

Proceedings

# ACS/IEEE International Conference on Computer Systems and Applications

**25-29 June 2001**

**Beirut, Lebanon**

*Sponsored by*

Arab Computer Society

IEEE Computer Society Technical Council on Software Engineering (TCSE)

IEEE Computer Society Technical Committee on Computer Architecture (TCCA)

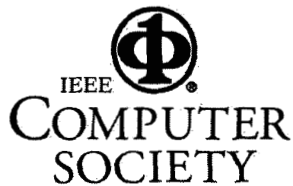
*In cooperation with*

ACM SIGSOFT, SIGART, SIGPLAN

Lebanese American University

American University of Beirut

Lebanon's National Council for Scientific Research (LNCSR)



Los Alamitos, California

Washington   □   Brussels   □   Tokyo

---

Copyright © 2001 by The Institute of Electrical and Electronics Engineers, Inc.  
All rights reserved

*Copyright and Reprint Permissions:* Abstracting is permitted with credit to the source. Libraries may photocopy beyond the limits of US copyright law, for private use of patrons, those articles in this volume that carry a code at the bottom of the first page, provided that the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923.

Other copying, reprint, or republication requests should be addressed to: IEEE Copyrights Manager, IEEE Service Center, 445 Hoes Lane, P.O. Box 133, Piscataway, NJ 08855-1331.

*The papers in this book comprise the proceedings of the meeting mentioned on the cover and title page. They reflect the authors' opinions and, in the interests of timely dissemination, are published as presented and without change. Their inclusion in this publication does not necessarily constitute endorsement by the editors, the IEEE Computer Society, or the Institute of Electrical and Electronics Engineers, Inc.*

IEEE Computer Society Order Number PR01165  
ISBN 0-7695-1165-1  
ISBN 0-7695-1166-X (case)  
ISBN 0-7695-1167-8 (microfiche)  
Library of Congress Control Number 2001089477

*Additional copies may be ordered from:*

IEEE Computer Society  
Customer Service Center  
10662 Los Vaqueros Circle  
P.O. Box 3014  
Los Alamitos, CA 90720-1314  
Tel: + 1 714 821 8380  
Fax: + 1 714 821 4641  
[http://computer.org/  
csbooks@computer.org](http://computer.org/csbooks@computer.org)

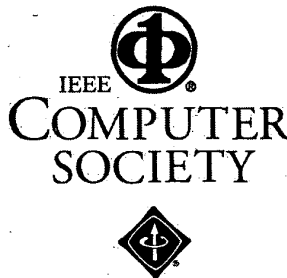
IEEE Service Center  
445 Hoes Lane  
P.O. Box 1331  
Piscataway, NJ 08855-1331  
Tel: + 1 732 981 0060  
Fax: + 1 732 981 9667  
[http://shop.ieee.org/store/  
customer-service@ieee.org](http://shop.ieee.org/store/customer-service@ieee.org)

IEEE Computer Society  
Asia/Pacific Office  
Watanabe Bldg., 1-4-2  
Minami-Aoyama  
Minato-ku, Tokyo 107-0062  
JAPAN  
Tel: + 81 3 3408 3118  
Fax: + 81 3 3408 3553  
[tokyo.ofc@computer.org](mailto:tokyo.ofc@computer.org)

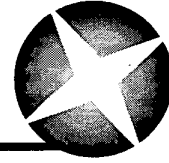
Editorial production by Danielle C. Young

Cover art production by Joe Daigle/Studio Productions

Printed in the United States of America by Applied Digital Imaging



# Table of Contents



*ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 2001)*

Foreword .....	xiv
Conference Chairs.....	xvi
Program Committee.....	xvii
Organizing Committee .....	xix

## Artificial Intelligence

### Session AIF1: Genetic Programming

Highlights on the Evolutionary Engineering Approach: The EE-method .....	5
<i>A. Lehireche, A. Rahmoun, and A. Gafour</i>	
A Generic Functional Genetic Algorithm .....	11
<i>J. Hawkins and A. Abdallah</i>	
A Genetic-Based Neuro-Fuzzy Generator: NEFGEN.....	18
<i>A. Rahmoun and S. Berrani</i>	

