circularity-satisfied if:

$$N \text{ modulo } c = 0$$

Where N is the current count of the number of packets generated (RTS/CTS) or DIFS and c is the circularity value for the particular node. Essential characteristics of applying circularity to RTS/CTS packets and DIFS time interval include the following:

- By identifying certain RTS/CTS packets or DIFS interval as circularity-satisfied, we induce these packets or interval to behave differently from the rest. The structure of the packet (size, headers etc) remains the same, so there are no explicit changes which have to be made to the standard IEEE 802.11 protocol.

- There is no absolute grouping of packets taking place to identify the circularitysatisfied packets. The packets are identified through the simple mathematical formula above.
- There are different RTS/CTS and DIFS interval circularity values. For simplicity, in this paper, we have considered all of them to be the same. Thus, for the above mentioned example, every fourth RTS packet, fourth CTS packet emanating from one particular node will be the circularity-satisfied packet and fourth DIFS interval is Circularity satisfied time interval.

Essentially, circularity is just a scheme to identify certain packets which behave differently than the rest. Thus, if the RTS circularity value is considered to be four, then a node which sends out a total of 20 RTS packets during the network lifetime will in effect have identified five (20/4) of them as circularity satisfied. Similarly, considering CTS and DIFS circularity as five, then a node sending out 20 CTS packets and 20 DIFS interval would have identified four (20/5) of them as circularity-satisfied. In our scheme, each node is assigned a specific circularity value for both their RTS and CTS packets and also for DIFS time interval. These value pairs are used to discard circularity-satisfied RTS packets, irrespective of the existing scenario. Similarly, the circularity satisfied CTS packets are delayed to allow parallel DATA or ACK transmissions and Circularity satisfied DIFS extended to obviate ACK/DATA collisions. In our experimental setup we have restricted the delay of the CTS packets to one SIFS time interval.

C. *RTS Packet Discarding with Circularity*

Dropping RTS packets is a technique of selectively disengaging the RTS/CTS dialogue for a particular transmission session.

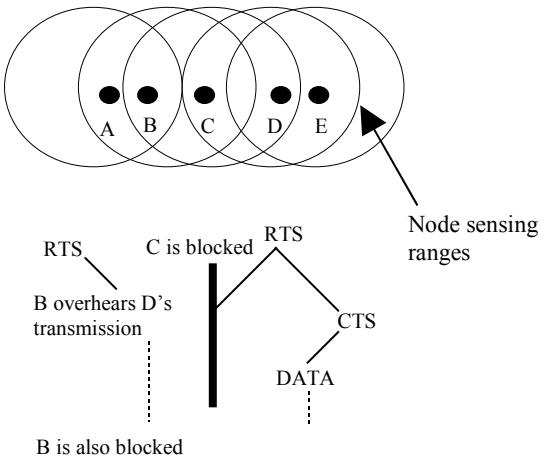


Figure 1: Hidden Node Scenario with sensing ranges being shown

Let us consider the scenario in Figure 1. In this figure we depict the circles as the sensing range (which is roughly 1.78 times the transmission range). Looking at the timeline sequence in the figure, it can be observed that D initiates a transmission to E with an RTS packet. This RTS packet is transmitted to E and C with E sending corresponding CTS. C then gets blocked from transmitting. This is followed by D starting its data packet transfer to E, but this can also be sensed by B as it is within carrier sensing range of D. At this point, B's neighbor A (who cannot hear beyond C) initiates a transmission by sending an RTS packet to B. However, as B can sense D sending a long data packet, it will back-off and not reply with CTS to A. Hence, a parallel transmission is prevented from taking place resulting in decreased network throughput. Another point to be noted is that the flow D to E may capture the channel for a long period of time which results in A giving up re-transmitting RTS packets after the retry count is exceeded and hence reporting a route failure to the routing layer. This will in turn lead to a new process of route discovery and increase network overhead considerably. Disabling the RTS-CTS exchange and simply transmitting the data packet would be of greater effect. We implement this selective RTS-CTS disengagement by identifying certain RTS packets based on their RTS circularity value and dropping them. We discuss possible techniques of setting the circularity values in a later section. The authors in [5] take a more conservative approach by waiting for a number of CTS timeouts to occur before sending the data packet directly. We propose a more aggressive approach of scheduling these packet drops based on the circularity value. The tradeoff is that there may be a higher number of packet collisions occurring but a higher overall network throughput could be obtained due to the parallel transmissions. We take the circularity values 2, 3, and 4 see how RTS-CTS with circularity works. We highlight the cases in the analysis that affect the throughput using **, * and @.

** refers to the cases where the dropping of one or more RTS packets due to circularity gives chance for another RTS packet to make its way through, without any collision.

* refers to the cases where the dropping of more than one RTS packets would lead to no RTS packet transmission. Although we think that we lose a RTS packet, we actually are gaining on the time lost due to collision.

@ refers to the case where the only RTS packet at that particular time would lead to a loss of RTS packet and hence these cases decrease the throughput.

Let t_c be the time lost due to collision.

Let t_r be the time for retransmission of RTS.

So, $t_c > t_r$.

For an example When circularity =4, we have 3 ** and 2 *.

So time saved due to RTS-CTS with circularity is

$4(t_c) + 2(t_c - t_r) = (6t_c - 2t_r)$

No. of additional RTS packets transmitted is (4 - 0) = 4

The above analysis gives a clear picture of how the number of MAC collisions avoided and number of RTS packets lost fluctuate with different values of circularity.

D. *CTS Packet Delay with Circularity*

The concept of delaying the CTS packet is also aimed at making possible the occurrence of parallel transmission within the network. By delaying the transmission of a CTS packet by a small time interval (one SIFS), we aim to help a neighboring

transmission to either continue or complete. Let us consider a sample scenario.

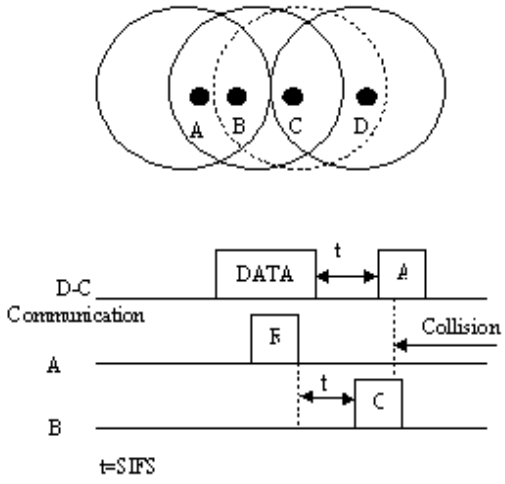


Figure 2: Scenario depicting CTS collision with ACK packet. The circles denote the transmission range of each node

In the Figure 2, we assume that D and C have had a successful RTS-CTS exchange and D has sent its data packet to C. During the course of this transmission, due to the nodes being mobile or if it was earlier a masked node, B comes into transmission range of C. At this point of time, A sends data to B by initiating an RTS-CTS handshake with an RTS packet. Since B is unaware of an ongoing transmission nearby, it responds with its CTS packet which collides with the ACK packet from C. However, if the CTS packet from B was delayed by a small interval, then the ACK packet would be correctly received by D. As only the circularity-satisfied CTS packets are delayed, the impact on the overall delays in the system is reduced. Also it does not guarantee that all CTSACK collisions will be avoided but the primary aim is to reduce the probability of these collisions as much as possible.

E. ***DIFS Extension with Circularity***

Deaf nodes cause both DATA and ACK packet collisions. Eliminating DATA packet collisions, without significantly changing the MAC protocol, appears to be difficult. However, it is possible to eliminate ACK packet collisions by changing the value of the DIFS parameter of IEEE 802.11. Specifically, from Figure 3, we see that if we had DIFS >SIFS + ACK transmit time, and then the ACK packet would not collide. For networks where the propagation delay is non-negligible, if DIFS satisfies the following inequality,

$$DIFS > SIFS + ACK_{transmittime} + 2t_{prop} \quad (1)$$

then ACK packet collisions are eliminated. Here, tprop is the maximum propagation delay between any two nodes in the network who are within the transmission range of each other. We remind the reader that every node is required to sense the channel to be idle for at least DIFS period of time before it is allowed to initiate a communication (by sending an RTS packet or the DATA packet itself). On the other hand, a node that sends a CTS or an ACK, is required to wait for SIFS period of time. Figure 3 shows why the choice of DIFS parameter according to inequality eliminates ACK packet collisions. There are three nodes in this figure, node A, B and C.

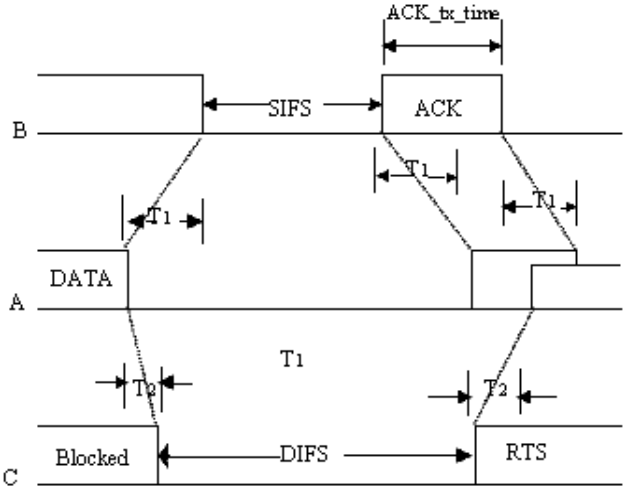


Figure 3: Justification of Equation (1): The RTS packet to reach node A after node A has received the ACK.

Node A is within the transmission range of both nodes B and node C, but node B and node C can not hear each other. T_1 is the propagation time between node A and B and T_2 is the propagation time between node A and C. Node A initially transmits a packet to node B. Node B after receiving the DATA packet, waits for SIFS period of time and transmits the ACK packet. On the other hand, node D, which wishes to transmit a packet, waits for DIFS period of time and transmits its RTS (or DATA packet) packet. Suppose that node A receives the ACK packet from node B during the interval I_1 and the RTS packet during from node C during the interval I_2. Then, from Figure.

$$I_1 = [2T_1 + SIFS;\ 2T_1 + SIFS + ACK_{txtime}] \quad (2)$$

$$I_2 = [2T_2 + DIFS;\ 2T_2 + DIFS + DATA_{txtime}] \quad (3)$$

Since we want the ACK packet to reach node A first, we need

$$DIFS > SIFS + ACK_{transmittime} + 2T_1 - 2T_2 \quad (4)$$

In the worst case, $T_1 = t_{prop}$ and $T_2 = 0$. Substituting these values in inequality (4), we get back inequality (1).

However, an important point to be considered here is that with an increase in DIFS value, the net session time increases with a corresponding probable drop in throughput. The tradeoff may work to an advantage in some cases and may not be there for others. We thus implement this idea with the circularity concept and simulate to observe the results.

IV. SIMULATION STUDY

A. ***Simulation Setup***

We carried out simulations using ns-2 [13] with wireless extensions from the CMU Monarch Group. Changes to the MAC source files were carried out to implement the circularity concept and enable the discarding/delaying and extending of circularity satisfied RTS-CTS packets and DIFS intervals.

Node movement is modeled by the random waypoint mobility model [14] with nodes moving at a speed between 10 and 100 m/s and a pause time of 20s. Each data point plotted is the average of 10 different scenarios with different initial network configurations. The following parameter values used throughout the simulations. The simulation metrics are net throughput, packet delay and packet loss ratio. We define net throughput as the aggregated throughput over all the flows in the network as the foremost performance metric.

Data Rate:	1 Mbps
RTS Size:	20 Byte
CTS Size:	14 Byte
Data Packet Size:	512 Byte
ACK Size:	14 Byte
Slot Time:	20us
SIFS:	30us
DIFS:	50us
CW Min:	31
CW Max:	1023
Long Retry Limit:	7
Adhoc Routing:	AODV
Simulation time:	20.0 sec

B. *Simulation Results*

In our simulations, the values of the circularity were varied from 1 to 300. For each circularity value pair, we plot the corresponding throughput obtained. This has actually been implemented as a two step method. In the first instance, both the RTS and CTS circularities are varied simultaneously. From the graph obtained here, we observe the circularity value for which one gets maximum throughput. Now keeping this value of RTS circularity fixed, the CTS circularity is varied to obtain the optimum throughput. This RTS and CTS circularity value pair denotes the optimum values for which the maximum throughput is attained. DIFS extension circularity values use the same circularity value as that of the last RTS circularity. It should also be kept in mind that we use AODV as the prominent network layer routing protocol for most of our simulations.

The Figure 4 denotes throughput versus circularity for 8 nodes. EMAC denotes the Extended MAC version incorporating circularity (Green and Red) whereas MAC 802.11 is the standard IEEE 802.11 MAC protocol. The EMAC-1 and EMAC-2 denote the two instances where in the first one we vary both circularity values at the same time whereas in the next the RTS circularity is kept constant (at the point of highest throughput) and the CTS value varied.

The graphs obtained have many significant points. The straight line in Figure 4, for the default mechanism denotes a constant throughput obtained as expected, but we can see that the net throughput oscillates for differing circularity values. At points it is higher and also lower at other points than the default mechanism. However, what is of significance are the higher points. We are looking for the perfect circularity value pair for which the net throughput obtained is the highest and observe its increase over the default mechanism. Points where the value is lower means that due to RTS dropping or CTS delaying or DIFS extension, the number of successful packet transmissions are lesser. But these circularity values can simply be ignored and the points where throughput is maximum be considered.

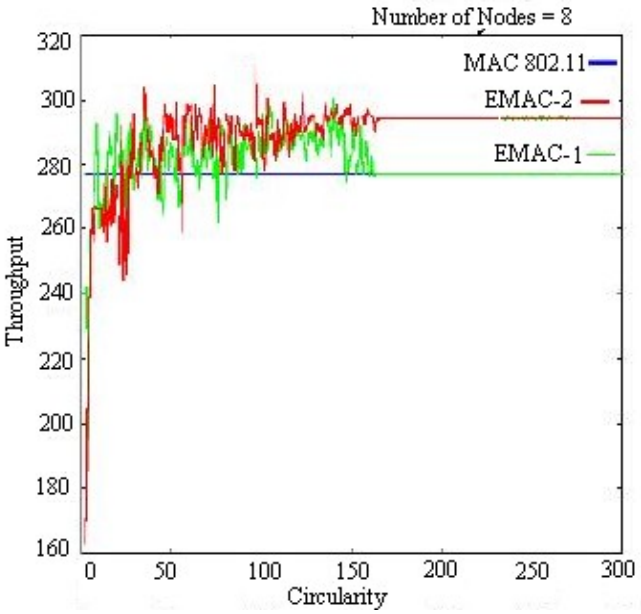


Figure 4: Throughput v/s circularity for 8 nodes

Finally, another important performance metric we have considered in our simulations is the total number of packet collisions. These packet collisions are the determining factor in the calculation of the successful packet transmissions and play a vital role in such a situation. From the Figure 6, we can see that in the Extended mechanism, the total number of packet collisions are significantly lesser than the standard for greater network complexity and link traffic.

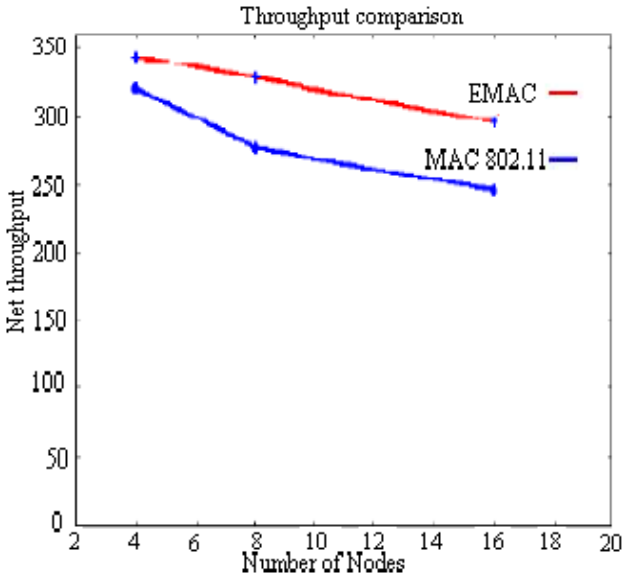


Figure 5: Throughput with respect to number of nodes.

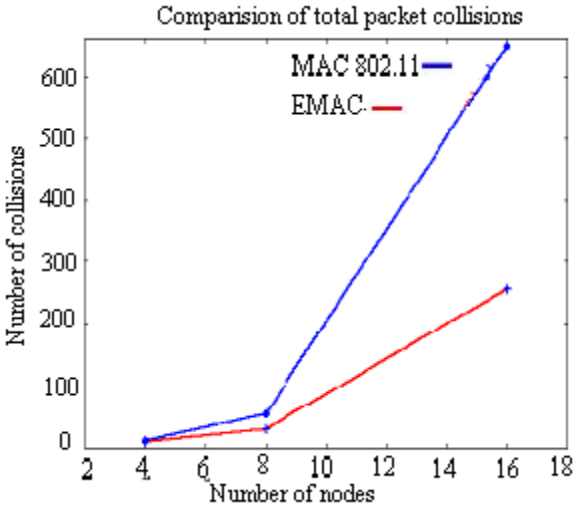


Figure 6: Packet collision with number of nodes

V. CONCLUSION

In this Paper,we have successfully designed and evaluated enhancements to the IEEE 802.11 mechanism for Mobile Ad hoc Networks for Pervasive Computing Envirnoment which uses a new and simple concept of circularity to effectively counter the pitfalls of the standard protocol. The concept of Circularity has been embedded in the existing RTS-CTS Handshake for MAC 802.11.On a positive note, we can confirm that the RTS-CTS Handshake with Circularity does give better performance over MAC 802.11 for the high MAC contention scenarios. We conclude this Paper with the hope that this spurs more research into the area and itself stands out as an important contribution.

ACKNOWLEDGMENT

The authors would like to thank Mr. Om Prakash Deshwal for preparing manuscript and Mr. Sameer Bhatia for his support in completing this paper. We would also like to thank the colleagues and entire HSC family.

VI. REFFERENCES

[1] IEEE Std 802.11b-1999. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Standard 802.11 (1999)

[2] Kleinrock, L., Tobagi, F.: Packet switching in radio channels: Part I - carrier sense multiple access modes and their throughput-delay characteristics. IEEE Transactions on Communications COM-23 (1975) 1400-1416

[3] Bharghavan, V., Demers, A., Shenker, S., Zhang, L.: MACAW: A media access protocol for wireless LAN's. In: Proceedings of ACM SIGCOMM '94. (1994) 221-225

[4] Ray, S., Carruthers, J.B., Starobinski, D.: Evaluation of the masked node problem in ad-hoc wireless lans. IEEE Transactions on Mobile Computing 4 (2005) 430-442

[5] Ju, H., Rubin, I., Kuan, Y.: An adaptive RTS/CTS control mechanism for IEEE 802.11 MAC protocol. In: Proceedings of IEEE Vehicular Technology Conference. Volume 2. (2003) 1469 - 1473

[6] Xu, K., Gerla, M., Bae, S.: Effectiveness of RTS/CTS handshake in IEEE 802.11 based ad hoc networks. Ad Hoc Networks 1 (2003) 107-123

[7] Crow, B.P., Widjaja, I., Kim, J.G., Sakai, P.T.: IEEE 802.11 Wireless local area networks. IEEE Communications Magazine 35 (1997) 116-126

[8] Bianchi, G.: Performance analysis of the IEEE 802.11 Distributed Coordination Function. IEEE Journal on Selected Areas in Communications 18 (2000) 535-547

[9] Deng, J., Liang, B., Varshney, P.: Tuning the carrier sensing range of IEEE 802.11 MAC. In: Proceedings of IEEE Global Telecommunications Conference. Volume 5. (2004) 2987-2991

[10] Haas, Z.: On the performance of a medium access control scheme for the reconfigurable wireless networks. In: Proceedings of the IEEE MILCOM. Volume 3. (1997) 1558-1564

[11] Yang, X., Vaidya, N., Ravichandran, P.: Split-channel pipelined packet scheduling for wireless networks. IEEE Transactions on Mobile Computing 5 (2006) 240-257

[12] Kuang, T., Williamson, C.: A bidirectional multi-channel MAC protocol for improving TCP performance on multihop wireless ad hoc networks. In: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems. (2004) 301-310

[13] VINT: The UCB/LBNL/VINT network simulator-ns (version 2). (URL http://www.isi.edu/nsnam/ns)

[14] Broch, J., Maltz, D.A., Johnson, D.B., Hu, Y., Jetcheva, J.: A performance comparison of multi-hop wireless ad hoc network routing protocols. In: Proceedings of Mobile Computing and Networking. (1998) 85-97

Energy-Balanced Cooperative Routing Approach for Radio Standard Spanning Mobile Ad Hoc Networks

Matthias Vodel and Mirko Caspar and Wolfram Hardt
Chair of Computer Engineering, Dept. of Computer Science
University of Chemnitz, Chemnitz, Germany
{ vodel | mica | hardt }@cs.tu-chemnitz.de

***Abstract*— In this paper, a new Energy-Balanced, Cooperative Routing approach (EBCR) for radio standard spanning mobile Ad Hoc networks (MANETs) will be introduced and evaluated. Past research approaches are limited to the usage in homogeneous topologies on basis of a unique radio standard. The proposed reactive EBCR provides an efficient routing of data in heterogeneous multi-standard network topologies. In addition to an improved reachability, a primary objective is to balance the overall power consumption in each operating node to prolong the lifetime of the whole topology. Instead of using static cost factors for the route path calculation, EBCR integrates a dynamic cost vector for the handling and processing of a data packet. This includes parameters like the required field strength for the data transmission and the node's current energy level to calculate the cooperative, optimal route path. Thereby, EBCR operates only on the basis of local network information. For evaluating the conceptual advantages of this approach, multiple scenarios with static and high-dynamic network topologies have been analysed in a dedicated simulation environment. The simulation results verify the significant improvements in the topology lifetime and the reachability up to 15%.**

I. Introduction

An optimal design of a mobile Ad Hoc network can be characterised by a good distribution of net load over all nodes, a good scalability and the robustness against disturbances and partial losses. Due to different application areas of wireless communication networks in industrial and private environment a multiplicity of radio standards has been developed. Each standard has application optimised characteristics in regard to transmitting range, data transfer rate, power consumption and the used frequency band. Almost every radio standard uses link-based transmission techniques to send and receive data. Accordingly, capable methods for the routing and forwarding of data packets between different network nodes are necessary. The current approaches for connectivity and communication in wireless mobile Ad Hoc networks are, however, only occupied with one radio standard each. The radio standard spanning interaction is not possible.

Looking forward to the next generation of wireless mobile technologies, one essential ability will be the integration of different wireless communication standards. *Figure I* illustrates related wireless communication techniques, represented by IEEE 802.x protocol standards, for different application areas and transmitting ranges [1]. Optimised 4G wireless mobile technologies must provide a interoperability between these application areas to create a multi-standard heterogeneous network topology.

One central point of such a concept is an efficient, radio standard spanning routing algorithm, which allows a reliable, fast and adaptive data packet transport in the available network topology. This paper presents an energy-balanced, cooperative routing approach (EBCR) to offer an abstract solution. After this introduction, section *II* will give an overview on past publications in the research fields of routing in mobile Ad Hoc networks, wireless multi-standard communication and interface synthesis. The following section *III* offers detailed information about the EBCR algorithm. Some key facts about the simulation environment are summarised in section *IV* and, later, relevant definitions for the cost function and scenario-specific weighting schemes are presented. Accordingly, the received simulation results are discussed in section *VI*. Finally, the paper concludes with a summary and an outlook for future work in this research area.

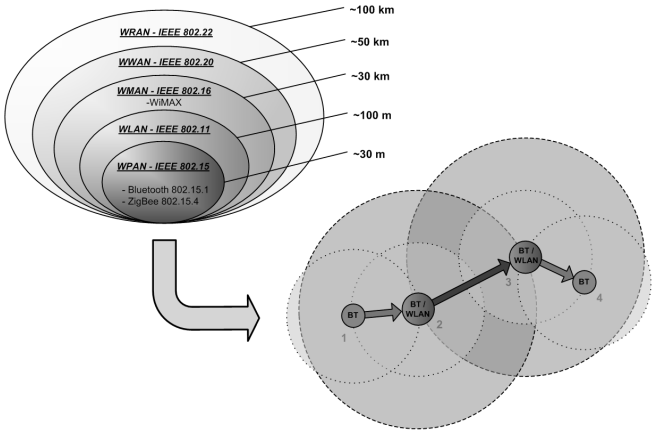


Fig. 1. Illustration of the serveral wireless communication standards IEEE 802.x with optimised characteristics for different transmitting ranges. Next generation wireless networks (bottom right) integrate different radio standards into one heterogeneous topology

II. Related Work

Based on a dynamically optimised topology, data between arbitrary nodes can be transmitted in the network. For the efficient package-oriented data communication, routing algorithms are used, which lead the packets over a preferably optimised path to the destination node [2]. For the usage in mobile Ad Hoc networks, proactive routing algorithms have

been developed, which try to update a local routing table also in a dynamic network topology. Thus, valid route paths are already found before they are needed for communication. At the time of a communication request, each node has a current route, without causing latencies. A representative of proactive routing algorithms is the Optimised Left State Routing (OLSR [3]) or the Destination-Sequenced Distance-Vector Routing (DSDV [2]). Since the administration complexity of such routing methods strongly rises with increased dynamics, reactive routing protocols have been conceptualised, which search valid route paths not until there is a connecting request (Ad hoc On demand Distance Vector AODV [2], Dynamic Source Routing DSR [2]). Also hybrid algorithms, like the Zone Routing Protocol (ZRP [4]), have been developed.

Xia et. al. propose in [5] an interesting approach to increase the lifetime of a wireless sensor network by the integration of a node's actual energy level into the routing algorithm. So this gradient-based routing protocol prefers route paths with a high energy level of the several nodes. This enables an self-adjusting balance of the energy consumption in the whole topology. The approach has its focus on the application field of wireless sensor networks with a uniform radio standard. The typical network traffic scenario is a balanced, periodical transmission of sensor data.

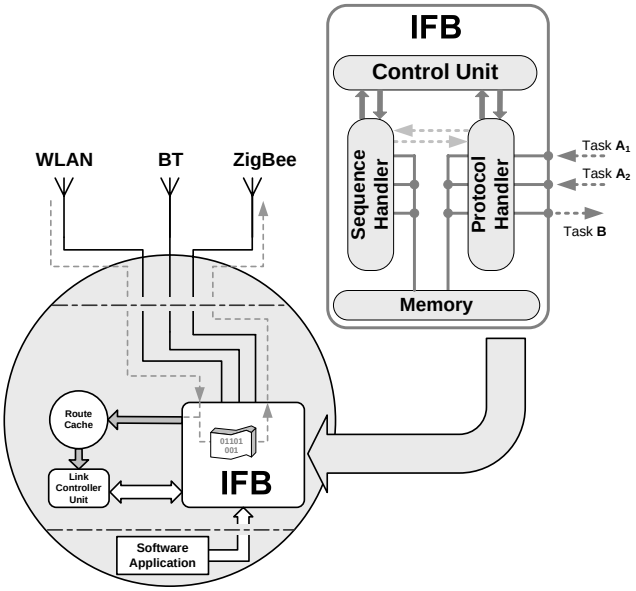


Fig. 2. The Interface Block (IFB) Macrostructure (top right corner) with Control Unit, Protocol Handler and Sequence Handler. Two tasks A (with the subtasks A_1 and A_2) and B with incompatible protocols communicate with each other using the IFB. In the bottom left corner the Basic layout of a radio standard independent node. Exemplary with three antennas for the radio standards Wireless LAN, Bluetooth and ZigBee. A central IFB connects the radio modules. Based on cached routing information, the link controller unit (LCU) manages the data flow in the IFB. Software applications use only one well-defined interface to the hardware block.

To achieve a radio standard spanning communication in MANETs, one central challenge is the interaction between the communication standards with different protocols. A promising solution for a this problem is the hardware-near coupling of individual standardised radio modules. For this purpose a special hardware block, which provides a conversion of incompatible protocols, is necessary [6, 7]. Such an interface block (IFB [8]) analyses incoming packets and extracts the user data. Subsequently, these data are adapted on desired protocols and passed on accordingly. Due to the IFB macro structure (*Figure 2*) a modular expandability is ensured.

Based on the IFB approach, a concept for the radio standard spanning communication in MANETs was proposed in [9]. The individual nodes (*Figure 2*) of a network topology can use advantages of different radio standards to get a higher degree of connectivity. They do not need a high arithmetic performance like it is necessary in research approaches of software defined radio [10, 11]. An energy-efficient and multi-functional applicable possibility of wireless communication is created. The presented approach is divided into three primary objectives, which have to be solved: topology construction, protocol conversion and routing. In the following sections of this paper an efficient routing algorithm will be introduced and analysed.

III. Cost Vector-Based Cooperative Routing

In order to provide a routing algorithm, which is able to make decisions about the choice of the used radio standards and the optimal route path on the basis of functional connection requirements like required bandwidth or available energy resources, the EBCR was developed. The primary objective is the integration of multiple cost indicators in a dynamic cost vector to prolong the average node lifetime in the network topology. Parameters about the actual node status permit a cooperative choice of the route paths.

Basic Information

For the primary usage in applications of mobile miniature devices, which have strongly limited energy resources and usually small arithmetic performance, a reactive routing algorithm is preferred. EBCR is able to minimise the resources for storage and administration of routing information in the devices. Furthermore, EBCR does not need any global network information like the actual node position or movement parameters. Every node in the network sends a periodic heartbeat signal to allow the administration of lists with the direct attainable neighbourhood. On the basis of these local network information and some basic status parameters, EBCR creates a dynamic cost vector for the route path calculation.

In consequence of the primary aim of EBCR, the generated route paths are not hop-optimal in many cases. The approach prefers energy-efficient route paths, which prolong the reachability and the lifetime of the entire network topology by balancing the energy level of each node regardless of the available radio standards. Due to the combination of cooperative characteristics of wireless sensor network applications and related routing approaches for MANETs, an optimal usage of the available resources can be ensured.

Figure 3 illustrates a simple routing scenario from the source node A to the sink Z. There are three possible route

paths, which generate different costs. By the fact that node B has a very low battery charge, EBCR will avoid this choice. For the decision about the route path via C or D, EBCR calculates the required transmission costs based on the radio standard, the required transmitting power, a possible swap of the radio standard (for example in node C during the path $A \rightarrow C \rightarrow Z$) and additional functional connection requirements.

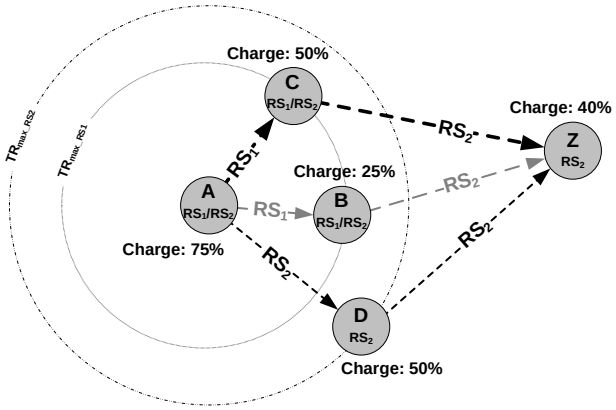


Fig. 3. A simple routing scenario with three possible routes from the source node A to the sink Z. Node A, B and C integrates two different radio modules RS_1 and RS_2 with specific maximum transmitting ranges $TR_{max\_RS_1}$ and $TR_{max\_RS_2}$. Node D and Z have only one available radio standard.

To balance the disadvantages of reactive routing methods like broadcast storms and in consequence radio interferences on the physical layer, EBCR provides the possibility of a counter-based retransmitting scheme [12] and a randomised retransmitting delay T_{b_delay}.

Broadcast/Forward process

With every route inquiry, the source node broadcasts the request including a unique time stamp. Each broadcast packet has a unique time stamp for a identification of equal broadcast messages. The several nodes caches these broadcast IDs and allows only a limited number of rebroadcasts over the selected radio modules. These simple techniques reduces the packet transmission error rate by radio interferences significantly [12]. Each node, which receives the request, updates the cost vector with the own parameter. In a defined waiting period T_{r_req} the node receives alternative sub-paths and compares the cost vectors. If the new cost vector is better than the stored one, the node updates its own route parameters. Bad sub-paths will be rejected. After T_{r_req} is expired and the maximum number of equal incoming broadcast request is not reached (counter-based retransmitting scheme [12]), the node rebroadcast the stored cost vector with the best parameters. If a broadcast request will be received by the sink node, the sink sends a corresponding response packet along the received route path. For the following bidirectional communication between the source and the sink, each node knows the next hop of the route path to forward the data packets. If the sink is not reachable, each node has a predefined timeout for broadcast requests. To reduce the number of route requests in the topology, valid route paths are temporary stored in a route cache with a maximum lifetime before they will be deleted. EBCR allows the additional caching of alternative route paths over different radio modules to provide backup functionality during a data transmission.

IV. Simulation Environment

Due to the designed cost vector model and the IFB-based integration of multiple radio modules, an implementation with related network simulators like NS2 is too complex. To clarify the necessity of such a heterogeneous, interoperable communication concept, a modular, platform-independent simulation environment was designed and implemented for the proof of concept. Every network node is emulated by a dedicated thread with own defined parameters and properties. Thus, the simulator is scalable to different node densities and topology scenarios. An event-driven graphical user interface allows the visualisation of topology modification in realtime. To analyse the dynamic of mobile Ad Hoc networks, a movement model (*Random Waypoint Mobility model - RWP* [16]) was implemented. A global topology control layer provides a statistical interpretation of the entire network topology. An essential difference to other approaches is the possibility to configure network nodes with more than one interacting radio modules, which are controlled by the Link Controller Unit (LCU) and the IFB (*Figure 2*).

To evaluate the conceptual advantages of a radio standard spanning communication in MANETs, a complexity model on basis of cost vectors was developed. Thereby, each transmission of data generates costs that are dependent on the used radio standard and the transmission range. During the route path of a data packet to its destination node, a change of the communication standard also generates costs for the data conversation by the node's internal interface block. The several parameters of the cost vector are predefined uniformly and are adaptable in a topology. Based on these cost vectors and the usage of TCP/IP, the modified EBCR algorithm finds cost-optimal route paths in the network. The simulator does not have a physical calculation model for interferences in the used frequency bands. To avoid such disturbances and problems like broadcast storms, the EBCR algorithm provides a feature for a randomised packet forwarding delay as described in section *III*.

Each node has predefined energy resources, which provide a limited lifetime. Furthermore, the simulation environment provides the possibility to generate a periodic, basic power consumption for nodes in the idle mode. For the proof of the EBCR approach, this time-based feature is disabled. If the available resources are expired, the node switches its current operational mode to offline and is no longer available in the topology.

To evaluate the advantages of EBCR, several test scenarios were analysed. The following simulation results include topology scenarios with uniform and random node distribution (*Figure 4*) and two node densities of 30 and 60 nodes. For all

scenarios, the maximum number of interacting radio modules has been restricted to the amount of two. It must be pointed out, that the theoretical maximum number of integrated radio standard is not limited upwards with the proposed concept [9]. Each node has an initial battery charging state of 100% and one or two radio modules for the notional radio standards RS_1 and RS_2. $TR_{max\_RS_1}$ represents the default low-power radio module. The transmission range of the second radio standard $TR_{max\_RS_2}$ is predefined with $TR_{max\_RS_2} = 2 \cdot TR_{max\_RS_1}$. For the chosen test scenarios, 50% of the nodes has two available radio standards. The other 50% transmit with only one radio standard RS_1. In several test cycles the simulator generates randomised network traffic with a uniform or random number of data packets per message.

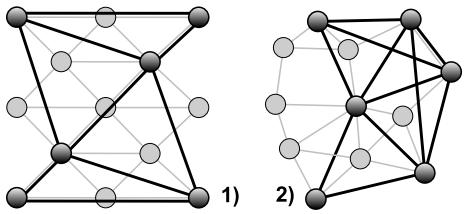


Fig. 4. Two exemplary network topologies with a uniform distribution (net 1) and a randomised distribution (net 2). Both topologies have about 50% of multi-standard nodes.

V. Simulation Scenarios

Cost vector definition

Parameter	RS_1	RS_2
Basic Costs per Hop C_{base}	20	20
Current Energy Level C_{e_level}	1-10	1-10
Transmitting Power Consumption C_{tpow}	0.8-8	5-50
Overall Data Transmission Time C_{ttime}	2-20	0.1-1
Data Packet Latency C_{lat}	20	10
Swap radio standard C_{swap}	10	10

Table I. Two defined radio standards with the separated costs for each factor.

For the proof of concept there are only two notional radio standards RS_1 *and* RS_2 with specific communication characteristics available. On the basis of these characteristics, *table I* describes the several parameters of the EBCR cost vector for the handling and processing of data. The three parameters C_{e_level}, C_{tpow} and C_{ttime} are defined in an interval. The individual values are calculated in each node as follows:

$$\begin{aligned} C_{e\_level} &= \frac{1}{P_{current}}, 0.1 \leq P_{current} \leq 1.0 \\ C_{tpow} &= 1 + (C_{tpow_{max_{RS_x}}} \cdot P^2_{transmit}) \\ & \quad 0.1 \leq P_{transmit} \leq 1.0 \\ C_{ttime} &= C_{ttime_{max_{RS_x}}} \cdot P_{transmit} \\ & \quad 0.1 \leq P_{transmit} \leq 1.0 \end{aligned}$$

$P_{current}$ ... node's current energy level
$P_{transmit}$ ... required transmitting power level

Accordingly, based on these cost factors, a cost function is defined as follows:

$$\begin{aligned} C_{overall} &= \sum_{i=1}^{\#_{hops}} (\sum_{j=1}^{n} (w_j \cdot C_j)) \\ &= \sum_{i=1}^{\#_{hops}} (w_{base} \cdot C_{base} + w_{e\_level} \cdot C_{e\_level} \\ & \quad + w_{tpow} \cdot C_{tpow} + w_{ttime} \cdot C_{ttime} \\ & \quad + w_{lat} \cdot C_{lat} + w_{swap} \cdot C_{swap}) \end{aligned}$$

n ... number of parameters
w_x ... weighting of parameter x

Cost vector weighting

By the usage of individual weighting factors, the cost function is adaptable to different fields of application. For the chosen simulation scenarios, the weighting W_{S1} is defined as follows:

W_{S1} ... power consumption optimised testing scenario

Weighting Parameter	W_{S1}
w_{base}	1.0
w_{e_level}	1.0
w_{tpow}	0.6
w_{ttime}	0.3
w_{lat}	0.1
w_{swap}	1.0

Table II. The defined weighting scenario is based on the defined cost function $C_{overall}$.

Energy adaptation model

Each process for the handling and transmission of data decreases a node's energy resources dependent on parameters like the used radio standard or the required electromagnetic field strength for reaching the selected node. The initial charge state of each node is assumed with 10000 logical units. The function for the adaptation of the energy level ΔP_{node} is defined as follows:

$$\begin{aligned} \Delta P_{node} &= f(M, P_{transmit}, RS_x) \\ \Delta P_{node} &= \#_{packets} \cdot P^2_{transmit} \cdot g(RS_x) \\ g &: (RS_x) \rightarrow N \\ g(RS_x) &= \begin{cases} 1, \text{Transmission } RS_1 \\ 6, \text{Transmission } RS_2 \end{cases} \end{aligned}$$

M ... Message size in packets
$P_{transmit}$ ... Required transmitting power level
RS_x ... Used radio standard

VI. Simulation Results

The following results are divided into two subsection for uniform and random node distribution (*Figure 4*). For each distribution there are two simulation scenarios.

Topology lifetime scenario:

The first scenario chooses randomised source and destination nodes and tries to find a valid and optimal route path. Through every valid route path a traffic generator transmits a message with a predefined number of data packets. The simulation ends if no more node is reachable and all available energy resources are exhausted. The simulator counts the number of successful transmitted data packets in the topology and, accordingly, the number of dead or isolated nodes.

Static end-to-end scenario:

The second scenario simulates a static communication stream between two predefined nodes. After each successful transmission of a message with n data packets, EBCR calculates a new optimal route path to balance differences of the energy level in topology. A primary objectives of this scenario is to minimise the number of dead nodes to preserve a fully connected topology in which every node keeps reachable. The number of transmitted data packets must be maximised. *Figure 5* illustrates these objectives.

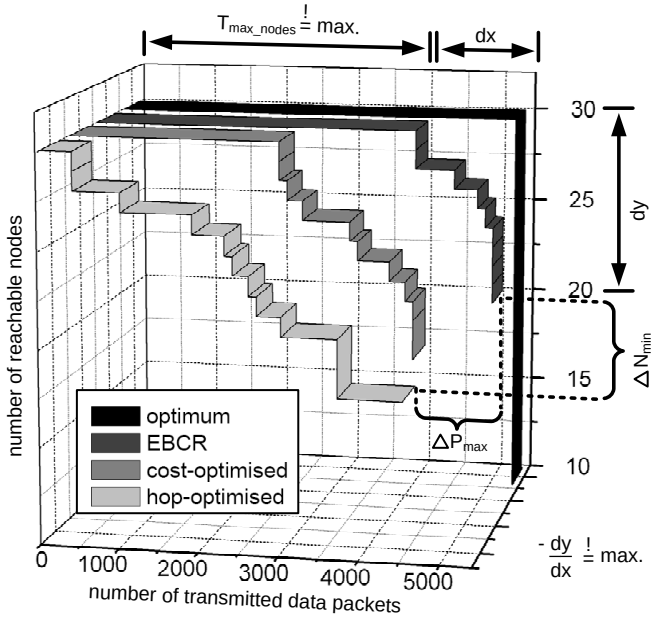


Fig. 5. The diagram shows the simulation results of three different routing approaches in a random distribution of 30 nodes. The theoretical optimum is illustrated as the black line. A primary goal is the maximisation of the time when all nodes in the topology are available (T_{max_nodes}). Simultaneously the negative slope of the curve must be maximised. An important measurement are the variations of the transmitted data packets (ΔP_{max})with several routing approaches. Differences in the minimal number of available network nodes (ΔN_{min}) are meaningful for all test cycles of the second scenario with a static source and sink.

As already mentioned in section IV, each process for the data processing and transmission decreases a node's energy level. In this simulations, two major processes influence this resource. Regardless from the result of a route request, each inquiry decreases the energy level of the reachable nodes as a consequence of the broadcasting process. If EBCR has found a valid route path, the generated traffic must be handled and forwarded in each node of the route path. Every data packet decreases the available energy resources dependent on the selected radio standard and the required transmitting power.

To verify the conceptual advantages of this approach, the lifetime of a valid route in the route cache is defined very short. With this precondition, the nodes in the topology are forced to use the routing algorithm for each transmission inquiry. EBCR is compared with two related routing approaches. The first one is a simple hop-optimised routing, which chooses route paths with minimum number of hops between source and sink. The second routing algorithm calculates an optimal route path with a non-cooperative, static cost function, which does not consider a balancing of the available energy resources or a node's current energy level. The several cost parameters are equal to the used parameters of section *III*.

