Proceedings

COMPUTER SOCIETY

The Fourth Workshop on Information and Computer Science

17-18 March, 2002 (3-4 Muharram, 1423)

Internet Computing

Information and Computer Science Department

King Fahd University of Petroleum & Minerals

In Cooperation With
IEEE Computer Society Chapter, Saudi Arabia

Designed & Printed by KFUPM Press

WICS 2002 Committees

Editorial Board

- Dr. Kanaan Faisal
- Dr. Muhammad Sarfraz
- Dr. Khalid Salah

Organizing Committee

Dr. Kanaan Faisal	**Chairman**
Dr. Muhammad Sarfraz	**V. Chairman**
Dr. Moataz Ahmed	Member
Dr. Suliman Al-Bassam	Member
Dr. Bakr A. Hassan	Member
Dr. S. Junaidu	Member
Dr. Najib A. K. Kofahi	Member
Mr. Husni Al-Muhtaseb	Member
Dr. Khalid Salah	Member

International Advisory Committee

Dr. Muhammad Sarfraz	KSA	**Chairman**
Dr. Moataz Ahmed	KSA	Member
Dr. A. Abdesselam	Qatar	Member
Dr. Ebad Banissi	UK	Member
Prof. Ernesto Damiani	Italy	Member
Dr. Jarallah Al-Ghamdi	KSA	Member
Dr. Fiaz Hussain	UK	Member
Prof. Ali Jaoua	Qatar	Member
Prof. Juan Llorens	KSA	Member
Dr. Zakaria Maamar	UAE	Member
Prof. Ali Mili	USA	Member
Prof. Serge Miranda	France	Member
Prof. Ahmad H. Nasri	Lebanon	Member
Prof. David Rine	USA	Member
Prof. Mark Pullen	USA	Member
Dr. Khalid Salah	KSA	Member
Prof. Joseph Urban	USA	Member
Prof. Paul P. Wang	USA	Member

Scientific Committee

Dr. Muhammad Sarfraz	**Chairman**
Dr. Khalid Salah	Member

Sponsorship Committee

- Dr. Kanaan Faisal
- Dr. Jaralla Al-Ghamdi

Program & Schedule Committee

Dr. Moataz Ahmed	**Chairman**
Dr. Bakr A. Hassan	Member
Dr. M. Paul D. Manual	Member
Dr. M. Shafique	Member
Dr Abdallah Al-Sukairi	Member

Publicity, Public Relation, and Guest Reception Committee

Dr. Talal H. Al-Maghrabi	**Chairman**
Mr. Abuosba	Member
Dr. Mohammed Balah	Member
Mr. Ali A. Alatif Busbait	Member
Mr. B. Ghandi	Member
Mr. L. Ghouti	Member
Mr. Khalid Al-Jasser	Member
Dr. S. Junaidu	Member
Mr. Mohsin Siddiqui	Member
Dr. Najib A. K. Kofahi	Member
Dr. K. Qureshi	Member
Dr. Nowduri Srinivas	Member

Site Committee

Dr. S. Al-Bassam	**Chairman**
Dr. Naser Darwish	Member
Mr. Mutlaq Al-Mutlaq	Member
Mr. Sharafuleslam	Member

Social Activities

Mr. Lahouri Ghouti	**Chairman**
Mr. Adel F. Ahmed	Member
Mr M. Minhas	Member
Mr. I. Putu Danu Raharja	Member
Mr. Mohammed Wasiq	Member

Registration Committee

Mr. Husni Al-Muhtaseb	**Chairman**
Mr. M. W. Aslam	Member
Mr. Abdallah M. Said	Member & Treasurer
Mr. Muhammad Saleem	Member
Dr Mohammad Al-Suwaiyel	Member
Mr. M. Atif Taher	Member

Publication & Printing Committee

Dr. Khalid Salah	**Chairman**
Mr. Faisal Alvi	Member
Mr Khalid Al-Badawi	Member
Mr. M. Buhari	Member
Dr. Talal Al-Maghrabi	Member
Mr. Raimi Rufai	Member
Dr. Muhammad Sarfraz.	Member