### Session AIF2: Applications of Artificial Intelligence

Genetic Optimization of Variable Structure PID Control Systems.....	27
<i>W. Bedwani and O. Ismail</i>	
FBLLB: A Fuzzy-Based Traffic Policing Mechanism for ATM Networks.....	31
<i>M. Salamah and H. Lababidi</i>	
DB-TUTOR: An Intelligent Tutoring System Using a Troublemaker Companion.....	36
<i>T. Bouhadada and M. Laskri</i>	

### Session AIF3: Machine Learning

Absorbing Stochastic Estimator Learning Algorithms with High Accuracy and Rapid Convergence.....	45
<i>G. Papadimitriou, A. Pomportsis, S. Kiritsi, and E. Talahoupi</i>	
Neural Expert System with Two Engines of Rough Set.....	52
<i>M. Yahia and R. Mahmood</i>	

A Planning System for Blocks-World Domain .....	59
<i>B. Prasad</i>	
<b>Session AIF4: Problem Solving</b>	
Constraint-Based Timetabling — A Case Study .....	67
<i>A. Abbas and E. Tsang</i>	
An Intelligent Approach for Solving the Airlines Crew Rostering Problem.....	73
<i>W. El Moudani, C. Cosenza, and F. Mora-Camino</i>	
<b>Session AIF5: Pattern Recognition</b>	
Hand-Written Indian Numerals Recognition System Using Template Matching Approaches .....	83
<i>F. Al-Omari</i>	
Principal Component Analysis of Multispectral Images Using Neural Network .....	89
<i>S. Chitroub, A. Houacine, and B. Sansal</i>	
Combination of Vector Quantization and Hidden Markov Models for Arabic Speech Recognition .....	96
<i>H. Bahi and M. Sellami</i>	
<b>Session AIS1: Pattern Recognition</b>	
Segments Matching Using a Neural Network Approach.....	103
<i>O. Djekoune, K. Achour, and H. Zoubiri</i>	
Arabic Hand-Written Text Recognition.....	106
<i>S. Saloum</i>	
A Neuro-heuristic Approach for Segmenting Handwritten Arabic Text .....	110
<i>A. Hamid and R. Haraty</i>	
Human Face Detection Using Fast Co-operative Modular Neural Nets.....	114
<i>H. El-Bakry</i>	
<b>Session AIS2: Neural Nets</b>	
Neural Speech Synthesis System for Arabic Language Using CELP Algorithm.....	119
<i>S. Ben Sassi, R. Braham, and A. Belghith</i>	

Restoration Method Using a Neural Network Model.....	122
<i>N. Zenati and K. Achour</i>	
Generation of the Sense of a Sentence in Arabic Language with a Connectionist Approach.....	125
<i>K. Meftouh and M. Laskri</i>	
Toward a Model of Agents Society: A Descriptive Approach .....	128
<i>H. Beilleili and M. Sellami</i>	

## Databases and Data Engineering

### Session DBF1: Data Engineering

Generating Implicit Association Rules from Textual Data .....	137
<i>Ch. Latiri and S. BenYahia</i>	
Discovery of Fuzzy Sequential Patterns for Fuzzy Partitions in Quantitative Attributes .....	144
<i>R.-S. Chen, G.-H. Tzeng, C. Chen, and Y.-C. Hu</i>	
Adaptive Query Processing in E-commerce Environment .....	151
<i>Y. Teh, A. Zaitun, and S. Lee</i>	

### Session DBF2: Multimedia Databases

A Hyper-spaced Data Model for Content and Semantic-Based Medical Image Retrieval.....	161
<i>R. Chbeir, Y. Amghar, A. Flory, and L. Brunie</i>	
Modeling Data Repository and Formalizing Content-Based Queries for Image Databases.....	168
<i>S. Atnafu, L. Brunie, and H. Kosch</i>	

### Session DBS1: Database Structures

A Three-Tier Architecture for Ubiquitous Data Access.....	177
<i>S. Helal, J. Hammer, J. Zhang, and A. Khushraj</i>	
Transformation Rules from Semi-structured XML Documents to Database Model.....	181
<i>Y. Badr, M. Sayah, F. Lafortest, and A. Flory</i>	
The Cost Model for XML Documents in Relational Database Systems .....	185
<i>J.-S. Kim, W.-Y. Lee, and K.-H. Lee</i>	