Results - Uniformly Distributed Topology

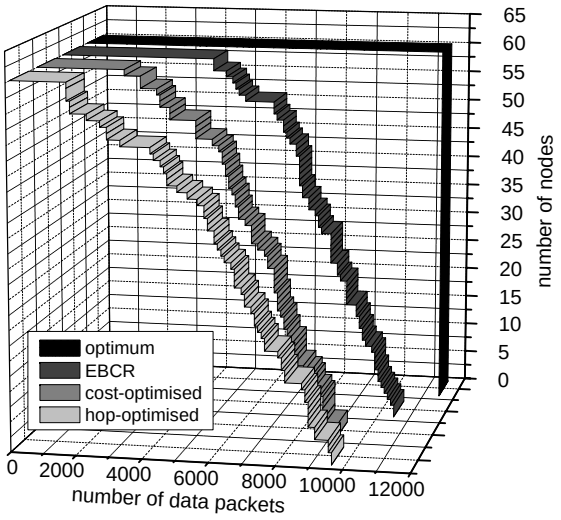


Fig. 6. Topology liftime scenario: uniform node distribution, 60 nodes, 50% of multistandard-nodes, message size: 100 packets

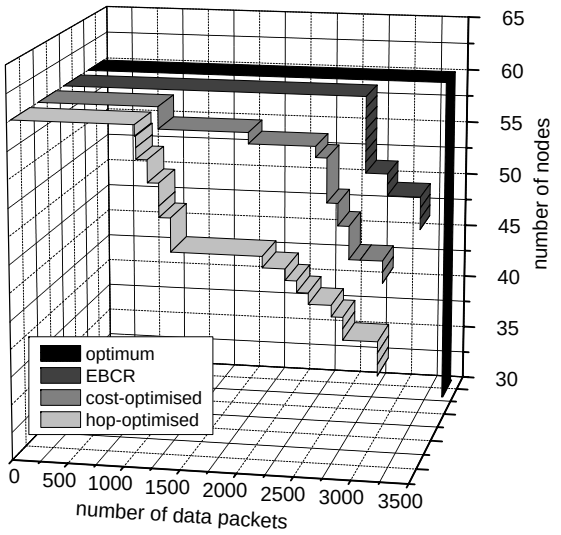


Fig. 7. Static end-to-end scenario: uniform node distribution, 60 nodes, 50% of multistandard-nodes, message size: 100 packets

Figure 6 illustrates the significant extension of the successful transmitted messages till the first nodes dropped out. Furthermore, with EBCR, the overall transmitted network traffic was increased of about 1000 data packets. An even higher degree of improvements shows the simulation results in the static end-to-end scenario. The EBCR curve in *Figure 7* is very close to the theoretical optimum. In addition, the number of dead nodes with depleted energy resources was reduced from 21 (hop-optimised routing) to 12, which is equivalent to an improvement of 15%.

Results - Randomly Distributed Topology

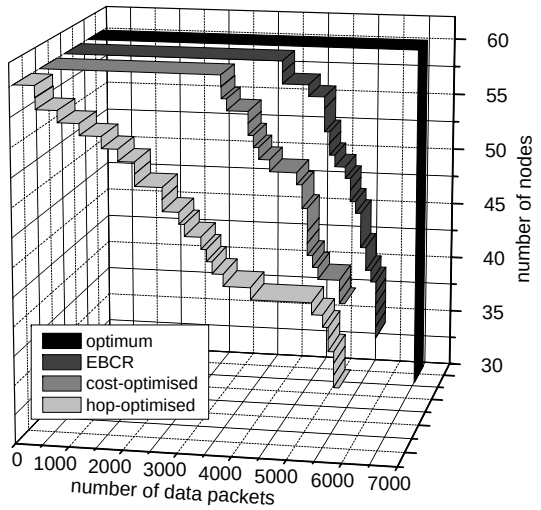


Fig. 8. Static end-to-end scenario: random node distribution, 60 nodes, 50% of multistandard-nodes, message size: 100 packets

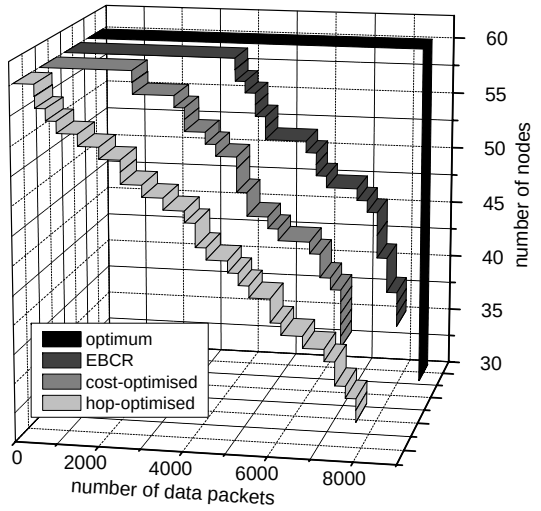


Fig. 9. Static end-to-end scenario: random node distribution, 60 nodes, 50% of multistandard-nodes, message size: random from 50 to 500 packets

Figure 8 and 9 represent the dependency on the simulation results from the defined message size. The simple hop-optimised routing approach shows a very high number of nodes with depleted energy resources in the second test cycle with a randomised message size. Independent on the defined packet size, EBCR provides the best results in the number of transmitted data packets. Furthermore, the simulations verify the primary objective of EBCR to maximise the reachability of each node in the topology.

VII. Conclusion

The presented simulation results clarify the importance and the significant advantages of this cooperative routing approach. The choice of the used routing algorithm is essential for a sufficiently stable, adaptive and scalable network topology [13, 14]. EBCR realises a radio standard spanning routing. This offers completely new possibilities for an efficient communication in high-dynamic MANETs. With an outlook to the next generation of wireless mobile technologies, EBCR provides a basic concept to integrate a multiplicity of radio standard into a single heterogeneous network topology. This enables various new application scenarios with wireless communication techniques.

References

[1] T. Cooklev. *Wireless Communication Standards - A Study of IEEE 802.11., 802.15., and 802.16.*. ISBN 0-7381-4066-X, IEEE Press, New York, USA, 2004.

[2] E.M. Royer, C.-K. Toh. *A review of current routing protocols for ad hoc mobile wirelessnetworks*. Personal Communications, IEEE, Volume 6, Issue 2, pages 46-55, April 1999.

[3] T. Clausen, P. Jacquet, A., Laouiti, P. Muhlethaler, A. Qayyum, L. Viennot. *Optimized Link State Routing Protocol*. IEEE INMIC, Pakistan, 2001.

[4] Zygmunt J. Haas, Marc R. Pearlman. *The Zone Routing Protocol (ZRP) for Ad Hoc Networks*. INTERNET-DRAFT, IETF MANET Working Group, November 1997.

[5] L. Xia, X. Chen, X. Guan. *A New Gradient-Based Routing Protocol in Wireless Sensor Networks*. International Conference on Embedded Software and Systems (ICESS 2004), LNCS 3605, pages 318-325, Hangzhou, China, 2005.

[6] W. Hardt, Stefan Ihmor. *Schnittstellensynthese - volume 2 of Wissenschaftliche Schriftenreihe: Eingebettete, selbstorganisierende Systeme*. ISBN: 3-398863-63-3, TUDpress, Dresden, Germany, July 2006.

[7] W. Hardt, T. Lehmann, M. Visarius. *Towards a Design Methodology Capturing Interface Synthesis*. In Monjau, Dieter, editor, 4. GI/ITG/GMM Workshop: Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen, volume 1, page 93-97, ISBN: 3-00-007439-2, Meißen, Germany, February 2001.

[8] S. Ihmor. *Modeling and Automated Synthesis of Reconfugurable Interfaces*. Dissertation, Universität Paderborn, Heinz Nixdorf Institut, Entwurf Paralleler Systeme, HNI-Verlagsschriftenreihe, Paderborn, Band 205, January 2006.

[9] M. Vodel, M. Caspar, W. Hardt. *Performance Analysis of Radio Standard Spanning Communication in Mobile Ad Hoc Networks*. Proceedings of the 7th IEEE International Symposium on Communications and Information Technologies (ISCIT) (accepted) , Sydney, Australia, October 2007.

[10] J. Mitola II, Z. Zvonar. *Software Radio Technologies*. New York: IEEE Press, 2001.

[11] S. Bhattacharya. *SDR Based End-to-End Communication*. DataSoft Corporation, Technical Committee, January 2005.

[12] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, J.-P. Sheu. *The Broadcast Storm Problem in a Mobile Ad Hoc Network*. Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, pages 151-162, ISBN: 1-58113-142-9, Seattle, United States, 1999.

[13] M. Gerharz, C. de Waal, P. Martini, P. James. *Strategies for Finding Stable Paths in Mobile Wireless Ad Hoc Networks*. Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks (LCN'03), Bonn, Germany, October 2003.

[14] F. Ye, A. Chen, S.W. Lu, L. Zhang. *A Scalable Solution to Minimum Cost Forwarding in Large Sensor Networks*. Proceedings of 10th International Conference on Computer Communications and Networks, pages 304-309, 2001.

Ludos Top: A Web-Based 3D Educational Game Multi-User Online

Everton Souza, Marlene Freitas, Luciano Ferreira, Alexandre Cardoso and Edgard Lamounier.
Departamento de Engenharia Elétrica, Laboratório de Computação Gráfica – Universidade Federal de Uberlândia (UFU) Caixa Postal 38.400 – 34.3239-4148 – Uberlândia – MG – Brasil

eevesou@ieee.org, {fsluciano,marlene_roque}@hotmail.com, { alexandre, lamounier}@ufu.br

Abstract

This paper presents Ludos Top - an educational 3D game that use virtual reality techniques, which can support multi-student with a new design model of networking on the web.
The project has actively involved end-users to focus on increase interactivity through the use of versatile system architecture.
We present a quick prototyping of a multi-user virtual world through the employment of Ajax, X3D and Web Services provides an efficient, flexible and robust means for distributed application. Results show improved network capabilities, in terms of interactive, ease of use, enjoyability, playability and usability.

Keywords: Ajax3D, Distributed Virtual Worlds, e-Learning, Online Games, Rich Internet Application, Virtual Reality, X3D, Web Games.

1. Introduction

The word games refer to activities of which nature or finality is recreative, diversion and entertainment. Theses activities exists officially since 776 B.C., and they began in Olympia, old Greece, with the Olympic Games. The games perpetuates till nowadays, nevertheless, the way which they are realized and its propose changed.

Research work argues that computer games are an engaging medium for learning since they can stimulate cognitive processes as reading explicit and implicit information, deductive and inductive reasoning, problem solving, and making inferences from information displayed across a number of screens [Pillay, 1999]. According to constructivist approach [Von Glaserfeld, 1990], learning depends on the active engagement of the subject that learns and on his ability to construct knowledge and understanding on the basis of interaction with the environment.

The Web environment has matured to support real-time delivery of web-based 3D content to increase interaction and integration with others systems. The Virtual Reality demonstrates the same evolution through of new standards like: X3D, the successor of VMRL standard for Web-based 3D graphics.

The X3D is an open standard file format and API for representing and communicating with Scene Authoring Interface (SAI). A major goal of X3D is to support 3D web applications, in addition to 3D on networking with portability.

This show news paradigms e.g. integration of X3D with Ajax3D programming model which is the W3C Document Object Model (DOM). The Ajax3D consists of a web-page embedded JavaScript program which allows an architecture asynchronously given flexibility for manipulation.

Through technologies Ajax3D and Web Services on architecture application-to-application communication via network or the web does solve many of the problems, it also creates many new problems and open new possibilities. The focus is support the movement towards an efficient environment that enables the development of multi-user applications, providing functionality with the highest level of compatibility.

This work motivated the development a game with pedagogical objectives. It is called by Blaise Müller of Quatro and particularly useful to test observation and thought abilities.

This paper presents works related. Section Three give a view about background. Section Four provides an overview of our proposed architecture. Section Five provides the case study Ludos Top. Section Six shows system implementation of this technology in a simulation. Section seven provides a conclusion.

2. Related Works

The increasing number of broadband users, and demand for service quality and diversity, especially in the entertainment area, drive to development of the new games like:

Strike Fighter was developed by Larry Rosenthal of Cube Productions. It has been running a science fiction virtual world/online community called StarbaseC3 for nearly that long. In 2001 were developed the first version of a 3D web game called “Strikefighter,” in VRML programming. Last year Strkefighter was updated for X3D and showed new model based on architecture with Ajax3D, the game connected with web server (running PHP and MySQL) that implements a scoreboard. When the game is over, it checks the current score against a database of high scores residing on the server. Strike fighter is similar to the Flash-based mini-games that have proliferated on the web over the last several years. Until Ajax3D, was not able to deploy a 3D game like this based on a

royalty-free, open platform that fully integrates with a web server and runs in a page.

Figure 1: Snapshot from Strike Fighter game in Ajax3D.

The Strike Fighter involved by: JavaScript, XMLHttpRequest, the DOM and SAI running in a web browser-independent fashion that will work with IE, Firebox and others browsers.

Road Rider is an interactive first person 3D game, where the player controls a virtual character whose task consists in reaching the site of a rock concert. During her/his trip, the user walks around a city (that is a 3D reconstruction of a portion of the Genoa city center - figure 3). The game plot consists of a number of "missions". Missions involve finding a car, getting money to buy a ticket for the concert, driving the car to visit friends who live in different cities to another, and finally reaching the destination site. Every mission features an increasing level of difficulty. In order to enhance the player's engagement, the game plot is dynamic. The missions are not be predefined and do not follow a fixed sequence. Conditions and events change in every mission. These situations and conditions are similar to those of state of the art commercial videogames. This makes this activity be perceived pleasantly by the user as a game.

However, there are two main aspects that differentiate Road Rider: the road settings and the score mechanisms. All the important game situations are tied to road safety (road-signs, vehicles, roads, cross-roads, pedestrians, etc.). And this is true also for the score rules. Score is a fundamental element of the game, since it provides the main motivation for a user to improve her/his performance. So, the criteria according to which points are assigned are very important because they define what layers' operations, actions and behaviors are positive and what are not relevant or even negative. In Road Rider, the system penalizes hazardous behaviors and rewards safe road-behaviors.

Figure 2: Snapshot from Road Rider.

The game has been realized with the Torque Game Engine, a high quality/price ratio free source game engine [Garage Games, 2007]. This tool provides a valid support to realize:

• A realistic 3D visualization, which is quite complex to achieve in a networked environment;

• An effective narrative, which is important in order to engage the user in interesting and compelling situations;

• A realistic simulation, which is important to increase the likelihood that the user may transfer in the real-life techniques and skills learned through simulation.

The last feature is a user can play the game through mouse and keyboard, but also through low cost steering-wheel and pedals mock-ups, which are available on the market for game consoles.

The NICE project is an effort to build Narrative-based, Immersive, Constructionist/Collaborative Environments for children. Developed at the Interactive Computing Environments Laboratory (ICE) and the Electronic Visualization Laboratory (EVL) of the University of Illinois at Chicago, NICE aims to create a virtual learning environment that is based on current educational theories of constructionism, narrative, and collaboration, while fostering creativity within a motivating and engaging context.

NICE is an outgrowth of two previously designed systems, CALVIN and the Graphical Storywriter. CALVIN (Collaborative Architectural Layout via Immersive Navigation) is a networked collaborative environment for designing architectural spaces [Leigh and Johnson, 1996]. The Graphical Storywriter [Steiner and Moher, 1994] is a shared workspace, where young children can develop and create structurally complete stories. Extracting and building on elements from these previous works, we have created a prototype learning environment for young children which presents simplified ecological models of various ecosystems within a fantasy setting.

Figure 3: Snapshot from Narrative, Immersive, Constructionist/Collaborative Environments.

Equally important to the construction of one's knowledge is the experience gained by participation in group activity. Collaboration is emphasized in our framework through the combination of collaborative learning across both virtual, as well as physical communities. Collaboration of virtual communities refers to communication and shared experience between children who are geographically separated.

The network component of NICE allows multiple networked participants at different locations to interact with each other and share the same virtual space. The representation of each tracked child in the virtual space is established through the use of an avatar with a separate head, body, and hand. As each person's head and hand are tracked, this allows the environment to transmit gestures between the participants such as the nodding of the user's head, or the waving of the user's hand to the other participants. Visually, the wand is mapped to the arm of the child's avatar. As the child waves her hand in the real world, her avatar waves its hand in the virtual world. As these avatars have sufficiently detailed representations, the children can communicate notions of relative position to one another with phrases such as "it is behind you" or "turn to your left." The use of audio (with wireless microphones) between the various sites enhances the communication process.

The number of participants is limited only by bandwidth and latency of the network. Multiple distributed NICE applications running on separate VR systems are connected via the central LIFE server to guarantee consistency across all the separate environments. The communications library uses multicasting to broadcast positional and orientation information about each child's avatar, and uses TCP/IP to broadcast state information between the participants and the behavior system [M. ROUSSOS, A. E. JOHNSON, J. LEIGH, C. R. BARNES, C. A. VASILAKIS, AND T. G. MOHER, 1997].

3. Background

The 3D web environment has begun to demonstrate increased utility which include higher use of participation, scalability and compatibility.

3.1 Ajax3D

The technology Ajax has architecture neutral approach due new standard of web-based development. One of the most popular trends in web applications is enriching client functionality through the means of Asynchronous JavaScript and XML (Ajax) [CRANE, DAVID, PASCARAELLO, ERIC, 2005]. AJAX is a combination of existing technologies rather than a completely new innovation. The primary component is the XMLHTTPRequest (XHR) object that provides for the ability to perform asynchronous communication with a server via Hypertext Transfer Protocol (HTTP). Used in coordination with either Dynamic Hypertext Markup Language (DHTML) or use of HTML FRAMES, the XHR request object allows a high level of performance thus avoiding the call-and response mechanism of earlier web applications.

Replacing the integration of DHTML functionality with the X3D event model, applications have applied the use of X3D markup technologies in the context of an AJAX-based application [PARISI, TONY, 2006]. This approach has identified a trend of scripting the 3D scene hierarchy in the same fashion as the Document Object Model (DOM) and handling the events through JavaScript. By relying on AJAX-based network calls, a user interacts with backend tiers via a 3D scene in asynchronous fashion previously provided by employment of HTML 'DIV' tags. This technique has supported a much higher level of interactivity and efficiency in web-based 3D [A. OSTROWSKI, DAVID, 2007].

3.2 Multi-User Worlds

Recently, the term Post-Nintendo kids have been coined, referring to the fact that today most children have been exposed to 3D computer games. With ease they navigate through virtual environments interpreting audio-visual hints as they go. Just, as WIMP (windows, menus, pull-downs) metaphor rules the design of graphical user interface (GUI) of today's software and replaced the command line control, 3D environment and 3D navigation with new GUI metaphors for operation systems and desktop applications.

Multi-user worlds are a new way for people to collaborate. In Multi-user worlds they can form new communities which have common interests and are not restricted by the locality, mobility or status of the person in the real world.

Clarker-Willson (S. Clarker-Willson., 1998.) Suggest that virtual environment designers should apply principles that made even early games so appealing: third person view, discovery and exploration, player control, maps, landmarks, closed environments (limited space), complexity management (reduced number of active objects) and constant positive feedback.

A platform of increasing importance for multi-user worlds is game console. Since 1995 Sony has sold over 20 million Play stations and SEGA sold 500.000 dream cast consoles in the US in just two weeks. Nowadays,

the dream cast console can be connected to the internet and trust make possible multi-user games. Given the low price of theses console, they might even be able to supersede personal computers as internet stations.

3.3 Trends in Multi-user and Gaming Applications

The multiplayer games are notoriously difficult to implement correctly of effectively because they are multiple traditional types of software rolled into a single application [SINGHAL, SANDEEP. AND ZIDA, MICHAEL. 1999], like:

- **Distributed Systems:** They must contend with all of the challenges of managing network resources, data lost, network failure, and concurrency.
- **Graphical applications:** They must maintain smooth, real-time display frames rates and carefully allocate the CPU among rendering and others tasks.
- **Interactive applications**: They must process real-time data input from users. Users should see the virtual environment as if it exists locally, even though its participants are distributed at multiples remote hosts.

The multiplayer game industry has influenced applications including simulation, education and training [Jain, Sanjay, McLean, Charles R., 2005].

In principle, developers have relied on C and C++ exclusively for gaming development [Davison, Andrew, 2005].

While providing a tight coupling with operating systems, this approach is slowly changing towards development in languages that are more portable and provide for faster development cycles e.g. Java, JavaScript, VB, Python, and Lua. [JAWORSKI, JAMES, 1999][HARBOUR, JONATHAN S., 2002][RILEY, SEAN, 2004][GUTSCHMIDT, TOM, 2004][TATE, BRUCE A.,GEHTLAND, JUSTIN, 2004]. Nowadays, already exist game's architecture implemented for compatibility and ease of development, for WWW through of new technologies shows next-generation gaming architecture with rapid development.

4. Overview of Architecture

We now present our architecture and more thoroughly describe the pros and cons in building a game system.

The architecture shows the integration between Ajax, X3D and Web Services that introduces a solution for multi-user gaming system. This solution provides some advantages like: scalability, security and interactive responsiveness.

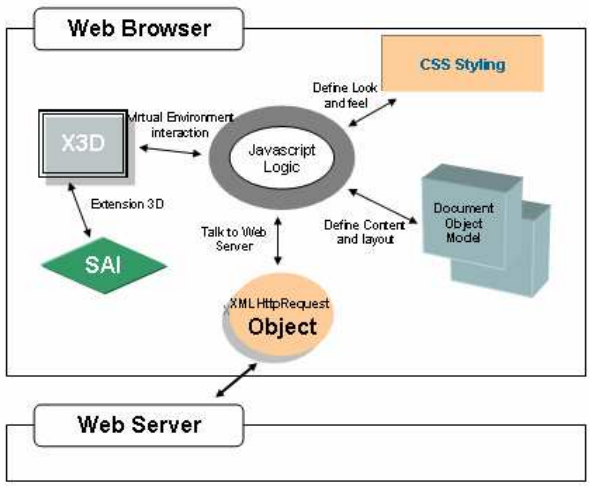


Figure 4: Architecture based X3D and Ajax

AJAX have found acceptance across a number of web based games starting from board based games such as WEBGOGGLE and working towards 3D interactive games including Strike Fighter. While some of the architectures surveyed at this writing maintain similar qualities to our application, direct comparisons are difficult to perform.

Advantages of our approach include support of web services, a portable solution that could be implemented across a number of architectures and a small program size completely implemented in scripting languages. The architecture with Ajax communicating with a web server without refreshing the whole page and another important driver and a significant benefit to Ajax architecture is the fundamental orientation around data-versus-documents and aligns well with a Service-Oriented Architecture (SOA).

The technique adopted a hybrid peer-server communication that when disseminating information, a host transmits data to some hosts directly and to other hosts through a central server. This provides a more natural model for network communication and has the potential for scalability as it more readily removes a centralized, potentially bottlenecked, server. Bottlenecked servers that seek to host more and more players on the same world, will incorporate increasingly sophisticated back-end architectures.

The main goal this project is propose a open architecture distributed portable across platforms, providing support for communication to non X3D applications, allows the routing of events over a network using the same routing, high efficiency transfer of data or node, secure communication and support the main Web/internet standards.

4.1 Authentication

Authentication and access control are necessary for any multiplayer game. For example, a game which requires a monthly subscription should only allow players who have paid to join the game. A player who has been banned from a game should not be allowed to

rejoin the game. The challenging problems with distributed authentication are security and scalability.

4.2 Communication Architecture

Conceptually, a web service provides an alternative way of exposing application logic to a heterogeneous client. Web Service enable clients and services to communicate regardless of the object model, programming language, or runtime environment used on either side of the communication link and that differentiates it from others remote access mechanisms such as RMI and CORBA. The web service exposes application functions over the Internet.

The primary advantages of using a distributed communication are scalability, responsiveness and resiliency. Distributed communication allows players to send messages directly to each other, creating a more responsive game. The key challenges in developing communications components for a distributed architecture are a consistency, event ordering, interactive responsiveness, security and scalability.

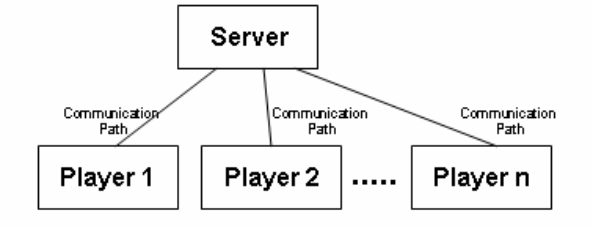


Figure 5: Multi-User client/server (logical architecture)

As showed in Figure 6, the logical view how messages are being passed through the gaming system. The advantage is change code in one place and then all players using theses objects instantly have access to the new code. The disadvantage here is that if you make a coding error, all system now accesses that error.

Ludos Top was designing to use the standard XHR object to request data from the server and update the server with changes to the data. These requests to server and made asynchronously, thus leaving the user interface on the client responsive as opposed to synchronous requests in which the client user interface appears to lockup as it is awaiting a response from the server.

Due Ajax architecture, you reduce the application latency experienced by the end user. To work around the fact that JavaScript is a single threaded, one generally uses asynchronous request to the server. Asynchronous requests are sent to the server; at which point rather that blocking the program wait for the server response, the JavaScript thread continue to execute. When the response from the server is received by the web browser, the single JavaScript thread is used to execute a callback function that was registered for that particular request before is sent. This further improves the user experience because the application continues to be responsive while data is passed to and from server behind the scenes.

The key to this AJAX-based user interaction is that it is focused on sending small pieces of data, not a rendered HTML web page, to and from server rather than a monolithic web page assembled completely by the server.

4.3 Computation

The computation component, is used to schedule game computations among the players. The ability to hardness million of players' processors create exciting possibilities for the virtual experience that player can participate in.

The server in client/server architecture typically does not have enough power to simulate complex interactions between players and the artificial intelligence (AI) because inter-process communication is slow over the internet.

4.4 Network Utilization

A recent study on Development.com [Developer, 2007] found that Ajax had the potential to reduce the number of the bytes transferred over the network by 73 percent, and total transmission time by 32 percent. In a sample application, user experienced the following measurable benefits:

- ***Reduce time spent waiting for data to be transmitted*** – Over many repetition, the time users spend waiting for the page load can add up to significant costs.
- ***Time spent completing a particular task –*** Increased efficiency in the user interface can often mean that time is saved at the task level.
- ***Bandwidth consumed for the entire task –*** Reducing the amount of data that must be processed and transferred over the wire.

4.5 Comparison between Architectures

The W3C's Architecture of the World Wide Web proposed three kind architecture to integrate distributed application. The table1 - provides a comparing between three main approaches.

	Web-Based	Script Node	Direct Networking
Support Web Browsers	**High**	**Medium**	**Low**
Communication Architecture	**Flexible**	**Limited**	**No Supported**
Heterogeneity Capability	**Yes**	**Yes**	**Yes**
Callback Techniques	**Yes**	**No**	**No**
Network I/O	**Yes**	**No**	**Yes**
Portability Capability	**Yes**	**No**	**No**
Maintenance Cost	**Low**	**Medium**	**High**

Table 1: Comparing three kind of distributed architecture.

These conceptual architectures of a 3D Browser showing external interfacing network communication capabilities.

4.6 Comparison with Other Remote Component Access Mechanisms

The table 2, show how a web service distinguishes itself from other remote component access mechanisms, such as CORBA, RMI, and Remote Procedure Call over Inter-OBR Protocol (RPC-IIOP) in several important ways.

Traditional RPC	Web Service
Within enterprise	Between enterprises
Tied to a set of programming language	Program language independent
Procedural	Message driven
Usually bound to a particular transport	Easily bound to different transports
Tightly coupled	Loosely coupled
Firewall unfriendly	Firewall friendly
Efficient Processing (Space/time)	Relatively inefficient processing

Table 2: Comparing Web Services to Others Type of RMI

5. Ludos Top

The system proposed based in game Quarto, created by Frenchman Blaise Müller in 1985. The sixteen game pieces show all combinations of size (short or tall), shade (light or dark), solidity (shell or filled), and shape (circle or square). Two players take turns placing pieces on a four by four board and the object is to get four in a line with the same characteristic - all short, for example. Only one piece can go in a cell and, once placed, the pieces stay put. Blaise Müller's brilliant twist is that you choose the piece that your opponent must place and they return the favor after placing it.

Figure 6: Game board at start game

5.1 Functioning of the game

As the objective of the game is establish a line of four pieces, with at least one common characteristic on the board. The line of pieces may be across the board, up and down, or along a diagonal.

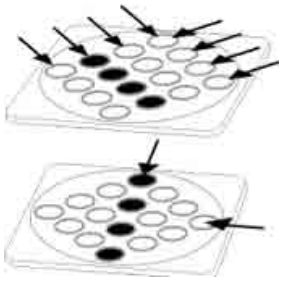


Figure 7: Objective of the game

Game sequence when the first player selects one of the 16 pieces and gives it to his opponent. That player places the piece on any square on the board and chooses one of the 15 pieces remaining and gives it to his one. The game is won by the first player to call "QUARTO!".

This project was motivated because is educational work, nowadays only 5% of the school public's students can access for Internet. Currently the Brazil has 54 millions of the students in public school, and only 2,5 million has this access to internet. But the govern has an audacious plan goal which is make almost universal access internet into next four years. According to with the Plan denominated Plano de Desenvolvimento da Educação – PDE, (Plan of Education's Development) till 2008, more than 50% of the students will have access for WWW by Broadband. In 2010, is reach 95% of students. It is an emphatic resolution from President Luíz Inácio Lula da Silva, say the Distant Education Secretary Ronaldo Mota. The first goal is be every 140.000 public school to have computer laboratory with minim of the ten computers.

5.1 Aspect Pedagogical

David reported [David, 1997] that there is an increasing demand for greater interactivity to be built into learning materials. There is a clear need to offer a variety of different knowledge presentations and to create opportunities to apply the knowledge within the virtual world, thus supporting and facilitating the learning process. To achieve that, it is necessary to provide a complex level of interactivity that stimulates users' engagement, apply different interactivity concepts as object, linear, construct or hyperlinked interactivity, and non-immersive contextual interactivity as well as immersive virtual interactivity.

When using computer games, and games in general, for educational purposes several aspects of the learning process are supported: learners are encouraged to combine knowledge from different areas to choose a solution or to make a decision at a certain point, learners can test how the outcome of the game changes based on their decisions and actions, learners are encouraged to contact other team members and discuss

and negotiate subsequent steps, thus improving, among other things, their social skills.

The Game provides a wealth of opportunities for children to acquire and practice the use of a wide range of problem solving skills and strategies. In games children encounter problems, often of their own making, under sets of conditions that are clearly defined and well understood by the players. Games are micro worlds for learning and could have a valuable place in the school curriculum.

The main aspect pedagogical are problem solving skills such as trial-and-error investigation, hypothesis testing and searching for relationships, reasoning skills such as inference, deduction analysis and evaluation, social skills of cooperation, communication and constructive argument and life skills such as perseverance, and the ability to see initial failure as a challenge and to learn from it.

The games Quatro was denominated Ludos Top System in this project, which may be played against the computer or a human opponent. The level of expertise for the computer can be selected. The technical advantages are employing interactive through of Ajaxian techniques that provides better enjoyability, playability and usability

6. System implementation

The implementation of the system denominates Ludos Top which introduces the game Quatro and some systems like: Ajax Techniques, Flex Flux plug-in 3D, Internet Explorer browser and Tomcat.

6.1 Ludos Top Web User Interface

Ludos Top Web User Interface is a HTML interface programmed using JavaScript. The user "Master" acts as moderator of game and operation. The functions like "Login", "Moving" and "Chat" available in main interface like in Figure 9.

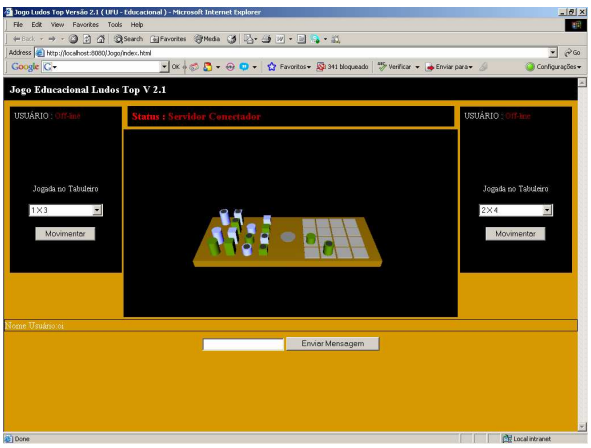


Figure 8: Snapshot from Ludos Top

After an authentication in Ludos Top software, the player has to establish a connection available with game Quatro, these actions are performed by adding a Web reference to the Web service Tomcat.

The Ludos Top interface (Figure 1) is a HTML programmed using JavaScript, Ajax and X3D. The communication between X3D and HTML page is through the X3D Script SAI, after this, the parameters are forwarded outside world for Web Services using Ajax request with XMLHttpRequest.

7. Conclusion

In this paper presents the possibilities of fully distributing multi-user games online through of standards such as X3D have made it possible deploy rich 3D content in real time over the Internet.

At the same time, Ajax has emerged as a worldwide phenomenon and an interest of new application development. By bringing these two technologies together, Ajax3D promises to be good open platform for creating a next-generation 3D web experience.

With Ajax3D, immersive virtual worlds can be deployed within a web browser, integrated with pages with more interaction, compatibility and can communicate with standard web servers using XML and Ajax technologies, enabling networking on web. The architecture with Ajax can improve and empower the user experience for the end users, making them more effective and satisfied, reduce the demands on networks and server infrastructure, saving money by reducing maintenance and even bandwidth, and improve quality of service for all users.

This work presents the possibility for new kinds of functionality not possible or practical in a traditional application model, giving users new tools to achieve their goals.

The Ludos Top is in its infancy. It will need to be actively developed. Ludos Top can become an example of Educational Virtual World on World Wide Web for help research and development personal.

Acknowledgements

The authors would like to thank the Edgard Lamounier Jr. and Alexandre Cardoso at the University Federal of Uberlândia for supporting the establishment of the Computer Graphics lab and orientation.

References

A. OSTROWSKI, DAVID, 2007. *A Web-Based 3D Gaming Style Multi-User Simulation Architecture*. Ford Research and Advance Engineering.

LEIGH, J. AND JOHNSON, A. E.1996a. *Supporting Transcontinental Collaborative Work in Persistent Virtual Environments*. In IEEE Computer Graphics and Applications. July 1996, pp. 47-51.

M. ROUSSOS, A. E. JOHNSON, J. LEIGH, C. R. BARNES, C. A. VASILAKIS, AND T. G. MOHER, 1997. *The NICE project: Narrative, Immersive, Constructionist/Collaborative*

Environments for Learning in Virtual Reality. In IEEE Computer Graphics and Applications.

STEINER, K. E. AND MOHER, T. G. 1994. *Scaffolding Story Construction with Interactive Multimedia*. In The Journal of Educational Multimedia and Hypermedia, pp.173-196.

SINGHAL, SANDEEP. AND ZIDA, MICHAEL. 1999. *Networked Virtual Environment*. In The Addison Wesley Longmann book, pp.08-41.

ZYDA MICHAEL, 2005. "From Visual Simulation to Virtual Reality to Games", IEEE Computer, pp. 25-32.

GARAGE GAMES WEB SITE, 2007. www.garagegames.com

KARTCH, D., 2000. *Efficient rendering and compression for full-parallax computer-generated holographic stereograms*. PhD thesis, Cornell University.

S. CLARKER-WILLSON., 1998. *Applying Game Design to Virtual Enviroments*. In C. Dodsworth (editor), Digital Illusion: Entertaining the Future with High Technology. ACM Press.

JAIN, SANJAY, MCLEAN, CHARLES R., 2005. *Integrated Simulation and Gaming Architecture for Incident Management Training*. PROCEEDINGS OF THE 2005 WINTER SIMULATION CONFERENCE.

DAVISON, ANDREW, 2005. *Killer Game Programming in Java*, OReilly.

DUCHENEAUT, N., YEE, N., NICKELL, E. AND MOORE, J.R., 2006. *"Alone together?": exploring the social dynamics of massively multiplayer online games. In: Proceedings of the SIGCHI conference on Human Factors in computing systems, 22-27 April 2006 Montreal*. New York: ACM Press, 407-416.

JAWORSKI, JAMES,1999. *Matering Javascript*, Sybex pub.

HARBOUR, JONATHAN S., 2002. *Visual Basic Game Programming with Direct X*, Premier Press.

RILEY, SEAN, 2004. *Game Programming with Python,* Charles River Media.

GUTSCHMIDT, TOM, 2004. Game Programming with Python, Lua and Ruby, Premier Press.

TATE, BRUCE A.,GEHTLAND, JUSTIN, 2004. Better, Faster, Lighter Java, O'Reilly.

CRANE, DAVID, PASCARAELLO, ERIC, 2005. James, Darren, Ajax in Acton, Manning pb.

H. Pillay, J. Brownlee, 1999 and L. Wills, *"Cognition and Recreational Computer Games: Implications for Educational Technology"*, Journal of Research on Computing in Education, Vol. 32 No. 1, pp. 203-215.

E. VON GLASERSFELD, 1990. "*Constructivist Views on the Teaching and Learning of Mathematics*". Journal for Research in Mathematics Education. Monograph, Vol. 4, pp. 19-29+195-210.

DIEHL, S., 2001. *Distributed Virtual Worlds*. Páginas 10 – 51. Editora Spring, 2001

RIBEIRO, M., LAMOUNIER, E. CARDOSO, A., 2004 *Uso de Corba na Distribuição de Ambientes Virtuais para Suportar Multidisciplinaridade no Processo de Educação"*. IV Seminário de Realidade Virtual. SVR´2004.

GRENVILLE, A., MARK, C. E PHILIP, B., 2006. *Networking and Online Games: Understand and Engineering Multiplayer Internet Games,* pages 12 – 15, Editora Wiley.

S. S. A. R. BHARAMBE,, S. RAO., MERCURY, 2002. *A scalable publish-subscribe system for internet games*. In Proceedings of the First Workshop on Network and System Support for Games.

PARISI, TONY, 2006. http://www.ajax3d.org/ whitepaper.

BUSCHMANN, F. MEUNIER, R., ROHNERT, H., SOMMERLAD, P., STAL, M. (1996): PATTERN-ORIENTED SOFTWARE ARCHITETURE – A SYSTEM OF PATTERNS. JOHN WILEY & SONS, CHICHESTER.

EMMERICH, W. (2000): ENGINEERING DISTRIBUTED OBJECTS. JOHN WILEY & SONS, CHICHESTER.

GAMMA, E., HELM, R., JOHNSON, R., VLISSIDES, J. (1995): DESIGN PATTERNS: ELEMENTS OF REUSABLE OBJECT-ORIENTED SOFTWARE. ADDISON-WESLEY LONGMAN.

KIRCHER, M., JAIN, P. (2004): PATTERN-ORIENTED SOFTWARE ARCHITECTURE, VOL.3: PATTERNS FOR RESOURCE MANAGEMENT. JOHN WILEY & SONS, CHICHESTER.

CARDOSO, A E LAMOUNIER, E, *" A Realidade Virtual na Educação e Treinamento.* In Realidade Virtual: "Conceitos e Tendências ", Ed. Mania de Livro Pré-Simpósio SVR 2004. , pp.256–264.

BERNARDI, G., CASSAL, M. L., *"Proposta de um ambiente de Ensino-Aprendizagem Utilizando Jogos e Realidade Virtual",* XIII Simpósio Brasileiro de Informática na Educação – SBIE – UNISINOS, 2004, pp. 535.

MACEDO, L. ET *al "Aprender com Jogos e Situações Problemas*", Ed. Artes Médicas, São Paulo- SP, 2000.

DAVID, 1997 ET AL.: *"Integrated Development and production Tools for Building Hypermedia Courseware and Interactive Scenarios". Proc. Of ED-MEDIA'97.*

GARRIS,R., AHLERS,R., AND DRISKELL,J.E. 2002. *"Games, motivation and learning, Simulation & gaming*; An Interdisciplinary Journal of Theory, Practice and Research". Vol33, No.4.

QUINN, C.N. 1999 *The Play's the Thing: Enhancing Learning Design Through Game Elements.* Tutorial at the AI-ED99, LeMans, France.

DEVELOPER , 2007. *http://www.developer.com/xml/article.php/3554271 accessed: September 28th, 2007.*

Hybrid OpenMP/MPI Parallel Programming of a Finite Elements Method Application

Leonardo Nunes da Silva
Dept. of Computer Science
University of Brasília, Brazil
nsleonardo@unb.br

Flávia Romano Villa Verde
Dept of Mechanical Engineering
University of Brasília, Brazil
flaviarvv@unb.br

Gerson Henrique Pfitscher
Dept. of Computer Science
University of Brasília, Brazil
gerson@unb.br

***Abstract* - In parallel processing and parallel algorithms several processors are used together to execute a single application faster. From the programmer's point of view there are two major programming paradigms: Shared Memory and Message Passing. Each of them fits into a specific physical model, but there are multiprocessors architectures whose mapping to one of these paradigms is not so simple. SMP clusters, for example, are built connecting some shared memory machines through an interconnection network. Applications on SMP clusters can be programmed to use message passing between all processors. However, it's possible to achieve better performance using a hybrid model which uses multithreading with shared memory communication inside the SMP node and message passing communication between nodes. The objective of the research presented in this paper was to implement and evaluate a hybrid model of parallel programming for an engineering application in order to evaluate and compare this model with a pure message passing version. Finite element formulation of linear elasticity problems results in linear equation systems that can be solved numerically by the conjugate gradient method. This article presents some numerical simulation results of a structural analysis, where the stiffness matrix of the system is a full matrix, using a 3D parallel code with two strategies for parallelization of the conjugate gradient method. The solution is obtained without any preconditioning technique to provide a performance comparison between a hybrid and a pure version of the same application.**

I. INTRODUCTION

In the area of parallel processing, several processors are used together to execute a single application faster. There are two major programming paradigms: shared memory and message passing [1]. The shared memory programming model targets a shared memory architecture, in which multiple processors share single memory space. The communication between processors takes place through reading and writing in this memory space. In the distributed memory programming model, processors do not share memory; instead, they explicitly send and receive messages. The only way to acquire the data that are not in the local memory is to request and receive them from the processor that has them.

In order to tell the underlying machine that a program should be executed in parallel, we need some form of programming language constructs. These constructs control data sharing, synchronization, and so on. The two paradigms offer different sets of parallel constructs to achieve this.

In the message passing model, the constructs typically come in the form of library of functions. The library includes functions for sending and receiving messages, synchronizing execution, and so on. The Message Passing Interface (MPI) is an important standard that is implemented in the form of such libraries. The parallel programmer's task in the message passing model is to incorporate these functions into the algorithm. Programmers need to devise ways to split data, communicate, and synchronize, and write or modify the program based on the idea. On the OpenMP model for shared memory programming, programmers inserts "directives" into the code. These directives do not affect the program semantics. They dictate how work and data shall be shared by the parallel processors. The source code is then compiled by a compiler which provides support for the model and it generates code that creates threads to run on several processors [1].