WEB Site Committee

Mr. Muhamad Saleem	**Chairman**
Mr. Akbar Badusha	Member
Mr. Murtaza Khan	Member

Table of Contents

Keynote Speeches

1. Applications Development: What Changed with the Introduction of the Internet? 3
Ibrahim S. Mishari

2. Next Generation Internet: Present Status and Future Directions 5
Morteza A. Rahimi

3. Databases, E-commerce, the Web, and XML: How Are They All Related 7
Ramez A. Elmasri

4. E-Commerce – The Saudi Arabian Case 9
M. I. Al-Suwaiyel

Invited Speeches

5. OMTI Project: Mediterranean Showroom Of Wireless Internet Services Of The Future .. 13
Serge Miranda

6. Information Visualisation Metaphors 15
Ebad Banisse

7. Agents Of Organizational Learning 17
David Rine

8. Empowering the Internet Generation 19
Sam Alkharat

Session 1: ***Electronic Learning***

9. Workgroup Computing Model: A Case Of Implementation In Higher Learning Institution 25
R. Abdullah

10. Developing On-Line Mathematics Course For Distance Education 35
M.F. Abdul Karim

11. E-Education and Distance Learning: Virtual Learning Environment in the Age of Global Information Networks and Web Learning Environment Evolution 47
S. Parasuram, V. Ganapthy and M.S. Sohail

12. Internet/intranet Based Higher Education 59
S.I. Sheikh and J. Siddiqui

13. Distance Learning: Major Issues and Challenges 67
N.A. Kofahi and Srinivas

14. Outcome-Based Learning Program Model for Information Systems at Zayed University . 83
A. Lansari, N. Nada and A. Abonamah

Session 2: *Systems Development*

15. A Strategy For Integrity Preservation In Temporal Relational Databases 95
F. Hanandeh, H. Ibrahim and Z. Muda

16. A Data-Centric Design For N-Tier Architecture 105
P.D. Manuel and J. AlGhamdi

17. A Knowledge-based System for Best Practices Reusability 115
E. Hattab, A. Papantoniou, N. Alsaid and R. Curwen

18. An Analysis On Software & Hardware Components Involved In The Design Of Embedded Medical Record Keeping System. 125
R. Sridaran, S. Annapoorani and K. Alagarsmy

19. Empirical Studies of System Development Methodology Practices In Malaysia 135
B.K. Siang

20. Conceptual Data Reduction. Application For Reasoning And Learning 149
A. Jaoua, A. Alrashdi, H. AlMuraikhi, M. AlSuaiey, N. AlGhanim, S. AlMisaifri

21. Best Practices for Deployment and Management of Institutional Systematic Software Reuse 159
N. Nada

Session 3: *Character/Speech Recognition*

22. Recognition of spoken digits: A Statistical Analysis 183
S. Rizvi, D. Nadeem, H. J. Naqvi and R. Biswas

23. Fingerprint Image Segmentation Using Hierarchical Technique 189
I. Abdullah and H.M. Nor

24. Arabic Text-To-Speech: Speech Units 199
M. AlGhamdi, M. Elshafei and H. AlMuhtaseb

25. Issues And Considerations For Sorting Arabic Script 213
F.A. Masoud, K. Shihab and N.A. Kofahi

Session 4: *Web-based Systems/Tools*

26. WEB GIS: Its Current Issues And The Future 227
I. Abdullah and N.A.F. Ismail

27. Web Based Automation for Capturing Outlines of Arabic Fonts 237
M. Sarfraz and F.A. Razzak

28. A Web Based System for Data Visualization 253
M. Sarfraz and S.A. Raza

29. T-Cubic and Arc/T-Cubic Spirals for Web-based Visualization of Planar Data 267
Z. Habib and M. Sakai

30. Web Operating System ... 279
A. Mufti and K. Salah

31. The SLXS Specification Language For Describing The Consistency of XML Documents 289
Y. Kotb, K. Gondow and T. Katayama