A Distance-Based Approach for Database Re-engineering .....	188
<i>A. Barbar and M. Collard</i>	

Static and Dynamic Reverse Engineering of Relational Database Applications: A Form-Driven Methodology .....	191
<i>M. Malki, A. Flory, and M. Rahmouni</i>	

### **Session DBS2: Database Processing**

Exploration Techniques of the Spatial Data Warehouses: Overview and Application to Incendiary Domain.....	197
<i>H. Zghal, S. Faïz, and H. Ghézala</i>	

Data Mining Using Classification Techniques in Query Processing Strategies.....	200
<i>Y. Teh, A. Zaitun, and S. Lee</i>	

Spatial Decision Tree — Application to Traffic Risk Analysis .....	203
<i>K. Zeitouni and N. Chelghoum</i>	

## Distributed Computing

### **Session DCF1: Analysis of Distributed Systems**

Scalability Analysis and Domain Decomposition of Large Eddy Simulations of Ship Wakes.....	213
<i>A. Osman, H. Ammar, A. Smirnov, S. Shi, and I. Celik</i>	

Implementation of DDARC: Software Architecture for Debugging Distributed Programs .....	220
<i>S. Rai, D. Sampath, and N. Srivathsa</i>	

Performance Analysis of Extended Vector-Scalar Operations Using Reconfigurable Computing.....	227
<i>I. Damaj and H. Diab</i>	

### **Session DCF2: Techniques of Distributed Systems**

On a Parallel Extended Euclidean Algorithm .....	235
<i>S. Sedjelmaci</i>	

A Functional View of Parallel Computer Graphics .....	242
<i>A. Abdallah and T. Theoharis</i>	

### Session DCF3: Synthesis of Distributed Systems

Distributed Mutual Exclusion Token-Permission Based by Prioritized Groups.....	253
<i>A. Housni and M. Trehel</i>	
Perceived Consistency .....	260
<i>C. Griwodz, M. Liepert, A. El Saddik, G. On, M. Zink, and R. Steinmetz</i>	
A New Routing Policy for Load Balancing in Communication Networks .....	267
<i>M. Heusse and Y. Kermarrec</i>	
A Survey and a Taxonomy of Approaches for Testing Parallel and Distributed Programs .....	273
<i>M. Al-ladan</i>	

### Session DCS1: Distributed Computing

On Designing a Systolic Network for the Resolution of Linear Systems Using the "Gauss-Seidel" Method .....	283
<i>F. Hadjam, A. Rahmoun, and M. Benmohammed</i>	
Location Management in Mobile Computing .....	287
<i>R. Subrata and A. Zomaya</i>	
An Environment Physically Distributed in Java .....	290
<i>D. Lopes and Z. Abdelouahab</i>	

## Software Engineering

### Session SEF1: Software Metrics

Measuring the Coupling of Procedural Programs .....	297
<i>J. Al-Ghamdi, S. Al-Nasser, and T. Al-Zubaidi</i>	
A Measure for Component Interaction Test Coverage .....	304
<i>A. Williams and R. Probert</i>	
Formalizing COSMIC-FFP Using ROOM .....	312
<i>H. Diab, M. Frappier, and R. St-Denis</i>	