Each of them fits into a specific physical model, but there are multiprocessors architectures whose mapping to one of these paradigms is not so simple. SMP clusters, for example, are built connecting some shared memory machines through an interconnection network. Applications on SMP clusters can be programmed to use message passing between all processors. However, it's possible to achieve better performance using a hybrid model with shared memory communication inside the SMP node and message passing communication between nodes. A hybrid programming model is defined as a model which uses multithreading for shared memory inside the node and message passing between SMP nodes.

The objective of the research presented in this paper is to develop and evaluate a hybrid model of parallel programming for a real engineering application based on the finite elements method [2]. Besides that we aim to provide a performance comparison between a hybrid and a pure version of the same application.

II. PARALLEL PROGRAMMING

A. *Message Passing Programming*

The Message Passing Interface (MPI) is a standard for writing message-passing programs. The goal of MPI is to

provide standard library of routines for writing portable and efficient message passing programs. MPI provides a rich collection of point to point communication routines and collective operations for data movement, global computation and synchronization [3].

An MPI application can be visualized as a collection of concurrent communication tasks. A program includes code written by the application programmer that is linked with a function library provided by the MPI software implementation. Each task is assigned an unique rank within a certain context: an integer number between 0 and n-1 for an MPI application consisting of n tasks. These ranks are used by the MPI tasks to identify each other in sending and receiving messages, to execute collective operations and to cooperate in general. MPI tasks can run on the same processor or different processors concurrently [3].

B. *OpenMP Programming*

OpenMP is a set of compiler directives and callable runtime library routines to express shared memory parallelism. OpenMP is ideal for developers who need to quickly parallelize existing scientific code without rewrite them. It also can be used to rewrite an entirely new application [4]. The programmer includes directives on the code instructing the compiler how to divide data and computation among processors. Then, when the code is compiles, the compiler generates code to be executed in parallel by the threads.

An OpenMP program executes according to a fork-join model (Fig. 1). This means that an OpenMP program begins execution as a single sequential thread, the master thread. When a parallel directive is encountered by that thread, execution forks and the parallel region is executed by a team of threads in parallel. When the parallel region is finished, the threads in the team synchronize at an implicit barrier, and the master thread is the only thread that continues execution. Figure 1 shows the OpenMP execution model [5].

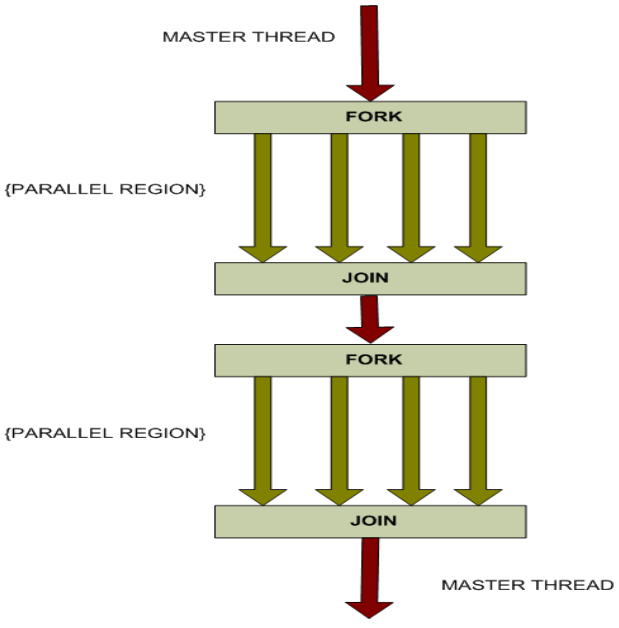


Fig.1. OpenMP execution model.

C. *Hybrid OpenMP/MPI Programming*

In [6], a hybrid programming model is defined as a model which uses multithreading for shared memory inside the node and message passing between SMP nodes. By using a hybrid programming model we should be able to take advantage of the benefits of both shared memory and message passing models

Hybrid programs should give better performance than pure message passing ones for three reasons: 1) message passing within a node is replaced by fast shared memory access; 2) there is smaller communication volume on the interconnect since internode's messages are not necessary; 3) fewer processes are involved in communication, which should lead to better scalability, particularly for global communications [7, 8].

Applications may or may not benefit from hybrid programming depending on some applications parameters [9]. In [10] is presented an efficient parallel iterative method for unstructured 3D solid mechanic grids developed for SMP cluster architectures with vector processors. The method is based on a 3-level hybrid parallel programming model, including message passing for inter-SMP node communication, loop directives by OpenMP for intra-SMP node parallelization and vectorization for each processing element. The method is focused on vector/parallel efficiency rather than robustness of the preconditioners themselves.

The advantages are clearly application dependent [8]. In [8] has been presented a classification based in the programming efforts: fine-grain and coarse-grain parallelization.

The fine-grain approach is the simplest one. It consists in OpenMP parallelization of the loop nests in the computation parts of the MPI code. In this case, is important to choose loops that contribute significantly to the global execution time. In the coarse-grain approach, OpenMP is still used to take advantage of the shared memory inside the SMP nodes but a SPMD programming style. At the beginning of the main program, N OpenMP threads are created and each of them acts as it was an MPI process [8].

III. CASE STUDY

Linear elasticity problems modeled by a system of equations generated by the finite elements method can be solved by the conjugate gradient method. With the finite elements method, instead of assuming some properties for the entire body, we divide it into smaller elements and assume these properties for these individual elements [2].

These individual elements are now analyzed and instead of carrying out integration over the entire body we carry out summation over the body consisting of finite number of elements of finite dimensions. The FEM leads to a system of equations of the form

$$\boldsymbol{A}\boldsymbol{x} = \mathbf{b}. \qquad (1)$$

Where A is a square, symmetric, positive-definite matrix, x is an unknown vector and b is a known vector [2].

The conjugate gradient method is then applied to iteratively solve the system of equations. The conjugate gradient method

consists of a loop over a fixed number of matrix vector and vector-vector operations. In linear elasticity problem simulation, a geometry is selected and some boundary conditions are applied to constraint the movement of the solid in space. After that, forces are applied in some points of the structure. At the end of the conjugate gradient method execution, the vector x give the resulting displacement for the points of the geometry.

A. Methodology

We first applied the incremental approach to obtain a fine-grained version of the original MPI code. We focused on parallelizing the loops on the parallel conjugate gradient method because this consumes about 90% of the application time. The code has been instrumented using the RDTSC instruction and the total execution time has been decomposed in computation time and communication time [12]. This decomposition is important to investigate our initial hypothesis that a hybrid solution would decrease the communication time and consequently decrease the computation time.

In this paper we refer to tasks as units of processing. Here, not only MPI processes but also OpenMP threads are called tasks. Thus the comparisons have been made between executions with the same number of tasks in both models. For example, 1 MPI process with 2 Threads in the hybrid model against 2 MPI processes (running on the same SMP node) in the pure model.

IV. PARALLEL CODE

The application input is a mesh for the selected geometry. First, this mesh is partitioned using the library METIS [10]. The number of partitions is equal to the number of MPI processes. Thus every process handles about the same volume of data. Then every process mount its matrix and some auxiliary data structures based on the partition it receives. Finally a parallel conjugate gradient method is applied to solve the system of equations.

During this last step, there is intensive communication between the processes because adjacent elements share nodes on their boundary and they need to exchange information about them during the computation. At the end of the parallel conjugate gradient method, each process has got part of the solution and a reduce operation is performed to create the resulting x vector. Fig. 2 shows the application flowchart.

V. SIMULATION RESULTS

The geometry selected to our experiments is shown on Fig. 3. One face of the solid has been restricted and a force of 5.000 N has been applied on the z axis. The measures have been made for four 3d meshes detailed on TABLE I. Fig. 4 shows this geometry divided in 933 elements for mesh 2. The validation of the parallel 3D code was made through the comparison of the results with those obtained from the execution of the sequential program ef++ [13], as it can be seen in Fig. 5. For the two programs, the Young´s module was considered to be E=21.0MPa and the Poisson coefficient υ = 0.2.

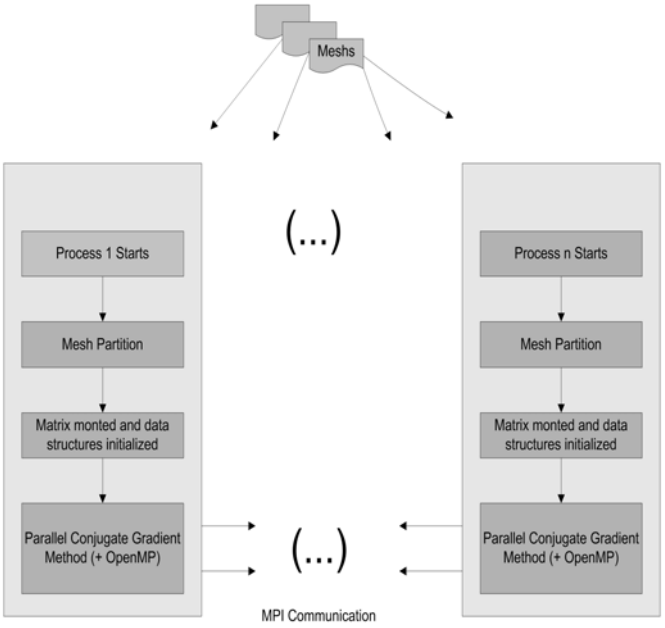


Fig. 2. Application flowchart.

TABLE I
MESHES CHARACTERISTICS USED IN SIMULATIONS

Mesh	Nº Equations	Nº Elements	Nº Nodes
1	1,266	1,882	467
2	2,577	4,014	933
3	5,567	9,126	1,984
4	10,581	18,031	3,717

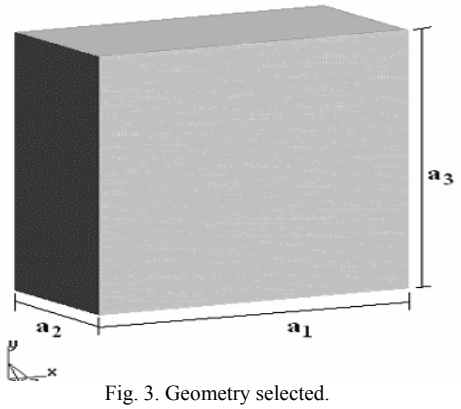


Fig. 3. Geometry selected.

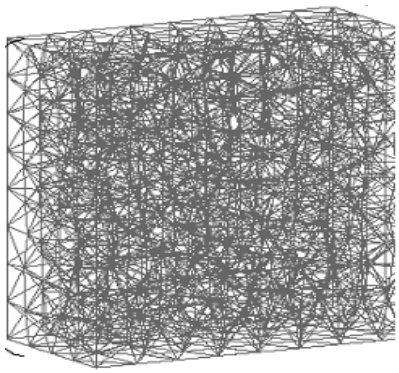
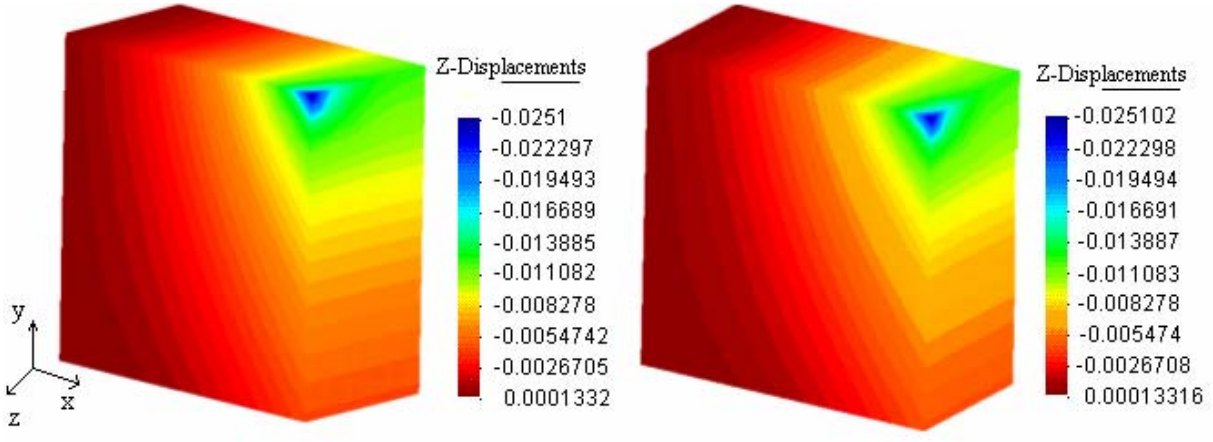


Fig. 4. Mesh 2 with 933 nodes and 4,014 elements.

The generated mesh was the same used to validate the code, resulting in 2,577 degrees of freedom. The results were obtained after 245 iterations of the CGM with a tolerance of 1x 10^{-4}. Figures 6 to 9 show total time comparison for each of the meshes for hybrid and pure versions of the application, and Figs. 10 to 13 show execution times for each of the meshes, decomposed in computation time and communication time.

The calculations has been performed on a cluster of 8 AMD 1,7 GHz dual processor nodes with 1,0 GB of RAM connected by a Ethernet *Switch* of 1,0 Gb/s. The application has been compiled using the Intel 9.0 Compiler with OpenMP support. The MPI library used was the mpich-7.2.



(a) (b)

Fig. 5. Resulted displacements in z (meters);. (a) sequential ef++ program, (b) parallel 3D code.

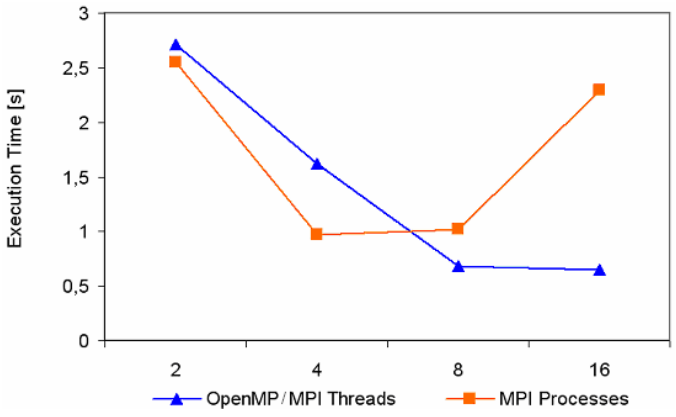


Fig. 6. Total execution times for Mesh1.

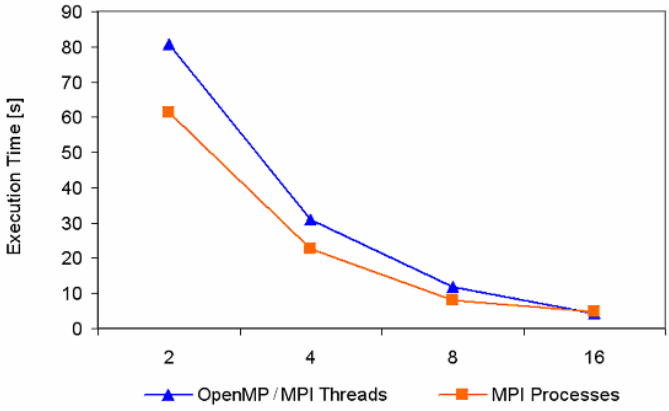


Fig. 8. Total execution times for Mesh3.

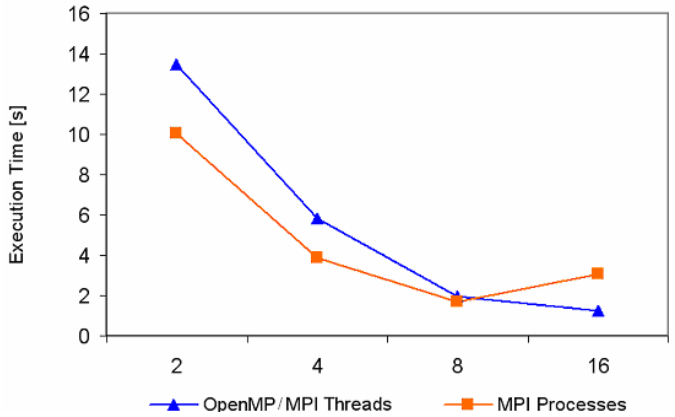


Fig. 7. Total execution times for Mesh2.

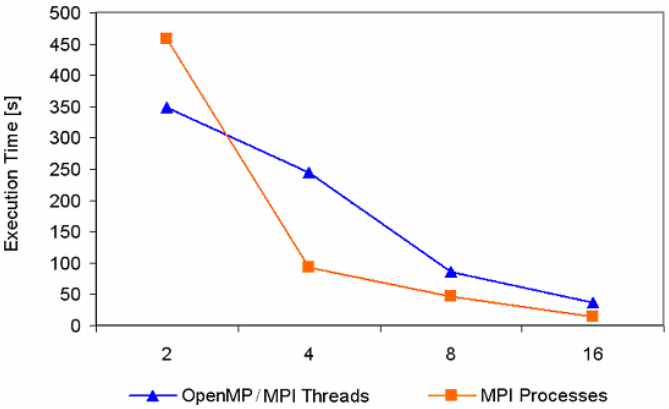


Fig. 9. Total execution times for Mesh4.

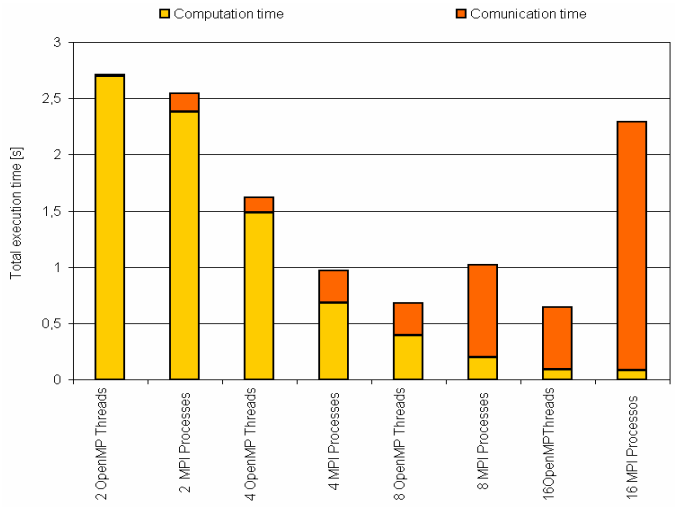


Fig. 10. Decomposed execution time for Mesh1.

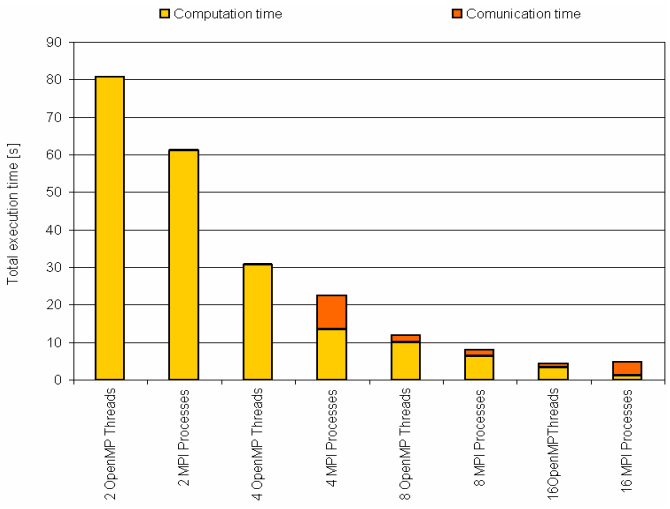


Fig. 12. Decomposed execution time for Mesh3.

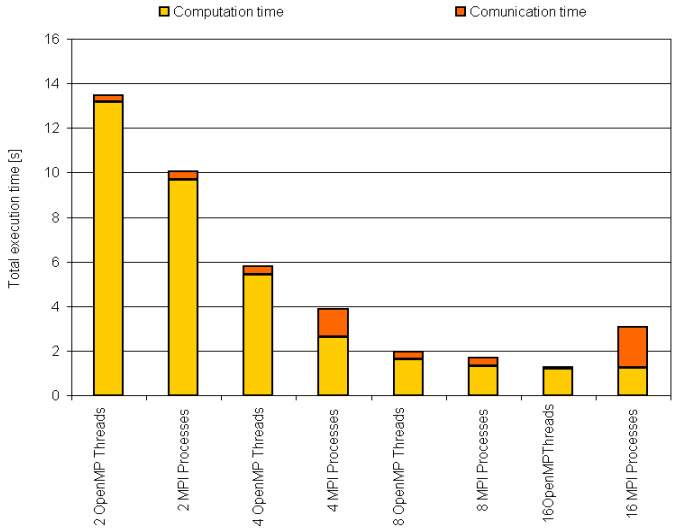


Fig. 11. Decomposed execution time for Mesh2.

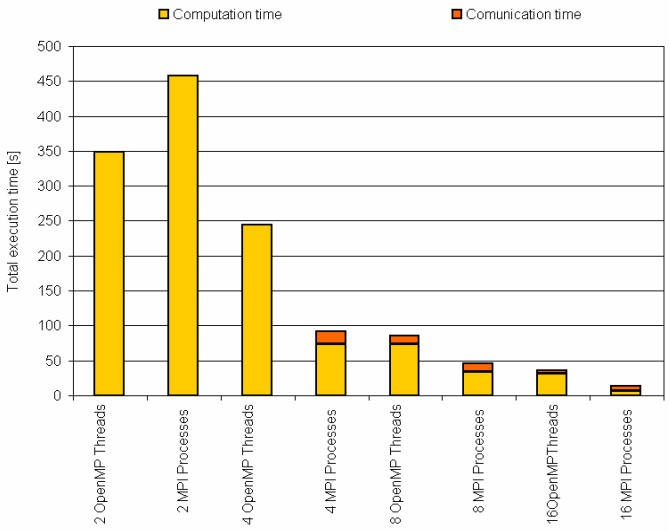


Fig. 13. Decomposed execution time for Mesh4.

The execution time results in Figs. 6 to 9 shows that in most cases the MPI total time is less than the hybrid one. On the other hand, in Figs. 11 to 13 we can see, as expected, that in general the communication time on the pure version is higher than the communication time in the hybrid one. We can see also that the hybrid model has a worst computation time in most cases and the communication advantage do not compensate the computation disadvantage.

The bad results can be explained by the major weakness of the model. They are [14]: (a) the replication of the OpenMP parallel regions implies a high thread management cost; (b) parallel regions lead to a bad utilization of the memory hierarchy. The factor (a) is especially important in this application because we have used a fine grain approach. Besides that, because the application uses an iterative method to solve the equations, the threads must be forked several times.

VI. CONCLUSIONS

In this paper we developed and evaluated a hybrid model of parallel programming for a real engineering application based on the finite elements method. Hybrid models on SMP clusters have been used in several applications. Although some works like [15, 16 and 17] have reached better performance, most studies like [7, 9 and 14] come to a conclusion that the hybrid version loses to pure MPI versions in most cases. In [8] the author shows that the comparison results are clearly application dependent.

However there are some optimizations like the one presented in [17] that can lead to very good results, [10] is focused on vector/parallel efficiency rather than robustness of the preconditioners themselves. Most of these optimizations, however, require a high programming effort and sometimes the original application must be completely rewritten to use them. In our results we have seen that in most cases the performance

of the pure MPI model is better than the performance of the Hybrid model. However, we saw that with a hybrid model the communication overhead is reduced.

This result shows that the use of a shared memory model inside the SMP node effectively decreases the communication overhead. Thus, with some optimizations this technique can be very useful for some applications, especially those with intensive communication.

REFERENCES

[1] W.P. Petersen, and P. Arbenz, *Introduction to Parallel Computing*, Oxford, Oxford University, 2004.

[2] Villa Verde, F.R., *Parallel solution for a finite element code applied to a linear elasticity on a cluster of PCs* . M. Sc. Thesis in Mechanical Engineering, University of Brasilia, 2004.

[3] M.Snir, S. Otto, S.Huss-Lederman, D. Walker, and J. Dongarra, *MPI: The Complete Reference*, Cambridge, The MIT Press, 1996.

[4] Dagum, L. and Menon, R., "OpenMP: An industrystandard API for shared-memory programming". *IEEE Computational Science & Engineering*, 5(1):46--55, 1998.

[5] Inter Corporation; *Extending OpenMP to Clusters*. Techinical Report 2006.

[6] L. Smith, and M. Bull, "Development of Mixed Mode MPI / OpenMP Applications", *Proc. of the Workshop on OpenMP Applications and Tools*, July 2000.

[7] E. Chow, and D. Hysom, *Assessing performance of hybrid MPI/openMP programs on SMP clusters*, Technical Report, Lawrence Livermore National Laboratory, May 2001.

[8] F. Cappello, and D. Etiemble, "MPI versus MPI+OpenMP on IBM SP for the NAS benchmarks", *Supercomputing*, 2000.

[9] R. Rabenseifner, "Hybrid Parallel Programming: Performance Problems and Chances", *Proceedings of the 45th CUG Conference*, Columbus, Ohio, May 2003.

[10] K. Nakajima, H. Okuda, "Parallel Iterative Solvers for Unstructured Grids Using an OpenMP/MPI Hybrid Programming Model for the GeoFEM Platform on SMP Cluster Architectures", *Proceedings of the 4th International Symposium High Performance Computing, ISHPC 2002*, Kansai Science City, Japan, May 2002.

[11] G. Karypis, and V. Kumar, *METIS – A Software Package for Partitioning Unstructured Graph, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices*, University of Minnesota, Department of Computer Science, 1988.

[12] Intel Corporation, *Using the RDTSC instruction for performance monitoring*, Technical Report, 1996.

[13] A. T. S., Bernardo, *Pós processador* ef++: *Manual do Usuário.* Department of Mechanical Engeneering, University of Brasilia 2003**.**.

[14] G. Krawezik, F. Capello, "Performance Comparison of MPI and three OpenMP Programming Styles on Shared Memory Multiprocessors", *SPAA 03*, San Diego California, June 2003.

[15] A. Grabysz, and R. Rabenseifner, "Nesting OpenMP in MPI to Implement a Hybrid Communication Method of Parallel Simulated Annealing on a Cluster of SMP Nodes", *Proceedings EuroPVM/MPI* 2005.

[16] R. D. Loft, S. J. Thomas, J. M. Dennis, "Terascale Spectral Element Dynamical Core for Atmospheric General Circulation Models", *Proceedings SC 2001*, Nov 2001.

[17] T. Q. Viet, T. Yoshinaga, B. A. Abderazek, and M. Soway, "A Hybrid MPI-OpenMP Solution for a Linear System on a Cluster of SMPs", *Symposium on Advanced Computing Systems and Infrastructures* 2003.

A Method to Estimate Physical and Cognitive Effort in Chording Data-Entry Systems

A.M. Marcon[1], G.F.G. Yared[2],
A.F.Silva[1], H.M.Bilby[1], N.S Bezerra[2], E.B.Rodrigues[1],V.C. Marques[1]
[1] BenQ Mobile Brazil
[2] Universidade do Estado do Amazonas
Av Djalma Batista, 536 - Manaus - Amazonas – Brazil – CEP: 69053-270

***Abstract*- As computer technology has become ubiquitous in nature, research in text and data entry is relevant to address new user needs, according to the evolution of computing equipments and changes in user interface. This paper presents a method to estimate the physical and cognitive cost of usage for chording text-entry based systems.**

I. INTRODUCTION

While the QWERTY system [2] governs supreme as the primary text entry device for most computing systems, the evolution in computing equipments poses new challenges in man machine interfaces. Many trials have been done to address such changes. Some of them resulted in prediction algorithms, which convert numeric entries into alphanumeric characters based on statistical models. Other initiative shrinks the traditional QWERTY keyboard to a level where it would be supposed to fit for specific device models. A comparative detailed study from MacKenzie & Soukoreff [3] presents a broad list of text input techniques. It must be considered that in parallel there are investigations in course regarding the adoption of voice-speech recognition (VSR) systems. Though there are many areas where VSR would help, like speaking a name to trigger a call, or use a dictate machine to generate a text file, there are many areas where typing text can not be so easily replaced. Situations where privacy is necessary and speaking is not an option, or situations where speaking is much more complex then typing, like editing a Unix source code file, or filling up a government form for tax declaration, also require the typing based approach.

There are two major competing paradigms for text input in computer based systems: pen-based input and keyboard-based input. Both emerged from ancient technologies ("ancient" must be understood as the pre-computer era): machine-typing and handwriting.

An alternative for text entry is to explore Chord Keyboard Systems. According to Weber [1], a Chord Keyboard is a keyboard that takes simultaneous multiple key pressings at a time to form a character. In chord keyboards, the user presses multiple key combinations to enter an input instead of using one key for each character. Pressing combinations of keys in this way is called chording. Since chord keyboards require only a small number of keys, they do not need large space, nor the many keys of regular keyboards such as the QWERTY keyboard.

It is controversial to establish an ergonomic hands-on standard for Chording Systems, mainly due to the multiple ergonomic behaviors of users, differences in the environment of use and particular user preferences, so this study is not about ergonomics.

Assuming that a certain level of effort is necessary to learn and use chord keyboards, in order to achieve effective usability, this study presents a method to express effort level in terms of tangible parameters and investigate the user performance obtained from different purpose maps: minimum physical effort and minimum cognitive effort. In the section II this paper introduces the proposed method. In the section III the experimental conditions are presented, followed by the results in section IV, and finally discussion and conclusion, are presented in sections V and VI, respectively.

II. COMPOSING: THE METHOD

In order to formalize the requirements, the first step is to propose a Method for a generic chording data entry system. Assuming that chords are combinations of entry keys, they can be represented typically as a binary sequence of digits (0-released, 1-pressed). Most likely, the physical effort level will range from *minimum effort* (no key pressed or idle) to *maximum effort* (all keys pressed). Assume **K** is generic array representing a generic chording data entry system based on ***N*** entry keys. The M symbols (characters) can be expressed through M chords represented in the K array, so that M! symbol/chord pairs are possible which are called maps. Therefore each map contains the logical relation between symbols and chords.

The Effort for any given chord ***Ec(j)*** would be expressed by Equation (1) as follows:

$$Ec(j) = \sum_{i=0}^{N-1} [W(i).C(i)].T \,, \qquad (1)$$

in which **C** is the binary sequence that uniquely represents each chord, ***T*** is the total number of active keys for a specific chord ($T \leq N$), ***i*** is the array index for any given digit in **C,** ***j*** is the index for any given symbol in a map and ***W(i)*** is the weighted effort for a single entry key in **C.** To make it more precise ***T***

should be considered in Equation (1) since the effort is minimum for ***T=0***, and maximum to ***T=N***.

For any given map its respective *Map mean effort* (***Mme***) expressing the mean cost of physical usage can be determined.

Let ***P(j)*** be the probability of the ***j***th symbol represented in the map , so:

$$\sum_{j=0}^{M-1} P(j) = 1. \qquad (2)$$

The *Map mean effort* for a given map chosen from ***M!*** possible permutations is given by Equation (3):

$$\boldsymbol{Mme} = \sum_{j=0}^{M-1} [Ec(j).P(j)] \qquad (3)$$

Choose the best combination to compose a complete symbol representation for a given Chord Keyboard **K**, including alphabet, numbers and special characters is not trivial. If the ASCII standard symbols are considered, for instance, then M should be 256 and there are 256! possible maps. This work intends to make this matter simplest as possible, for what a certain level of abstraction is proposed, where two basic symbol maps are proposed: AEOSRI map and MARCON map.

The AEOSRI map is derived from a symbol combination that minimizes physical effort (minimum Mme) according to Table I.

TABLE I
THE AEOSRI MAP

T = 1 / N ° Active Bits	Individual Key Effort: Thumb 7%	index 13%	medium 20%	ring 27%	little 33%	Chord Effort (Linear) Ec(j)	Frequency (%) P(j)	AEOSRI MAP (Portuguese) Char	Ec(j) * P(j)
0.20	1	0	0	0	0	0.0133	14.63	a	0.00195067
	0	1	0	0	0	0.0267	12.57	e	0.00335200
	0	0	1	0	0	0.0400	10.73	o	0.00429200
	0	0	0	1	0	0.0533	7.81	s	0.00416533
	0	0	0	0	1	0.0667	6.53	r	0.00435333
0.40	1	1	0	0	0	0.0800	6.18	i	0.00494400
	1	0	1	0	0	0.1067	5.05	n	0.00538667
	1	0	0	1	0	0.1333	4.99	d	0.00665333
	0	1	1	0	0	0.1333	4.63	u	0.00617333
	1	0	0	0	1	0.1600	4.74	m	0.00758400
	0	1	0	1	0	0.1600	4.34	t	0.00694400
	0	1	0	0	1	0.1867	3.88	c	0.00724267
	0	0	1	1	0	0.1867	2.78	l	0.00518933
	0	0	1	0	1	0.2133	2.52	p	0.00537600
	0	0	0	1	1	0.2400	1.67	v	0.00400800
0.60	1	1	1	0	0	0.2400	1.30	g	0.00312000
	1	1	0	1	0	0.2800	1.28	h	0.00358400
	1	1	0	0	1	0.3200	1.20	q	0.00384000
	1	0	1	1	0	0.3200	1.04	b	0.00332800
	1	0	1	0	1	0.3600	1.02	f	0.00367200
	0	1	1	1	0	0.3600	0.40	j	0.00144000
	1	0	0	1	1	0.4000	0.47	z	0.00188000
	0	1	1	0	1	0.4000	0.21	x	0.00084000
	0	1	0	1	1	0.4400	0.02	k	0.00008800
	0	0	1	1	1	0.4800	0.01	w	0.00004800
0.80	1	1	1	0	1	0.5867	0.01	y	0.00005867
								Mme	0.09951333

Therefore the AEOSRI map is a logical map based on the probabilities ***P(j)*** according to the Theory of Frequency Analysis and Information Theory [4],[6]. Notice that variations in frequency of characters are deeply influenced by language-context and there is a specific distribution for each language. On other hand, The MARCON map suggests a symbol combination that minimizes the cognitive effort (best learning curve), disregarding the level of physical effort. Despite the fact that 256! permutations are available to choose among cognitive maps, this work selected only one specially designed by the authors to be more intuitive, according to Table II.

For convention, assume minimum effort (***Min***) for a given chord ***Ec(j)=0*** when ***T=0,*** and the maximum effort (***Max***) ***Ec(j)=1*** when ***T=N***. Based on this assumptions Equation (4) is valid:

$$\sum_{i=0}^{N-1} W(i) = 1. \qquad (4)$$

The elements in the array **W** are expressed as a sequence of an arithmetic progression containing ***N*** elements, so any ***W(i)*** is given by Equation (5):

$$W(i) = i.\frac{2.Max}{N.(N+1)} \qquad (5)$$

The present Method is expected to be useful to estimate the mean effort for generic chording systems based on ***N*** keys assuming that ***Mme*** can be calculated for any map.

TABLE II
THE MARCON MAP

T = 1 / N ° Active Bits	Individual Key Effort: Thumb 7%	index 13%	medium 20%	ring 27%	little 33%	Chord Effort (Linear) Ec(j)	Frequency (%) P(j)	MARCON MAP Char	Ec(j) * P(j)
0.20	1	0	0	0	0	0.0133	14.63	a	0.001950667
	0	1	0	0	0	0.0267	12.57	e	0.003352
	0	0	1	0	0	0.0400	6.18	i	0.002472
	0	0	0	1	0	0.0533	10.73	o	0.005722667
	0	0	0	0	1	0.0667	4.63	u	0.003086667
0.40	1	1	0	0	0	0.0800	2.52	p	0.002016
	1	0	1	0	0	0.1067	2.78	l	0.002965333
	1	1	1	0	1	0.2933	0.01	w	2.93333E-05
	0	1	1	1	1	0.3733	1.04	b	0.003882667
	1	0	0	0	1	0.1600	0.40	j	0.00064
	0	1	0	1	0	0.1600	3.88	c	0.006208
	0	1	0	0	1	0.1867	1.28	h	0.002389333
	0	0	1	1	0	0.1867	5.05	n	0.009426667
	0	0	1	0	1	0.2133	1.67	v	0.003562667
	0	0	0	1	1	0.2400	6.53	r	0.015672
0.60	1	1	1	0	0	0.2400	7.81	s	0.018744
	1	1	0	1	0	0.2800	0.01	y	0.000028
	1	1	0	0	1	0.3200	1.30	g	0.00416
	1	0	1	1	0	0.3200	4.34	t	0.013888
	1	0	1	0	1	0.3600	0.02	k	0.000072
	0	1	1	1	0	0.3600	4.74	m	0.017064
	1	0	0	1	1	0.4000	1.20	q	0.0048
	0	1	1	0	1	0.4000	1.02	f	0.00408
	0	1	0	1	1	0.4400	0.21	x	0.000924
	0	0	1	1	1	0.4800	4.99	d	0.023952
0.80	1	1	1	1	0	0.5333	0.47	z	0.002506667
								Mme	0.153594667

III. THE EXPERIMENT

A Text Input Experiment was conducted and two different Chording maps were selected for the experiment: a minimum

Mme map, and a minimum cognitive effort map. A generic Chord Keyboard of ***N=5*** keys was chosen. These experiments were performed to measure the input speed and accuracy for text input using the chording text entry system.

A. Subject Requirements

The experiment was conducted in Brazil so selected subjects were Brazilian Portuguese native speakers. Two groups of eight people were recruited among the students body. All the subjects had previous experiences in use of computer keyboard (minimum average use of 20 minutes per day), and keypad of mobile devices for sending and storing short text messages (minimum average use of 1 time per week). Right-handed and left-handed volunteers participated.

B. Apparatus

The experiment was conducted in the usability laboratories of BenQ Mobile Brasil and UEA – Universidade do Estado do Amazonas. A single 5 key device connected to a PC computer, running a Java application on top of Windows XP, was specially designed for this experiment.

C. Procedure

The first group of students was assigned to the chord keyboard equipped with AEOSRI map while the second group was assigned to the chord keyboard equipped with MARCON map. Basic training, about three minutes, including free trial was provided for all participants. Printed helps were available through all trial sessions. Eight trial sessions per group were hold for the whole experiment cycle, 30 minutes expected for each individual session. A minimum interval time of ten minutes between sessions was mandatory for resting.

D. Test Cases

Test cases were expressed as 8 corpus growing in complexity from the first session to the last. Each corpus had up to 450 characters, including spaces, line breaks and special characters typical of Brazilian Portuguese Language. The corpora were representative of the frequency of symbols for the chosen language according to the Theory of Frequency Analysis [4],[6].

E. Metrics collection

The respective ***Mme*** for each map was calculated, though the physical cost could be expressed. The metrics were separated in two main groups: speed and accuracy. To compute speed, the time for typing was recorded. Accuracy was measured by counting errors. Text input time was later converted to input speed in terms of characters per minute (CPM) and words per minute (WPM) for statistical analysis.

F. Data Analysis

Accuracy and session duration composed the mean learning curves, though learning functions could be derived from it in order to obtain a quantitative measure for the cognitive cost. The results for each map were to verify whether there are significant differences between the learning curves for the given groups, though they were assigned to specific maps.

G. Satisfaction

A questionnaire was distributed for all participants, so user satisfaction using chord keyboard could be evaluated, based on System Usability Scale (SUS) [5].

IV. RESULTS

To measure physical effort applying the proposed Method, the ***Mme*** of 0.0995 and 0.1536 were found for the AEOSRI and MARCON maps, respectively. For SUS there was no statistically significant difference between the maps, after applying the ANOVA test with 0.05 of confidence level. In a range of 0 to 100, the SUS [5] for AEOSRI map scored up to 74 points and the MARCON map scored 64 points. As observed in Figure 1, the first derivative (slope) of the straight-line obtained after linear regression were -2.02 for MARCON and -1.79 for AEOSRI. This parameter is supposed to be somehow assigned to the learning speed, thus it will be defined as the cognitive Learning Gradient (**LG**).

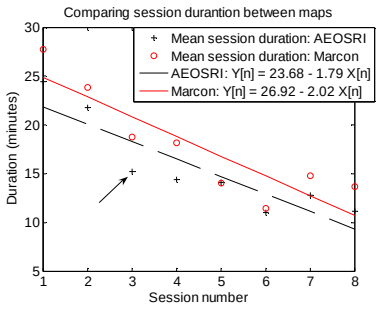


Figure 1 – The session duration between maps.. The arrows indicate statistically significant difference between maps.

The results have shown that despite the increasingly complexity through sessions for the experiment, the character accuracy and the character/word rate seemed to increase with time, what was observed for both maps.

Accordingly, it is worthwhile to investigate which map may give the best performance during user interaction with chording systems, since the AEOSRI map is supposed to give the smallest physical cost (compared to the MARCON map) while the MARCON map is supposed to give the smallest cognitive cost.

The mean character accuracy for each session is presented in Figure 2. The arrows indicate a statistically significant difference between maps, after applying the ANOVA test with 0.05 of confidence level, for the corresponding session.

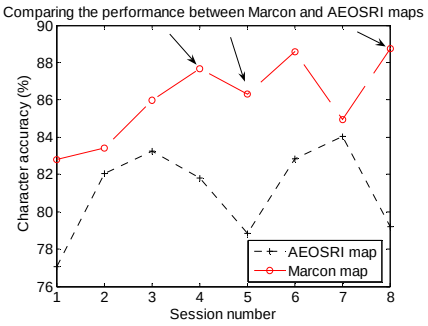


Figure 2 – The mean character accuracy for each session. The arrows indicate statistically signisicant difference between maps.

The results showed that the MARCON map outperformed the AEOSRI map at least in three sessions (including the last one) when analyzing the mean character accuracy. On the other hand, the character and word rate presented in Figures 3 and 4 have shown that the only difference between maps was observed in the first session.

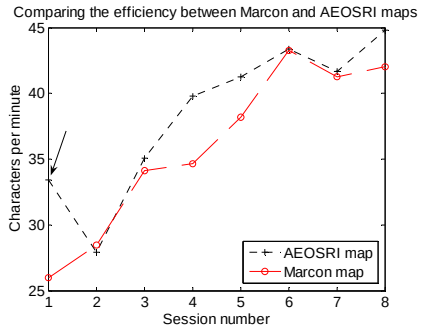


Figure 3 – The mean character rate for each session. The arrows indicate statistically significant difference between maps.

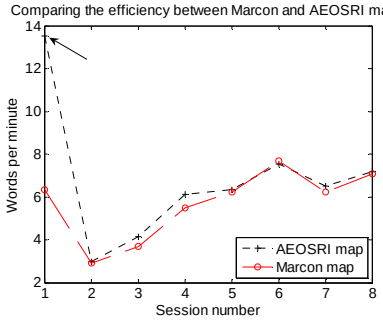


Figure 4 – The mean word rate for each session. The arrows indicate statistically significant difference between maps.