32. Students Information Systems: Removing Cobwebs with Web ... 305
M.W. Aslam and T.H. Maghrabi

Session 5: *Internet Agents*

33. Using Agent in a Web-based Software Process Environment ... 317
J.J. Chen and C. Lin

34. AgentGateway: A Communication Tool for Multi-Agent Systems ... 329
J.J. Chen and S. Su

35. A Method of Search using Fuzzy Logic ... 345
A.Q. Ansari and R. Biswas

Session 6: *Electronic Commerce*

36. Electronic Banking and Customer Preferences in Malaysia: An Empirical Investigation . 355
M.S. Sohail and Balanchandran

37. Process Systems Integration Over the Internet/Intranet ... 365
R.A. Fan

38. The Impact of Hacking and Other Computer Related Offences on the Growth of Electronic Commerce ... 377
M. Abdul Jalil

39. Enabling Virtual Communities: Gulliver and Aveiro ... 391
A. Nijholt

Session 7: *Infrastructure, Security, and Performance Evaluation*

40. The Role of Intranet in a University: The KFUPM experience ... 405
T. Maghribi and J. Yazdani

41. Performance Issue Of ATM-Ethernet Network An Analysis ... 415
A. Jantan

42. Adaptive Decentralize Task Scheduling Strategy for Heterogeneous Distributed Raytracing System ... 421
K. Qureshi and M. Bozyigit

43. Parallel Polygon Rendering with a Cluster of PCs ... 429
A. Ait-Aoudia, A.Bachir, and M. Mana

44. A Framework for Enabling Collaboration over Heterogeneous Networks 435
P. Bhandarkar, R. Chowdhury and M. Parashar

45. Optimizing Web Servers Using Page Rank Prefetching for Clustered Accesses 451
V. Safronov and M. Parashar

46. Optimizing Resource Usage Using Parallel Active Networks 467
M.I. Buhari, S. Junaidu, Kalamulla, M.A. Badhusha, H.M. Saleem and Hunger

47. The Cognitive Self-Structuring Connectionist Machine 481
A.A. Fadol and M.K. Ibrahim

48. Starvation Problem in CPU Scheduling For Multimedia Systems 489
M.O. Saliu and K. Salah

49. Boolean Routing on High Degree Chordal Ring 499
A. Abbas, M. Othman, and M.H. Selamat

50. A Method for Defeating DoS/DDoS TCP SYN Flooding Attack: The SYNDEF 511
N.A. Noureldien and I.M. Osman

Referees 521

Author Index 523

WICS 2002
Internet Computing
The 1st Workshop on Information and Computer Science, KFUPM, Dhahran
Saudi Arabia
• March 17 – 18, 2002

Optimizing Resource Usage Using Parallel Active Networks

MI Buhari[1], S Junaidu[1], Kalamullah[2], MA Badhusha[1], HM Saleem[1], Hunger[2]
[1]King Fahd University of Petroleum and Minerals, Saudi Arabia.
[1]Gerhard- Mercator University, Duisburg, Germany.
Email: {mibuhari, akbar, sahalu, msaleem}@ccse.kfupm.edu.sa

Abstract

In traditional packet networks, like the Internet, the amount of computation performed on the packets is extremely limited or is performed in an ad hoc manner. As the Internet grows there is an increasing demand by users to perform user-driven computations or services inside network nodes. In many cases, these services are implemented at nodes, such as firewalls, which adopt the facade of routers but perform application-specific processing that transcends conventional architectural guidelines.

In an active network, the routers or switches of the network can be programmed to perform customized computations, in a systematic manner, on data packets. For example, routers in active networks can perform functions like data compression, issuing source quench messages to the source or previous router and so on.

This paper proposes the use of active networks to address the problem of congestion in computer networks. A second goal of the paper is to use parallel processing to optimise resource usage amongst the active nodes of the network. We report experimental results of a simulated active network as well as a preliminary real implementation of an active network.

Keywords: Active Network, Parallel Processing

1 Introduction

The emergence of Active Networks technology has attracted many academics and researchers to become involved in the development of this technology. Active networks have potentially much to offer the Information Technology world. First, making use of this technology can reduce the time required for the standardization process of new network services. Second, active networks shift the conventional network paradigm: from a passive node that only transfers bits to a more general processing engine like an end station which supports customized computations on user's data. Furthermore, active networks can also be used for enabling on-the-fly modification of network functionality, for example to adapt to changes in link conditions.