### Session SEF2: Software Testing

A Knowledge Base for Program Debugging .....	321
<i>A. Tubaishat</i>	

An Integrated Testing Technique for Component-Based Software .....	328
<i>S. Beydeda and V. Gruhn</i>	
Some Parameters for Timed System Testability .....	335
<i>S. Salva and H. Fouchal</i>	
<b>Session SEF3: Software Engineering Management</b>	
Towards a New Approach on Software Process Evolution.....	345
<i>M. Ahmed-Nacer</i>	
Facilitating Group Formation and Role Allocation in Software Engineering Groups .....	352
<i>G. Dafoulas and L. Macaulay</i>	
A Service Oriented Standardised System for Virtual Private Networks.....	360
<i>H. Antonopoulou, N. Bogonikolos, K. Giotopoulos, S. Likothanassis, A. Tsakalidis, and B. Vassiliadis</i>	
<b>Session SEF4: Object Based Software Engineering</b>	
Corba Views: Distributing Objects with Views.....	369
<i>H. Mili, H. Mcheick, J. Dargham, and S. Sadou</i>	
Enhancing Schedulability: A Real-Time Object Slicing Approach .....	379
<i>M. Kimour, M. Bendelloul, and Z. Bouras</i>	
Object Based Software Engineering with Hidden Sorted ECATNets and Hidden Labeled Rewriting Logic .....	386
<i>M. Bettaz</i>	
<b>Session SEF5: Software Engineering Techniques</b>	
A Reuse Description Formalism .....	395
<i>Z. Houhamdi and S. Ghoul</i>	
An Execution Model for the Seamless Integration of Control Flow and Data Flow .....	402
<i>B. Ibrahim and H. Randriamparany</i>	
Modeling Resources in a UML-Based Simulative Environment .....	405
<i>H. Ammar, V. Cortellessa, and A. Ibrahim</i>	
Automatic Verification of Concurrent Object Properties .....	411
<i>R. El-Baïda and J.-P. Bahsoun</i>	



**Session SES1: Programming Languages**

On the Interaction of Programming by Contract and Liskov  
Substitution Principle..... 421  
*W. Al-Ahmad*

The Arabic Object-Oriented Programming Language Al-Risalh ..... 424  
*M. Amin*

Seamless Integration of Control Flow and Data Flow in a Visual Language..... 428  
*H. Randriamparany and B. Ibrahim*

Taxonomy of Process Modeling Languages ..... 435  
*K. Zamli and P. Lee*

UML-L: An UML Based Design Description Language ..... 438  
*H. Hamed and A. Salem*

**Session SES2: Software Engineering Techniques**

Interactive Electronic Technical Manual Object Model (IETMOM)..... 445  
*D. Ghobashy and H. Ammar*

A Reuse Based Approach for Requirements Engineering ..... 448  
*O. Roudiès and M. Fredj*

An Environment for Formal Lingware Engineering ..... 451  
*B. Gargouri, M. Jmaiel, and A. Hamadou*

Building UML-Based Use Case Sub-tool for Component-Based  
Software Development..... 454  
*F. Eassa and O. Abulnaja*

Automatic Signature Files Update in Antivirus Software Using Active  
Packets..... 457  
*A. Badhusha, S. Buhari, S. Junaidu, and M. Saleem*

**Session SES3: Software Testing and Reliability**

Methods and Metrics for Selective Regression Testing..... 463  
*R. Bahsoon and N. Mansour*

A Unified Symbolic Execution System ..... 466  
*C. Koutsikas and N. Malevris*

Prediction of Software Reliability: A Comparison between Regression and Neural Network Non-parametric Models .....	470
<i>S. Aljhdali, A. Sheta, and D. Rine</i>	
CORBA-Based E-commerce Application Testing Architecture.....	474
<i>R. Probert and W. Li</i>	
GOMSE: A Group Oriented Modeling and Simulation Environment.....	478
<i>B. Belattar and A. Korichi</i>	
<b>Session SES4: Software Engineering Management</b>	
A Case Study with Relational Database Metrics.....	485
<i>C. Calero, M. Piattini, and M. Genero</i>	
Toward a Comprehensive Framework for Software Process Modeling Evolution .....	488
<i>O. Eljabiri and F. Deek</i>	
An Analysis of Software Engineering Metrics in OO Environment.....	492
<i>N. Debnath, R. Lee, and H. Abachi</i>	
Distributed Object Computing and Its Application in Electronic Commerce Systems .....	495
<i>K. Saleh, C. Demarais, and R. Koppler</i>	
CSPRSGs Strength Measures Based on the Classes of Transformation .....	499
<i>H. Mohamed and I. Osman</i>	

## Web Engineering

### Session WEF1: Web Applications Management

Automated Detection of Internet Usage Policy Violation .....	507
<i>S. Al-Khayatt and R. Neale</i>	
Connectionist Approach for Website Visitors Behaviors Mining.....	511
<i>K. Benabdeslem, Y. Bennani, and E. Janvier</i>	