V. Discussion

Given the results, MARCON map presented superior accuracy. That could be explained by the fact that MARCON map was designed to be more intuitive and achieve better cognition. On other side, the AEOSRI map presented superior performance for CPM and WPM in the first session. This observation is possibly due to the lack of knowledge about the maps in the first session and consequently the cognitive load is apparently higher at this point. However, both maps give statistically equivalent results for the other sessions with an almost monotonic increase in efficiency, which may indicate that the cognitive cost was compensated and/or lowered after some learning period. This also suggests that the physical cost is not critical as the cognitive cost, since after the initial session both maps present the same character/word rate. Another interesting point to observe is that although there are no significant differences between mean session duration for the maps, the LG for MARCON map is higher than the rate for AEOSRI map.

VI. Conclusion

The use of traditional parameters for performance evaluation like Characters per Minute (CPM) and Words per Minute (WPM) might be useful when comparing different data entry systems, but in this particular case where different maps for the same chording data entry system were evaluated, they did not provide significant information when compared to the accuracy, the ***Mme***, or even Learning Gradient (**LG**).

As AEOSRI and MARCON maps were designed to achieve different purposes, the first - minimum physical effort, and the second - minimum cognitive effort, the satisfaction level was found equivalent from user point of view.

Future work is necessary to balance cognitive and physical effort. It would be useful to build a common map that achieves better physical performance with minimum learning effort.

Acknowledgment

Thanks to Alexandre Eisenmann for his kindness and technical contribution in the mathematical modeling aspects. Thanks to BenQ Mobile and UEA for providing infrastructure, equipments and support. Also thanks to the volunteer students who shared their time with us for the experiments.

References

[1] G. Weber, *Reading and pointing-New interaction methods for Braille displays*. In: A. Edwards. (ed.): *Extra-Ordinary Human-Computer Interaction*, Cambridge University Press, Cambridge, pp.183-200, 1995

[2] E. Matias, I. S. MacKenzie and W. Buxton, “Half-QWERTY: A one-handed keyboard facilitating skill transfer from QWERTY”, *Proceedings of the INTERCHI '93 Conference on Human Factors in Computing Systems*, pp. 88-94, 1994.

[3] I. S. MacKenzie and R.W. Soukoreff. “Text entry for mobile computing: Models and methods, theory and practice. Human-Computer Interaction”, vol.17, pp.147-198, 2002

[4] Huffman, D. A. A Method for the Construction of Minimum-Redundancy Codes, In: Proceedings of the Institute of Radio Engineers, 40(9):1098-1101, September 1952.

[5] J. Brook, *SUS A Quick and dirty usability scale*, In: Jordan, P.W et al., pp.189-94. London, UK: Taylor & Francis, 1996.

[6] J. G. Proakis, *Digital Communications*, 4th ed., McGraw Hill, 2000.

JIDS: An Intrusion Detection System using Agents

M. Canderle, F. Piccoli, G. Aguirre
Universidad Nacional de San Luis
Ejército de los Andes 950
5700 - San Luis - Argentina
e-mail: {*gaguirre, mpiccoli*}*@unsl.edu.ar*

***Abstract*—Intrusion Detection technology is designed to monitor computer activities with the purpose of finding security violations. What constitutes a security violation will depend of each organization. Thus, a *universal* Intrusion Detection System doesn't exists. Although the principles, goals and methods of security are standardized, the specific application of security is different for every organization. Taking this as a start point, we will describe JIDS: an architecture that provides a multiagent platform as a basis for the development of different Intrusion Detection techniques. JIDS allows the implementation of the specifics security policies for every organization, without having to worry about the general functionality of the IDS within the computer network.**

I. Introduction

The Intrusion Detection Systems (IDS) are very important in the security policy of every organization with a local network connected to the Internet. The common approach for the architecture of most commercial and research IDS is to centralize the data collection and analysis over a single monolithic entity.

In [1], the Intrusion Detection is defined as: *The problem of identifying individuals who are using or attempting to use a computer system without authorization, or those who are abusing of their access privileges to the system. In every cases, the integrity, confidentiality or availability of the system resources are compromised.*

An IDS is a computer system (possibly a combination of software and hardware) that attempts to detect intruders, alerting the system administrator when an intrusion is detected. It can be defined as defense system that detects hostile activities in a particular computer or computer network. The key is to detect and alert about activities that may compromise the system safety, or "hacking" attempts in progress, including reconnaissance and data collection activities, like port scans [2][3][4].

Note that this definition of IDS doesn't include prevention of intrusions, just the detection and subsequent alert to the administrator about these facts. In some cases, the IDS could try to stop the detected intrusions, taking the necessary measures to contain and stop the damage, for example ending the connection between the intruder and the compromised network[5].

Intrusion Detection Systems can be classified in different ways[6], [5], [3], [4], [7], according to their: *Architecture*, *Technique* used to perform intrusion detection, *Method* to obtain data from the system, *Behavior* in the presence of an attack and *Source* of information. We pay special attention to this last point.

Depending on the information source gathered, an IDS can be classified in Host based IDS (HIDS) or Network based IDS (NIDS). HIDS collects and analyses data from a specific computer. They are highly effective for detecting internal intrusions: attackers from inside local network or regular users making bad use of available computational resources or exceeding their privileges. HIDS are simple to implement and control.

NIDS monitors activities that occurs in the local network, analysing every data packets traveling in the network. This technique is called "packet sniffing"[3]. The data packages are examined and, sometimes, compared with empirical data to verify its nature: malicious or not. NIDS is very good in detecting intrusion attempts from outside local

Group supported by National University of San Luis and the National Agency for Science and Technology Promotion(ANPCYT)

network. But it has drawbacks, one of then is the difficulty to protect the local resources of a particular computer.

As HIDS is good in areas where NIDS is not and vice versa, the combination of both approaches results in more robust IDS. The HIDS architecture places in each computer a local IDS. Hence, the attacks from the local network inside are covered. As a complement, NIDS assures a wider view of the complete system, the local network, detecting more easily attacks coming from outside local network.

A good IDS must: *impose a minimal overhead*, *run continually*, be *fault tolerant*, *resist subversion*, *maximise configuration capabilities*, *be able to adapt to changes*, be *scalable* provide an *graceful degradation of service*, *allow dynamic re-configuration* and *provide historical data of the detected events*. Although they may be conflicting with each other, it is wise to try satisfy most of them [6].

The most common IDS architecture proposed are single monolithic entities which have some shortcomings. An alternative is distributed IDS architecture based on multiple entities. These entities work independently, cooperating to detect intruders. The multiagent systems are a suitable architecture for distributed IDS. In section II, we review the characteristics of multiagent systems.

To summarise, in order to obtain a robust IDS, it is advisable to base our IDS design on the Host, but also incorporating the advantages NIDS provides[4]. The objective of our IDS is to try to cover every defense line. In this paper, we first explain the multiagent systems and the motivation to apply this approach in JIDS. In section III we describe the JIDS, its design and some issues related with its implementation. Finally, we present the conclusion and future works.

II. Multiagent Systems & IDS

We start with two basic definitions: Autonomous Agents and Multiagent systems[8]:

An *agent* is a software entity situated in some environment. It is capable of performing autonomous actions over this environment, in order to meet its design objectives. It also has the ability of interacting with other agents of the environment to fulfill a pre-defined objective that can only be satisfied with the cooperative work of agents.

A *Multiagent system* is composed by a number of agents that interact among them, normally interchanging messages through some infrastructure of computer networks. In general, the agents in a multiagent system will represent or will act according to their own objectives. To successfully interact, every agent will have the ability to cooperate, coordinate and negotiate with the rest, in the same way humans do it.

In our context, an autonomous agent will be an independent software entity that performs a designated task. The task resolution can involve only one agent or an agents group. This definition implies the autonomy of the IDS components. An agent can perform one single specific function, or its functionality can be part of a more complex activity. In both cases, the agent can interact with other agents to exchange data that allows it to complete its tasks.

The multiagent architecture can be considered as a subclass of concurrent and distributed systems[9]. This architecture based on autonomous agents helps satisfy the desirable characteristics of an IDS[1], [10].

In the intruder detection area, the use of autonomous agents allows to design, develop and test different intrusion detection techniques, independently from the IDS. Once obtained the desired results, the new techniques can be integrated to the rest of the actual IDS.

III. JIDS

JIDS (**J**ava **I**ntrusion **D**etection **S**ystem) is a framework of research and development for any intrusion detection technique (including the combination of more than one technique, through an agents group). The main objective is to build a solid platform over which develop an IDS that: *Fulfills* the desirable characteristics, *Facilities* the adaptation of the IDS with the system reality and the security policies of the organization, and *Allows* the uses and combination of different intrusion detection techniques, to protect the integrity of an organization's network as well as every computer connected to it.

JIDS allows a separate implementation of functionalities, without having to worry about general IDS issues like inter-agent communication, system scalability, or user interface, between others. In the next section we explain the JIDS architecture and implementation details

A. JIDS Architecture

JIDS presents a Multiagent Architecture. It uses a hierarchy of autonomous agents to perform the different activities. It combines the advantages present in HIDS and NIDS.

JIDS architecture is composed by four hierarchical levels of autonomous agents. The agents interact with other agents to detect intruder and to alert to the end user, generally the system administrator, of the anomaly. Figure III-A shows different agents level and their interactions, level *n* agent with level *n+1* agent.

Every agent in JIDS is said to have a "boss" designated to which it reports any news arisen from its action. The boss can send it commands to perform maintenance, control, configuration or other similar tasks. The Level 4 agent is the only one in JIDS that interacts with the end user. Each level agent is described in next sections.

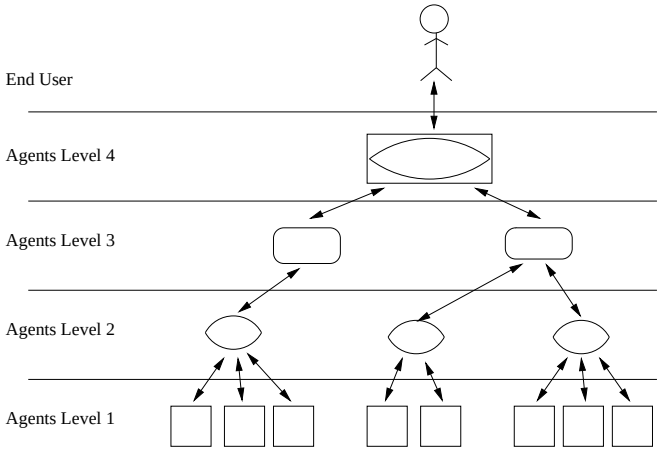


Fig. 1: JIDS Architecture - Logical View

1) Level 1 Agents: Sensors end Detectors of the IDS: Level 1 agents implements the majority of the several intrusion detection techniques at host level as well as at network level. The objective of a Level 1 agent can be one of the following:

- *Individual task*: The agent is responsible of detecting a single type of intrusion, without intervention of other agents. These agents class implements HIDS techniques: control of the local resources usage, control of the access according to user privileges, integrity of sensible information for local host, etc.
- *Data Collection task*: The agent captures system data and provide them to another Level 1 agent. The data collector is responsible for gathering information and could be called *IDS sensor*. Its work is, for example, to capture the network traffic to or from hosts in local network, and to send the information available to another agent.
- *Analysis task*: The agent analyses the data obtained by another agent. It controls the occurrence of network attack, like port scanning, remote execution of malicious commands, vulnerability of network protocols, among other possible attacks.

As a consequence, Level 1 agents can work individually or in groups. When any of them finds signs of a possible intrusion, it issues a security alert to its "boss", the Level 2 agent. A host can have several Level 1 agents assigned to different tasks of data collection or analysis.

2) Level 2 Agents: Mediators: A Level 2 agent plays the mediator role between Level 1 agents and the rest of the IDS . There is only one Level 2 agent per host. Its main job is to collect the alerts issued by their "subordinates", level 1 agent, and to send them to the corresponding Level 3 agent, previously defined as its "boss". It has to be sure that Level 3 agent receives correctly the alerts. Every Level 1 agents in a host will be subordinated to the only Level 2 agent of the same host.

The other functionality is to receive queries for information. The requirement comes from the Level 3 agent. The Level 3 agent sends a query to the Level 2 agent in the corresponding host. This agent sends the query to the respective Level 1 agent. Finally it sends back to the Level 3 agent the results given by Level 1 agent.

The layer of Level 2 agents is necessary to divide the responsibility of data collection and analysis from the alert communication to the upper levels. As a consequence, a Level 1 agent does not need to worry about the issued alerts arriving to the end user, the Level 2 agent takes care of this matter.

Level 3 agent by the other hand, does not need to worry about knowing which Level 1 agent resolves its query, it only needs to send the requirement to the corresponding Level 2 agent. Hence, there is a clear separation between specific intrusion detection and the inter-agent communication, provided by the IDS architecture.

3) Level 3 Agents: Global Monitors: Like Level 2 agents, Level 3 agents has the functionality of gathering alerts sent by its subordinates and passing them to the upper level. However, Level 3 agents can have other functionalities:

- They can detect attacks or intrusions more complex and react in consequence. These are not detectable by lower agents levels.
- They can exchange information with other agents at its same level, possibly placed in different sub-networks, to find out about other detected attacks in other parts of the network and even to decide contention measures to avoid the propagation of a given attack.

This horizontal communication with its peers, allows Level 3 agents to have a more complete view of the system.

Level 1 agents, that are network sensor, can have access to a limited part of the network traffic, hence their view of the system is restricted only to certain areas of the network. This restricted view can be caused by:

- Design issues to minimise overhead and favour the scalability of the IDS: the computers connected to the network are distributed in groups, so that every Level 1 agent protects only a part of the network and does not have to cope with the entire network traffic.
- Physical reasons, caused by the network topology: a Level 1 agent placed in a given sub-network is not be able to see the traffic traveling within another sub-network.

For these reasons, Level 3 agents are able to detect certain attacks that would pass undetected for lower level agents. To perform its work, they uses data obtained by distinct sensors placed over different host in the local network. They also interchange information with other Level 3 agents to obtain data through other sensors near them.

With both information sources and the alerts received from Level 2, a Level 3 agent has a global view of system state. For this reason, the level 3 agents are called *Global Monitors*. Some kinds of attacks can be more simple to detect at this level, like Net Scanning [6].

4) Level 4 Agents: User Interface: In this layer, there is only one agent, it interacts with the JIDS Users. Its tasks are

- To show information about the system state, global as well as discriminated state, and the alerts generated. A Level 4 agent receives the alerts sent by Level 3 agents, its subordinates, process them and finally informs to the end user.
- To allow configuration of different aspects of the IDS, like setting of the boundaries between a normal state and an alert state.
- To give the possibility to add new computers at the network. With that information, Level 3 agents could take the necessary actions to create new agents in the new hosts and to change some existing agent configuration.

This Level of agents is necessary because the nature of its tasks, the human-computer interaction is totally different from the issues related to Intrusion Detection. With this layer we accomplish a total independence between User Interface (front-end) and JIDS functionality (back-end).

B. General JIDS Operation

Having revised the agents hierarchy in JIDS architecture, we can now describe its general operation. Figure III-B shows an example of a physical distribution of the agents.

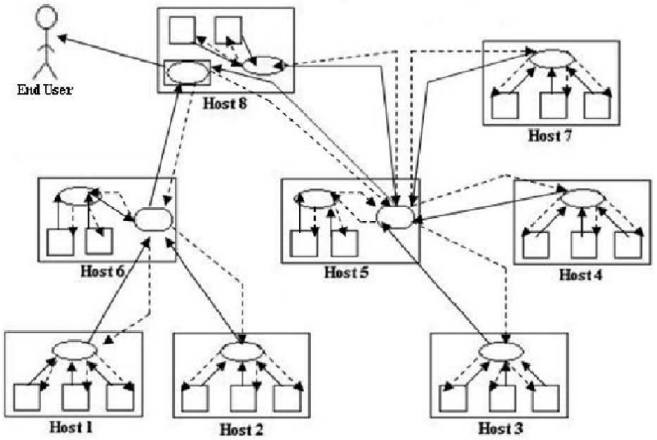


Fig. 2: JIDS Architecture - Physical View

How the agents are really distributed among the network hosts, will depend of the particular case (organization's security policies). The network topology must be taken into account, as well as the workload of every host and various practical details that arise according to each case. In the next section, we give some guidelines to implement an IDS using JIDS architecture.

1) Agents Distribution: JIDS can serve as a basis for integral protection against intruders in a computer network. The combination of HIDS and NIDS techniques implies local protection to every host in the network.

Therefore, in every computer in the network there will be Level 1 agents to protect host's resources (HIDS functionality). Their implementation are subject to the host architecture.

To cover the network level attacks, Level 1 agents will be placed in some strategical hosts. The selection of hosts will depend of each particular case too. The developers experience will be of much help in this kind of issues.

Each host has one single Level 2 agent. This agent is the "boss" of every Level 1 agent in the local host.

The next step is to place Level 3 agents in some hosts of the network. One of the goals JIDS has, is to distribute the workload among different hosts avoiding to impose too much overhead on some computers. A good idea is to group machines so that every group have computer physically close favouring communication between agents in Level 2 and 3. In one of the computer in every group, a Level 3 agent will be placed. This Level 3 agent is the "boss" of all Level 2 agents belonging to the group.

Finally, the last decision is to place the Level 4 agent, nexus between the end user and JIDS. The issue about user interface hasn't being treated with too much detail yet. It is a point to develop in the future. However, we imagine that a logical location would be a powerful computer, for example a network server.

2) Inter-agent Communication: The communication between agents has a central importance in multiagent systems, therefore in JIDS architecture is of central importance too. This topic has been widely studied[8]. We decide to employ a technology already implemented to communication among agents.

JIDS bases its development in JADE (Java Agent DEvelopment framework) technology, developed by TILAB S.p.A.[11]. This technology provides the full support to multiagent systems, through a middle-ware that is compliant with FIPA specifications[12].

The communications between agents in JIDS is entirely administrated by JADE technology, which has been widely tested and used in many multiagent systems. Thus, its correct operation is guaranteed and we can concentrate in the implementation of JIDS architecture, without having to worry about inter-agent communication.

However, it is necessary to stress that communication between agents in Level 1 layer in the same host can use a shared buffer or repository, without the need of JADE technology to exchange data.

C. JIDS Implementation

The objective of this work is to develop a stable architecture to facilitate the implementation of different IDS. The development is still in research and laboratory testing phase.

We have developed a first prototype that serves as a *proof of concept* for the JIDS architecture. It is entirely implemented in JAVA Programming Language[13] and works under Windows platforms, for the moment, but its design is portable.

The idea of this first prototype was to prove that JIDS really could work as a platform to develop an IDS with characteristics of host and network IDS. As a first step we implemented the hierarchy described in III-A. For every layer of agents, we defined the necessary functionality for JIDS to work: boss-subordinates structure, alert issuing, handling of issued and received alerts, communication between Level 1 agents dedicated to singular detection task and other internal details.

Concerning agent's functionalities:

1) To include host level protection, we implement an agent whit the responsibility to detect the presence of Rootkits[14], [15], [16], [17] inside the host where it resides.
2) To include network level protection, we developed a group of collaborative agents to

detect port scanning[7], [18], [19] over the computers group assigned to protect.

3) To have a simple user interface as system output, we implemented a rudimentary Level 4 agent, whose only purpose was to communicate the alerts received.

For detecting rootkits, the method chosen for the development of a Level 1 agent was Integrity Based Detection[1].

For Port Scanning Detection, the classic way to detect it is through pattern recognition by observing the network traffic. For this, it is necessary to control every packet in and out the system. All connections, or connection attempts, are analyzed.

The tests were conducted on the laboratory's network with 20 computers. JIDS worked over many environment: different network loads and computers functions.

IV. Conclusions and Future Works

We have introduced a Multiagent Architecture for an Intrusion Detection System, *named JIDS*. It works with a hierarchy of autonomous agents to perform different activities in a distributed fashion, avoiding the various problems present in centralized architectures.

We combined HIDS and NIDS techniques without trouble using autonomous agents. They were developed totally isolated from JIDS and then integrated to the architecture. Their task was to issue alerts correctly and to assure that these alerts arrived correctly to the end user. Therefore, we could also prove that JIDS is a good framework to facilitate the research, development and testing of different techniques for intrusion detection.

The JIDS development is still in laboratory testing phase. First, it is necessary to test the operation and behaviour of JIDS to assure its viability as a basic architecture to develop functional IDS. Our first prototype had this purpose and showed that JIDS is perfectly viable as a platform for developing IDS.

However, there is still a lot of work to do. It's necessary to optimize the general operation of JIDS, in order to reduce its impact over the normal load in the network traffic. JIDS involves communications between agents, necessary for the functionalities inherent in different intrusion detection techniques or inherent to JIDS architecture. In order to optimize JIDS, It is necessary for example:

- To measure the extra load that could be imposed to the network traffic by horizontal communication among Level 3 agents.
- To improve the scalability capabilities of JIDS.
- To develop a complete user interface for JIDS. The user interface should provide ways to configure different aspects of the IDS, as well as methods to indicate changes in the network topology, or security policies, to improve the adaptability of the IDS to changes in the system, among others.
- To analyse the real possibilities of multiplatform support, JIDS is entirely developed in JAVA, so it is possible, theoretically, to accomplish a platform independence for the implementation of agents hierarchy.
- Giving to the agents capabilities for Artificial Intelligence (AI), would help agents to adapt themselves to attacks against the IDS, and to fight back or recover from the damage caused by an attacker.

Finally, once JIDS architecture becomes a solid platform for an IDS, the final step is to make a deeper study of the different Intrusion Detection techniques, in order to develop agents to implement them. Furthermore, the addition of learning capabilities to all level of agents would give the IDS better possibilities of adaptability to changes.

[1]Tripwire is an example of an anti-rootkit tool that uses this technique[20]

References

[1] E. H. Spafford, D. Zamboni: *Intrusion Detection Systems Using Autonomous Agents.* Computer Networks 34. Pp 547 - 570. 2000.

[2] P. Kazienko, P. Dorosz: *Intusion Detection Systems (IDS) Part 1 and Part 2: Classification, Methods, Techniques.* WindowSecutiry.com

[3] J. Novak, S. Northcult. *Network Intrusion Detection*. Sams Publishing. 2002

[4] R.Gurley Bace. *Intrusion Detection*. Sams Publishing. 1999.

[5] C.F. Endorf, J. Mellander, E. Schultz. *Intrusion Detection & Prevention*. McGraw-Hill Professional. 2003.

[6] T. Crothers: *Implementing Intrusion Detection Systems.* Wiley 2003

[7] C.R. McNab. *Network Security Assessment*. O'Reilly. 2004
[8] M. Wooldridge: *An Introduction to Multi-Agent Systems.* Wiley 2002
[9] A. S. Tanenbaum: *Computer Networks*. 4th Edition, Prentice Hall 2003
[10] J. S. Balasubramaniyan, D. Zamboni: *An Architecture for Intrusion Detection using Autonomous Agents.* Coast Technical Report 98/05
[11] F. Bellifemine, G. Caire, T. Trucco and G. Rimassa. *JADE, A White Paper.* http://jade.tilab.com
[12] Foundation for Intelligent Physical Agents. *FIPA RDF Content Language Specification.* http://www.fipa.org
[13] H. Schildt. *Java: The Complete Reference, J2SE TM.* 5 Edition. McGraw-Hill Professional. 2005
[14] M. Burnett and L. J. Locher and C. Doyle and C. Amaris and R. Morimoto. *Maximum Windows 2000 Security*. Sams Publishing. 2001
[15] J. Scambray, S. McClure. *Hacking Exposed Windows Server 2003*. McGraw-Hill Professional. 2003
[16] J. Rutkowska. *System Virginity Verifier: Defining the Roadmap for Malware Detection on Windows Systems.*
[17] N. Petroni, T. Fraser, J. Molina, W. Arbaugh: *Copilot - a Coprocessor-based Kernel Runtime Integrity Monitor.* Proc. 13th Usenix Security Symposium. Pp 179-194. August, 2004.
[18] Insecure. *NMAP: a free open source utility for network exploration or security auditing*. http://insecure.org/nmap/
[19] Thomas H. Ptacek, Timothy N. Newsham: *Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection.* Secure Network Inc. 1998
[20] E.Cole. *Hackers Beware*. Sams Publishing. 2001

Risk Management Applied to Internet Banking Environment of Financial Institutions in Brazil

Adriano de Melo Pouchain, Ricardo Staciarini Puttini
Electrical Engineering Department, University of Brasilia, Brasilia DF Brazil

***Abstract*-We deal with a risk management model applied to internet banking. By performing a survey about attacks performed against financial institutions in Brazil, we identified the main risks of internet banking. The conclusions identified critical environments, the behavior of the hackers, the most significant vulnerabilities and the best way to manage the correspondent risks.**

I. INTRODUCTION

Risk management reduces the exposure of a project or a process by means of identifying the main threats and vulnerabilities, qualifying for adopting measures directed to control and to minimize undesired events.

In this paper we explore a model for risk management, including the steps of identification, analysis, planning, monitoring and communication. The whole process is understood as a continuous cycle, where each step provides information to the next one and receives new information from the previous one.

Our main contribution consists in determining the main threats and the types of the most frequent internet attacks against the financial institutions in Brazil. We have carried out researches to identify the more critical environments, as well as the more significant threats and vulnerabilities.

By analyzing these data, based on series of 04 years, and comparing the number and types of attacks with new security solutions proposed in the period, we could find certain patterns in hacker's behaviors.

We also identified the main security measures that are being adopted by the financial institutions in Brazil in order to reduce the success probability of attackers.

A risk management model can greatly improve the efficiency of security management in these institutions, as they can map the events of risk and evaluate them. It makes possible to rate the risks according to their level of criticality. The best strategy to apply effective management is to concentrate in managing the most critical risks, i.e., those that may significantly threaten the accomplishment of established objectives.

Statistics of the last two years have shown a growth of around 50% in the using of internet banking in Brazil [10]. Other countries of Europe and North America also have registered considerable increase in the number of electronic bank transactions [5]. However, the lack of security perceived by the users has continuously been pointed out as a strong inhibiting factor for reaching even grater rates [5]. Therefore, good management of risks inherent in Internet banking is a key point towards for more competitive, secure and lucrative financial institutions.

II. RISK MANAGEMENT IN THE INTERNET BANKING ENVIRONMENT

The risk identification stage starts with knowing the current process, establishing its objectives, and interacting with the internal and external environments where the process is inserted. E-banking aims at providing clients with banking services, products and information through the Internet.

To accomplish this objective it is fundamental that banks may be able to identify their clients in a positive way, so that financial institutions can offer the correct site to allow and assist users in realizing their transactions, besides ensuring protection to the clients' information.

A. Risks Identification

The very first step in risk management is to know the whole environment where internet banking transactions take place.

Actually, three complementary environments can be identified: the user environment, that embodies the transaction flow from the client's computer to the tcp/ip package prepared to be sent to the destiny; the internet environment itself, that comprises since the reception of the tcp/ip package from the source until the delivery to the destiny throughout the worldwide network; and, finally, the financial institution environment, where the package is received and the transactions are processed. We have carried out a research about the Brazilian bank's approaches regarding attacks in the users' environment (first environment) and in the financial institutions (third environment).

Once the environments have been very well known and the objectives have been established, risk identification stage proceeds with the mapping of possible threats that may jeopardize business success.

B. Operational Risks

Operational risk can be considered the risk of losses resulting from: internal or external frauds; inadequacy or failures in internal processes, individuals and systems; and natural events (catastrophes, etc.).

In this area, the main threats for banks concern information security. In other words, they encompass aspects of confidentiality, availability and integrity of information. Among others, the following concerns represent serious threats: destruction/damage of information or other resources; unauthorized modification; theft, removal or loss of information; server interruption; and disclosure of information to non-authorized individuals.

One common type of attack involves sweeping information about a computer network or about a user's computer. It is known as 'passive threat'. The hackers search information about data of users or about a network configuration - made available by the banks to provide information or to realize transactions - that may be used in a real attack to a system or a set of systems. They do not provoke any modification in the system's information or in its operation way.

One example of attack against banks' servers is the network analysis. The hacker makes a systematical and methodical approach, known as 'foot printing', to become familiar with the security structure of the network. During this exploration stage, information about network addresses, gateway and firewall localizations are obtained, and it is also carried out an analysis of the traffic in the most common port numbers in order to detect information or services being executed in the system. The analysis of the information collected allows identifying vulnerabilities to be explored afterwards.

On the client's side, such kind of attack may result in leakage of personal bank information that will make it possible to the hacker to execute fraudulent transactions. It is relevant to notice that the hackers generally are not interested in changing or destroying any information. For them it is extremely important to keep themselves "invisible" during the attack, since this situation will increase their chances of success as well as the use of the data obtained.

With possession of the bank data, obtained by those techniques, the hacker comes to a position of performing transactions "on behalf of" the real client.

We monitored, from May 1, 2006 to May 31, 2006 the trojans sent to the bank clients in Brazil. The results are showed in the Table I. The numbers represent the daily quantity of malicious codes sent to clients.

The figures do not necessarily quantify new trojans. Sometimes, the same malicious code is sent masked into different phishing messages, like:

- "you are being betrayed, click here to see the photos";
- "your elector's card has been cancelled, click here for information".

The trojans behind those "clicks" are usually aimed at collecting bank information and/or to lead the user to execute actions that will disclose his or her personal data by sending them to the hacker. Unfortunately the Internet passes an illusion of security, maybe justified by the individual interaction between the man and the machine. Not knowing the technology 'behind" this environment, the naive user imagines he or she is the only one who can access the information passed through the computer. The user relies on the message received and eventually yields to the presented request. Attacks by trojans are the greatest threats to the customer environment.

TABLE I
QUANTITY OF TROJANS SENT TO BANK CLIENTS

1/5/2006	178	17/5/2006	115
2/5/2006	419	18/5/2006	133
3/5/2006	163	19/5/2006	86
4/5/2006	81	20/5/2006	97
5/5/2006	85	21/5/2006	20
6/5/2006	23	22/5/2006	61
7/5/2006	8	23/5/2006	149
8/5/2006	96	24/5/2006	65
9/5/2006	24	25/5/2006	200
10/5/2006	144	26/5/2006	129
11/5/2006	137	27/5/2006	95
12/5/2006	157	28/5/2006	51
13/5/2006	2	29/5/2006	195
14/5/2006	38	30/5/2006	56
15/5/2006	190	31/5/2006	98
16/5/2006	139	Total	3434

Table II shows the result of the monitoring carried out from May 5, 2006 to June 5, 2006, when we plotted the attacks against banks settled in Brazil. In all the cases the patches for the vulnerabilities explored were available in the sites of the software firms. The weakness is marked by the lack of installation of the patches provided, what should be done by system managers.

If, on one hand, the users are usually not aware of the threats in the internet environment, on the other hand there are system managers which do not have enough knowledge to ensure that a web site is safely ruled.

In addition to the type of attack mentioned above, we highlight three others: "transaction poisoning", internal attacks and attacks by "social engineering".

The poisoning attack is aimed at capturing and modifying the original message without the user's perception and bypassing the financial institution security system. Such kind of attack is technically known as "man-in-the-middle" or, more recently, "man-in-the-browser".

The problem in such type of attack is that the trojan acts after the regular authentication of the user by the bank, misleading the security walls installed to ensure user's authenticity. For the bank, the transaction is authentic.

In the same way internal attacks represents exactly high risks, because the hacker operates inside the institution. Either temporary staff or regular employees have already overcome many of the barriers placed against external hackers, thus

increasing the risk of non-authorized access to critical environments or systems.

TABLE II
QUANTITY OF TROJANS SENT TO BANK CLIENTS

Identification of the attack	Ocurrences	%
Synflood	1.758.020	25,90
HTTP_Cross_Site_Scripting	714.457	10,52
IP_Unknown_Protocol	534.306	7,87
Stream_DoS	469.548	6,92
BGP_Route_Unreachable	449.138	6.62
DNS_Query_All	274.311	4,04
TCP_Port_Scan	223.576	3,29
HTTP_IIS_Unicode_Wide_En-coding	150.952	2,22
RealSecure_Kill	140.230	2,07
Email_Outlook_URL_Spoof	131.296	1,93
Other	1.942.847	28,62
Total	6.788.681	100,00

In this type of attack the "confidence" factor is largely explored by users with malicious intention. In considering reliable their colleagues, the managers many times focus their attention to attacks coming from outside of the institution. Therefore, the internal environment gets weaker in respect to security measures, that should have be implemented to protect systems, processes and stored information.

The social engineering attack deserves to be highlighted. It certainly is the greatest challenge to financial institutions, since it deals with internet banking user's personal behaviors. The non-familiarity of individuals regarding the use of worldwide computers network is a highly explored vulnerability.

In Brazil, financial institutions stand out for their massive investment in technology. Security - particularly information protection - is amongst the major concerns of managers.

All over the years many security measures have been implemented, which include solutions in areas such as user authentication, systems and data protection, and security to users environment. Users' environment has precisely been the largest apprehension of bank executives lately.

The security measures implemented by the financial institutions in e-banking environments have been providing a comfortable protection level against invasions. As a result, hackers have directed their attacks to the "weakest link of the chain". In the research we have observed that hacker action is more concentrated in "social engineering" than in technological methods to gain access to the systems and to the user's information. Using an efficient persuasion process, hackers mislead the users into providing or disclosing their banking data.

We could also identify a large number of attacks with trojans hidden in links supposedly related to contest applications, credit card promotions, debt notices, caricatures, denounces of conjugal infidelity, etc. These attacks are directly related to the data showed in table I.

Unfortunately hackers have a great range of appealing subjects to explore. The more banks implement stronger security solutions, the more criminals create new forms of social engineering. Let's analyze two cases of great national banks in Brazil.

Banco do Brasil has implemented a security solution that consists of registering its client's computers. When adhering to the solution, banking transactions that involve funds only can be made from the computers previously registered by the client.

One of the social engineering attack created to bypass this solution consists of sending an email to the customer, (supposedly sent by the bank), requesting the registry of a new computer - the correspondent code is attached to the message - under the promise of some reward. By doing so, the customer "legitimates" the hacker's machine into the system, and give to the fraudster full access to his/her banking data.

In its turn, Banco Bradesco has implemented a solution which generates dynamic passwords, which are supposed to be used by clients to confirm critical transactions. Bradesco's clients receive a card that contains 70 dynamic passwords of three digits each. Besides the traditional password, the dynamic one is requested to be typed in order to authorize the transaction. For this particular case, the social engineering attack consists of requesting to the customer - also by email supposedly sent by the bank - to inform all the 70 passwords received from the bank.

C. Legal Risk

It is the risk related to non-compliance with law and other regulations. The potential impact to the organization consists of penalties that may be imposed by the regulators. In some countries, regulators may even revoke the license given to the company to operate in their country.

D. Image Risk

It is the risk connected with a public negative opinion about the company due to involvement of the corporation in illegal operations, in damage to the environment, in serious system failure, in scandals, bad news about its financial health, etc.

The image risk is usually linked to the materialization of operational or legal risk, reflecting directly in the business results, either by loss of clients or a decrease in its operations. Moreover, besides the financial losses provoked by the impact of the other risks, the organization ends up with its reputation "scratched".

By offering internet banking services to their clients, financial institutions increase their image risk exposure. Every time a bank makes available a different channel for either communicating or dealing with its clients, specific measures to manage the inherent risks are required.

E. Analysis and Planning to Manage Risks

Once identified and collected all the information from the activity monitoring (company and client environments) as well as the risks that the institutions is exposed, it is time to proceed a risk analysis and to plan the actions to deal with each threat adequately.

By analyzing the researched data we could verify that hackers have been posing serious threats do the user security, through passive monitoring. The number and diversity of trojans sent to bank clients to obtain their critical personal data has been increasing continually.

It is important to emphasize that, although the monitoring attack is passive, it represents high risk since it allows hackers to collect users banking data and, later on, to perform fraudulent transactions by authentication spoofing.

According to the figures in Table I we can observe an average of 111 daily occurrences of trojans being sent to bank clients. Despite of the difference between the instances, all of them share the same objective: obtaining user data. Regarding this particular type of attack, we could say, in short, that the great number of different "traps" is one of the main factors that increases hackers' likelihood of success.

The situation raises even more worry when we add to that the little awareness of internet users. Many of the users are completely unaware of the inherent risks in a virtual environment. Besides that, delays in updating systems and networks, as well as the poor qualification of some system/security managers also contributes to increasing risks.

F. Hackers Behavior

Along the research about attacks against financial institutions, we have performed a research about hackers' behavior regarding the preferred modus operandi in their attacks.

We have concluded that hackers always look for the easier way to proceed their attack. For instance, we did not find real evidences of direct attacks against cryptosystems. That kind of attack requires more time, good level of knowledge and a reasonable money investment. Hackers always look for the simplest way to achieve their objectives. We have noticed that immediately after a bank implements a security solution, the attack to that particular bank decreases in terms of number, revealing an "escape" of the hackers to other banks whose security level is possibly weaker.

A research published by the Gartner Group [3] reproduces the abovementioned. In figure 1 we can observe the attack evolution during the last four years. All the types of attack showed in the chart act in the client's environment, i.e., in the weakest link of the security chain.

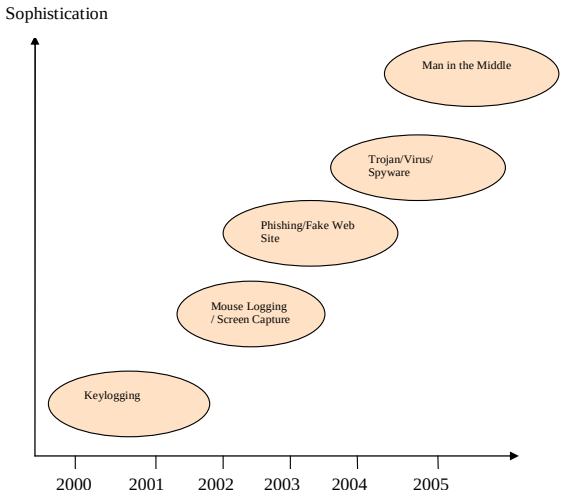


Fig. 1. Electronic Fraud Trends. Gartner Group [3].

Still based on Gartner Group we can verify, in figure 2, that whenever the bank adopts additional security measures, the incidence of occurrences decreases immediately, and resumes increasing when the hackers either change their strategy or find out some other vulnerability.

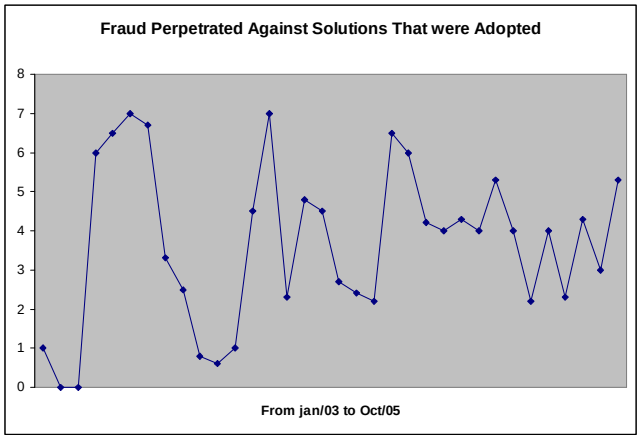


Fig. 2. Variation in number of incidents along the last 03 years versus HSBC internal security measures.

This variation also can be observed in figure 3. At this time the graphic was produced from our research about the attacks against the financial institutions in Brazil, during the last four years.

Security solutions built by the banks in the past as virtual keyboard, hidden click, transaction limits, active monitoring, clients and computers pre-registering, amongst others, had reflected in immediate reduction of attacks.

This fact is demonstrated by the visible variations shown in the chart of figures 2 and 3. Each fall off in losses is usually associated with security measures implemented by the financial institution. It does not mean that attacks had ceased. They probably had only been directed to other banks.

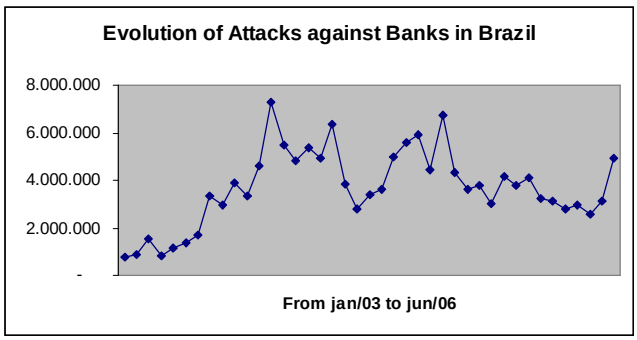


Fig. 3. Variation in number of incidents along the last 03 years versus banks in Brazil internal security measures.

Besides that, not all the customers adopt the security measures offered by the institutions, making it possible that attackers keep on having success.

If we join the figures 1, 2 and 3, as showed in the table III, we can identify the threats and the measures adopted by banks. We can also perceive the delay from detecting the threat and adopt some security measure. This delay could be explained by some problem in the risk management process. Some steps can not be executed appropriately.

TABLE III
QUANTITY OF TROJANS SENT TO BANK CLIENTS

Threat	detection	Security measure	Implementation
Keylogger	2001-2002	Virtual keyboard	2003
Screen Capture	2002-2003	Browser defense	2004
Fake web site	2003-2004	Monitoring	2004
Trojan	2004-2005	Monitoring	2005
Man in the middle	2005-2006	Cryptography	2006

G. Security Measures

In order to guarantee authentication of its customers, banks have been adopting many security solutions in addition to the traditional password. Virtual keyboards (keyboards that appear on the computer screen and are stroked by clicking on the mouse), browser defense, dynamic passwords, registering of computers, biometry and digital certification are examples of actions taken by the banks.

Among all the solutions, it has been currently standing out a technique known as "continuous authentication". It is more efficient comparing to the "only-in-the-beginning authentication", because it difficult attacks of the types "man-in-the-middle" and "man-in-the-browser". These kind of attacks basically take advantage of the single authentication (only in the beginning), since the trojan "poisons" the transactions completed after the authentication.

Following the line of reasoning made explicit in the analysis of aggressors' behavior, it is reasonable to foresee that banks must invest in strong authentication methods as an additional security layer.

In this direction, some Brazilian banks like Banco do Brasil already offer digital certification to their customers, in addition to other measures already implemented as: password, virtual keyboard and browser defense.

Digital certification provides an advanced level of security, once the solution uses a combination between a private key of the customer's certificate and a public key. In the authentication process, even the bank cannot access the customer's private cryptography key, which is stored only in the chip of a smart card or token.

There is an additional advantage in using certificates stored onto smart cards. A unique card is sufficient to authenticate an individual in several different institutions (banks and others).

We present in table IV some security measures, describing the main advantages and disadvantages.

TABLE IV
ADVANTAGES AND DISADVANTAGES OF SOME SECURITY SOLUTIONS

Measures	Advantages	Disadvantages
Password	Easy use and comprehension	Phishing scan and keylogger attacks.
Cognitive Password	Easy use and comprehension	Social engineering, phishing scan and keylogger attacks.
Virtual Keyboard	Protection against keylogger	Mouselogger and screenlogger attacks
Dynamic Password	Low cost	Social engineering attack, storing and transportation problems;
Grid Cards	Low cost and easy use and comprehension	Social engineering attack, storing and transportation problems;
Tokens	High security	High costs, mobility;
Identification of devices	High security	Social engineering attack, mobility;
Transaction anomaly detection	No applications needed in the client computer	False/Positive ratio. Do not assure the user authentication.
Biometry	High security	High costs, mobility, intrusive technique.