### Session WEF2: Web Applications Design

Improving Web Database Access Using Decision Diagrams.....	519
<i>D. Popel and N. Al Hakeem</i>	
A Security Analysis for Label Switching Routers.....	526
<i>M. Al-Ibrahim, M. Sausar, and W. Adi</i>	

Democratic Data Fusion for Information Retrieval Mediators.....	531
<i>Y. Tzitzikas</i>	
Keep It Small and Smart.....	538
<i>A. El Saddik and R. Steinmetz</i>	
<b>Session WES1: Web Engineering</b>	
Bargaining in E-commerce: A Special Negotiation Scheme.....	547
<i>R. Mourtada and F. Szidarovszky</i>	
Evaluation and Validation Tests of Multimedia ISO9004 Arabic Prototype .....	550
<i>Y. Monsef</i>	
A New Digital Report for Disseminating Meeting Outcomes .....	554
<i>M. Aparicio and C. Costa</i>	
CBR-Based Collaborative Interface Agents: Architectural Issues .....	557
<i>M. Malek and R. Kanawati</i>	
Creation of a Personal Space with HyWebMap .....	560
<i>I. Saleh, F. Papy, and N. Bouhaï</i>	
<b>Author Index</b> .....	<b>563</b>

# Automatic Signature files update in Antivirus software using Active Packets

Akbar<sup>1</sup>, Buhari<sup>1</sup>, Sahalu<sup>2</sup>, Saleem<sup>1</sup>

<sup>1</sup>Lecturer, ICS Dept., King Fahd University of Petroleum and Minerals,

<sup>2</sup>Assistant Professor, ICS Dept., King Fahd University of Petroleum and Minerals,  
31261 Dhahran, Saudi Arabia.

Phone: 966 3 860 4374

Fax: 966 3 860 2174

Email: [mibuhari@ccse.kfupm.edu.sa](mailto:mibuhari@ccse.kfupm.edu.sa)

## Abstract

*The field of information technology is growing very fast and so there are more and more updates regularly. At the same time, the field of viruses are also going on the increase and even people complain that attacks on computers are mostly from within the intranet structure. So, if we are able to update the signature files of the antivirus that exist in the various computers that are used by the various members of the locality concerned, we can prevent to some extent, the problems of viruses. In order to do this, now a days, the system administrators send remainders to the concern people or the computers have to be set to update every few days or so. These systems have their own drawbacks. To avoid this hazard, we are providing an option of active packet oriented automatic signature file update.*

**Keywords:** Active networks, parallel virtual machine

## 1. Introduction

Active networking refers to the placement of user-controllable computing and other resources in the communication network, where they can be utilized by applications that need those capabilities. An active network supports a user-network interface allowing the nodes of the network to be programmed by the application (user) to provide a desired functionality, such as routing. This programming might be done on a per-packet basis (as in the capsule approach of Tennenhouse and Wetherall) or through an out-of-band signaling mechanism (e.g., injection of user-specific programming into the switch, as in Switchware). Active networks allow users to increase the likelihood that the service offered by the network will be useful to them, by providing an interface that supports multiple (or programmable) services. There are costs associated with such a flexible interface, and they affect all of the network's users whether they have advantage of active network support or not. The (monetary) cost of

providing the interface, though likely to be significant, is paid once and can be amortized over all users for a period of time. The performance cost of using the interface should vary with application; this is the end-to-end argument [Sam97]. Applications running on end systems are allowed to inject code into the network to change the network's behaviour in their favour [Dan99].

Instead of the system administrator sending emails to the users stating them to update their anti-virus software or the system administrator himself moves upto the users machine to update the required anti-virus software regularly, the process of anti-virus software updation can be done automatically using active networks. This will save improper wastage of man power as well as delays that occur in regular updation.

The rest of this paper is organised as follows: Section 2 discusses about the current existing and the related works done by others. In section 3, We discuss about the drawbacks of the current system and how the drawbacks are overthrown by the use of active networks. Our proposed model is discussed in length in section 4 and conclusion follows it in section 5.

## 2. State of work

The current world of anti-virus update is fully manual. Now a days, the licensed user is notified of some additional bugs found and their respective remedies handled with the help of the new patches or new set of data file updates. The patch is used to update the existing version so as to be make the antivirus able to identify more viruses and safeguard the system in an efficient way. The update of the antivirus software is done with the help of data files which contain signatures of the various newly appearing viruses. This process is now handled with manual update and the network administrator concerned asks the other users of the network to do the update on their own.

One disadvantage of manual update is that a user who is not around in the campus, may not be able to do the update on time. One fine example of this was when the CIH Virus attacked the machines, the attack happened in the night when most of the users were not around to take the necessary action. Even if everyone can do the update, it can be monotonous and takes up precious personnel time. On the other extreme, some (experienced) users might use the option of automatic update for antivirus datafiles. In this case, users set their own update dates. This case will cause connections to be set-up to the main server regularly. Instead of polling the server at random (at user-set update times) when there might not be updates, it is better to ensure that there is update whenever the server is polled.

### 3. Active networks capsule

There are two methods by which signature update can be done. They are:

#### a. Out of bound method

Each packet contains a set of headers, which specify,

1. The identifier to one or more functions to be applied to the packet,
2. The type of application that ought to be executed, and
3. Parameters to be supplied to those functions.

In this case, the signature oriented data file is not sent along with the capsule. Only the position of availability of the concerned data file is sent along with the capsule. The concerned host or router, gets connected to the server that possesses the concern data file and downloads it and executes it on its machine. This is automatically done along with the packets.

#### b. In bound method

In this case, the capsule itself contains the signature file inside. When the node receives the packet, the concern signature file is taken and it is executed on the node. This is faster to be handled as the capsule itself has the signature. But there are certain drawbacks of overloading the network with more data, if we are going to send the data file along all the capsules. Maybe a multicast option in this case will make the job faster.

### 4. Our Network Model

Our network model is made of 4 subnets with each subnet made up of 4 nodes named from 0 to 3, which are connected to each other through a hub as shown in Figure 1. The hubs of all the subnets are interconnected. The host

computers handle the packet that carries destination address of the packet. The packet generation is handled by the hosts, which possess the ideal packet generators in them. The generated packets are then processed and transmitted through the transmitter. The hub is made up of a processing node connected to many transmitters and receivers, depending on the number of hosts that ought to be connected to it. The node part of the host and the hub are shown in Figure 2. The system has the option of being active or non-active, which could be set up by the user during the simulation time.

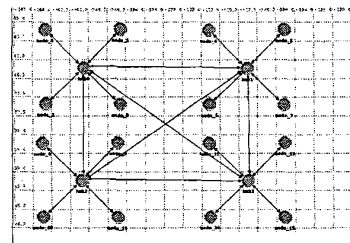


Figure 1. Network model

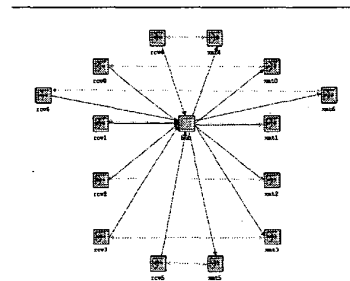


Figure 2. Hub Structure

Since each hub is connected to 4 nodes and three hubs it has a set of receiver and transmitter for each connection. The program segment for process can be written into the receiver and the transmitter.

### 5. Our Process model

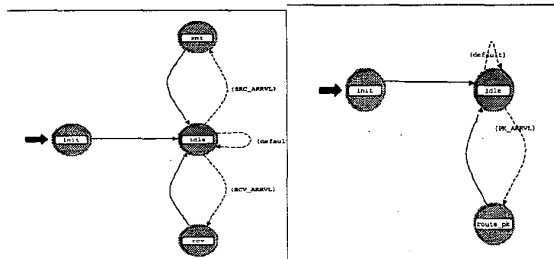


Figure 3. Process model of the host and the hub

- Processing part of host and hub

The processing portion of the host and the hub are shown in Figure 3. The host's "init" part is used to make the packet destination address generation to be between 0 and 3 because the network has 4 nodes only. As we have 4 subnets the subnet address in which the node belong to is also added into the packet. The generated packets are then forcibly sent to the idle part. The transmitter then transmits the packet along any one stream depending upon the destination subnet and the destination address specified.