Combining security measures is a best practice of securing online service, such internet banking. This is what we want to say about "strong authentication method" as an additional security layer.

III. CONCLUSIONS

Managing risks is managing uncertainties. The model initially presented in this paper proposed a continuous cycle of risk management. Summary, the central point of the model is the importance of performing a careful risk identification, in order to rate and rank all the threats according to their relevance, thus optimizing resource allocation.

By following the proposed steps it is possible to reduce the time between the identification of a threat and the adoption of a security measure.

Some financial institutions, in Brazil, perform a joint action of monitoring, analysis and reporting of hostile events detected.

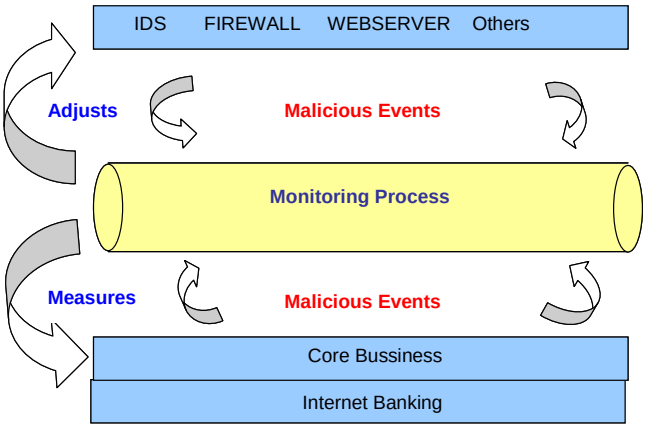


Fig. 6. Monitoring Process.

The figure 6 shows a monitoring process. As soon as a malicious event is detected, it's possible to implement adjusts in security devices or in the business rules, reducing the number of successful attacks.

By adopting a risk management model, the institutions can reduce the number of frauds and financial losses. Something much appreciated by stakeholders.

Moreover, it reduces the risk of image and raises the satisfaction of the clients with the security provided by the institution.

Looking at the figure 7, we can see that the gap between the detection of an event and the adoption of some security measure has decreased during the last four years. As consequence the financial losses are reduced too. This is the result of the adoption of a risk management process, by the banks in Brasil. The graphic is the result of our searches and analyses over information collected during four years. By reducing the "ΔT" it's possible to minimize the financial losses "Δ$".

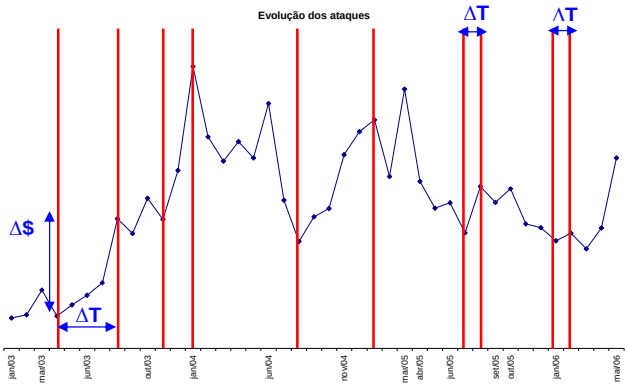


Fig. 7. Fraud Behavior – Decreasing ΔT and Δ$.

The results of our researches pointed out the personal environment of clients as the most critical environment and, naturally, the favorite of hackers to attack internet banking services. The most common type of attack is social engineering that is highly effective. This threat represents the greatest challenge to the banks since it deals with individuals' behavior. Bank clients have been proving to be ingenuous, thus facilitating the disclosure of their personal and banking data.

Regarding hackers' behavior it has been evidenced that they always look for the easier way to proceed to their attacks. Every time a bank implements a security solution, the number of attacks against that specific bank decreases and attackers tend to direct their action to other institutions which have not implemented similar measure yet.

Concerning the banks' environment, we could conclude that the attacks were well-known, but despite of patches to treat the vulnerabilities were available in the software firm's websites, in most of the cases the systems had not been updated by the system/security manager.

Internet banking has provided facility and comfort to bank clients and definitely have brought new opportunities of business for the banks, incrementing their profitability. Frauds and information theft have always existed, even before the arising of the virtual environment

In the virtual world, aggressors are able to reach much more people than they would by approaching physically the clients inside a bank agency. Trojans are launched "in the space" in large loads and in a wide variety of ways, either to infect/damage personal computers or simple to obtain banking data hence increasing the chances for attack success.

What is difficult to the banks is to make clear to all their internet banking clients that although banks are working hard to improve security solutions, it is essential that users do their job, to help protecting their money and to avoid distress. To be acquainted and aware of the internet environment risks and also to adopt basic security measures may certainly contribute to reduce the number of frauds.

REFERENCES

[1] Dorofee, Andrey J., Walker, Julib A., Alberts, J.Christopher, Higuera, Ronald P., Murphy, Rochard L., Williams, Ray C. - Continuous Risk Management Guidebook, SEI – Software Engineering Institute - Carnegie Mellon University.

[2] Gartner Group – Conduct a SWOT Analysis to Shape Your Sourcing Practice, DREYFUS, Publication date: 1 September 2004.

[3] Gartner Group - HSBC Bank Brasil Turns to Back-End Fraud Detection to Curb Cybercrime, Publication date: 2 June 2006.

[4] Gartner Group – Regulators Tell U.S. Banks to Adopt Stronger Risk-Based Authentication, Publication date: 27 October 2005.

[5] GLOBAL MARKET INSITE, Online Banking Gaining Worldwide Momentum, Avalilable: http://www.gmi-mr.com/gmipoll/release.php?p=20051019. Access: 01.03.2007.

[6] GRIFFITHS DAVID, Risk Based Internal Auditing. Available: http://www.internalaudit.biz/files/introduction/Internalauditv2_0_3.pdf.

[7] HOLTON, GLYN. Defining Risks – Financial Analysts Journal Volume 60 Number 6, page 22 (2004, November).

[8] ISO - International Organization For Standardization / IEC Guide 73 – Risk Management Standards page 03, Ferma – Federation of European Risk Management Associations, 2003.

[9] Theiia - The Institute of Internal Auditors. Global Techonology Audit Guide – Management of IT Auditing, Michel Juergens, Publication date: March 2006.

[10] Transações Bancárias e Automação. Available: http://www.febraban.org.br/Arquivo/Servicos/Dadosdosetor/2006/item05.asp (URL). Access: 01.03.2007.
[11] Weihrich, H. (1982) The TOWS matrix: a tool for situational analysis, Journal of Long Range Planning, Vol. 15 Issue 2, pp.12-14

A New Trust Model for Reasoning about PKI

Hanan El Bakkali*

*Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes
Université Mohammed V - Souissi , Rabat, Morocco
Tel: +212 37 712167, Fax: +212 37 7730
E-mail: elbakkali@ensias.ma

***Abstract*— Participants authentication and crucial information privacy are often required in electronic transactions. Public-Key Infrastructures (PKIs) are, generally, essential for providing them these security services in open networks like Internet. This paper proposes a new trust model for reasoning about PKI and compares it to some existing ones. This model reasons in a global way about certificates and entities beliefs with regard to public key authenticity and certification authorities (CAs) trustworthiness. It takes into account the number of intermediates that have participated in an entity belief, the trust level in a statement, the application context and other parameters. Indeed, there has been too much focus on PKI technologies and relatively little on trust in these technologies. Particularly, trustworthiness of PKI's CAs has to be proved and not imposed. It is also necessary to permit a trust level evaluation of such beliefs which depends on several parameters like policies constraints and application context.**

I. INTRODUCTION

Electronic signature and thereby the public-key cryptography are the essential elements in the process of authenticating participants in an electronic transaction via open networks like Internet. Some of these transactions, as those of e- payment, require strong authentication of the participants and must use, for that, reliable and easy to use electronic signature mechanisms. However, without a global, reliable and trustworthy PKI for the keys management and publication, these mechanisms will lack to ensure the security requirements of such transactions.

In this paper, we are particularly interested to design a PKI trust model that will permit reasoning about trust in PKIs. Our model does not enforce any particular initial trust relationships between end users also called end entities (EEs) offering the possibilitie of building a large PKI required by the applications of e-commerce. However, some initial 'trust relationship' between end entities and nearest CAs is required. The goal is to allow the building of proofs concerning entities public key authenticity and trustworthiness on the basis of certificates and statements about entities initial beliefs with regard to CAs public key authenticity and CAs trustworthiness. Such proofs have to consider the number of intermediates that have participated in an entity belief, the trust level in a statement and other important parameters.

We have already presented in [3] a predicate calculus logic for reasoning about PKI trust model.

This paper is a continuity of this work with the main contribution that attributes certificates and application context are also considered and trust aggregation is tacked into account, in addition of trust propagation.

The major difference of this paper with related work is discussed in Section 4. In Sections 2 and 3 we describe briefly some Trust and PKI background and in Section 5, we present the proposed PKI trust model. We conclude in Section 6.

II. TRUST BACKGROUND

A. Trust definitions

'Trust' and 'Trustworthiness' can seem to be well understood, but, in reality there is no agreeing on what they mean, how to measure them and how to reason about them. Even in the context of security, we find number of definitions. We present here some ones of them:

Definition 1[12]: Trust is a directional relationship between two parties that can be called 'Truster" and "Trustee". The truster must be a "thinking entity" whereas the trustee can be anything (persons, systems, abstract notions…).

Definition 2[7]: Trust is the expectation that a service will be provided or a commitment will be fulfilled.

Definition 3[18]: Trust is the firm belief in the competence of an entity to act dependably, reliably and securely within a specific context.

In our paper, we adopt the following combined definition: Trust is a relation that permits to a Truster to believe that a Trustee will do a certain 'positive' effort to act as the truster expects within a specific context and under some constraints. The trust level can then be expressed as being an estimation of the degree of this effort. Generally, the trust level can be improved, if the truster disposes of more favourable and trustworthy opinions about the trustee, via trust aggregation.

B. Trust propagation

Trust propagation is essentially due to the transitive nature of trust. However, this transitivity must be limited to the same context and submitted to some constraints. Indeed, as we have seen, trust itself is never defined in absolute terms. Rather, it is limited only to some context (or scope) and restricted by certain constraints. In PKI context, trust in a CA can be limited to particular domain or to one kind of application and so on.

III. PKI BACKGROUND

A simple definition of a PKI may be: "A set of entities that communicate with protocols and provide services for managing the public-keys and their certificates." Generally, the role of a PKI is to provide the necessary mechanisms to establish trust relationships between end entities and offer

them security services such as integrity, authentication and non-repudiation.

A. Certificate

A Certificate is used to prove the authenticity of the binding between a public-key and some information concerning its owner. Thus, a trustworthy CA must sign it. At a minimum, a certificate must contain the issuer CA identifier, the subject (key holder) identifier and/or other attributes, its public-key and the expiration date. A certificate user (often called, relying party) must trust the issuer CA and must know with certainty its public-key in order to verify the certificate signature. Otherwise, he can use another certificate of this key issued by a second CA and so on. In this case, we talk about certification path validation.

The X.509 certificate format is the most widely adopted format in the PKIs. The 3rd version [8] is more flexible than the precedents by the use of the extensions field that can convey information about the certificate policy and the key usage.

There are, essentially, two types of certificate: identity certificate and attribute certificate. According to X.509 specifications [4], an attribute is defined as "information that describes the qualifications and authorities granted to the target entity". Attributes can include identity, group membership, role, clearance level, domain-specific properties, etc.

B. Certificate Policy

According to X.509, a certificate policy is “a named set of rules that indicates the applicability of a certificate to a particular community and/or class of applications with common security requirements”[22]. Its role is to help a relying party in deciding whether a certificate and the binding therein are sufficiently trustworthy for a particular application [1]. Thus, it is important to enable each entity to be aware of the certificate policies governing any certificate it may encounter.

C. Entities relationships

In a PKI, the main entities are the CAs; the others are generally end entities(EEs) that use the PKI services. The trust relations among PKI entities are based on preliminary relations between them that are of two sorts: existing trust relations and necessary relations to a PKI “adhesion”. The first relations include, for example, those between a bank and its account holders, between a government and its citizens and so on. The second relations must be established between the EEs that want to join a PKI and at least one of its CAs. This is the case of a person who wants to obtain his first certificate; he can be gated in certain PKIs to present himself to the issuer CA with his identifying papers. Generally, the degree and the kind of the previous relations depend on the certificate policies of PKI’s CAs. So, it appears that preliminary trust relations are the starting point in building new ones based on public key technology.

D. PKI trust model architecture

PKI trust model is the organization of CAs that provides a security service user with confidence in using that service [2]. A PKI includes several CAs linked by certification paths considered as trust paths. There exist different PKI trust model architectures: the CAs may be arranged in distributed manner, hierarchically or in hybrid architecture. PKI trust models are generally represented as directed graphs in which the nodes correspond to entities and the edges represent the certificates and implicitly trust relationships.

IV. RELATED WORK

There has been considerable work on reasoning about trust models [7,13,18] and in a less measure about PKI trust models. Lot of them are logic-based as Maurer’s approach [17] that is one of the prior works in this area. It proposes a logic-based approach to modeling a PKI from a user Alice’s point of view. This approach has been readopted in [14] and was the basis of the work of [16] which focuses on other issues such as time, revocation, delegation, and heterogeneous certificate formats.

Ref. [17] uses confidence values that concern Alice’s statements about public-keys authenticity and entities trustworthiness. They are measured on a continuous scale between 0 and 1 and are interpreted as probabilities. The use of similar continuous measures in the [0,1] interval was adopted in others works like in [10, 11] where is question of degree of belief, degree of disbelief and degree of uncertainty and in [9] that uses a formal concept of weighted credentials that represent evidences in providing authenticity and trustworthiness relatively to a given credential network. All these approaches require that users (of the model) determine the numerical trust measures that are needed as input. For example, in [11] there is the base rate value that determines the a priori trust that would be put in any member of the concerned community and in [18] there is a value called k that determines the rate of change of trust with time and it is assigned by the truster based on its perception about the change.

Furthermore, the majority of this kind of approaches requires user-defined thresholds in order to interpret their outputs as in [6, 9].

All logic-based trust model approaches, build a trust model which allows a derivation of new trust relations from initial (or direct) ones essentially on the basis of the trust transitivity. To limit the use of this principle, some works use the previously discussed notion of trust scope or context as in [11, 18].

In our model, we consider a trust model from a global view taking into account the model architecture. We also take into account other parameters and constraints that may influence each entity’s belief. Our motivation is to give PKI designers a formalism which can be used to reason about a PKI in a global way.

Furthermore, we don’t recommend the use of continuous trust measures which make it difficult to understand the meaning of statements concerning an entity belief. We rather recommend the use of discrete trust levels and more meaningful parameters, such as constraints concerning certificate policies, application context and certification path length, to limit the trust in a statement.

This is also the opinion of the authors of [19] which say that “The output of the metric should be intuitive” in the sense that it should be possible to the average user to understand what the output means. However, our approach can deal with continuous trust measures if the application context requires their use. It suffices that the trust aggregation function be adapted to this kind of measures.

Finally, in the proposed trust model, users don't need to choice any thresholds values or to assign initial continuous trust measures which is very difficult to do in a non- arbitrary way.

V. THE PROPOSED PKI TRUST MODEL

A. Introduction

We describe in this section our new trust model for reasoning about PKI. As we have mentioned above, we use a logic similar to that used in our previous work [3]. However, we handle additional issues like those related to certificate attributes, trust aggregation and application context. The emphasis of this approach may be summarized as follows:

- It allows the modeling of the attribute certificates as well as identity certificates and the entities beliefs about both authenticity binding and entities trustworthiness.
- Any certificate format can be represented, but we recommend the use of formats that can contain useful information about the subject.
- The axioms are intuitive and correspond to the PKI foundations. The role of these axioms is to derive more conclusions from a set of initial statements.
- The assumptions are of two kinds: Assumptions specifics to the analyzed trust model where they are generally satisfied (For instance, in a hierarchical model, we assume that every entity trusts the root CA and knows with certainty its public key) and assumptions about certificate validity that are similar to those made in [17].

In fact, we are not concerned with certificate obtaining and revocation problems. We suppose here that all the certificates are available, valid and not revoked. This assumption, at the contrary to what it is mentioned in [16], is not a real restriction because, normally, in a 'well managed' PKI, there are almost always periods of time where it is satisfied at least with regard to CA-certificates. We are interested in analyzing a PKI trust model in such periods in order to focus on issues like certificate policies, application context and their influence on entities beliefs.

In effect, we are not interested in a specific entity's view at a specific time; on the contrary, we think that it is more suitable to separate the problem of certificates validation/revocation of our problem of analyzing, in a global way, a PKI trust model.

B. Formalization of PKI's concepts

In the following, we will describe the representation of the concepts used in our formalization (in Artificial Intelligence terminology, our universe of discourse). Some of these concepts are used to limit an entity's belief in a statement concerning public keys authenticity or entities trustworthiness.

1) Entities and Keys :

As we have said previously, we distinguish between a CA and an EE which mustn't issue certificates. However, we will consider only the set E of all entities without any distinction between the two kinds, because, as it will be explained below, we use the CA-flag attribute for this purpose. K will denote the set of all valid public keys.

2) Constraints :

The constraints retained in this paper are considered as elements of a set Ct of all possible and useful constraints in the PKI context. Each constraint is represented with the form: 'nc=vc' where nc represents a constraint name and vc its value.

Examples of useful constraints inspired from X.509 certificate specifications are:

- SDN = *d*, where *d* is a valid domain name; means that the subject domain name is *d*.
- CDNs= *{…, di,…}*, where each *di* is a valid domain name; means that the subject key (respectively, the subject entity) can only sign (respectively issue) certificates for EEs that pertain to domains listed in CDNs.
- CA-DNs = *{…, di,…}*, where each *di* is a valid domain name; means that the subject key (respectively, the subject entity) can only sign (respectively issue) certificates for CAs that pertain to domains listed in CA-DNs.
- KU =*{…, ui,…}*, where *ui* is a valid key usage (exp: e-mail signature, messages encryption, certificate signature,...).
- CPs = *{…, pi,…}*, where each *pi* is a valid policy identifier; means that the certificate policy that governs the trust in the formula in question is an element of CPs.
- SCPs= *{…, pi,…}*, where each *pi* is a valid policy identifier; means that the certificate policy respected by the 'subject' of the formula in question is an element of SCPs.
- TP=v, where v is either 'yes' or 'no'. When v='no' this means that the formula in question cannot be used to propagate a trust relation.

For the sake of simplicity, if C1 is a subset of C, we note by 'C1 $_{\dots\ nci,\ \dots}$', the subset of Const that only contains the constraints of the type nci. For example, if C1={SDN=D, KU=encryption, CPs={p1,p2}}, then $C1_{SDN,\ CPs}$ = {SDN=D, CPs={p1, p2}}and $C1_{KU}$={KU=encryption}.

P(Ct) will denote the set of subsets of Ct.

3) Attributes :

The attributes retained in this paper are considered as elements of a set Att of all possible, useful and not highly dynamic (potentially short-lived) attributes in the PKI context. In effect, the use of highly dynamic attributes is in contradiction with our assumption concerning the certificates (valid and not revoked).

Each attribute is represented with the form: '*na=va*' where *na* represents an attribute name and *va* its value.

Examples of useful attributes (some ones are inspired from X.509 specifications) are :

- CA-flag = *v*, where *v* is either 'true' or 'false'; means that the subject is a CA or not.
- Profession = *pr*, where *pr* is a valid profession name; means that the subject profession is *pr*.
- Member = *{…,gi,…},*, where each g*i* is a valid group name; means that the subject is member in each group *gi*.
- Maximmum_value = *mv*, where mv is a valid maximum value (for example, between 0 and one million); means that the subject is reliable in businesses up to *mv* $.

- Certifications=*{…,ci,…}*, where each *ci* is a valid certification id; means that the subject holds each *ci* certification.
- Meaningful-IDs=*{…,idi,…}*, where each *idi* is a valid identity that can be mapped to the subject and that is meaningful in some particular context.

Lets P(Att) denotes the set of all subsets of Att, we define a function At from E to P(Att) which associate to each entity the set of its attributes.

4) Application context :

In our approach, the general context of all statements is the 'PKI context'. We introduce the concept of 'application context' as a more specific context of each statement and this is especially helpful to limit trust propagation.

Lets AC denotes the set of all possible and useful PKI Application Contexts. Examples of elements of AC are : S/MIME e-mail, IPSEC, SSL/TLS, GSS-API, Payment Protocols, etc. P(AC) will denote the set of subsets of AC.

Note that some attributes are only meaningful in a specific application context so that it must be ignored in others contexts to minimize liability concerns.

5) Path length constraint :

In accordance with X.509, the Path Length Constraint indicates "the maximum number of CA certificates that may follow this certificate in a certification path". A value of zero indicates that the CA can issue only EE certificates.

In this paper, we prefer that the value of zero indicates that the subject is considered by the statement issuer as an EE and mustn't issue certificates. We use also a value LC_{max} (as a constant that must be assigned with a value before analyzing a PKI trust model) as a maximal limit of a certification path length.

Note that a non null path length constraint only makes sense if the subject of the statement is a CA which means that the CA-flag attribute must be true.

6) Trust level :

Lets T denotes the set of possible trust levels associated by an entity to a statement. In this paper, we suppose that T is {0, 1, 2, 3}. The value 0 means that there is no trust, the value 1 means that the trust is marginal, the value 2 means that the trust is medium and the value 3 means that the trust is maximal. The choice of four discrete trust levels is inspired from PGP's Web of trust [21] and can be modified if it is necessary in a particular context.

We remind that trust is difficult to quantify and we believe that the choice of discrete levels of trust is more suitable for PKI context than continuous measures of trust.

7) Trust aggregation :

In trust models, trust reasoning is based essentially on trust propagation but can also be based on trust aggregation. Trust aggregation concentrates on giving an estimated trust level for a trustee based on trust levels given by different entities for this trustee.

In our model, we consider a function Ag that has as arguments two trust levels and a set of application contexts and returns one estimated trust level :

$$Ag : \{0, 1, 2, 3\}^2 \times P(AC) \rightarrow \{0, 1, 2, 3\}.$$

For example, we can choose that : Ag(Tl1, Tl2, {e-payment})=min(Tl1,Tl2) , Ag(Tl1,Tl2,{e-mail})= max(Tl1,Tl2).

Note again that if continuous trust levels are used in a specific context, appropriate aggregation function must be used.

C. Formalization of the Statements

In our model, statements and their representations (in Predicate calculus) are one of the followings forms:

1) *Certificate:* Cert(At(X), K_X, A, K_A, LC, C, Ap, Tl) *means* that there is a certificate signed by K_A that claims that the CA A authenticates, whit the trust level Tl and in context of Ap, the binding between At(X) and the public key K_X with the path length constraint LC and under the constraints of C.

If the certificate is an identity certificate then At(X) must contain{meaningfulID=X (or other identity of X)}.

2) Authenticity : Auth(X, At(Y), K_Y, LC, C, Ap, TL, L) *means* that X believes in the binding between Y's attributes At(Y) and the public key K_Y that can be used in the context of Ap for the verification of certification paths only if their length is ≤ LC and under the constraints of C. The trust value of this belief is TL and it is made with the support of L CAs. L=0 means that X has a direct trust about K_Y authenticity.

3) Trust : Trust(X, Y, At(Y), C, Ap, Tl, L) *means* that X believes with the support of L entities that Y is a trustworthy entity with At(Y) as attributes in the context of Ap and under the constraints of C. The trust value of this belief is Tl.

4) Absence of conflict : Noconflict(C, Ap) means that there is 'no conflict' between the constraints of C in the context of Ap.

Noconflict(C, Ap) is false for every context Ap, in the cases where one of the followings pairs of elements is in C :

- SDN= d and CDNs= {d1,d2,…}, where d∉ CDNs.
- SDN= d and CA-DNs = {d1,d2,…}, where d ∉ CA-DNs.
- CPs ={p1, p2,…}and SCPs={p'1, p'2,…} where ∀ i, j : pi≠ p'j.
- TE= 'yes' and TE= 'no'.

An example where there is conflict between C and Ap is when SCPs= {p1, p2,…} is an element of C and the applications of Ap are not mentioned in any policy pi in SCPs.

5) Different : Diff(X, Y) means that X and Y are two different entities.

D. Axioms

We admit all logical axioms of predicates calculus. We consider also as axioms the following formulas :

1) Key-authenticity axiom :

Axiom (A1) expresses the role of a certificate issued by a trustworthy CA A (cf. CA-flag='yes') to prove the authenticity of the binding between the key K_Y and the certificate subject attributes **At**(Y) by a relying party X. It expresses then the basic principle of public key certification. Nonetheless, with regards to the use of constraints, application contexts, trust levels and intermediates number in the axiom's formulas, the axiom validity is less evident, but remains intuitive since it translates the fact that more a certification path progresses

more it becomes constraining and implies more intermediates(CAs).

$\forall X, Y, A \in E, \forall K_X, K_Y, K_A \in K, \forall C1, C2, C \in Ct, \forall Lc, Lc1, L1 \in [0,LC_{max}] \cap N, \forall L2 \in [0,L_{max}] \cap N, \forall Tl, Tl1, Tl2 \in T$

Cert (At(Y), K_Y, A, K_A, Lc, C, Ap, Tl) ^
Auth(X, At(A), K_A Lc1, C1, Ap1, Tl1, L1) ^{CA-flag='yes'}⊂ At(A)
^**Trust**(X, A, At(A), C2, Ap2, Tl2, L2) ^ Ap∩Ap1∩Ap2 ≠∅ ^ (Lc<Lc1)^(L1+1<LC_{max})^**Noconflict**($C_{SDN,CPs}$∪ $C1_{CA\text{-}DNs,CDNs,SCPs}$ ∪ $C2_{CA\text{-}DNs,CDNs,SCPs}$, Ap∩Ap1∩Ap2)
⇒ **Auth**(X, At(Y), K_Y, Lc, C, Ap∩Ap1∩Ap2, Ag(Tl, Ag(Tl1,Tl2, Ap∩Ap1∩Ap2), Ap∩Ap1∩Ap2), L1+1) (A1)

In (A1), the condition Ap∩Ap1∩Ap2 ≠∅ guaranties that at least one application context concerns the use of the authentic key K_Y. The condition (Lc<Lc1) concerns the respect of the certification path length.
According to (A1), if there are conflicts between the 'SDN' and 'CPs' constraints of C and the 'CA-DNs', 'CDNs' and 'SCPs' constraints of C1 and C2, we cannot derive the authenticity of K_Y.

2*) Trust propagation axiom :*

Axiom (A2) concerns the trust propagation that prevents that each relying party must trust 'directly' all the CAs in a certification path. We don't treat the trust propagation between EEs such as in reputation systems. Nonetheless, our formalism can be used to express it.

$\forall X,A1,A2 \in E, \forall C1,C2 \in Ct, \forall L1,L2 \in [0,L_{max}] \cap N, \forall Tl1, Tl2 \in T$

Trust(X, A1, At(A1), C1, Ap1, Tl1, L1) ^{CA-flag='yes'}⊂ At(A1) ^ **Trust** (A1, A2, At(A2), C2, Ap2, Tl2, L2) ^ Ap1∩Ap2 ≠∅ ^{CA-flag='yes'}⊂At(A2) ^ L1+L2+1<L_{max} ^ **Diff**(X, A2) ^ **Noconflict**($C1_{CA\text{-}DNs,SCPs,TE}$ ∪ $C2_{SDNs,CPs,TE}$ ∪{TE='yes'}, Ap1∩Ap2)
⇒ **Trust**(X, A2, At(A2), C2, Ap1∩Ap2, Ag(Tl1,Tl2, Ap1∩Ap2), L1+L2+1) (A2)

As you can remark, we use a constant L_{max} (in addition of LC_{max}) that must be assigned with a value before analyzing a PKI trust model. L_{max} is a maximal limit of a trust propagation chain.

In (A2), the conclusion that states that X trusts A2 is less 'strong' than the trust statements in the conditions. For example, it concerns only the applications of Ap1∩Ap2 and the trust level is Ag(Tl1,Tl2, Ap1∩Ap2) which is always inferior (or equal) to Tl1 and/or Tl2.

Note also that the use of the condition **Diff** (X, A2) prevents the derivation of statements that concern the trust that an entity can have in itself.

3) Trust aggregation axiom :

Axiom (A3) concerns the trust aggregation when an entity X via two different paths has established the trustworthiness of a CA A. The resulting belief encloses the two application contexts Ap1 and Ap2 under constraints of both C1 and C2 and can be seen as derived via a 'virtual' path of (L1+L2) div 2 as length. However, the resulting trust concern only the intersection of the sets of attributes At(A) and At'(A).

$\forall X, A \in E, \forall C1, C2 \in Ct, \forall L1, L2 \in [0,L_{max}] \cap N, \forall Tl1, Tl2 \in T$

Trust(X, A, At(A), C1, Ap1, Tl1, L1) ^ **Trust** (X, A, At'(A), C2, Ap2, Tl2, L2) ^{CA-flag='yes'}⊂ At(A)∩At'(A) ^ (L1+L2)div 2<L_{max} ^**Noconflict**(C1∪C2, Ap1∪Ap2)
⇒ **Trust**(X, A, At(A) ∩At'(A), C1∪C2, Ap1∪Ap2, Ag(Tl1,Tl2, Ap1∪Ap2), (L1+L2) div 2) (A3)

In (A3), when the two statements about the trustworthiness of A concern different sets of attributes, the resulting statement concerns only the intersection of the two sets.

4) Certification axioms :

Axiom (A4-1) states that a –good- CA, before issuing a certificate, must be sure (with a certain trust level Tl) that the key belongs to the subject and that he holds the corresponding attributes. A CA must specify the constraints and the application context related to the use of this certificate and its key.

$\forall X, A \in E, \forall K_X, K_Y, K_A \in K, \forall C \in Ct, \forall LC \in [0,LC_{max}] \cap N, \forall Tl \in T$
Cert(At(X), K_X, A, K_A, LC, C, Ap, Tl)
⇒ **Auth**(A, At(X), K_X, LC, C, Ap, TL, 0) (A4-1)

Axiom (A4-2) concerns the case where the subject is another CA (CA-flag='yes') and asserts that the issuer must be sure of the subject trustworthiness and its capability to act as a CA with regard to some application context and under some constraints.

$\forall X, A \in E, \forall K_X, K_Y, K_A \in K, \forall C \in Ct, \forall LC \in [0,LC_{max}] \cap N, \forall Tl \in T$
Cert(At(X), K_X, A, K_A, LC, C, Ap, Tl) ^ (0<LC) ^ {CA-flag='yes'}⊂ At(X)
⇒ **Trust**(A, X, At(X), C, Tl, ,Ap, 0) (A4-2)

E. Analyzing a simple PKI trust model

The analysis of a PKI trust model based on our approach consists of two phases:

- Expression of all the certificates and the hypothesis of the model with the logic formulas. Our initial global view is the set Δ containing these formulas, our axioms, all valid formulas (as 1<2) and all valid 'No-conflict' formulas.
- Application of a direct resolution method to these formulas that stops when no new formulas can be derived or a resolution by refutation in order to know whether some interesting formulas are satisfied and under which conditions.

As it is well known in predicate calculus logic, both the direct resolution and the refutation are sound. Furthermore, the refutation is complete, if all the formulas are Horn clauses which is the case of the formulas of Δ.

Lets consider the simple PKI trust model of Fig.1 that consists only of two CAs (A1 and A2) and two EEs (X and Y).

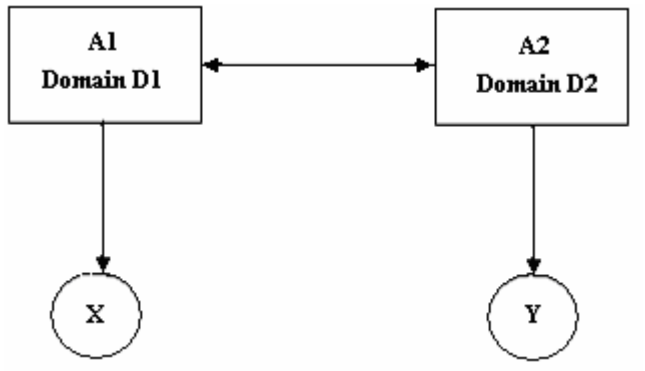


Fig.1. A simple PKI Trust Model

The CAs A1 and A2 pertain to different domains D1 and D2 and they have performed a cross-certification. The user X has a certificate issued by A1 and also pertains to D1. The user Y has a certificate issued by A2 and pertains to D2.

1) Representing the certificates :

The certificate c1 (resp. c2) signed by KA1(resp. KA2) and issued by A1 (resp. A2) to attest its belief about the authenticity of the binding between the X's attributes (resp. Y's attributes)) and the key KX (resp. KY) for signature purposes (resp. for encryption). This issuance was made with respect of the certificate policy p1 (resp.p2) :

Cert({meaningful-ID=X}, KX, A1, KA1, 0,{KU=signature, SDN=D1, CPs=p1}, {email},2) **(c1)**
Cert({member=G1}, KY, A2, KA2, 0, {KU=encryption, SDN=D2, CPs=p2}, {email}, 2) **(c2)**

The certificate c3 (resp. c4) signed by KA1(resp. KA2) and issued by A1 (resp. A2) to attest its belief about the authenticity of the binding between the A2's attributes (resp. A1's attributes)) and the key KA2 (resp. KA1) :

Cert({meaningful-ID=A2, CA-flag=yes}, KA2, A1, KA1, 1, {KU= certification, CDNs=D2, SDN=D2, SCPs=p2, CPs=p1}, {email,, VPN}, 3) **(c3)**
Cert({meaningful-ID=A1, CA-flag=yes}, KA1, A2, KA2, 1, {KU= certification, CDNs=D1, SDN=D1, SCPs=p1, CPs=p2}, {email }, 3) **(c4)**

2) Representing the hypothesis :

The hypothesis concern trivial statements about the trust relationships between the EEs and the CAs they choose to obtain certificates and the fact that they are also sure about the authenticity of the keys of these CAs.

According to CA-DNs and CDNs values, in this example, X (resp. Y) trusts A1 (resp. A2) for issuing certificates to EEs in the domain D1 (resp. D2) and to CAs in the domain D2 (resp. D1) :

Trust(X, A1, {meaningful-ID=A1, CA-flag=yes}, {CA-DNs= D2,CDNs=D1,SCPs=p1}, {email, VPN}, 3, 0) **(t1)**
Trust(Y, A2, {meaningful-ID=A2, CA-flag=yes}, {CA-DNs=D1,CDNs=D2, SCPs=p2},{email, VPN}, 3 , 0) **(t2)**

X believes that the binding between A1's attributes and KA1 is authentic and so does Y with regard to A2's attributes :

Auth(X, {meaningful-ID=A1, CA-flag=yes}, KA1, 2, {CA-DNs=D2,CDNs=D1,SCPs=p1}, {email, VPN}, 3, 0) **(a1)**
Auth(Y, {meaningful-ID=A1, CA-flag=yes}, KA2, 2, {CA-DNs=D1,CDNs=D2,SCPs=p2}, {email, VPN}, 3, 0) **(a2)**

3) Application of the resolution method :

First, we must choose two values for the constants L_{max} et Cl_{max}. Lets 3 be the value of these two constants which is suitable for the size of the model of our example. The set Δ of our initial formulas include the formulas (c1), (c2), (c3),(c4), (t1), (t2), (a1) and (a2).

In this example, we will apply two refutations to the clausal form of the set Δ, in order to try to prove that useful formulas as **Auth**(X, At(Y), KY, Lc_{req}, C_{req}, Ap_{req}, Tl_{req}, L_{req}) and **Auth**(Y, At(X), KX, Lc_{req}, C_{req}, Ap_{req}, Tl_{req}, L_{req}) are logical consequences of Δ. The formulas/goals we have chosen contain both constants and variables in order to find the binding of these variables.

Given space constraints, the details for these two refutations are omitted. The results are (under some conditions):

Δ |= Auth(X, {member=G1}, K_Y, 0, {KU=encryption, SDN=D2, CPs=p2}, {email}, 2, 2) **(RA1)**

Δ |= Auth(Y, {meaningful-ID=X}, K_X, 0, {KU= signature, SDN=D1, CPs=p1},{email} , 2, 2) **(RA2)**

The constraints therein a formula are very important, for example, note that (RA1) and (RA2) cannot be derived if there is conflict between the policies p1 and p2 or between them and the e-mail application context or if in (a1) we have CA-DNs = D3 instead of CA-DNs=D2, the formula Noconflict({CA-DNs=D3, **S**CPs={p2}}∪{SDN= D2, CPs={p2}}∪{TE=yes}) will be false (there is a conflict between CA-DNs=D3 and SDN=D2) and then we will be unable to derive the formula Trust(X, A2, {KU=certification, CDNs=D2, SDN=D2, SCPs={p2},CPs={p1}}, 3, 1) by using the axiom (A2) and then (RA1).

Note that the utility of (RA1) and (RA2) is limited by the specified constraints and application context. For example, (RA1) is without any utility for X if she receives by email a signed facture by the private key associated to K_Y, due to the constraint (KU=encryption).

Remark that the example above is so simple that it doesn't need automation of the refutation method. However, more complex PKI trust models will need the automation of the refutation and the choice of an effective strategy. The use of Prolog as programming language can be useful but we must modify the default strategy (SLD) of Prolog for more efficiency, for example, we must add the new derived formulas at the top of the database (initially the formulas of the set Δ) . We have already tested this kind of automation which seems to be promising and we are working on its amelioration.

VI. CONCLUSION AND FUTURE WORK

Until recently, the use of PKIs essentially resided with governments and a few large organizations. The growth of electronic transactions over Internet such as e-payment transactions requires a global and a trustworthy PKI that can provide security services and especially authentication.

In this paper, we have proposed an approach that is based on a specific logic which will allow us to analyze a PKI trust model and then verify whether it responds to these applications requirements and under which conditions.

In effect, we have presented a new logic-based trust model for reasoning about PKI. We have shown how statements about entities beliefs can be expressed with the formulas of this logic by taking into account different parameters like certificate policy constraints, application context and certification path length.

The main difficulty of our approach is the formulation of rigorous definitions about other possible types of application contexts and other useful constraints. It is also useful to specify with more details the Noconflict relation. We have presented, in this paper, some of these definitions.

At the moment, we focus on the amelioration of this model and its applicability to different kind of PKIs and different security applications including wireless security applications[20].

REFERENCES

[1] S. Chokhani and W.Ford, "Internet X.509 Public Key Infrastructure :Certificate Policy and Certification Practices Framework" , 1999.

[2] A. Colleran, "Final Report: Standardization Issues for the ETS", Quercus Information Ltd, 1997.

[3] H. EL Bakkali, B. Idrissi Kaitouni, **"**A Predicate Calculus Logic for the PKI Trust Model Analysis", Proc. IEEE International Symposium on Network Computing and Applications, Springer-Verlag, pp. 386-372, 2001.

[4] S. Farrell and R. Housley, "An Internet Attribute Certificate Profile for Authorization", IETF, RFC 3281, 2002.

[5] M.R. Genesereth and N.J. Nilsson, "Logical Foundations of Artificial Intelligence", Morgan Kaufmann Publishers, pp. 9-62, 1987.

[6] R. Haenni, Using Probabilistic Argumentation for Key Validation in Public-key Cryptography", International Journal of Approximate Reasoning,, Vol. 38, No. 3, pp.355–376., 2005.

[7] L.J. Hoffman, K. Lawson-Jenkins and J. Blum M., "Trust Beyond Security: An Expanded Trust Model", Communications of the ACM, Vol. 49, No. 7, pp. 94-101, 2006.

[8] R. Housley, W. Ford and D. Solo, "Internet Public Key Infrastructure; Part I: X.509 Certificate and CRL Profile", IETF X.509 PKI (PKIX) Network Working Group, 1999.

[9] J. Jonczy and R. Haenni, "Credential Networks: a General Model for Distributed Trust and Authenticity Management", Proc. International conference on Privacy, Security and Trust, A. Ghorbani& S. Marsh (Eds), pp. 101-112, 2005.

[10] A. Josang, "An algebra for assessing trust in certification chains", Proc. 6th annual symposium on network and distributed system security, 1999.

[11] A. Josang, R. Hayward and S.Pope, "Trust Network Analysis with Subjective Logic"**,** Proc. Twenty-Ninth Australasian Computer Science Conference, CRPIT, Vol. 48. V. Estivill-Castro and G. Dobbie (Eds), pp. 85-94, 2006.

[12] A. Jøsang, "Prospectives for Online Trust Management", unpublished.

[13] L. Kagal, T. Finin and Y. Peng, "A Framework for Distributed Trust Management"**,** Proc. IJCAI-01 Workshop on Autonomy, Delegation and control, 2001.

[14] R. Kohlas and U. Maurer, "Confidence Valuation in a Public-Key Infrastructure based on Uncertain Evidence", Proc. Third International workshop on Practice and Theory in Pubic Key cryptography, H. Imai & Y. Zheng(Eds), pp. 93-112, 2000.

[15] R. Kohlas, J. Jonczy, and R. Haenni, "Towards a Precise Semantics for Authenticity and Trust", Proc. International Conference Privacy Security and Trust, McGraw-Hill (Eds), pp., 2006.

[16] J. Marchesini and S. Smith, "Modeling Public Key Infrastructure in the Real World", Proc. European public Key Infrastructure Workshop : research and applications, pp. 118-134, 2005.

[17] U. Maurer, "Modeling a Public-Key Infrastructure", Proc. European Symposium on Research in Computer Security, Springer-Verlag, Vol.1146, pp. 325-350, 1996.

[18] I. Ray and S. Chakraborty, **"**A Vector Model of Trust for Developing Trustworthy Systems", Proc. European Symposium on Research in Computer Security, Springer-Verlag, pp. 260-275, 2004.

[19] M.K. Reiter and S.G. Stubblebine, "Authentication Metric Analysis and Design", Proc. ACM Transactions on Information and System Security, Vol. 2, pp. 138-158, 1999.

[20] N. Sklavos and X. Zhang, "Handbook of Wireless Security: From Specifications to Implementations", CRC-Press, A Taylor & Francis Group, ISBN: 084938771X, 2007.

[21] P. Zimmermann, "PGP User's Guide", vol. 1 and 2, 1994.

[22] ISO/IEC 9594-8/ITU-T Recommendation X.509, "Information Technology, Open Systems Interconnection - The Directory: Authentication framework", 1997.

An Autonomic Computing Approach to Server Virtualization

Dan Ionescu, Bogdan Solomon,
Network Computing and Control Technologies Research Laboratory, School of Information Technology and Engineering, University of Ottawa
bsolomon@ncct.uottawa.ca,
ionescu@ncct.uottawa.ca

Marin Litoiu, Mircea Mihaescu
IBM CAS Toronto
marin@ca.ibm.com, mihaescu@ca.ibm.com

Abstract

Virtualization imposed itself most recently as a disruptive technology changing concepts related to data centers, corporate transparency, and paving the road to Software as a Service (SaaS), yet another disruptive technology. Autonomic computing, on the other hand, has imposed itself as a new research area whose aim is to embed "intelligent algorithms" in the IT infrastructure management software. Mechanisms such as self-configuring, self-protection, self-healing and self-optimization have been investigated and appropriate algorithms proposed. More recently, a real-time adaptive architecture has been introduced and experiments conducted which show that the autonomic computing becomes more and more accomplished as a research area. Despite some initial resistance, Linux has been penetrating the user community needing a stable and threat free operation system. In conjunction with it the Software as a Service has seen quite a successful expansion. Solutions leading to run different servers on different operating systems running virtually on the same processing unit showed a dramatic improvement in their utilization. Server Virtualization opens new perspectives on datacenter management, while the self-management concepts introduced by autonomic computing researchers can provide solutions for the monitor and control tasks applied to virtual machines. This paper introduces an autonomic computing approach to virtual computing environments in regards to the automatic provisioning of virtual servers. The paper presents the main autonomic solution to server virtualization, detailing the model to be used for representing virtual applications as running software processes. The paper also introduces the control system approach to the server virtualization provisioning. A real-time architecture is proposed and experiments performed on virtualizing clusters of WebSphere servers deployed on a distributed system Xen servers are given.

I. INTRODUCTION

Virtualization has become one of the hottest areas of research and marketing in the IT industry. Driven by industry analyst reports and by marketing hypes the virtualization began to be the most "must have" solution for efficient server, space and power utilization/consumption. First attempt to an abstract definition for computing process virtualization was recorded in [1] where a pseudo-formal definition was introduced. The following was stated: "For any computer a virtual machine monitor may be constructed if the set of sensitive instructions for that computer is a subset of the set of privileged instructions". From a technical point of view, the virtualization beginning traces back to the mainframe era when the IBM VM/370 allowed to virtually use the CPU for many jobs at the same time through its time sharing technique [3]. The virtual machine introduced by IBM was defined as a fully protected and isolated copy of the underlying physical machine's hardware [2] where applications and even operating system behave as they would behave on the physical machine. There is a software layer, the *virtual machine monitor (VMM)* which virtualizes hardware resources, exporting a virtual hardware interface that reflects the underlying machine architecture. VM/370 supported multiple concurrent virtual machines, each of which believed it was running natively on the IBM System/370 hardware architecture [4].

Thus multiple virtual machines can be enabled on request and more than one OS can run simultaneously on a single CPU - every instance running within its own virtual hardware. This independence of the resources used makes the server movable and easier to recover from failures. However, with the increase in hardware efficiency the economy realized it has a downside in regards to the hardware failures - once a hardware resource is lost the concentration of virtual

applications running on the same resource increases exponentially. This slows down the time response of the applications running under the virtual machine or operating system. There are also other limitations when applying a virtualization solution. Applications which have a large amount of I/O operations and databases are not good candidates for this technique. Recent research and company products provide virtualization platforms which break the bond between the hardware and the operating system running on it. The approach discussed in this paper relates only to the VMM based server virtualization. As such products such as Plex86 [5], User Mode Linux [6], Virtual PC [7], and VMWare [8] will be the candidates for the automation of the server virtualization through autonomic computing technologies. It is notable that a special Xen-compatible version of Linux, XenoLinux [9] can run in the Xen environment for up to 100 XenoLinux instances, all in a single Xen VMM with minimal performance degradation. Xen-compatible versions of Windows XP and NetBSD are also available.

On the other hand, autonomic computing is yet another methodology for improving the IT infrastructure efficiency and automation. Autonomic computing is yet another hot research topic initiated by IBM [10]. Principles such as self-configuration, self-protection, self-optimization, and self-healing are combined into a software platform responsible for the management of the software and hardware infrastructure. These self-management attributes of a computing system allows it to adapt at run time its resource usage to the demands put on them by ever changing user needs, thus continuously optimizing itself in a transparent way. Recent works in the autonomic computing area [11], [13] envisage autonomic computing models as real-time systems equipped with transducers, controllers, and actuators whose goals are the automation of the IT infrastructure, while the dynamics of the Information Infrastructure itself is described by dynamic mathematical systems. This view of autonomic computing processes allows for an infusion of principles and methodologies specific to control systems into the architecture and implementation of autonomic computing. The real-time system view of an autonomic computing system is related to the feedback control of a cluster of IT applications. A real-time reference architecture for autonomic computing platforms has been proposed in [12], while an advanced adaptive mechanism for the adaptation of the provisioning of computing resources to the reality of processing requests has been introduced in [14]. It is on this ground that an autonomic computing control technique for the provisioning of virtual servers on a given hardware platform will be introduced and developed in this paper. This paper provides a solution for the dynamic load control on VMM based virtual computing platforms. Section 2 introduces an autonomic computing architecture for the server virtualization. Section 3 presents the components of the autonomic control loop while Section 4 discusses the computational model of virtual servers which run on the same CPU. Section 5 shows in more details the implementation of the system and finally Section 6 presents some experimental results obtained with this platform.

II. A REAL-TIME AUTONOMIC COMPUTING ARCHITECTURE FOR SERVER VIRTUALIZATION

As mentioned in Section I in this paper only VMM based server virtualization solutions will be investigated and an autonomic computing architecture will be designed and its implementation commented upon. As in [12], a pattern based architecture will be devised for the autonomic computing self-provisioning of virtual servers. The architecture for sever virtualization will contain the same type of elements as the reference real-time architecture for autonomic computing such as transducers, filters, estimators, controllers, decision elements, and actuators. The process to be controlled is not the addition or removal of a server, rather the CPU "partition" made by the VMM. As opposed to the regular usage of the VMM where the CPU partitions once configured are fixed until maybe another configuration command is performed by the system administrator, in this case, based on the need for client services, the partition is controlled by the autonomic computing control loop. The proposed architecture allows for the autonomic solution for the virtual server to modify itself during runtime, as components can be dynamically added, removed or modified as needed by the system. The components used are distributed, residing on any node of the virtual server infrastructure, thus allowing for scalability which is one of the major issues in server farm deployment. The system self configuration/reconfiguration [12] allows for the automatic deployment or removal of virtual sever components when needed. A state persistence mechanism allows to smoothly manipulating components such that the functionality of the whole virtual server or of the applications running in the virtual server environment is not affected by autonomic computing decisions.

In Figure 1 below a classical VMM virtualization architecture is depicted. The VMM is used to partition the CPU and transparently run the corresponding operating system on it. As VMM, was selected Xen,

and Debian Linux 4.0 was used as the operating system. A Java Virtual Machine 1.5 was loaded on both operating systems. The applications selected for illustrating the way that the autonomic computing architecture helps to improve the efficiency of the hardware and decrease operating costs are related to Web Services. The justification stems from the fact that "consolidating Internet Services to a single machine or group of machines by using virtual machines is an elegant solution that provides the benefit of isolation while simultaneously reduces wasted computing power and maintenance of additional computers" [15].

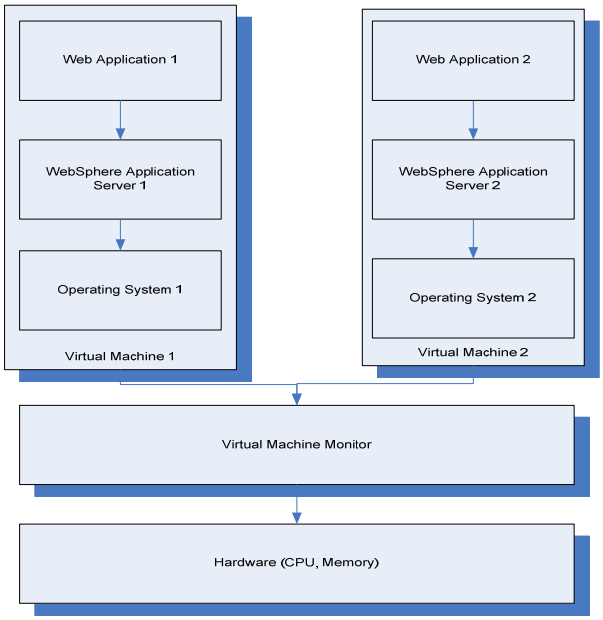


Fig. 1. A Virtual Machine Monitor based solution for Virtual Servers.

The autonomic computing architecture proposed for CPU load distribution and control for the virtual servers running under the same VMM, is presented in Figure 2, where only two such servers were considered. The architecture is based on the standard components of the autonomic computing architecture presented in [12]. As such, the architecture contains sensors for the CPU load and for the response time of each of the servers, filters to smooth the rapid variations of the above parameters, estimators to estimate the predicted values for the measured parameters which characterize the activity of the virtual server, coordinators responsible for the main logic of the control loop and drives the whole autonomic computing process, and the decision maker which translates the results of the coordinator in commands sent to the actuators. The actuators are nothing else then the provisioning part of the autonomic computing loop for the virtual server CPU load control. In real cases the actuators are implemented as Tivoli Provisioning Managers. In a real case a CPU can be loaded with more than two servers. Studies and tests conducted with Xen showed that a number of 100 Linux machines running different servers were function with acceptable performance on the same CPU.

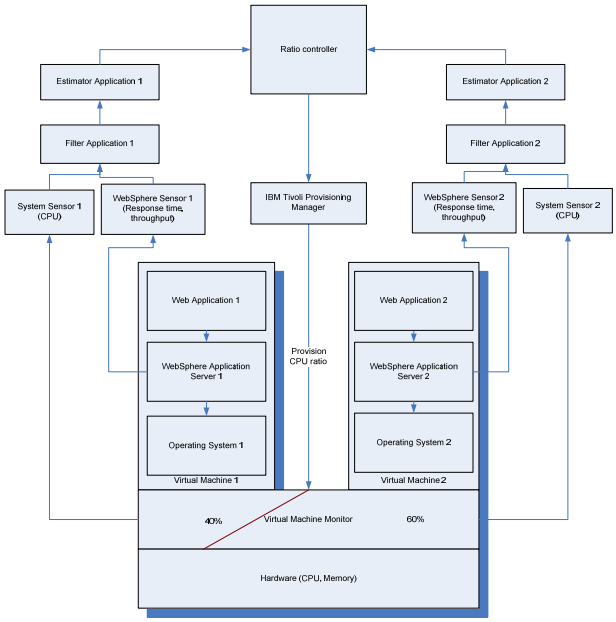


Fig. 2. An autonomic computing architecture for the CPU load control in the case of two virtual servers

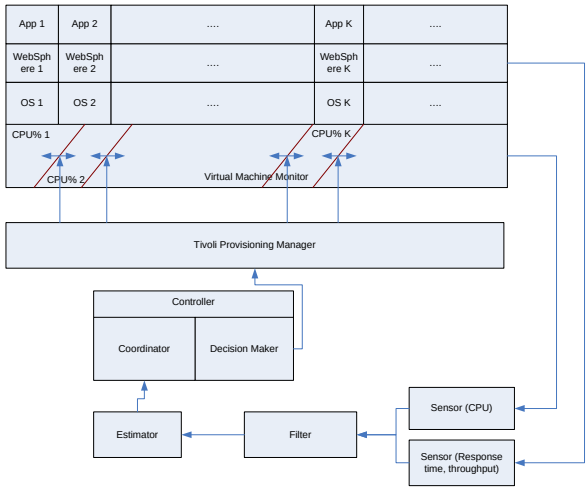


Fig. 3. A generalized architecture for multiple servers sharing the same CPU

III. THE COMPONENTS OF THE REAL-TIME ARCHITECUTRE FOR VIRTUAL SERVERS

A. Sensors

The system described in this paper, uses two types of sensors for the measurements. The principal measured parameters are the response time which is obtained from the WebSphere Application Servers for each of the virtual servers, and the CPU load which is obtained from the Virtual Machine Monitor. From the application servers the sensors measure response time, for the monitored application, as well as the number of invocations received by a certain application and the timestamp of the data acquisition. From the Virtual Machine Monitor the sensor retrieves the percentage of CPU time used by each of the virtual machines. The information coming from the sensors represent absolute values from the start of the application server and as such they have to be processed by the filters introduced in the loop.

The sensors are implemented as Web Services Distributed Management (WSDM) resources based on the Apache Muse 2.2 framework. The WebSphere sensor exports information obtained from the application server via JMX; while the VMM sensor retrieves the time each of the VM domains has run for.

B. Filter

The filter component implements a filter chain. Based on the data retrieved from the sensors, the filter chain calculates the average throughput of users, by dividing the number of invocations by the time period.

$$Throughput = \frac{\Delta Users}{\Delta t}$$

The average think time is also calculated as follows:

$$Thinktime = \frac{\Delta t - users * Avg\,\mathrm{Re}\,sponsetime}{users}$$

where Δt represents the time period, AvgResponsetime is the average response time for the last time period and users is the average number of invocations made by one user.

Furthermore the filter chain calculates the average response time, number of users, throughput and think time across all the servers in the cluster. For example for a cluster of N servers, the average of V across the servers is calculated as follows.

$$Avg = \frac{V1 + V2 + \ldots + VN}{N}$$

C. Estimator

The estimator estimates the state values of the virtual server for each time step, such that the coordinator can elaborate the proper decision based on the predictions of the state of the virtual server whose CPU load is to be modified accordingly. The estimation is based on a model for the virtual server which is considered to be non-linear. The system of equations for this model is:

$$\begin{aligned} \hat{x}_i(k) &= f(\hat{x}_i(k-1), u_i(k-1), 0) \\ \tilde{z}_i(k) &= h(\tilde{x}_i(k), 0) \end{aligned} \quad (1)$$

where x∈ X (dim n), u ∈ U (dim m), z ∈Z (dim p) and called the state vector, input function vector, the number of resources required, and measured output vector, while f, and h are nonlinear functions over the state space and input vectors respectively, while i ∈ N represents the index of the virtual server.

Through linearization the model (1) above becomes:

$$\begin{aligned} x(k) &= \tilde{x}(k) + A(x(k-1) - \hat{x}(k-1)) + Ww(k-1) \\ z(k) &\approx \tilde{z}(k) + H(x(k) - \tilde{x}(k)) + Vv(k) \end{aligned} \quad (2)$$

i=1,..K

where A, W, H, and V are NxN, NxM, PxN, and PxM matrices respectively, given by :

$$A_{[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[j]}}(\hat{x}(k-1, u(k-1), 0)$$

$$W_{[i,j]} = \frac{\partial f[i]}{\partial w[j]}(\hat{x}(k-1), u(k-1), 0)$$

$$H_{[i,j]} = \frac{\partial h[i]}{\partial x[j]}(\bar{x}(k-1), 0)$$

$$V_{[i,j]} = \frac{\partial h[i]}{\partial v[j]}(\bar{x}(k), 0)$$

The state obtained in (2) is passed to the next block of this virtual server self configuration system, he coordinator.

D. Coordinator

The controller subsystem is formed from the coordinator, and the decision maker. All components act together to reach a provisioning decision based on the filtered data, but they are also separate components which can run on separate nodes.

The coordinator is responsible for the main logic of the control loop and drives the whole process. As such the coordinator is a thread. The control loop logic for the virtual machines is an iterative control loop. After obtaining the data from the filter, the coordinator uses the estimator repeatedly, until either a certain number of iterations is passed or two consecutive estimations are within a certain threshold. Once one of the two events happens, the estimator requests a decision from the decision maker. Finally that decision is sent to the actuator for provisioning.

D. Decision Maker

The decision maker reaches a decision for how the CPU load should be split among the VMs as well as whether a new VM is needed dependant on the state of the model and the estimated values.

E. Actuator

The actuator component is responsible for taking the actual provisioning decision on the managed entity or entities. The actuator maps the command issued by the coordinator in regards to the modification required to the CPU load of each server in an acceptable command/script sent to the VMM via SSH (in this case the Tivoli Provisioning Manager). Once the actuation action is completed it returns the information about the entities that were modified back to the coordinator. This feedback is required in some instances because the coordinator needs to create or remove sensors based on what happened to the managed entities or needs to update the model with the new values. In the case of a new server addition, the coordinator would have to add new sensors to monitor the newly added virtual server.

IV. LOGICAL ARCHITECTURE OF THE AUTONOMIC COMPUTING SOLUTION FOR CPU LOAD CONTROL FOR VIRTUAL SERVERS

The predicted value of the x(k) – the virtual server state vector – represents as a vector, the expected CPU load, the response time, and the application throughput as its components. The expected CPU load to be used in the next computation phase by the Coordinator and Decision Maker blocks is processed by the Kalman controller which uses a linearized set of equations of the non-linear process model described by the equations (1). The action calculated by the controller in regards to the CPU load for each virtual server is based on a logic decision according to which a server is added or not, while its predicted response time is considered as well. From a logical point of view the system general architecture is given in Figure 4 below.

A layered queuing model (LQM) was used for representing users' interactions with the system such that the linear approximation errors are minimized. In the experiment conducted there were 2 separate virtual servers, each allowing 2 application servers to respond to clients' request, and which were given a thinking time with a mean of Z ms. (as a default, Z= 1000). The 2 application servers each modeled with a certain number of threads, run on processor whose CPU demand is also modeled through the LQM.

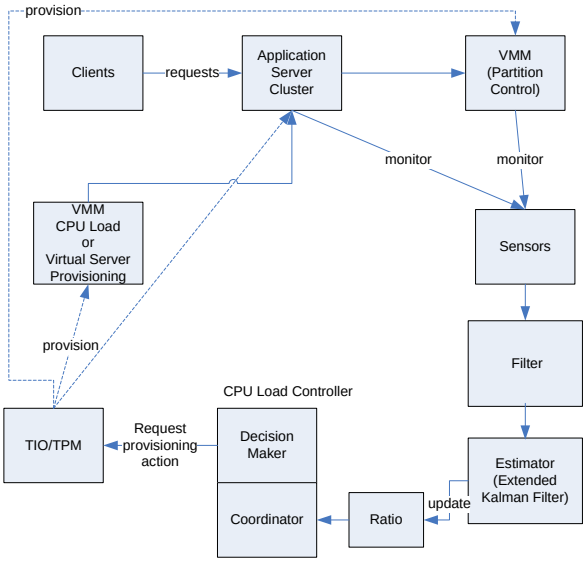


Fig. 4. Logical architecture of a tracking filter for autonomic computing control of the CPU load in a virtual server environment

This example was extended to 3 virtual servers where 3 CPU loads were dynamically controlled. Experiments are conducted with a larger number of servers on Sharcnet [15]; however, these experiments will be reported in a subsequent paper.

The logic of the Coordinator contains also rules through which the Controller (the combination of the Coordinator and Decision Maker) decides whether or not to deploy another virtual server or to remove it from the list of existing virtual servers running on the same CPU. The rules are also used to implement minimal threshold in regards to the CPU partition assigned to a virtual server and also to decide the threshold regarding the optimal utilization of the assigned CPU partition. Besides the above decisions, the actions sent to the Tivoli Provisioning Manager include commands through which Tivoli Provisioning Manager deploys or removes application servers.

V. IMPLEMENTATION

The architecture for the CPU load self configuration introduced in this paper was implemented in a test-bed environment at NCCT (www.ncct.uottawa.ca). Figure 5 describes the test bed set-up used in the demonstration of the solution proposed in this paper.

The virtual server environment was set by installing a Xen VMM on an Intel Core 2 Duo PC, while the TPM was installed on another IBM PC with 2.6 GHz processor. The virtual machines were paravirtualized Debian Etch Linux guests. The components of the autonomic computing environment were implemented as Web-Services all running on a 3.2 GHZ Intel processor PC.

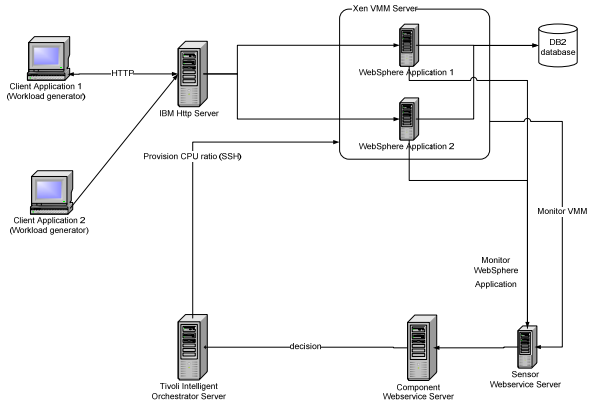


Fig. 5. An implementation of the autonomic computing self-configuration of the CPU load of virtual servers

The configuration of the system was done via an interface based on JavaServer Faces technology. This interface is shown in Figure 6 below.

Fig. 6. The GUI of the autonomic computing self-configuration solution for virtual servers.

The GUI presents the user a very intuitive view of the system, while a series of wizards allow the user to configure the autonomic system itself. As the autonomic computing solution can be applied to a diverse spectrum of self-configuration solutions for various processes, the user is given the possibility to choose the proper process such that the autonomic computing self-configuration loop can be adapted to it.

At the same time, through simply drag-and-drop actions the user can build the appropriate real-time control architecture for the process type chosen. A tree of control loop elements is present in the interface. By dragging-and-dropping each of the elements in the loop the user has only to connect them by setting up the arrow lines between the above elements.

In Figure 7 below the wizard for initial configuration of the sensors used is given. The CPU load, the response time and the number of invocations are set to provide the coordinator the appropriate measured values. These values are filtered with a filter as specified in Section III. The filter set-up is implemented via a wizard as shown in Figure 8 below.

Fig. 7. The sensor wizard

Fig. 8. The filter wizard

The estimator is configured using the wizard shown in Figure 9 and 10. The estimator configuration process includes the choice for a model which is illustrated in the Figure 9 below. In Figure 10 the configuration of the estimator maps the filtered variables to the modeled variables.

For the autonomic system to appropriately self configure and change dynamically the configuration of the Virtual Server environment a wizard allowing the user to build the coordinator is required. The parameters to choose are the number of iterations needed to come up with a sound estimation for the modeled variables, the convergence measure and the sampling time as explained in section III. An example of such configuration process is given in Figure 11 below. The process of properly provisioning the decision maker such that the control loop sends appropriate commands to the actuator – in this example the Tivoli Provisioning Manager, is given in Figure 12.

The parameters to be provided to the decision making block relates to the thresholds used for the response time, the CPU load, and the number of user requests per minute.

Fig. 9. The wizard allowing the user to chose an appropriate model

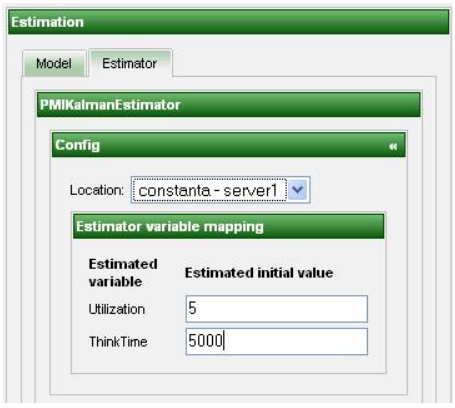


Fig. 10. The wizard allowing the user to chose the utilization and the think-time in the estimator wizard

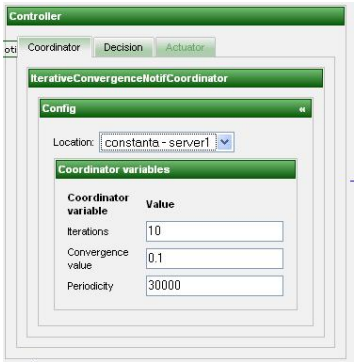


Fig. 11. The wizard allowing the user to chose the coordinator parameters

After such thresholds are over passed or under passed the system provides the actuator the action to be taken. Eventually the user has to tell the system which actuator is to be used and on which physical machine (computer) the actuator resides such that it can deploy all the commands related to the configuration of the whole autonomic computing environment and also which Tivoli specific parameter has to be passed for the initialization of the actuator. Figure 13 shows the above wizard.

Fig. 12. The wizard allowing the user to finely tune the decision making block

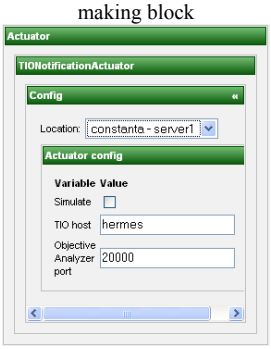


Fig. 13. The wizard allowing the user to configure the actuator

VI. EXPERIMENTAL RESULTS

A series of test were devised and performed on the architecture introduced above. Some results are shown below where in Figure 14 the CPU Utilization in the case of two Virtual Servers installed on the same hardware is presented, when the CPU load distribution is as given in Figure 15 below using a scale where 1 = 100% utilization.

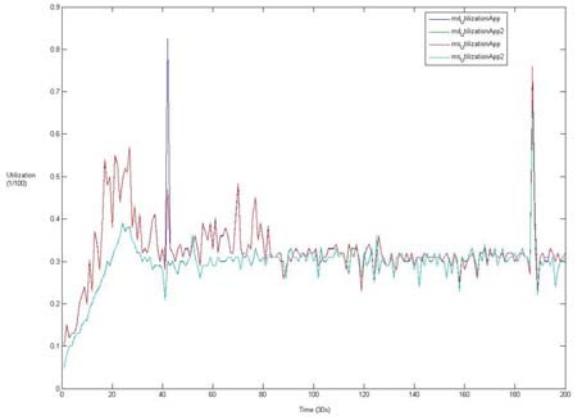


Fig. 14. The real-time values of the CPU load in the case of two Virtual Servers controlled by the autonomic computing platform

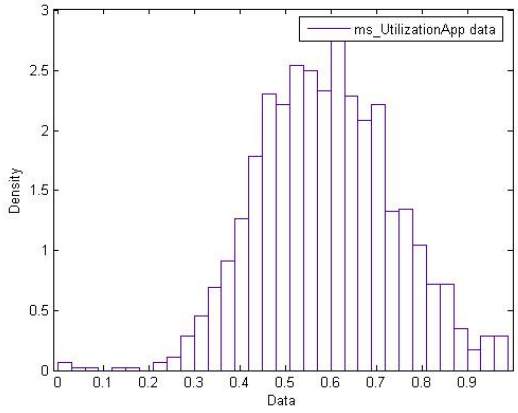


Fig. 15. The CPU load distribution as used in the case of two Virtual Servers

VII. CONCLUSIONS AND FUTURE WORK

In this paper, an autonomic computing architecture for real-time provisioning and configuration of a Virtual Server environment was introduced. The architecture introduces for the first time an intelligent control of the CPU load using demonstrated principles for autonomic computing systems. A control loop approach was used for the dynamic modifications of the CPU allocations through a Virtual Machine Monitor. The architecture, as introduced in a previous series of works, is based on seven different modules, each responsible for one specific action in the control loop. Each module is capable of running as a dedicated Web Service. The versatility of the solution introduced was increased dramatically through a solid user interface platform. Experimental results show that a dynamic CPU load control improves the hardware utilization and optimizes the Virtual Server solution. This is extremely important for the new trend of offering Software as a Service (SaaS) which is considered as one of the important new disruptive technology in the high tech industry.

Future works will explore other mechanisms for controlling the self-management of Virtual Server environments in which a blend of well known control strategies such as robust or output control with adaptive based methods will be devised.

ACKNOWLEDGMENT

The authors would like to thank the IBM Center for Advanced Studies (CAS) for the help given with TIO and the autonomic manager.

REFERENCES

[1] G. J. Popek and R. P. Goldberg, "Formal requirements for virtualizable third generation architectures," Commun. ACM, vol. 17, no. 7, pp. 412–421, 1974.
[2] IBM: "IBM Virtual Machine/370 Planning Guide" 1972
[3] R. J. Creasy, "The Origin of the VM/370 Time-Sharing System," IBM Journal of Research and Development, vol. 25, no. 5, p. 483, 1981.
[4] P. H. Gum, "System/370 Extended Architecture: Facilities for Virtual Machines," IBM Journal of Research and Development, vol. 27, no. 6, p. 530,
[5] "The Plex86 Project," 2003.
[6] S. T. King, G. W. Dunlap, and P. M. Chen, "Operating System Support for Virtual Machines," in USENIX Technical Conference, 2002.
[7] "Virtual PC Technical Overview," 2004.
[8] G. Venkitachalam and B.-H. Lim, "Virtualizing I/O Devices on VMware Workstation's Hosted Virtual Machine Monitor."
[9] U. of Cambridge Computer Laboratory, "The Xen Virtual Machine Monitor," 2004.
[10] IBM Research, "Autonomic Computing Manifesto" http://researchweb.watson.ibm.com/autonomic/manifesto/autonomic_computing.pdf
[11] Litoiu, M., Woodside, M., Zheng, T. "Hierarchical Model-based Autonomic Control of Software Systems", CASCON 2005
[12] B.Solomon, D.Ionescu, M.Litoiu, M.Mihaescu: "A Real-Time Pattern Based Approach to Autonomic Computing" SEAMS 2007 ACM Workshop on Software Engineering for Adaptive and Self-Managing Systems, May 26-27, 2007, Minneapolis, Minnesota, USA.
[13] B.Solomon, D.Ionescu, M.Litoiu, M.Mihaescu: "Decentralized Predictive Control of Autonomic Computing Environments" 4th International Information and Telecommunication Technologies Symposium (I2TS'2006), pp. 94-103, Cuiaba, Brazil, December 6-8, 2006
[14] B.Solomon, D.Ionescu, M.Litoiu, M.Mihaescu:" Adaptive Autonomic Computing" CACON 2007, Toronto October 23-25, 2007
[15] Sharcnet, http://www.sharcnet.ca
[16]Robert Rose: "Survey of System Virtualization Techniques" – unpublished

Analysing Parallelism into an Image Mining System

J. Fernández, R. Guerrero, N. Miranda, F. Piccoli
Universidad Nacional de San Luis
Ejército de los Andes 950
5700 - San Luis - Argentina
e-mail: {*jmfer, rag, ncmiran, mpiccoli*}*@unsl.edu.ar*

***Abstract*—Image gathering and the subsequent repository's generation is accomplished in a wide range of areas: commercial, institutional, military. The widespread use of images is because they are a potential information source to be analyzed and processed at later. As a consequence, it is necessary to support big collections of complex type information which includes complex objects data, spatial information or multimedia information, in others word, it involves identifying present characteristics in an image and its consistent evaluation. Many research works have focused on images and image mining. The main obstacle is to get knowledge of the whole existing image domain for, afterwards, be able to infer information, this demands greater processing power.**

Image mining, a relatively new and very promising field of investigation, tries to ease this problem proposing some solutions for the extraction of significant and potentially useful patterns from these tremendous data volume. This research field implies different stages, most of them demanding so many resources and computational time. The use of parallel computation is a good starting-point. Image mining process appears to be algorithmically complex requiring computing power levels that only parallel paradigms can provide in a timely way. As data sets involved are large, rapidly growing larger and images provide a natural source of parallelism, parallels computers could be organized to handle such big collection effectively. At this work we will examine the image mining problem with its computational cost, propose a possible global or local parallel solution and also identify some future research directions for image mining parallelism.

I. Introduction

Images can reveal useful information to human users when are analyzed. The explosive growth in applying images as data in many fields of science, business, medicine, etc.. With the advances in multimedia data acquisition and storage techniques, the need for automatically discovering knowledge from large image collections is becoming more and more relevant. Image mining deals with the extraction of image patterns from a large collection of images, whereas the focus of computer vision and image processing is in understanding and/or extracting specific features from a single image.

Group supported by National University of San Luis and the National Agency for Science and Technology Promotion(ANPCYT)

The most general misinterpretation is that image mining only involves applying already existing data mining algorithms on images. Investigations in the area are usually pointed out into two main directions. The first one involves specific authority applications focusing on extracting most relevant image features, so they could be used in data mining [14][17][18]. The second direction applies to general applications, where the aim is discovering image patterns that might be useful in the understanding of existing interactions between human perception of the images at high level and image features at low level. Investigations in this direction try developments with major certainty of success in recovered images from a general purpose databases [13][20][24].

Human visual system has the ability to extract significant image relationships which are not represented in low-level primitive image features. Complex information and its use on specific applications leads to describe new association rules to information. The big challenge in image mining is extracting implicit knowledge, image data relationships, or other features not explicity stored in a pixel representation. As knowledge representation method, *patterns* have already been used by human being for simulating diverse cognitive processes like intuition, intention and thinking. As long as the use of patterns can make the cognitive process more effective, they can be applied to describe the complexity and features of objects. Since the aim is to generate all significant patterns without any knowledge of the image content, diverse patterns types could be recognized: classification, description, correlation, temporal and spatial patterns.

Image mining deals with all aspects of large image databases including image storage, indexing schemes, and image retrieval, all concerning an image mining system [16]. Image databases con-

taining raw image data as information, cannot be directly used for image mining purposes. Relational databases, traditionally used in data mining, do not satisfy this need; that is why other types of databases are defined like spatial, temporary, documentary and multimedia databases [26].

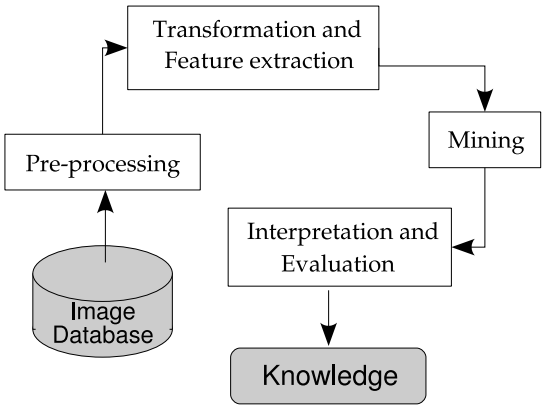


Figure 1: General Image Mining System

Figure I shows a general structure model for an image mining system. The system considers a specified sample of images as an input, whose image features are extracted to represent concisely the image content -Transformation and feature extraction phase-. Besides the relevance of this mining task, it is essential to consider the invariance problem to some geometric transformations and robustness with respect to noise and other distortions while designing a feature extraction operator -Pre-processing phase-. After representing the image content, the *model description* of a given image -the correct semantic image interpretation- is obtained. Mining results are obtained after matching the model description with its complementary *symbolic description*. The symbolic description might be just a feature or a set of features, a verbal description or phrase in order to identify a particular semantic.

The development of an image mining system is often a complex process since it implies joining different techniques ranging from data mining and pattern recognition up to image retrieval and indexing schemes. Besides, it is expected that a good image mining system provides users with an effective access into the image repository at the same time it recognizes data patterns and generates knowledge underneath image representation. Such system basically should assemble the following functions: image storage, image pre-processing, feature extraction, image indexing and retrieval and, pattern and knowledge discovery.

Image mining deals with the extraction of image patterns from a large collection of images, whereas the focus of computer vision and image processing is in understanding and/or extracting specific features from a single image. It might be thought that it is much related to content-based retrieval area, since both deals with large image collections. Nevertheless, image mining goes beyond the simple fact of recovering relevant images, the goal is the discovery of image patterns that are significant in a given collection of images. As a result, an image mining systems implies lots of tasks to be done in a regular time. Images provide a natural source of parallelism; so the use of parallelism in every or some mining tasks might be a good option to reduce the cost and overhead of the whole image mining process [1].

This works is structured as: the following section explains the differents difficulties and challenges involved in designing an image mining model. The section III explain the three main stages constituing a standard image mining system and their feasibility to be parallelize. Finally different parallel image mining models are proposed.

II. Difficulties and Challenges

Image mining deals with the study and development of new technologies that allow accomplishing this subject. A common mistake about image mining is identifying its scopes and limitations. Clearly it is different from computer vision and image processing areas. Moreover, the many knowledge discovery algorithms defined in the context of data mining are ill-suited for image mining. In image mining, there are many challengers still to overcome, some of them are:

- **Complexity of data:** To work with image and visual data is often to work with unstructured data, difficult to interpret and stored in a variety of different formats.
- **Scalability:** Image databases can easily reach hundreds of gigabytes and even terabytes in size. Scalable tools and algorithms for pre-processing and mining images that can manage such extremely large data in a reasonable time are yet to be developed. Massively parallel and high performance computing should help in this perspective for both image pre-processing and image mining.

- **Data inaccessibility:** Data adquisition and selection is fundamental in knowledge discovery process. The reasons for inaccessibility are multiple depending on the gathering means: sensors, satellites, among others. As incredible as it may seem, gathering images for research purpose or even industrial applications is not an easy task.
- **Privacy:** This has been an important issue with any data gathering and access. In some applications, image mining propels the problem of privacy a step further.
- **Minor support:** Image mining is relativity new, it relies heavily on fields such as vision and signal processing for data pre-processing and features extraction, fields which lack of adequate tool support, but in constant development.
- **Insufficient training:** Knowledge discovery and image mining tasks are related with many disciplines: artificial intelligence, databases, image and vision processing, high performance computing, visualization, etc. Interdisciplinary skills and work are required to process and cope with image.

To solve them in only one good application should be hard or impossible, but independent treatment of anyone could give notorious improvements to the whole image mining process.

III. Parallelism and Image Mining

Many issues of image mining can be optimized with different parallel techniques. Furthermore depending on tasks properties, different parallel paradigms could be applied in the same system. At a first glance, parallel applicant tasks will be: image storage, image processing, image indexing and retrieval and, pattern and knowledge discovery. In this section, the three main stages of an image mining system will be explained and then the feasibility of apply parallel paradigms at global and local level will be analized.

A. Processing Phase

Automatic image categorization involves experience on a real problem. The aim is to build a mining model using attributes extracted from and attached to the real problem, then evaluating the effectiveness of the model using new images. After the adquisition stage, the visual contents of the images in the database must be extracted and characterized by descriptive patterns -usually multidimensional feature vectors.

Orthogonal to challenges of developing specific image mining algorithms and models that operate on idiosyncrassy of images, one other major challenge for image mining is the pre-processing state previous to the extraction of relevant features. Generally, most of the images, if not all, are difficult to interpret, and a pre-processing phase is necessary to improve the quality of the images and make the feature extraction phase more reliable. The pre-processing state is arguably the most complex phase of the knowledge discovery process when dealing with images. If the pre-processing is well done, it can be decisive whether patterns could be discovered, or whether the discovered patterns could be interpreted at all. This phase often requires related expertise to computer vision, image processing, image interpretation, graphics and signal processing, domain knowledge or domain applications.

Pre-processing is always a necessity whenever the data to be mined is noisy, inconsistent or incomplete and it significantly improves the effectiveness of the data mining techniques. Nevertheless, a processing step is always done when applying any discovery technique for the extraction of relevant features. As a consequence, the process of building a mining model involves to split the processing phase into a pre-processing and extraction of visual features steps.

Figure III-A shows an overview of a categorization process. The first step is represented by the image acquisition and image enhancement, followed by feature extraction. The last one is the classification part, where different techniques for supervised learning are applied.

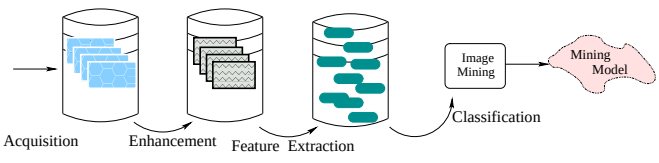


Figure 2: Image categorization process

The enhancement and feature extraction steps are usually refered as *Data Cleaning* and *Data Transformation* states that should be applied to the image collection. Data Cleaning is the process of

cleaning the data by removing noise or other aspects that could mislead the actual mining process. Image enhancement helps in qualitative improvement of the image and can be done either in the spatial domain or in the frequency domain.

The most common techniques applied for data cleaning are the tipical image processing techniques like smoothing and sharpening filters. All these techniques could be combined with respect to a specific application.

On the other hand, data transformation implies to get an image content descriptor by means of its visual and semantic content. Visual content can be very general or domain specific. *General visual content* refers to color, texture, shape, spatial relationships, etc.; while *Domain specific visual content* is application dependent and may involve domain knowledge.

A good visual content descriptor should be invariant to any accidental variance introduced by the imaging process. A visual content descriptor can be either global or local. A global descriptor uses the visual features of the whole image, whereas a local descriptor uses the visual features of *regions* or *objects* to describe the image content. Moreover, as a previous step to obtain the local visual descriptors, an image is often divided into parts. The simplest way of dividing an image is to use a *partition*, which cuts the image into tiles of equal size and shape. A simple partition does not generate perceptually meaningful regions but is a way of representing the global features of the image at a finer resolution. A better method is to divide the image into homogenous regions according to some criterion using *region segmentation* algorithms that have been extensively investigated in computer vision. A more complex way of dividing an image, is to undertake a complete *object segmentation* to obtain semantically meaningful objects (like ball, car, horse).

Some widely used techniques for extracting color, texture, shape and spatial relationships from images are: Color Moments, Color Histograms, Color Coherence, Color Correlogram, Gabor Filter, Tamura features, Wavelet Transform, Moment Invariant, Turning Angles, among others [10][19].

Semantic content could be obtained by textual annotation or by complex inference procedures based on visual content. We will not discuss in detail this topic, trying to focus on our subject.

B. Mining Phase

In the nontrivial process of knowledge discovery in databases (*KDD*), data mining in general, and image mining in particular, have the aim of extracting implicit knowledge from data. They try to define valid, novel, potentially useful, and ultimately understandable patterns, relations or rules from them. These relations draw a *Predictive* or *Descriptive* model. With a predictive model is possible to estimate future or unknown values of interest, while with a descriptive model is possible to identify patterns which explain or summarize the analized data. Mining tasks depend in the model to be applied. *Classification* or *Regression* techniques define predictable models, while *Association Rule Mining* or *Clustering*, among others, define descriptive models. Image mining refers to a set of methods dedicated to the extraction of hidden knowledge from within an assortment of images. The early image miners have adopted existing machine learning and data mining techniques to mine for image information. Very few achievements have been realized and the approaches can be grouped in two classes. Those that discover patterns from:

- Images in large collections using the processed and extracted features within images;
- The image database using general descriptors.

While the applications vary from creating suitable models for image indexing to recognizing objects, categorizing images or image segments, the general tasks are similar and can be summarized as grouping images or features, either supervised or unsupervised, and associating image features.

The techniques frequently used include object recognition, image indexing and retrieval, image classification and clustering, association rules mining and neural network or a combination [8], [18].

C. Interpretation and Evaluation Phase

This task is a crucial one, it is tightly related with mining phase because it measures the quality from obtained patterns. Model preciseness can be secured by guarantying data independency between the training data set and testing data set.

Different evaluation techniques and measures can be applied. Evaluation measures could be objective or subjective. Which one of them would be used will depend on the mining taks to be done. The

application context should be always considered when validating the obtained model [18], [13].

D. Global Parallel model

An image mining system(IMS) can be very computationally demanding due to the large amount of data to process, the response time required or the complexity of the involved image processing algorithms. Any parallel system requires dividing up the work so that processors can make useful progress toward a solution as fast as possible. The essential question is how to divide the labor.