The process of the packets is done at the hub. Once the packet reaches the hub, the initial settings for the analysis of the packet is done. Now, the process is handled based on the criteria whether the packet is active or non-active. The processing at the router on the packet is done only when the communication is active during the simulation time. If the simulation run is set as active, the packet is first read for the filename indication and the data is stored down on the packet. If there is data present in the data part, then it means that the filename indicates the name with which the data part needs to be stored in the router. After storing the packet data onto the router, the action is performed on the router based on the command present. The result of the processing is stored into the data part of the packet and packet is forwarded. This forwarding is mainly based on the destination subnet. If the current subnet is same as the destination subnet, then the packet is just send to the destination node or else, the packet is forwarded to the hub that is present in the destination subnet. If type of the packet is two, that means there is a request for a file from the receiving end, in that case, we need to check the whether the file is present in the intermediary router. If the file is present, then we just send the result from the router itself or else, the requests is send to the server to get the required file. In general, the processing includes caching as well as processing on the fly.

## 6. Analysis of the experimental results

### • Packet format

The packet structure is made as per the requirement of the application. The packet is made up of destination subnet, destination host address, type, command, filename, data. The destination subnet and destination host address are stored as integers as we number the subnet is marked from 1 to 4 and the nodes from 0 to 3. To handle the integer storage in OPNET, the necessary variable type is present. To handle the filename and the data storage as strings, the storage is done in the form of a structure which is made of an array of characters. The type option is used to indicate whether the source is sending a packet or the receiver is sending a request packet. The command option is used to indicate the command that is used to act upon the packet. The data part is used to store the result of the processing on the packet. The data part can also be made of files that need to be stored at the routers that the packets pass by.

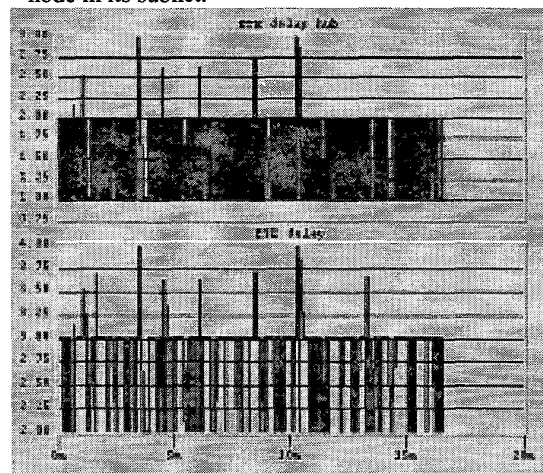
### • Analysis

The End-to-End (ETE) delay has been calculated both at the router level and the node level. The ETE delay is calculated by finding the difference between the time at the creation of the packet and the time at which it reaches the hub or the destination. Time synchronization problem does not persist here because the whole simulation process runs on a single machine. The ETE delay is calculated for both active as well as non-active system. The ETE delay at the hub and at the node is shown in Figure 4.

ETE value at hub and node vary but same while active or nonactive.

We can infer from the graph of Figure 4 that the ETE delay for the hub is less than that of the node because the hub is physically closer to the source than the node. Also the packet density is more at the hub because of the following reasons,

- The hub is connected to more than one node,
- A hub receives message from the 4 nodes which are present in the subnet,
- The hub also receives any packet that is destined to any node in its subnet.



**Figure 4. (x-axis: time, y-axis: simulation time) End to End(ETE) delay at the hub and the node.**

Each node receives its own packet and so less density of ETE entries, also see figure 4.

Figure 5 shows the Throughput (Packets/sec): The statistic represents the average number of packets successfully received or transmitted by the receiver or transmitter channel per second.

It is the same for both the active and the non-active cases. We can also infer that there is an influence at the hub whenever there is a change in ETE at the node. This

strongly supports any action to be taken at the hub. That is active processing.

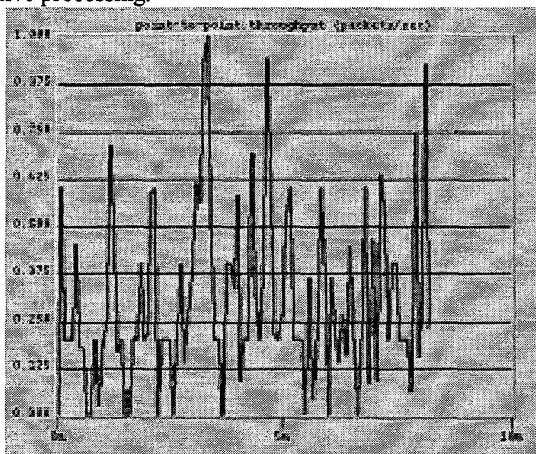


Figure 5. (x-axis: packets/sec, y-axis: simulation time) Throughput of the network for both the active and non-active cases.

The throughput at the node is almost same for active and non-active, as the number of packets destined to or send from the node are the same irrespective of active or non-active system.

Packet inter-arrival rate is the rate of packet arrived to time. If this rate is less then the packet generation is faster. Figure 6 shows the throughput of the node comparing for packet inter arrival rate of 40 and 4 simulation time. Here we can infer from the graph that the packets received is high in number for the inter arrival time of 4 than that of 40 which is obvious.

The simulation time is less for packet inter-arrival rate of 4 seconds because of the fast generation of packets.

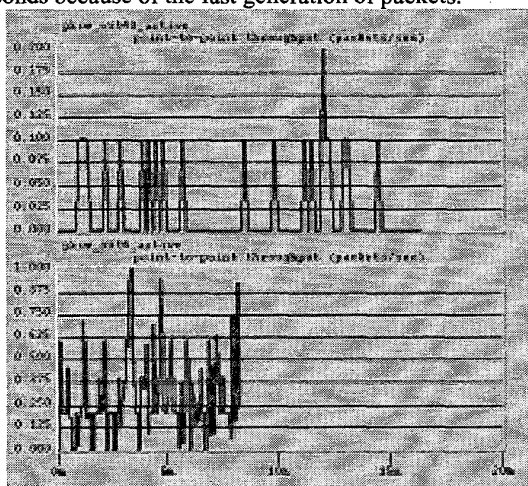


Figure 6. (x-axis: packets/sec, y-axis: simulation time)

## Throughput for the inter arrival time of 40 and 4 simulation time

### 7. Conclusion

Our work makes the system administration job easier, but using active packets to just activate and update any anti virus software. The same can be used for some other software systems which need some patches to update to the existing software. This process will be further developed to check for some security oriented implications.

### References

- [Dan99] Dan S. Decasper, Bernhard Plattner, Guru M. Parulkar, Sumi Choi, John D. DeHart, and Tilman Wolf, "A Scalable High-Performance Active Network Node", IEEE Network, May/June 1999, pp. 8-19.
- [Chi00] Chih-Lin Hu, and Wen-Shyen E. Chen, "A Mobile Agent-Based Active Network Architecture", Proceedings of the Seventh International Conference on Parallel and Distributed Systems (ICPADS '00), Japan, July 2000, pp. 445-452.
- [Dav96] David J. Wetherall and David L. Tennenhouse, "The Active IP option", 7<sup>th</sup> ACM SIGOPS European Workshop, Sept. 1996.
- [Dal96] David L. Tannenhouse and David J. Wetherall, "Towards an Active Network Architecture", Computer Communication Review, Vol. 26, No. 2, April 1996
- [Sam97] Samrat Bhattacharjee, Kenneth L. Calvert, and Ellen W. Zegura, "Active Networking and the End-to-End Argument", Proceedings of the international Conference on Network Protocols, Oct 1997, pp. 220-228.
- [Dan99] Dan S. Decasper, Bernhard Plattner, Guru M. Parulkar, Sumi Choi, John D. DeHart, and Tilman Wolf, "A Scalable High-Performance Active Network Node", IEEE Network, May/June 1999, pp. 8-19.
- [Sco98] Scott Alexander, William A. Arbaugh, Angelos D. Keromytis, and Jonathan M. Swith, "A Secure Active Network Environment Architecture: Realization in SwitchWare", IEEE Network, May/June 1998, pp.37-45.