There are three components to the work: computation, access to the data set, and communication among the processors [21]. These components are tightly related: dividing up the computation to make it faster creates more communication and often more data set accesses as well. Finding the best parallel algorithm requires carefully balance of the three named issues.

Parallelizing the image mining system showed at figure I involves to parallelize its three main areas: processing, mining and interpretation. Even though there exists a sequential line among them, it is a pseudo sequential line: after the first image set have been processed, the three named areas can be done in parallel.

Figure III-D shows the proposed global parallel arquitecture for the IMS. The model consists of four logical processing stages working in parallel: processing, mining, interpretation and coordination. The first three stages comes from the corresponding mining stages, and the last one from the derived management of the parallel model.

The coordinator process provides a GUI and task manager that directs the image mining process and is responsible for:

- At the starting of the mining process, it will coordinate the image mining tasks in a sequential way. First, the descriptor database generation through the processing phase, followed by the mining phase and finally the interpretation and evaluation phase.
- During the image mining process, it is responsible for the interaction with the image mining engine in terms of invoking, guiding and monitoring computations as well as visualization of the results.

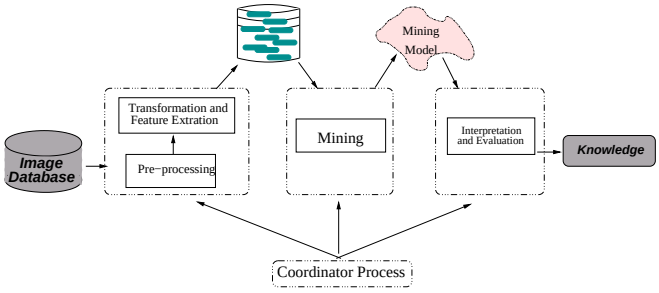


Figure 3: Global parallel arquitecture for the IMS

Different parallel stage relationships are done by data sharing. Processing and mining stages share the feature database, and mining and interpretation stages share the mining model. Because reading and writing data structure accesses are simultaneous, synchonization mechanisms are required [25].

E. Local Parallel Model

At a refined level, each global parallel stage could be resolved in a parallel way. As a first attempt, we will focus only on the processing stage. This section sketchs three parallel levels concerning different parallel programming models and grains that could be accomplish collaboratively. The parallel alternatives are presented in an increasing complexity parallel order.

1) Level 1: Embarrassingly Parallel: Applying the processing stage over the whole image set in the database could result in a time-expensive processing task. Besides this particularity, every internal data independence enables to draw a simple parallel model. The figure III-E1 shows the mentioned system architecture. N independent processes work on an image database partition, *DataBase_i* $(\forall i \quad 0 \leq i < N)$, making a feature data subset that will be joined to the whole working data set for the mining stage.

The parallel system has a coarse grain parallelism at data level following the $MDSP$ parallel programming model rules [2][15]. Moreover, as no particular effort is needed to segment the problem into a very large number of parallel tasks, and there is no essential dependency (or communication) between those parallel tasks, the problem is considered an embarrassingly parallel problem [25].

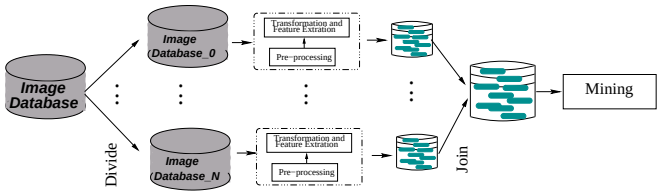


Figure 4: Level 1 Systems

As each step can be computed independently from every other step, they could be made to run on a separate processor to achieve quicker results. An a-priori system performance estimation points out that it could be optimal or cuasi optimal.

2) Level 2: Parallelism into Processing Stage: At previous section only incoming system data independence was considered. At this section parallelism inside the processing stage will be take into account. Inside processing stage, as feature extraction step must be done after pre-processing, a pipelined processing is proposed, see figure III-E2 [25]. The pipeline has two well defined steps, the first one for image enhacement and the following for image feature extraction. As a consequence, a stream of images is passed through a succession of processes, each of which perform one task.

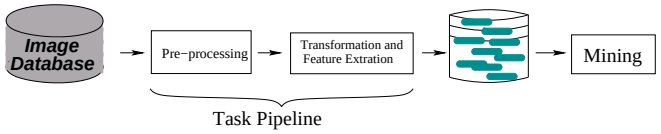


Figure 5: Level 2 Systems

An interesting point to be considered in a pipelined parallel computation is the work developed into every step. When tasks workload is the same for every step, the pipeline gets the best performance. Either the enhancement or feature extraction steps address different processing task that should be done in an specific sequence. Dividing the enhancement or feature extraction steps (or both) into ordered substeps, could lead to take advantage from the inner step parallel characteristics and diminishing processing time problems due overwork into each of them. The system shown at figure III-E2 arises from the concepts stated. It can be observed a tasks pipeline where every task could be any classic processing task or a set of them, depending on the workload.

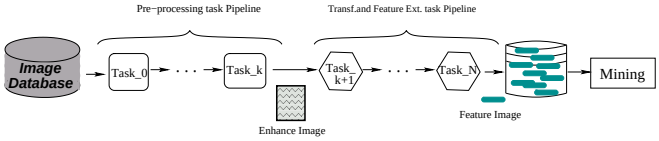


Figure 6: Refined Level 2 Systems

This last system has a finer parallel grain than the previous one where communications between tasks were increased.

3) Level 3: Parallelism Depending on Image Processing task: A sort of standard image processing tasks are commonly used at processing stage, like image smoothing, histogramming, 2-D FFT calculation, local area histogram equalization, local area, brightness and gain control, feature extraction, maximum likelihood classification, contextual statistical classification, image correlation (convolution, filtering), scene segmentation, clustering feature enhancement, rendering, etc. [6]. Many existing algorithmic implementions [3][5][7][9][11][12], could be done thru parallel solutions. Moreover, different techniques at different grain scale could be applied depending on the particular task; some of them are [4][22][23].

At this level any parallel model proposed not depends directly from the mining model itself, whereas it depends directly from any image processing task involved at the processing phase. As a consequence, any possible parallel model will be closely related to the specific image processing task to be done [10]; that is the reason because we do not suggest any model. The best solution could be to build a standard parallel image processing library that enables to make parallel processing at different combinations.

F. Some Results

Un analisis del sistema global (GPS) establece que su costo esta determinado por la etapa de mayor costo (introducir la formula de abajo)

$$cost(GPS) = cost(First\_Step) + max\{cost(Stage\ j)\}$$
$$\forall j\ 1..3$$

Dadas las caracteristicas de cada uno de los stages implicados en el sistema de mineria de imagenes, es posible aplicar otro nivel de paralelismo. El stage 1 es el primero en ser considerado por estara intrinsecamente relacionado con el procesamiento de las imagenes.

El primer modelo de paralelismo local propuesto (level 1), al ser embarransingly parallel, su costo es igual a aplicar el stage 1 a I/p imagenes, siendo I el nmero total de imagenes y p la cantidad de procesadores.

$$cost(L1S) = cost(stage\ 1(\frac{I}{P}))$$

Por otro lado, el costo del sistema de nivel 2 se deriva directamente de su estructura pipelineada, siendo:

$$cost(L2S) = cost(pipe\ fill) + max\{cost(Task\_j)\} \quad \forall j\ 0..N$$

Finalmente, Si bien los modelos previamente analizados optimizan al IMS, su costo puede ser sensiblemente mejorado aprovechando las caracteristicas propias de cada tarea involucrada en la etapa de preprocesamiento y extraccion de caracteristicas, y quizas, una combinacin de ellas. Es por ello que como primer paso en la implementacin de una propuesta integrada se resolvi comenzar paralelizando a un nivel ms refinado, es decir a Level 3. Inicialmente se consideraron algunas tareas bsicas de procesamiento de imgenes como la 2D convolucin(FFT2D), el clculo del histograma de color(CH), la equalizacin del histograma(CHE), la deteccin de bordes(ED) y el difuminado de la imagen(S). Los algoritmos resultantes siguen el paradigma farm donde P procesos comparten la imagen de entrada $N * N$ y las comunicaciones involucran la tranferencia de valores enteros. Por consiguiente, cada algoritmo posee un costo individual que puede indicarse de la siguiente manera:

$$\begin{aligned}
cost(FFT2D) &= 2cost(FFT1D(\frac{N}{P})) + 3(P-1)comm(\frac{N}{P}) \\
cost(CH) &= cost(histogram(\frac{N}{P})) + (P-1)comm(\#color) + \\
&\quad + cost(join(parcial\ histogram)) \\
cost(CHE) &= cost(CH) + (P-1)comm(1+\frac{N}{P}) + \\
&\quad + cost(equalization(\frac{N}{P})) \\
cost(ED) &= cost(edging(\frac{N}{P})) + (P-1)comm(\frac{N}{P}) \\
cost(S) &= cost(smoothing(\frac{N}{P}+2)) + (P-1)comm(\frac{N}{P})
\end{aligned} \tag{1}$$

Del conjunto de alogritmos seleccionados, puede observarse que la FFT2D es la tarea ms costosa y que todos los dems se mantienen sensiblemente inferiores. Se podria generalizar que el costo de la etapa de pre-procesamiento depende directamente del conjunto de tareas seleccionadas. Una propuesta mas refinada deberia intentar definir un valor invariante de costo computacional para esta etapa.

IV. Conclusion

An image mining system presents numerous aspects feasible to be parallelized enabling not only the use of different parallel paradigms but also a combination of them at different levels.

This paper presents some proposed optimization alternatives El artculo presenta diferentes alternativas de optimizacin de la performance de un sistema de mineria de imgenes, una de ellas relacionada al sistema en general y las otras a una etapa en particular, la etapa de pre-processing y extraccion de caracteristicas de las imgenes.

Integration of parallel techniques into the image mining process was analized. It was considered from two points of view: global or local processing (each task into the global processing). Local analysis was focused on processing stage and three parallel image mining system models were proposed. They consider different parallel paradigms and granularity.

At this time, a first total system attempt is been developed over a cluster of 15 nodes. The earlier predicted costs, besides they are general, are very promising to be refined.

References

[1] H. Krawczyk A. Mazurkiewicz. A parallel environment for image data mining. In *Proceedings of the International Conference on Parallel Computing in Electrical Engineering (PARELEC'02)*, 2002.

[2] A.Grama, A. Gupta, G. Karypis, and V. Kumar. *Introduction to Parallel Computing*. Addison Wesley, 2003.

[3] D. Ballard and C. Brown. *Computer Vission*. Prentice Hall, Englewood Cliffs, 1982.

[4] J. Barbosa and J. Tavares A. Padilha. Parallel image processing system on a cluster of personal computers. *Lecture Notes In Computer Science*, pages 439 – 452, 2000.

[5] S. Beucher and F. Meyer. The morphological approach to segmentation: the watershed transformation. *Mathematical morphology in image processing*, pages 433–481, 1993.

[6] A. Choudhary and S. Ranka. Parallel processing for computer vision and image understanding. *IEEE Computer*, 25(2):7–9, 1992.

[7] J. Crespo, J. Serra, and R. Schafer. Theoretical aspects of morphological filters by reconstruction. *Signal Processing*, 2(47):201–225, 1995.

[8] C. Djeraba. *Multimedia Mining, A highway to Intelligent Multimedia Documents*. Kluwer Academic Publishers, 2003.

[9] C. Giardina and E. Dougherty. *Morphological Methods in Image and Signal Processing*. Prentice Hall, 1988.

[10] R. Gonzalez and R. Woods. *Digital Image Processing, 2nd Edition*. Prentice Hall, 2002.

[11] B. Jahne. *Digital Image Processing: Concepts, Algorithms, and Scientific Applications*. Springer Verlag, 1997.

[12] A. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall, 1989.

[13] Y. Keiji. Managing images: Generic image classification using visual knowledge on the web. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 167–176, November 2003.

[14] R. Kosala and H. Blockeel. Web mining research: a survey. *ACM SIGKDD Explorations Newsletter*, 2(1):1–15, June 2000.

[15] L.Yang and M. Guo. *High Performance Computing: Paradigm and Infrastructure*. Wiley-Interscience, 2006.

[16] R. Missaoui and R. Palenichka. Effective image and video mining: an overview of model-based approaches. In *MDM '05: Proceedings of the 6th international workshop on Multimedia data mining*, pages 43–52, New York, NY, USA, 2005. ACM Press.

[17] T. Mitchell, R. Hutchinson, M. Just, R.S. Niculescu, F. Pereira, and X. Wang. Classifying instantaneous cognitive states from fmri data. In *Proc. 2003 American Medical Informatics Association Annual Symposium*, pages 465–469, 2003.

[18] H. Orallo, R. Quintana, and F. Ramirez. *Introduccion a la Mineria de Datos*. Prentice Hall, 2004.

[19] J. Parker. *Algorithms for Image Processing and Computer Vision*. J. Wiley & Sons, 1997.

[20] A. Selim, K. Krzysztof, T. Carsten, and M. Giovanni. Interactive training of advanced classifiers for mining remote sensing image archives. In *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 773–782, 2004. (Industry/government track posters).

[21] D. Skillicorn. Strategies for parallel data mining. *IEEE Concurrency*, pages 26–35, 1999.

[22] W. Rapf M. ReinhardtL T. Braunl, S. Feyrer. *Parallel Image Processing*. Prentice Hall, Englewood Cliffs, Berlin Heidelberg, 2001.

[23] M. A. Vorontsov. Parallel image processing based on an evolution equation with anisotropic gain: integrated optoelectronic architectures. *Optical Society of America*, (16):1623–1637, 1999.

[24] Y. Wang, F. Makedon, J. Ford, L. Shen, and D. Goldin. Image and video digital libraries: Generating fuzzy semantic metadata describing spatial relations from images using the r-histogram. In *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, pages 202–211, 2004.

[25] B. Wilkinson and M. Allen. *Parallel Programming: Tecniques and Aplications using Networked Workstations and Parallel Computers*. Prentice Hall, New Jersey, 1999.

[26] Ji Zhang, Wynne Hsu, and Mong Li Lee. Image mining: Trends and developments. *Journal of Intelligent Information Systems*, 19(1):7–23, 2002.

Efficient Service Discovery System Based on Semantic Overlay Networks

Shanshan Jiang and Finn Arve Aagesen

Department of Telematics, Norwegian University of Science and Technology
N-7491 Trondheim, Norway
Email: {ssjiang, finnarve}@item.ntnu.no

***Abstract*-A network of autonomous directories forming a large-scale distributed service discovery system is considered. In order to locate services efficiently in such large-scale systems, we propose to organize directories into Semantic Overlay Networks (SON) based on service ontology. The various SONs are further organized in super-peer networking structures. The focus is on the functionality for organization of directories into SONs and the use of SONs for efficient service discovery and efficient SON management. Efficient service discovery means high recall, small number of messages-per-request and small number of hops-per-request. Efficient SON management is characterized by small management procedure overhead, small load factor and short self-organization time. Simulations indicate that, compared to a system based on a random overlay network, such super-peer based SON system can achieve high recall with small hops-per-request value and significantly reduced messages-per-request value. Moreover, such system requires only a small management procedure overhead and the self-organization time is short both for SONs initial construction and reconstruction under dynamic node joining and leaving situations. Simulations also indicate a small average load factor.**

I. INTRODUCTION

In a distributed service environment, service discovery is a core functionality to locate desired services. *Service discovery* is the process of finding the desired services by matching *service descriptions* against *service requests*. A *service description* provides service-related information which can be advertised by a service provider and searched during service discovery process. A *service request* represents user's service requirements. Both service descriptions and requests comprise information on functional and non-functional properties. Ontologies are the basis for adding semantic expressiveness to service descriptions and requests. An *ontology* is an explicit and formal specification of a shared conceptualization [1]. A *service ontology* is accordingly an explicit and formal specification of core concepts of the functional and non-functional properties of service. An *upper ontology of service* is a model of common service-related concepts applicable to a wide range of domains. *Semantic service discovery* is a service discovery process based on ontology concepts. Likewise, *semantic matching* is the matching of service requests and service descriptions based on ontology concepts.

Current service discovery protocols, such as SLP [2], Jini [3] and UPnP [4], are targeted for enterprise and home networks. They either rely on centralized entities for discovery or search entire local network by broadcast or multicast, thus do not suit for wide area discovery. UDDI [5], the standard for Web service discovery, is based on centralized registries, also facing the scalability and performance challenges when the amount and variety of services increase. Hence, an important research objective is the location of services in a large-scale system with a huge amount of a large variety of services *efficiently, i.e. to answer service requests fast with low overhead*.

A large-scale service discovery system is considered, which consists of autonomous directories, where each directory has its own local registered service descriptions. We propose to apply semantic service discovery and to organize directories into *Semantic Overlay Networks* (SON). SON is a flexible network organization that logically connects nodes with semantically similar contents. The concept of SON is introduced in [6], which aims to improve query performance while maintaining a high degree of node autonomy. In this paper, a super-peer [7] based networking architecture is proposed for the various SONs. Super-peer architecture aims to combine the efficiency of a centralized approach and the scalability, load-balancing and robustness of a distributed approach. A super-peer architecture organizes nodes into hierarchy by utilizing the different capacities of nodes. Super-peers are nodes with high capacity. The super-peers and other nodes constitute the service discovery system. The super-peers have special assigned functionality in the service discovery procedure. They are dynamically selected by some super-peer selection algorithm and do accordingly not constitute single points of failure.

The focus of this paper is on the functionality for organization of directories into SONs and the use of SONs for efficient service discovery and efficient SON management. For the organization of directories into SONs, a shared service ontology based on a predefined service category hierarchy is used. The service descriptions are classified into service categories based on service ontology, and directories with semantically similar descriptions are grouped into SONs, where *semantically similar* means belonging to the same service category subtree. Each service request also identifies one service category. Service discovery based on SONs is an iterative process that can consist of several loops. Each loop has two steps. In Step 1, SONs which may contain relevant directories are selected. In Step 2, all the directories in these SONs are searched based on semantic matching. In each loop, the results are checked to see if enough number of matches is found. If not, more SONs are selected in new loops and until

enough number of matches is obtained. For SON management, a self-organizing process based on gossiping [8] is applied for the construction and maintenance of SONs.

The rest of the paper is organized as follows: Sect. II discusses the system requirements, while Sect. III presents a model for such SON-based service discovery system. Sect. IV describes the super-peer based SON system. Experiments and results are described in Sect. V. Related work is discussed in Sect. VI followed by conclusions in Sect. VII.

II. Requirements to an Efficient SON-based Service Discovery System

An efficient SON-based service discovery system must both be efficient with respect to *service discovery* and with respect to *SON management*. Efficient service discovery requires that relevant quality answers must be returned fast with small discovery procedure overhead. Efficient SON management means fast construction and reconstruction of SONs and small management procedure overhead.

With respect to the efficient service discovery requirement two design requirements can be defined.

- **R1:** *The system should enable high probability searching with high recall*.

The number of SONs selected for answering a request should be small and at the same time contain directories that have a high number of matches. This is because that if the directories to which the request is sent have many semantically similar service descriptions (i.e. many possible matches), the request is answered fast.

- **R2:** *The number of SONs and the size of SONs must be small*.

Smaller SONs and fewer SONs reduce management procedure overhead. At the same time, fewer messages are required for answering a request, so the request is answered faster. Reducing SON connections per directory can reduce SON size. Shared service ontology with a predefined service category hierarchy is assumed. It is however inefficient to build SONs for each category. Aggregation of services into groups is needed, i.e. service categories at appropriate hierarchy level should be selected according to the service distribution so that the number of SONs and the SON connections can be reduced.

III. SON-based Service Discovery System Model

A *service discovery system* Ψ can be modeled as $\Psi = <O, D, L, F>$, where O is the *service ontology*, D is a set of directories, L is a set of *links* that logically connects directories based on semantic similarity and F is a set of *functionality*.

The service ontology O defines the service categories as well as other service related concepts in a hierarchical structure. The upper ontology of service *s* is defined as a tuple *<c, p, op, qos, sp>*, where *c* denotes the *service category* the service *s* belongs to, *p* the *policies* applied to *s*, *op* the *operations s* performs, *qos* the *QoS parameters* and *sp* the *service parameters*. As shown in Fig. 1, a *service* belongs to a *service category*, and has *operations*, *policies*, *service parameters* and *QoS parameters*. Each operation is defined by *inputs*, *outputs*, *preconditions* and *effects*. The set of categories $\{c\}$ is connected by a rooted tree, the *category hierarchy* $\mathcal{CH}$. The role and the use of the service categories are explained at the end of this Section. More details of definitions of service ontology elements and their application in semantic service discovery can be found in our previous work [9].

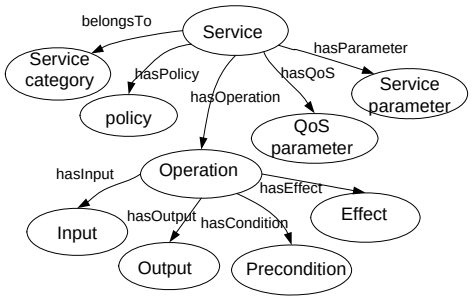


Fig. 1. Upper ontology of service.

D is the set of directories $\{d_i\}$. Each d_i maintains a set of *service descriptions* $SD_i = \{sd_j\}$. Each service description sd_j belongs to one service category *c*, denoted as $sd_j \rightarrow c$. Each d_i is *logically* connected with a set of directories, called its *neighbors* N_i.

L is a set of *links* $\{l\}$ that logically connects directories based on service categories. A *link l* is a triple $<d_i, d_j, c>$ where d_i and d_j are the logically connected directories based on a service category *c*. A link is symmetric, i.e., $<d_i, d_j, c>$ is the same as $<d_j, d_i, c>$. A *Semantic Overlay Network (SON)* is represented by the set of links with the same *c*, $SON_c = \{d_i, d_j \in D \mid \exists$ a link $l = <d_i, d_j, c>\}$. Each directory can join one or several SONs.

F is a set of *functionality* provided by the system, which includes seven operations as described as follows. Each SON_c supports three operations:

- $Join_c(d_i, c)$: a *directory* d_i is added to SON_c, which means to find a directory $d_j \in SON_c$ so that a link $<d_i, d_j, c>$ can be added to the system Ψ.
- $Search_c(r, c)$: the service request r is sent to all the directories in SON_c and a set of matches are returned.
- $Leave_c(d_i, c)$: a directory d_i leaves SON_c by simply dropping all the links d_i maintains.

In addition to the above three operations related to SONs, the service discovery system also supports the following operations:

- *Register*(sd_j, d_i, c): a service description sd_j is registered in a directory d_i with service category *c*.
- *Assign*(d_i): a directory d_i is assigned to a number of SONs according to the service descriptions registered. In addition, SON memberships can be updated according to runtime situation.

- *Discovery*(r, *c*): a service request r with a service category *c* specified by the user is sent to the system for discovery and the results are sent back to the user.
- *SONselection*(*c*): select the relevant SONs for a given service category *c*.

The role and the use of the service categories can be explained as follows. In service ontology, the category hierarchy $\mathcal{CH}$ is defined and is referenced by both users and service providers when they define service requests and descriptions. Each service description is mapped to one service category when it joins the system. Each service request contains one service category c_r which defines the range of services that the user is looking for. Each SON_c is related to one service category *c*, which is the root of the category subtree from $\mathcal{CH}$. Fig. 2 illustrates the category relations in a directory. The black circles represent categories related to service descriptions, while larger circles represent categories related to SONs. Each directory has a set of service categories from all the service descriptions registered in the directory (the set *ServiceDescriptions*). By aggregation of service descriptions according to their semantic relationships, a new set of categories (the set *Potential_SONs*) is obtained. This category set corresponds to potential SONs. Further applying join policies, a new set of categories of {*c'*} (the set *SONs*) is obtained (cf. Sect. IV.A). Each *c'* corresponds to one $SON_{c'}$ that the directory is member of. A directory joining $SON_{c'}$ implies that it has enough service descriptions that fall under the subtree *c'*.

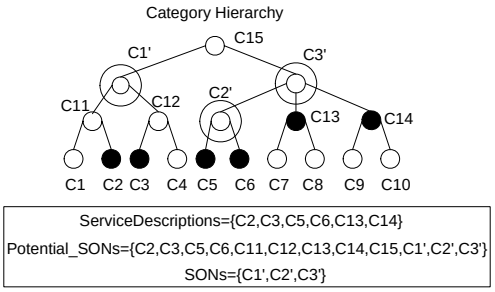


Fig. 2. Category set examples as seen from one directory.

IV. A Super-peer Based SON Service Discovery System

The architecture for a super-peer based SON service discovery system is illustrated in Fig. 3. Service descriptions are stored in directories, while service requests from a service discovery user are sent to the service discovery system via the *edge* directory. Directories are logically organized into SONs. Each directory can belong to several SONs. The *capacity* of a directory is characterized by the number of directories it can manage. Each SON has a *super-peer*, which is a directory with high capacity. Other directories in a SON are all connected with the super-peer and communicate via the super-peer. In addition, such super-peers are connected with each other for inter-SON communication. In other words, the super-peer can distribute messages within a SON as well as forward them to different SONs for global discovery. The inter-SON communication is basically broadcast-based, although other approaches can be applied, such as DHT (Distributed Hash Table) [10].

Each directory in the service discovery system can have one or several of the following roles: *service storage*, *edge* and *super-peer*. Each directory has at least the role of *service storage*, which has the following functionality: 1) Registering service descriptions, 2) Assigning the directory to SONs and adapting SON memberships to dynamic changes, and 3) Accepting service requests from a super-peer and carrying out local semantic matching procedure. The directory that a service discovery user contacts also assumes the *edge* role, which will 4) accept the service request from the service discovery user, forward it to its super-peer (can be the same directory), and return the answers to the user. Furthermore, each SON has a *super-peer*, which has the following functionality: 5) selecting relevant SONs for discovery, forwarding requests to other directories in the same SON and/or other relevant SONs accordingly, and returning results back to the edge. A sequence diagram showing the interactions between different roles during service discovery will be given in Sect. IV.C.

In addition, system functionality is needed for 6) SON super-peer network construction and maintenance, including directory joining and leaving.

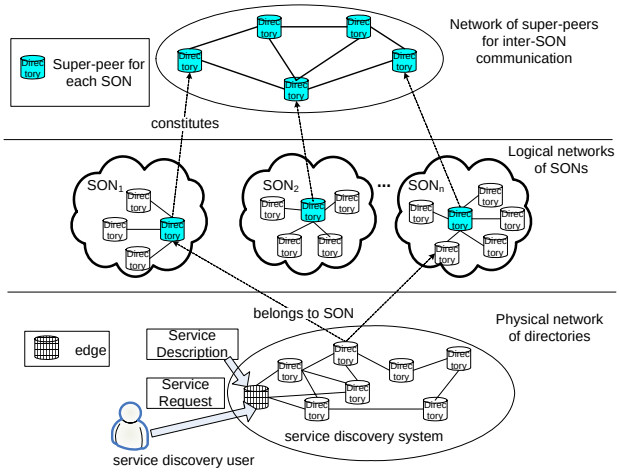


Fig. 3. Super-peer based SON service discovery system architecture.

Each service is registered in a local directory with a service category. Policies are applied when assigning directories to SONs. Service discovery requires the iterative selection of SONs and searching in each SON. In the following, assignment of directories to SONs, SONs construction and maintenance as well as service discovery are explained in more detail.

A. Assignment of Directories to SONs

To join the SON-based service discovery system, a directory needs to find out which SONs to join based on the service categories of the service descriptions contained in it. To meet requirement R2, a directory should first aggregate

service categories of all service descriptions stored in it into groups, and then select the best "representative" service categories to join SONs according to join policies. As services can dynamically join and leave the system, the services registered in the directories can change, causing SON membership to change. The join policies will thus not only be applied at startup, but also be applied dynamically for directories to adapt to runtime situation.

1. Aggregation

Aggregation is based on the semantic relationships between service categories. For two service categories c and c', $c' \leq c$ means c' equals to c or c' is a descendant of c in the service category hierarchy. To aggregate service descriptions, a service description sd will be placed in a group of service category c (denoted as $sd \Rightarrow c$) if ($sd \rightarrow c'$) and ($c' \leq c$). In other words, c is the root of subtree that c' belongs to.

2. Join Policies

Assignment of directories to SONs is based on the group information which is the result of the aggregation. In the following three join policies Join1-Join3 are defined. The selection of join policies is determined by the distribution of services. If services are clustered, each directory contains only a few service category groups. Both the number and size of SONs can be small if applying Policy Join1. On the other hand, if services are evenly distributed, Policy Join2 and Join3 need to be applied to reduce the SON connections.

Policy Join1: A directory d_i joins a SON_c if it has a service description that belongs to group c, i.e. ($d_i \in SON_c$) if ($\exists sd_j \Rightarrow c$, $sd_j \in d_i$). This is the most conservative policy, but it tends to produce many SON connections.

Policy Join2: A directory d_i joins a SON_c if the number of service descriptions that belongs to c is greater than the *join threshold* T, i.e. ($d_i \in SON_c$) if (Count($sd_j \Rightarrow c$) > T). Such policy is good and practical for wide-area service discovery. If a directory d_i joins a SON_c, this *implies* that d_i contains many service descriptions falling under the subtree c; therefore, selecting d_i for matching will provide higher number of matches than selecting the directories that are not members of SON_c. Policy Join2 thus meets the *high probability searching* requirement (R1). The join threshold T is an adjustable parameter. Policy Join1 can be viewed as setting T=0. The value of T is important for the performance of SON-based service discovery system. Taking a high T value, SON connections can be reduced, and accordingly SON sizes are reduced (R2). However, for small groups of service descriptions that are not assigned to SONs, there will be possibility that such descriptions never be searched if they are not associated with any SON. In this case, Policy Join3 can be applied.

Policy Join3: For service groups that do not have enough service descriptions, the directory will join a SON that corresponds to an ancestor of current service category.

B. SONs Construction and Maintenance

To meet requirement for efficient SON management, a self-organizing process is applied for both SONs construction and maintenance based on gossiping [8]. It is an autonomous functionality determined by directories themselves and enables the updates of topology information. In particular, failed or leaving nodes are automatically deleted by the absence of information exchange. Initially, all the directories in the system decide which SONs to join (i.e. select appropriate service categories) according to the function described in Sect. IV.A. For each SON, one directory with high capacity will be selected as super-peer, while other directories are *clients* of the super-peer. Each super-peer has a maximum capacity which determines the maximum number of directories it can manage. If a large number of directories decide to join SON_c, several super-peers may be needed. This means that there are several SONs corresponding to the same category c and each of them is managed by a super-peer.

A modified version of SG-1 algorithm [11] is adopted for super-peer selection. This algorithm is based on gossiping and assures the dynamic selection of super-peers according to runtime situation. Each node periodically exchanges its current status information (called *partial view*) with randomly selected peer nodes, which contains information such as identifier, capacity, neighbors, SON memberships, current role in a SON (super-peer or client), and the number of clients they are serving. After exchanging views, nodes with available connections and higher capacity will be changed into super-peers, and others are to connect to them as clients. Alternatively, a super-peer may decide to move all its clients to another super-peer with more capacity, and become a client itself. Such process continues until the system becomes stable, i.e., the minimum number of super-peers is selected. The super-peer selection process is also applied when directories join or leave the system (i.e. SON maintenance). In addition, each client periodically probes its super-peer to see if it is active. If the super-peer fails or leaves, the client can declare itself as a super-peer, and participate in the above super-peer selection process. For more details, please refer to our previous paper [12].

C. Service Discovery

When a request is sent to the service discovery system, the discovery operation *Discovery*(r, c) is carried out by an iterative procedure consisting of two steps in each loop.

1. Step 1: Select Relevant SONs

This corresponds to the operation *SONselection*(c). Given a service request with category c, the system needs to map it to SONs which may contain relevant service descriptions. The selection of relevant SONs is determined by the semantic distance between c and c_s (category for a candidate SON) in the $\mathcal{CH}$. An *iterative selection policy* is adopted, which selects SONs associated with c, and both its descendants and ancestors c' in an order that can enable *high probability searching* (R1). SONs most likely to answer the request are selected first to carry out Step 2 (cf. the following subsection). Then less probable SONs are selected until enough number of matches is obtained. In detail, given a request for category c, SON_c (if it exists) is searched first. If not enough number of matches is obtained, SONs that correspond to c's descendents

(if any) are searched. If still not enough number of matches returned, SONs that correspond to *c*'s ancestors are searched (if any, and from the nearest ancestor until the root of the $\mathcal{CH}$). Such process continues until enough number of matches is obtained or every possible SONs are searched. Since the maximum size of each SON is determined by the capacity of its super-peer, there may be several SONs corresponding to one category *c*. In this case, each such SON is selected until enough results are obtained.

2. Step 2: Search Each Directory in SON_c

This corresponds to the operation $Search_c(r, c)$. Since each SON is managed by a super-peer, the super-peer will forward the service request to each directory in the SON. Then semantic matching is carried out in each directory. This is based on other service concepts than service category in the service ontology, such as operations, service and QoS parameters. The semantic matching is based on semantic similarity between ontology concepts. The results of a semantic service discovery can be ranked according to functionality similarity based on degree of match, which can be distinguished as *Exact*, *Plugin*, *Subsume* and *Fail* [13]. A match between a service description *sd* and a service request *r* can then be modeled as a function $M(sd, r) = \{1, \text{if } degree \in (Exact, Plugin, Subsume) \mid 0, \text{if } Fail\}$. Our previous paper [9] proposed an approach for semantic matching which is based on service ontology defined in Fig. 1.

An UML sequence diagram for service discovery procedure is given in Fig. 4. The arrows with operation names represent operational messages, while arrows without names are the return messages, containing results for operational messages. The frame **loop** refers to a loop behavior, while frame **opt** means optional behavior. Fig. 4 shows that a service request is first sent from the service discovery user to the edge, then forwarded to super-peer1 (the super-peer of the edge, which can be the edge itself). The loop Discovery in super-peer1 consists of iterations of SON selection, search within the same SON as well as forwarding the request to corresponding SONs (super-peer2) for search. At the end of each loop, the results are checked to see if enough number of matches is obtained.

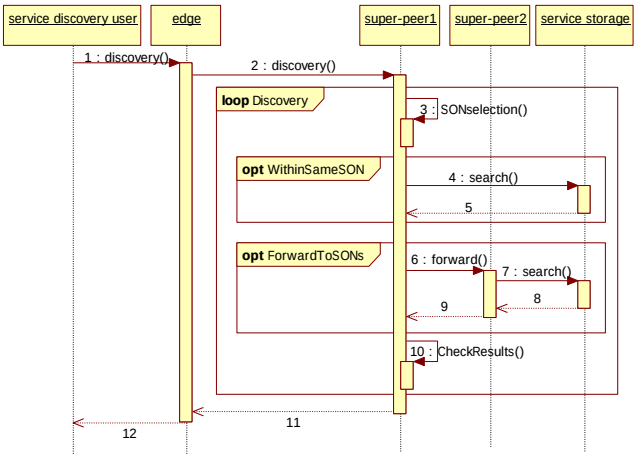


Fig. 4. UML sequence diagram for service discovery procedure.

V. Evaluation

A. Evaluation Measures

The service discovery system performance is characterized by: 1) *Service discovery efficiency* and 2) *SON management efficiency*. Measures for service discovery efficiency are: *recall*, *messages-per-request* and *hops-per-request*. Measures for SON management efficiency are *management procedure overhead*, *load factor* and *self-organization time*.

Recall of a discovery process is the proportion of relevant service descriptions that are retrieved out of all relevant service descriptions. Higher recall indicates better discovery quality. *Messages-per-request* is the number of messages generated by the discovery system to answer a service request. This represents discovery procedure overhead. The larger the messages-per-request, the more overhead and longer time it needs to answer a request. *Hops-per-request* is the average number of hops for answering a request. The smaller the hops-per-request, the faster the request can be answered.

Concerning SON management efficiency measures, the number of messages sent per directory per round in order to maintain super-peer based SON structures is applied as a measure for *management procedure overhead*. The *load factor* is defined as the value of *management procedure overhead* divided by directory *capacity*. A small load factor indicates a small work load for a directory to maintain the super-peer based SON structures. As a measure for *self-organization time*, the number of simulation rounds for the system to stabilize is used. A smaller number of rounds indicate that the system can stabilize faster.

B. Experiments

1. Service Discovery Efficiency

In these experiments, SONs are constructed first and then service discovery performance is evaluated in terms of recall, messages-per-request and hops-per-request. We have written a simulator in Java and used BRITE [14] to generate network topologies. Assume the classification of service descriptions and requests is correct and accurate, so that only one SON_c is selected for $Discovery(r, c_r)$. The network size varies from 1000 to 10000 nodes (directories), with 2500000 service descriptions falling into 250 service categories. 25000 service requests are randomly generated for each network size. Power-law distribution on node capacities is applied. The maximum capacity is 2000. Since the number and size of SONs are influenced by service distribution, two cases are simulated: 1) *uniform*, where service categories as well as service descriptions are uniformly distributed to directories. This represents the case when services are *evenly distributed*; 2) *power-law*, where the distribution of service categories on directories follows power-law and service descriptions are uniformly distributed to categories in the directories. This represents the case when services are *clustered*. For each case, 10 replications with different seeds have been conducted for each network size. The results of our SON-based system are compared with Gnutella-based [15] pure P2P system, where nodes are connected by a random overlay network and flooding is used for routing requests. Gnutella system is not

sensitive to service distribution. Its recall and messages-per-request are determined by TTL (time-to-live) parameter.

Fig. 5 shows the average number of messages-per-request for achieving recall of 1.0 for different network sizes. It shows that, to retrieve all relevant service descriptions, the average messages-per-request for Gnutella system is about 3 to 4 times of that needed for SON uniform system (i.e. SON system with uniform distribution), and about 7 to 46 times of that needed for SON power-law system (i.e. SON system with power-law distribution). SON uniform system needs about 3 to 10 times of the messages-per-request required for SON power-law system. As the network size increases, the overhead of Gnutella system in terms of messages-per-request increases much more than that of SON system. Assume the values of messages-per-request are normally distributed, the confidence intervals (CI) can be determined. Take an example, for SON uniform system with 1000 nodes, the average messages-per-request is 861.7, its standard deviation is 7.16, and the 95% confidence interval will be $856.58 \le CI \le 866.82$.

Fig. 6 shows the average messages-per-request for different recall for network size of 1000 and 2000. It shows that, to achieve the same recall (by adjusting TTL for Gnutella system and join threshold T for SON uniform system respectively), the average messages-per-request sent by Gnutella system is about 3 to 4 times of that needed for SON uniform system.

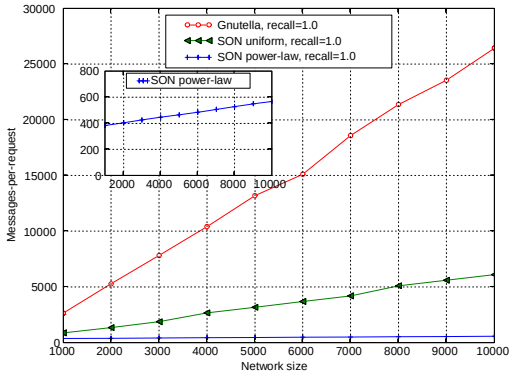


Fig. 5. The average number of messages-per-request for recall=1.0.

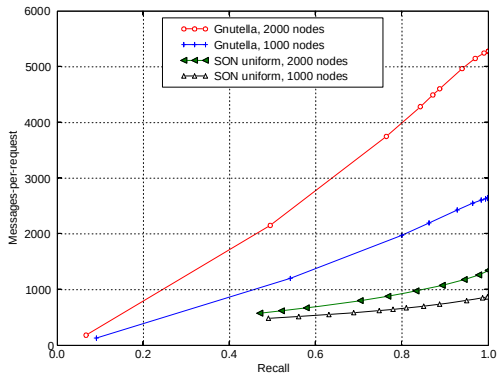


Fig. 6. The average number of messages-per-request for different recall for network size 1000 and 2000.

The number of hops for answering a request is determined by the hops to find the super-peer for SON_c, i.e. by the inter-SON communication. If flooding is used for inter-SON communication, the results from simulation give a hops-per-request value between 4 and 5 for different network sizes. This indicates a service request in SON system can be answered fast.

The size and number of SONs determine the messages-per-request for discovery. If the SON connections per node are reduced, the SON sizes as well as the messages-per-request will be reduced. When services are clustered, each node will have a small number of service groups and will have a few SON connections. Table 1 shows the effect of service distribution and join policies on system performance for network size of 2000. The applied policy will be Join1 when the adjustable join threshold parameter T=0 and Join2 with T>0 (cf. Sect. IV.A.2). When the threshold parameter T=7, a node will only join SON_c when it has more than 7 service descriptions belonging to group *c*. The SON connections per node and SON size will accordingly be changed by the variation of T. As a consequence, the service discovery efficiency measures will also change. Gnutella system is also listed for comparison and its recall and messages-per-request will be changed when adjusting TTL. As can be seen from the table, when applying Policy Join1 (i.e. T=0), the average SON size for SON uniform system is about half of the network size, while the average SON size for SON power-law system is about one-fiftieth of the network size. SON power-law system has smaller number of SON connections per node, and the messages-per-request is only about one-thirteenth that of Gnutella system (for same recall). SON uniform system has higher number of SON connections per node, and the messages-per-request is only about one-fourth of Gnutella system (for same recall). By applying Policy Join2 and adjusting T, SON connections are reduced, thus SON size and messages-per-request are reduced accordingly, as shown in Table 1 (SON Uniform, T=7). The recall is also reduced in this case; however, SON Uniform system needs only one-fourth of the messages-per-request value required by Gnutella system to achieve the same recall. The recall may be increased by iterative selection of SONs for discovery (e.g. SONs corresponding to an ancestor of category *c*), but the messages-per-request will also be increased.

TABLE 1. SOME SYSTEM PERFORMANCE MEASURES FOR NETWORK SIZE=2000.

SON system						
Distribution	T[1]	*SONs/node*[2]	*SON size*	*Recall*	*Msg*[3]	*Hops*[4]
Uniform	0	123	985	1.0	1349	4
Uniform	7	65	521	0.77	884	4
Power-law	0	5	42	1.0	405	4
Gnutella system						
TTL[5]				*Recall*	*Msg*[3]	
181				1.0	5281	
4				0.76	3751	

Note: 1. Join threshold parameter. 2. Average number of SON connections per node. 3. Messages-per-request. 4. Hops-per-request. 5. "Time-to-live" parameter.

The results indicate that when services are clustered, SON system can improve discovery efficiency with high recall and significantly reduced messages-per-request. Even when services are evenly distributed, SON system still has better

discovery performance than Gnutella system, requiring much smaller messages-per-request. Moreover, policies can be applied to ensure a few SON connections per node and keep SON size small, so that messages-per-request can be reduced. There is however a tradeoff between high recall and small messages-per-request when services are evenly distributed.

2. SON Management Efficiency

For these experiments, the PeerSim simulation framework [16] is used. To realize the $Search_c(r, c)$ and $Join_c(d_i, c)$ operations, a general super-peer based protocol for communication within individual SON is implemented and an existing implementation of Newscast – a gossip-based membership protocol [16] is used for inter-SON communication. Assume each directory joins one SON. SONs are constructed according to procedure defined in Sect. IV.B. Power-law distribution on directory capacities is applied. Simulation rounds for network sizes from 1000 to 50000 are carried out. To simulate dynamic behavior of the system, 100 nodes are deleted while 50 new nodes are added to the system in each round from round 30 to round 35.

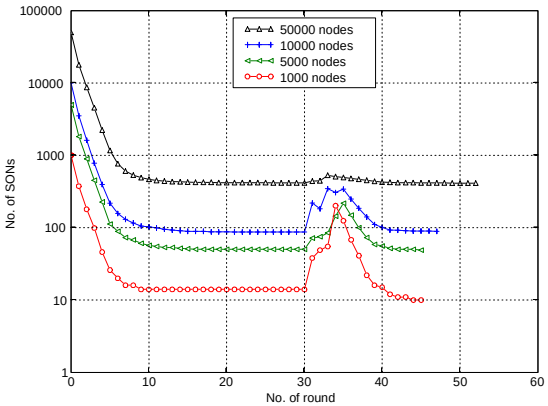


Fig. 7. Self-organization time (number of rounds) for different network sizes.

Fig. 7 shows that our proposed SON-based system becomes stable in approximately 10 rounds in both SONs initial construction and reconstruction under dynamic situation. Fig. 8 shows that average management procedure overhead is between 1 and 6. Fig. 9 shows that maximum management procedure overhead is usually below 15 for network size≤10000, and below 20 for network size=50000. Fig. 10 shows the maximum and average load factor for each round for network sizes of 1000 and 5000. The average load factor is below 1 and maximum load factor below 13. It also demonstrates strong correlation with Fig. 7 and indicates a very small value of load factor when the system is stable and a relative high value during the self-organizing process for initial SONs construction and reconstruction under dynamic situations. This can be explained as follows. Most of the management procedure overhead is related to the super-peer selection process. Initially, nodes with low capacity also participate in this process. This leads to a high load factor value. After one message exchange, such nodes will be changed into clients, and leave the selection process. Gradually, only nodes with higher capacity (potential super-peers) will participate in the selection process, reducing load factor (cf. Sect. IV.B).

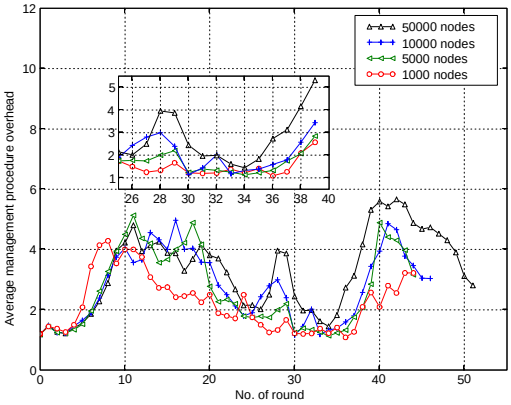


Fig. 8. Average management procedure overhead for different network sizes.

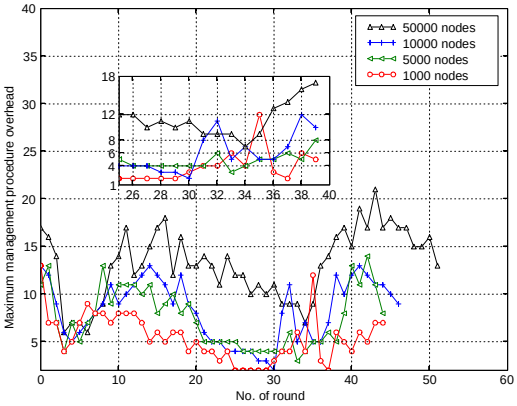


Fig. 9. Maximum management procedure overhead for different network sizes.

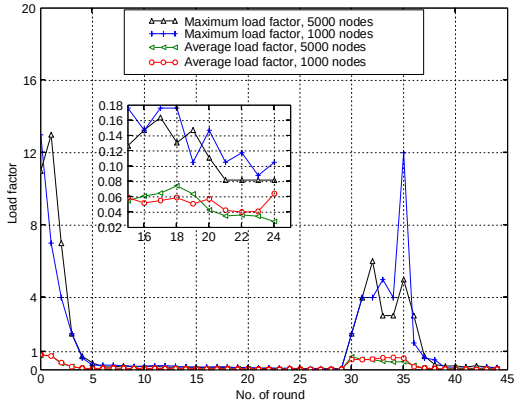


Fig. 10. Maximum and average load factor for network size of 1000 and 5000.

VI. Related Work

P2P-based service discovery systems have been proposed as promising solutions providing scalability and autonomy. Pure P2P-based service discovery systems [17] based on flooding for routing requests, do not rely on centralized entities and are well-suited for highly dynamic environments such as ad hoc

networks. However, flooding increases message overhead and is not suitable for wide area discovery.

DHT-based P2P systems [10][18] can locate relevant nodes with a bounded number of messages by applying the DHT algorithm. However, as DHT is based on a single key value, DHT-based system can not process complex queries in service discovery (e.g. based on ontology). Moreover, DHT techniques either require a specific network structure or assume total control over the location of the data. The high maintenance cost for DHT-based index and the lack of node autonomy limits the scalability and the application to wide area service discovery systems.

Lately several approaches have been proposed for creating and using clusters or SONs to improve search in P2P systems. SONs based on pure P2P systems (i.e. random overlay networks) have been proposed in [19][20]. In [21], clustering policies are proposed to generate semantic clusters in super-peer networks. Our approach differs by clustering directories based on service category hierarchy. Furthermore, our approach considers the adaptation of SONs, where the number of SON connections and SON membership can change dynamically by applying join policies on each directory at runtime. In addition, the super-peers in our system do not index data of their clients and are mainly used to route requests to proper SONs.

This paper adopts a modified version of SG-1 [11] for super-peer selection. A recent protocol and a natural evolution of the SG-1 algorithm is SG-2 [22], which introduces the notion of latency between peers and poses a QoS limit on it. In addition, SG-2 is strongly bio-inspired.

VII. Conclusions

Semantic Overlay Networks (SONs) have been proposed as the basis for improving the discovery efficiency in large-scale service discovery systems. The focus of the paper is on the functionality of the organization of directories into SONs and the use of SONs to locate services efficiently as well as efficient SON management. A super-peer based SON service discovery system is described. Aggregation and join policies are applied to ensure the SON size and the number of SONs can be small. The number of SON connections and SON membership can change dynamically by applying join policies at runtime. Iterative selection of SONs for discovery enables high probability searching. A self-organizing construction and maintenance of SONs ensures efficient SON management. The proposed system has been evaluated by simulations. Simulations indicate that such super-peer based SON system can significantly improve discovery efficiency by providing high recall with small messages-per-request and small hops-per-request. The simulations also indicate a small management procedure overhead and short self-organization time. It also indicates a small average load factor. The results, however, indicate that a SON system has better performance when services are clustered than when services are evenly distributed.

References

[1] Studer, R. et al. (1998). Knowledge engineering: principles and methods. *Data and Knowledge Engineering, 25(1-2)*:161–197, 1998.

[2] Guttman, E. et al. (1999). Service location protocol, version 2. RFC2608.

[3] Sun Microsystems. (2003). Jini architecture specification version 2.0. http://www.jini.org/

[4] UPnP Forum. (2000). UPnP device architecture version 1.0. http://www.upnp.org/

[5] UDDI. (2002). Universal description, discovery and integration of web services. version 1.0. http://www.uddi.org/

[6] Crespo, A., & Garcia-Molina, H. (2002). Semantic overlay networks for P2P systems. *Technical Report*, Computer Science Department, Stanford University, October 2002.

[7] Yang, B., & Garcia-Molina, H. (2003). Designing a super-peer network. In *Proc. of ICDE'03*.

[8] Demers, A. et al. (1987). Epidemic algorithms for replicated database maintenance. In *Proc. of POCD'87*.

[9] Jiang, S., & Aagesen, F.A. (2006). An approach to integrated semantic service discovery. In *Proc. of AN'06*.

[10] Stoica, I. et al. (2001). Chord: a scalable peer-to-peer lookup service for Internet applications. In *ACM SIGCOMM'01*.

[11] Montresor, A. (2004). A robust protocol for building superpeer overlay topologies. In *Proc. of P2P'04*.

[12] Jiang, S. et al. (2007). A self-organizing service discovery system based on semantic overlay networks. In *Proc. of CODS'07*.

[13] Paolucci, M. et al. (2002). Semantic matching of web services capabilities. In *Proc. of ISWC'02*.

[14] Medina, A. et al. (2001). BRITE: an approach to universal topology generation. In *Proc. of MASCOTS'01*.

[15] Gnutella. (2003). The Gnutella protocol specification v0.4.

[16] PeerSim. (2003). PeerSim: a peer-to-peer simulator. http://peersim.sourceforge.net/

[17] Paolucci, M. et al. (2003). Using daml-s for p2p discovery. In *Proc. of ICWS'03*.

[18] Rowstron, A., & Druschel, P. (2001). Pastry: scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Middleware, 2001*.

[19] Triantafillou, P. et al. (2003). Towards high performance peer-to-peer content and resource sharing systems. In *Proc. of CIDR'03*.

[20] Vazirgiannis, M. et al. (2006). Peer-to-peer clustering for semantic overlay network generation. In *Proc. of PRIS'06*.

[21] Loser, A. et al. (2003). Semantic overlay clusters within super-peer networks. In *Proc. of DBISP2P'03*.

[22] G. P. Jesi, A. Montresor and O. Babaoglu, Proximity-aware superpeer overlay topologies. In *Proc. of SelfMan'06*.

A Framework for Species Distribution Modeling: A performance evaluation approach

Jeferson Araujo, Pedro Luiz Pizzigati Correa, Antonio Mauro Saraiva
University of Sao Paulo
Sao Paulo, Brazil, 05508-900
jeferson@jefersonmartin.com.br, pedro.correa@usp.br, amsaraiv@usp.br

Abstract *- **Balancing social and economic development with environmental conservation is a major challenge. There is a strong demand for software applications to determine the fundamental ecological niche of organisms. Such tools can help us to understand the occurrence and distribution of biological species, such as invasive or endangered species. The openModeller framework was developed to provide a software environment for conducting such analyses.***

openModeller (oM) is an open source static spatial distribution modeling framework. Models are generated by an algorithm that receives as input a set of occurrence points (latitude/longitude) and a set of environmental layer files. One of several algorithms can be used to produce a model and generate a probability surface. The algorithms evaluate the environmental conditions at known occurrence sites and then compute the preferred ecological niche.

For optimizing the performance of oM a careful study of individual classes and the typical execution flow chart was conducted. In this study, the framework was divided into several components. A typical workload was defined with the help of end users. Preliminary results were produced that will be useful in identifying the components that can be processed in alternative parallel and distributed ways.

These results were also used to propose a performance analysis model. The aim of the performance analysis model is to aid the development of an infrastructure that will deliver optimal performance, maximum system availability, interoperability and that minimizes financial costs.

Aspect Oriented Programming and asynchronous message processing were used in order to identify the components causing bottlenecks. The results of this project will be used to design a parallel and distributed architecture to support oM analyses on a large scale (generating many thousands of models). On the other hand, the user interface application will be interoperable, allowing its utilization across different computer platforms. As the framework will be divided into components, and needs to be scalable, new functionalities can easily and efficiently be included.

Keywords: Performance evaluation techniques, measurement and analysis, performance prediction, Aspect Oriented Programming, high performance and distributed application.

I. INTRODUCTION

The openModeller [1] project aims to provide a flexible and user friendly desktop application on the client side and a high performance server side environment where the entire process of conducting a fundamental niche modeling experiment can be carried out. The software includes facilities for reading species occurrence and environmental data, selection of environmental layers on which the model should be based, creating a fundamental niche model and projecting the model into an environmental scenario. A number of fundamental niche modeling algorithms are provided as plug-ins, including GARP, Climate Space Model, Bioclimatic Envelopes, MinimumDistance and others.

To produce the model and generate the projected probability surface of species occurrences, the system uses computational resources in an intensive way. The modeling process is complex and demands a lot of processing and time. The use of several processing units and the parallelism concept is a solution for reducing the processing time. Clusters of computers have been used as scalable servers to get high performance [2]. In parallel computing, the first task is to identify the components which demand a lot of processing and could be parallelized.

Thus a detailed study of the system architecture is an important step towards configuring the computational resources for high performance.

One of the major problems in evaluating the performance of a monolithic application like openModeller is predicting the performance of each method to identify the bottlenecks. One approach to gathering these metrics is to divide the application into different functional components and then to host these components in a distributed architecture that implements inter process communication between the main process (the engine) and all the components. In this paper, several components were created, based on the existing libraries, and then hosted using Microsoft COM+ (Component Object Model) middleware architecture. The Goal of COM+ is to support the development

of components that can be dynamically activated and that can interact with each other [3]. COM+ itself is offered in the form of a library that is linked to a surrogate process specialized to accept a high volume of incoming inter process calls. Thus, from the main program, all the components are called in a distributed way, independently of their location. This technology, besides being a middleware platform for inter process communication, provides rich functionalities. For example, COM+ provides the interception mechanism that is essential for implementing the AOP (Aspect Oriented Programming) paradigm needed to instrument the openModeller components code. However, the COM+ mechanism by itself does not provide any support for performance analysis of component configurations. Instead, the AOP implementation provides performance analysis functionality.

Thus, a methodology was devised to enable analysis of the performance of component-based applications and to collect data from components. To implement this methodology, it is necessary to acquire performance parameters for the openModeller framework. The authors implemented a solution to profile the performance of each component and its methods, to monitor the code at runtime to find time consuming methods and to return metrics such as memory and processor consumption.

The two main techniques for evaluating application performance are sampling and instrumentation. The first approach usually maintains summary statistics of performance metrics, such as inclusive and exclusive time or hardware performance monitor counts [4], for routines in each thread of execution. Inclusive time means the amount of time spent in this function and all of its children. Exclusive time means the amount of time spent on this function alone. The function with the highest inclusive time is the function that has the most time between its entry and exit points, while the function with the highest exclusive time is the function that is actually doing most of the work. Performance evaluation tools either employ sampling of program state based on periodic interrupts or direct instrumentation. Sampling provides a fixed overhead and the accuracy of performance data generated depends on the inter-interrupt sampling interval. Here, direct measurement based on instrumentation techniques are considered, where instrumentation hooks are inserted at relevant program events. As the program executes, events or actions that take place in the program are inspected to characterize the program execution. In brief, sampling profiling captures program state by sampling at specified intervals (such as clock cycles or page faults) and instrumentation profiling inserts probes into the code to catch every single function call made by the program.

This problem domain is an ideal application for the concept of AOP described by [5]. The AOP approach, in conjunction with Object Oriented Programming techniques and asynchronous messages processing architecture approach, proved to be an efficient way to collect performance data for each component and method as these techniques offer control over the granularity of instrumentation and decrease the level of overhead due to instrumentation, as we will show in section V. Overall, we believe that bringing the full richness of an AOP language to application performance management allows us to implement a rich feature set, and offers a level of flexibility and customization that is harder to achieve with other approaches.

After running the application and recording performance metrics (referred to as *evidence* in this paper) the analysis of the evidence collected can be carried out without having to understand the work flow of the entire solution execution. Additionally, evidence can be analyzed on a per module bases removing the need to process enormous amount of performance information collected during the application execution. The other metrics, such as processor and memory utilization, was obtained from WMI API provided in the .NET Framework. The architecture and infra-structure prepared are detailed in sections V and VI.

A back office visualization tool was developed to allow the consolidation of the results collected. The visualization tool also enables the location of the most important performance problems that occurred during the execution of the application. After locating the candidate bottlenecks, the next step is to understand the causes of those performance problems. Typical causes are unnecessary memory allocation and extensive use of recurrence.

This paper provides an overview of openModeller system in Section III, explaining the workflow of a typical execution, from the reading of input data to building the projection. The performance evaluation process is detailed in Sec. IV, where were discussed the importance of collect performance evidences and the methods used, like AOP, WMI and message processing techniques. The infra-structure implemented to allow the performance evidence gathering and its analysis was presented in Sec. V. In section VI the models representation derived from the whole study is discussed. Finally, the Sec. VII concludes this work.

II. RELATED WORKS

There has been a significant amount of research in the area of computer systems performance analysis. The use of AOP as an instrumentation technique was studied by [6]. Davies et.al studied the capability of the AspectJ toolkit to analyze code, to explore caching opportunities and to improve application performance. But in the case of openModeller system, the data are always generated dynamically, so the use of cache to store results for prior computation is not a good strategy.

In reference [7] the authors discuss their experiences in using AOP in the field of application performance management that were gained during their development of an open-source J2EE performance management tool called InfraRED. According to the authors, application performance management (APM) is a classic crosscutting concern that is well suited to the use of AOP. There are several requirements for an APM tool such as basic timing and usage information, separating statistics by layers, and tracing remote calls, and

gathering statistics about the persistence tier, where AOP can be used as an effective implementation technique.

Reference [8] describes a methodology for predicting the performance of component-based applications, studying the characteristics of each assembly (in Microsoft Framework .NET, the term *assembly* is used to define a component that was developed using the .NET technology), building assembly performance profiles and connections to help in predicting performance. This methodology can be very useful only in early phases of development: at the conception or design stage of a process development life cycle and to measure performance against the required performance desired by the software architects and the system users. The openModeller architecture is still in development, so this first performance study was useful to get some ideas and to follow the recommendations to propose the new architecture and infrastructure for the openModeller system.

Research by [9] has shown that AOP does not necessarily introduce severe performance penalties. Experimental results were obtained from a benchmark, comparing code with no aspects against code with AOP techniques. The overhead with the introduction of aspects in component based software is acceptable and, in this way, can be used to collect logging information such as performance data.

To accurately answer the openModeller performance issues, we proposed a methodology which combines the AOP techniques to instrument the code and to collect the performance evidences with a multithreaded model to decrease the overhead introduced by AOP instrumentation and its interceptions. Although this work presents the openModeller as case study to apply the methodology and to validate our approach, the techniques can be used by others open sources projects to support the decision making in order to reach the highest scalable architecture and to answer questions about performance problems.

III. Framework OpenModeller: An Overview

openModeller is a fundamental ecological niche modeling framework that provides a uniform method for modeling distribution patterns using a variety of modeling algorithms. A number of fundamental niche modeling algorithms are provided as plug-ins, including GARP, MinimumDistance, Climate Space Model, Bioclimatic Envelopes, and others.

The openModeller framework provides an interface that aims at a flexible, user friendly, cross-platform environment where the entire process of conducting a fundamental niche modeling experiment can be carried out. The software includes facilities for reading species occurrence and environmental data, selection of environmental layers on which the model should be based, creating a fundamental niche model and projecting the model into an environmental scenario.

Fig. 1: The steps to build the probability distribution

Ecological niche modeling is typically a two-stage process. In the first stage, a model is constructed. In order to derive the ecological niche model, openModeller requires as input a set of points of species occurrence (latitude/longitude) for a given region. A second input required is a collection of one or more environmental layers (in the form of raster surfaces). Typically, these rasters will contain climatic variables such as temperature and rainfall. Abiotic rasters can also be used – for example representing topography or soil properties. The environmental layers are sampled at each site where an occurrence for the species under study has been observed. These environmental samples are then passed to an algorithm (for example Bioclim) which uses the samples to construct a model that describes the environmental preference or 'niche' of the species. In the second stage, this model is 'projected' into any geographic region containing a compatible set of environmental variables. The output from this process is a new raster layer, where each cell in the raster is assigned a value that represents the probability of the species occurring there.

The process of using the AOP profiling techniques follows the same steps described for the niche modeling. The researcher must:

1. provide a set of occurrence points for the species;
2. provide a set of environmental layers (rainfall, temperature etc.);
3. choose an algorithm to be used to construct the niche model, and select appropriate parameters for the algorithm;
4. generate the ecological niche;
5. use the generated model to calculate a probability of occurrence surface by projecting the model into a set of set of environmental layers for a given region.

The probability surface generated by openModeller is a gray-scale TIFF or Erdas Imagine '.img' file. The probabilities are expressed either as scaled integer values in the range [0,255] or as floating point numbers in the range [0, 1].

Thus, the architecture of openModeller can be described by the following items:

- A controller application: This could be a graphical user interface, a command line application or similar application that invokes operations on the openModeller library.
- openModeller Library: handles input / output of model data, sampling of environmental values at species occurrence points and delegates the task of constructing the model to the algorithm plug-in;

- Algorithms: implement a standard application program interface (API) and contain logic for constructing an ecological niche model and returning the probability of occurrence given a set of environmental conditions. Auxiliary libraries:

GDAL: Provides low level IO support for raster layers;

EXPAT: An XML parser for reading configuration files;

GSL: Mathematical library used the Climate Space Model algorithm.

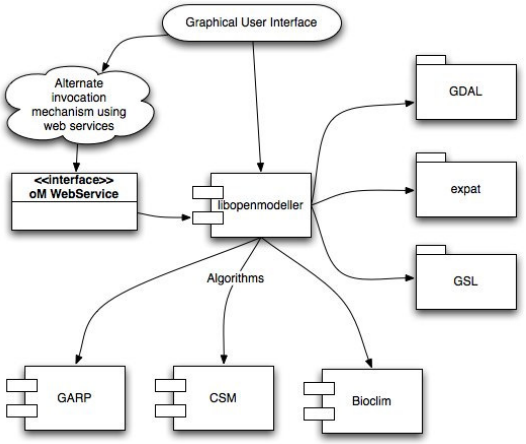


Fig. 2 – Architecture of *openModeller*

In certain cases the process model creation and projection can take several days of processing time. In many cases, the scientists and researchers need to build the projection using different algorithms to compare the results.

For example, to model the potential distribution of the Maned Wolf (*Chrysocyon brachyurus)*) in South America, a text file containing points of known occurrence was created. A data set of 13 environmental layers (maximum and minimum temperature, altitude, rainfall, etc) for the region was prepared.

#especie	long	lat
lobo guara	-38.5	-11
lobo guara	-50.083333	-29.166667
lobo guara	-47.866425	-21.598806
lobo guara	-47.776944	-21.587222
lobo guara	-50.004722	-25.235833
lobo guara	-45.847275	-23.229692
lobo guara	-47.550000	-15.533333
lobo guara	-45.867061	-22.978731
lobo guara	-45.500831	-22.853933
lobo guara	-45.874600	-22.988861
lobo guara	-45.933333	-14.233333
lobo guara	-47.500000	-14.000000
lobo guara	-46.500000	-15.833333
lobo guara	-52.750000	-18.316667
lobo guara	-49.650000	-24.166667
lobo guara	-42.283333	-22.516667
lobo guara	-55.613183	-19.506717
lobo guara	-47.933333	-15.950000
lobo guara	-45.833333	-15.050000
lobo guara	-43.326389	-21.864722
lobo guara	-55.833333	-15.2
lobo guara	-46.172222	-17.920556
lobo guara	-53.3	-18.083333
lobo guara	-47.907325	-21.553378
lobo guara	-54.016667	-23.433333
lobo guara	-43.583333	-19.266667
lobo guara	-45.867717	-23.038547
lobo guara	-46.526089	-20.236842
lobo guara	-43.883333	-21.7
lobo guara	-47.633333	-21.666667
lobo guara	-47.666667	-21.083333
lobo guara	-43.483333	-20.083333
lobo guara	-47.816667	-22.25
lobo guara	-48.75	-23.516667
lobo guara	-48.916667	-19.2
lobo guara	-46.9406	-19.5933
lobo guara	-47.95	-23.7
lobo guara	-50.013883	-25.429528
lobo guara	-47.979536	-15.710736

Fig. 3: Occurrence points used to create an ecological niche model for the Maned Wolf (*Chrysocyon brachyurus*)

Once the input data was prepared, the next step was the algorithm selection stage. The Climate Space Model (CSM) algorithm was chosen, with its parameters left at their default values to create the model. The CSM algorithm is a principle components analysis based approach to distribution modeling.

The result of the model creation process was an XML file containing the model definition, and .tiff image representing coverage of probability of occurrence for the Maned Wolf.

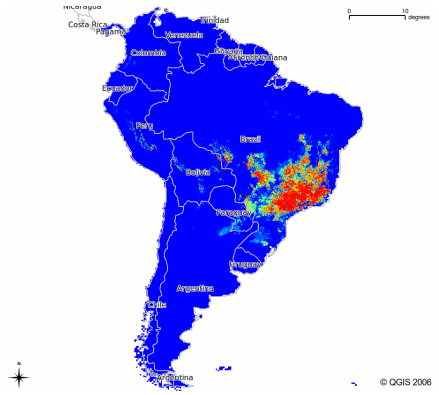


Fig. 4: The model projection rendered as a probability surface for the Maned Wolf (*Chrysocyon brachyurus*). Red indicates more suitable areas the species could potentially inhabit. Blue indicates less suitable areas.

In this simple example, openModeller took approximately one hour to generate the model and to create the projection. Using the same set of occurrence data, but with 19 environmental layers and using the GARP algorithm, the system took more than 5 hours to present the results.

In this context, it can be seen that building a performance profile and identifying parts of the code that can be optimized can be a useful way to reduce wait times for model computation. In order to minimize this execution time to enable analysis which demands a lot of processing time, as well as analysis using several algorithms and distinct set of layers, the computing system should provide the following features:

- Simultaneous processing of multiple jobs to provide information for analysis.
- Exploitation of the parallelism in modeling algorithms as well as in the projection of the model on any geographical areas.

So, in order to provide the requirements, the execution platform is a cluster of computers where each computer contains dual dual-processors.

In the next section, there is a description of how the work went about building the performance evaluation framework.

IV. The Performance Evaluation Process

Prior to starting the performance evaluation process, a detailed study of the openModeller execution flow was carried out. The scientists and researchers that actually use the system were interviewed. They were asked to identify time consuming activities and to provide some examples of workload that they

usually submit to the system. The user interviews provided important domain context while proposing a new openModeller system architecture that could meet the requirements of scientists and researchers.

The first step in conducting a performance evaluation was to divide the openModeller Framework into separate components (based on the libraries already used by the monolithic openModeller architecture). These components were then each installed and configured within a COM+ environment host. Following this component distribution and configuration, the code was instrumented using AOP techniques. This provided a detailed break down of the length of time that each method in each component ran for and thus, a preliminary estimate indicating which part of a submitted job takes the most time.

In addition to time based metrics, the AOP techniques used the resources from the WMI API in order to capture other metrics and application statistics. The Windows Management Instrumentation (WMI) API is a powerful and sophisticated instrumentation mechanism provided by the Microsoft .NET framework. These additional metrics included processor and memory utilization and the amount of traffic exchanged (the throughput) between the components. WMI provides consistent programmatic access to management information in the application. Thus it was possible to use a single consistent, standards-based, extensible and object-oriented model to make the openModeller system more manageable. By using the WMI object model, it was possible to discover and traverse the relationships between managed objects. Using the Windows administrative performance tool, some metrics were collected and persisted in to posterior analysis.

The performance evaluation process used real data for its workload. This approach ensured that the samples used in the performance evaluation process are a reliable indicator reflecting the real environment where the application will be used. At least three real workloads for each modeling algorithm were tested to be sure that all parameters and arguments that affect performance were considered.

A) The use of AOP for collecting performance evidence

The AOP techniques were used because no severe performance penalties were introduced in the openModeller application. Through AOP techniques it is possible to intercept, for instance, a method call and get the time that this method takes to execute.

AOP was developed to capture the concerns that are not captured by traditional programming techniques, like object orientated programming (OOP). A 'concern' is a goal, concept or area of interest. When a concern affects multiple implementation modules throughout a program, it is called a 'crosscut'. AOP provides a way to place the code of these crosscuts in a single place (for instance, a specialized logging class). A 'join point' is a defined point in the program flow. A join point is generally a method call or a method execution. A 'point cut' isolates one or more join points and retrieves run time values, like arguments, return values and thrown exceptions. 'Advice' provides a mechanism to define additional behaviour and is used to implement the crosscutting behavior. Advice brings together a pointcut and a piece of code, which runs at each of the join points defined by the pointcut. Most AOP suites have the following kinds of advice: 'before', 'after', 'after returning', 'after throwing' and 'around'.

In this research, a timer variable to zero is set before a method execution, the timer is started and then the counter is stopped when the method returns. The timer result, the component and method name are then passed as arguments to a specific method in the aspect.

An aspect is a container for all pointcuts and advice that implement a single crosscut. An aspect can usually be defined as a class, so it can have methods and fields in addition to the pointcuts and advice.

In the aspect class, a method was implemented to generate a message with the information received (the arguments). This method also instantiates the WMI objects that will monitor the computational resources consumed. A method in the aspect class is responsible for putting the generated message in a message queue application asynchronously (MSMQ). Another process is responsible for polling the queue of the MSMQ, capture the messages and persisting them in a database system. A back office application was developed to query the database and report on the execution flow with each method's duration. To analyze the resource consumption, the Perfmon snap-in (provided in most of Windows operating systems) was used.

With this information, it could be determined where the program execution takes the most time and research a way to optimize or parallelize the code. Through the evidence collected by the WMI instrumentation, a code review should be performed to avoid unnecessary memory use, avoid excessive round trip communication between the components and, based on the CPU utilization, define an ideal architecture and configuration to distribute the components among the servers.

Furthermore, as these libraries were divided into different components and those components hosted in a container to allow inter process communication, the throughput in megabytes exchanged between the components and the main application was monitored. This information was very important in the decision-making process about the distributed infrastructure of the components and the servers.

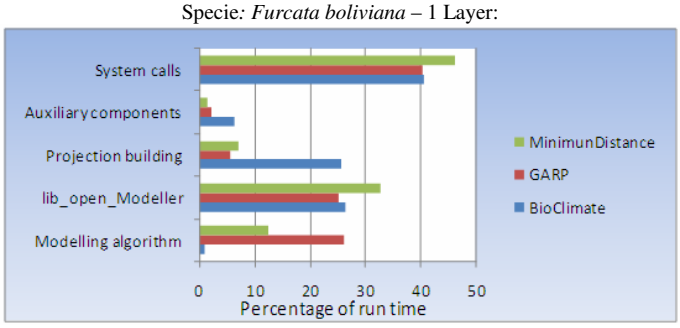
B) Presentation of collected evidence

One of the most important steps in every performance evaluation study is the way the final results are presented to the analysts. The openModeller back office application was developed to allow the analysts to consolidate the data collected in a structured way. The application queries the database where the performance data were persisted and then presents those data in a way that represents the workflow of the openModeller execution.

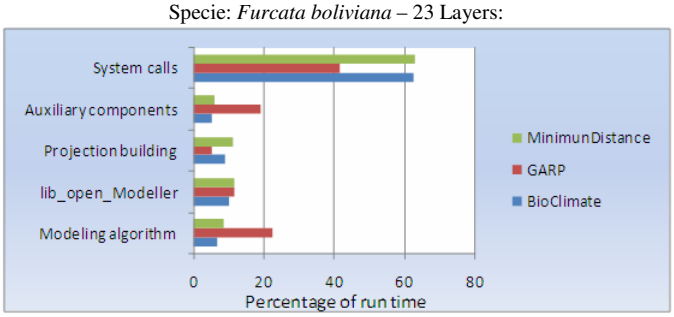
Based on the evidences collected, five categories (System calls, Auxiliary components, Projection building, lib_openModeller library and Modeling algorithm) were created with the purpose of discovering and dividing the bottlenecks in the program workflow. The next 3 figures show

the results analyzed and point the categories with high execution time:

Specie*: Furcata boliviana* – 1 Layer:



Specie: *Furcata boliviana* – 23 Layers:



Specie: *Furcata boliviana* – 41 Layers:

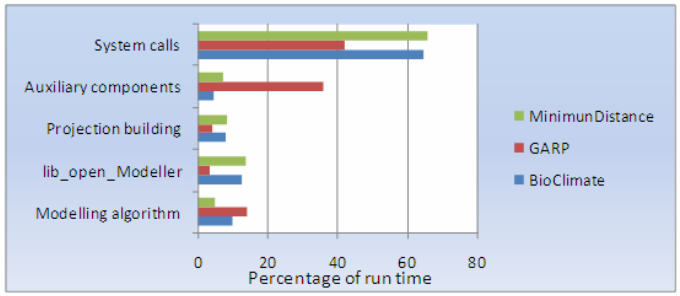


Fig. 5: The figures show where the openModeller whole process execution takes more time to be processed

After the knowledge of the results provided by this back office application, the methods with high execution time were studied one by one to inspect which approach can be applied to increase the whole application performance: parallelism, distribution between the cluster nodes, code optimization, etc. In case of parallelism approach, we concluded that GARP algorithm is a good choice to be parallelized due to its highest percentage of run time when compared with others modeling algorithms, as showed in the figures. The new resulting algorithm, the P-GARP (Parallel GARP), and its performance comparison is presented in [10].

V. Infra-structure Implemented

To minimize the impact of performance penalties with the introduction of instrumentation, an asynchronous message processing approach was implemented.

In this way, the user starts the process, and as the component methods are called, the AOP class implementations are called in a separate thread. A message is built containing the time consumed by the method. The WMI API is then called to profile the metrics used and, in sequence, puts this message in the Microsoft Message Queue. Parallel to this execution thread, the main thread, which is executing the openModeller algorithm, continues processing normally.

A Windows Service application (known as an NT service in Windows systems) was developed to catch the messages stored in a private queue at the MSMQ and persist them in a database for further analysis. The service continuously polls the message queue, gathering the messages in 5-minute intervals. The polling interval can be configured with an XML file. Thus, the messages generated by the AOP instrumentation are stored at the Message Queue and stay there until the service runs and retrieves the messages to be persisted. To avoid concurrence for computational resources between the openModeller system and the infra-structure process, like MSMQ and the LogService, both these services were configured to run in the background; in the other words, they were configured to run with less priority than the openModeller system.

The next figure shows the workflow of the architecture and infra-structure implemented to enable the gathering of evidence related to the performance evaluation process. To better understand the whole process, a brief explanation of each step, from 1 to 5, is provided below.

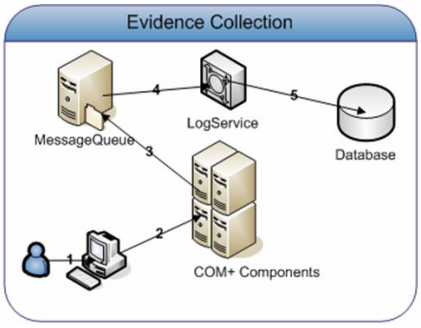


Fig. 6: The architecture and infrastructure designed and implemented for the openModeller performance evaluation process

The performance evaluation process consists of 5 steps, described bellow:

- Step 1: The user submits the input data to the openModeller engine installed in his desktop;
- Step 2: The openModeller engine, that is a console application, invokes the components installed at the COM+ container, to build the model and the projection. As the components process the input data provided by the user, these components return a progress status of the computation. This status is provided to the user in his screen;
- Step 3: Each method of the components hosted at the COM+ are intercepted by the COM+ mechanism and each crosscut is created by the AOP mechanism implemented in a different

processing thread. The AOP class implementations collect the metrics and synchronize with the WMI provider. In sequence, a message is generated with evidences and is sent to the MSMQ engine, to be stored in a private queue;

- Step 4: The Windows service, in this paper called LogService, polls the queue according to the configuration specified and retrieves the messages that were stored there. This processing is always executed in low priority, to avoid interference from the evaluation process and to avoid using excessive computational resources.
- Step 5: The process keeps running and catching the messages for the time specified in the configuration. After this time has elapsed, the LogService sends these messages to the database system to be persisted in an appropriate table. After all the messages are sent to the database, the LogService comes back to idle status.

In order to show the efficiency of this new technology devised, two experiments were tested. Both experiments used three modeling algorithms (BioClimate, GARP and MinimunDistance) with 23 layers as arguments. The specie studied was *Furcata boliviana.* The execution time of the first experiment (only the oM code) was compared with the execution time resulting from the other experiment, where we introduced the instrumentation code to get the evidences. The next figure presents the differences between two experiments and show that there is no great impact in the whole application execution time when the instrumentation code was inserted.

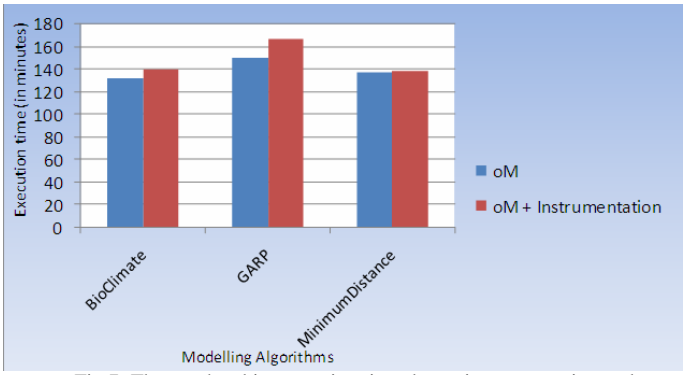


Fig.7: The overhead in execution time due to instrumentation code

VI. The Simulation Model Of The Distributed Architecture

The main idea of this model is to allow the simulation of the system and analyze different component system configurations over the high performance infra-structure. This model is a tool for support decision to define the best distribution of the components of openModeller and identify possible bottlenecks.

To define this model, an abstraction of the main components of the system that will use the high performance infra-structure was necessary.

The analysis of collected data allowed the building of a prototype of a distributed architecture to make openModeller more scalable and efficient. In this prototype, the researchers and scientists no longer need to install the whole openModeller, including the libopenmodeller library and the auxiliary libraries in their desktop, in order to build and project the model. Only the user interface and a simple component to read, parse and verify the consistency of the arguments are needed. This simple component also submits the necessary input data to the openModeller system, which is now a remote service, hosted in an infra-structure dedicated to processing openModeller requests. These remote servers provide more powerful computational resources than the user's desktop.

On the server side, architecture with two layers was proposed. The first layer consists of the Web Server with a web service proxy application and a component (represented as the message switch in the logical diagram above) responsible for deciding which modeling component should be invoked. In this layer, the web services security approaches, such as authentication and authorization, were out of scope.

The second layer is the core of the openModeller request processing subsystem and can be composed of several servers hosting different modeling algorithms. This layer represents the main module of the architecture.

The following diagram represents a logical division of the openModeller system in a distributed way:

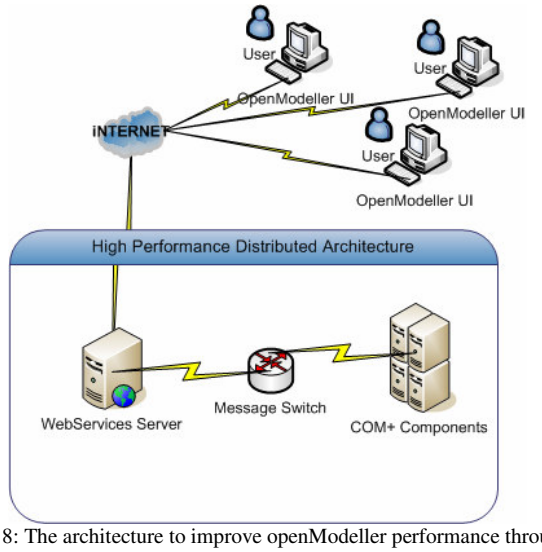


Fig. 8: The architecture to improve openModeller performance through componentization and processing distribution

Based on this architecture, this research represents the simulation model of the software components (see Figure 9). This model is based on closed network queues, where each queue represents a software component of the system. Following this model, the users send requests from Desktop openModeller to components in the High Performance Distributed Architecture (HPDA) (1) and receive responses from the HPDA (2). The HPDA server decides which

component will be requested, that could be an algorithmic request, projections or responses to the users.

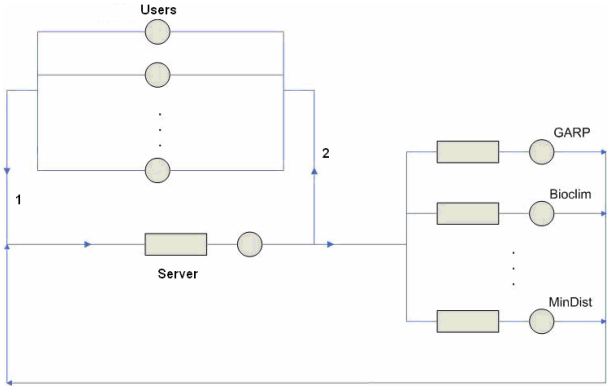


Fig. 9: Simulation model representation of openModeller System.

VII. CONCLUSION

This research was very useful to provide the openModeller application with architecture and infra-structure highly scalable and available and in agreement with the biological researches and scientists necessities, through parallel and distributed processing. Thus, these architecture and infra-structure implemented allowed simultaneous processing of multiple jobs and parallelism in the GARP modeling algorithm, implemented in [10].

The AOP technique to instrument the code has proved to be an efficient and original way to collect performance data as it could control the granularity of data collected through interception mechanism provided by the COM+. The asynchronous message processing used in the AOP implementation decreased the interference of this instrumentation in the end results. The application back office helped the analysts to quickly find the bottlenecks in the systems and to propose a way to turn the code more efficient.

References

[1] SOURCE FORGE. Project OpenModeller homepage. <http://openmodeller.sourceforge.net>. – Verified at 09/30/2007

[2] Buyya, R. 1999. *High Performance Cluster Computing: Architectures and Systems*. Prentice-Hall.

[3] Tanenbaum, A. S., van Steen, M.,*Distributed Systems – Principle and Paradigms*. Prentice Hall, Upper Saddle River, New Jersey, pp 526. 2002.

[4] Browne, S., Dongarra, J., Garner, N., Ho, G., Mucci, P., 2000. *A Portable Programming for Performance Evaluation on Modern Processors*. International Journal of High Performance Computing Applications. Volue 14, 189-204.

[5] Elrad, T., Filman, R. E., Bader, A.*Aspect-oriented programming*: Introduction. Communications of the ACM. Volume 44 Issue 10, 29-32. 2001

[6] Davies, J., Hiusmans, N., Slaney R., Whiting S., Webster, M., Berry, Robert., 2003. *An Aspect Oriented Performance Analysis Environment*. International Conference on Aspect Oriented Software.

[7] Govindraj, K. et al. *On Using AOP for Application Performance Management*. 5th International Conference on Aspect-Oriented Software Development. Bonn, Germany, 2006.

[8] Dumitrascu, N., Murphy S., Murphy L. *A Methodology for Predicting the Performance of Component-Based Applications*. Eighth International Workshop on Component-Oriented Programming. 2006

[9] Haupt, M., Mezini M. Micro-Measurements for Dynamic Aspect-Oriented Systems. .NET Object Days. 2004.

[10] Santana, Fabiana ; Bravo, César ; Saraiva, Antônio Mauro et al. *P-Garp (Parallel Genetic Algorithm for Rule-set Production) for clusters applications*. In: Ecological Informatics ISEI2006, 2006, Santa Barbara - California. Proceedings of Ecological Informatics ISEI2006, 2006.