As the routers in active networks are more general computing engines, computations that would otherwise be delayed and performed at the end station can be performed earlier at the router level. This can potentially lead to the reduction of congestion and the minimization of communication costs. The use of active networks potentially creates a huge amount of computations inside network routers. Parallel processing can be used to speedup those potential

9960-07-187-1 (©) KFUPM

computations. We envisage that parallel processing will be a powerful tool when used in conjunction with active networks. This is more so in a LAN environment when the load distribution on the nodes in a LAN is skewed with some machines heavily loaded and others lightly loaded. These two technologies can be used to good effect by taking advantage of the low-communication latency of a typical LAN.

In this article, we set out to use a synergy of these two techniques in a LAN environment; use active networks to address the issue of congestion and to use parallel processing to improve the performance and resource utilisation of the underlying active network.

The remainder of this paper is organised as follows. In the next section we present the general architecture of an active networks system and the two main implementation alternatives highlighting our implementation choice. The PVM architecture is described in Section 3. The experimental set-up and analysis using the simulation system is described in Section 4. Section 5 highlights the (simplifying) assumptions in our simulations and puts our results into proper perspective. The experimental analysis using the real-time system is described in Section 6. The advantages offered by implementing the hybrid technology are discussed in Section 7. Section 8 concludes.

2 Active Network Architecture

The general architecture of the active network is made of three main layers, as shown in Figure 1. These layers are the active application layer, the execution environment layer and the node operating system layer. The application layer is used to add certain applications that are to be processed at the node when active processing on the capsule is performed. The execution environment layer is to do the active processing on the capsule. The operating system that lies behind the execution environment is called as the node operating system layer. We note here that although our system works in a heterogeneous environment, some of the services supported at the node operating system layer are usually dependent on a particular operating system.

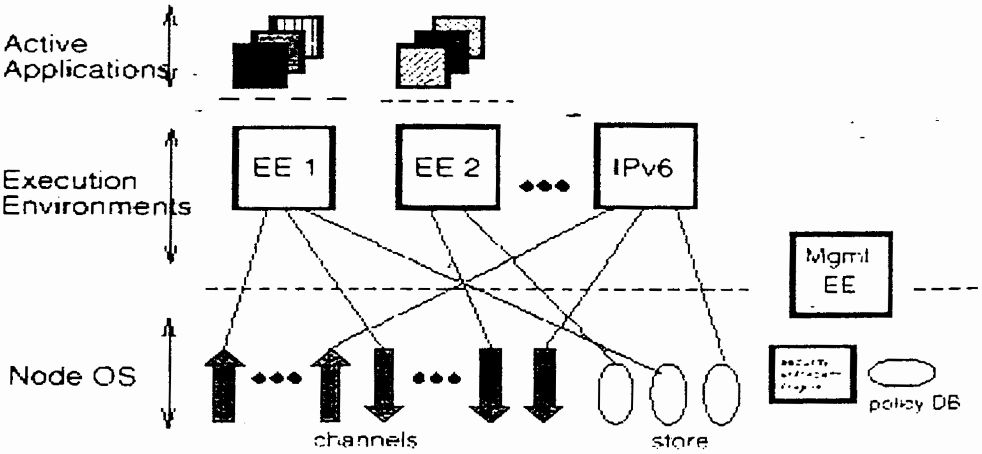


Figure 1: Active Network Architecture [1]

The most important layer that contributes most to the 'active' term of active network is the execution environment (EE). It is in this layer that a new service from the local machine is uploaded and the request from other machines for a new service is submitted and processed. The EE deals with network operating system and service layer. A packet is directed to the EE in order to access network services.

In Java architecture, the EE lies above the Java runtime and under the service layer. From the Java point of view the architecture of Active Network is as depicted in Figure 2.

The active network paradigm allows programs to be injected into an active node. A Program may reside in a switch/router [2, 3, 4] and is activated in the EE only after the packet (i.e., capsule) from the client that carries identifiers or reference to the code, is processed and verified. Another approach allows the packet carries the whole code [5, 6, 7]. In the latter approach, the node is also active in the sense that it allows computation up to the application layer.

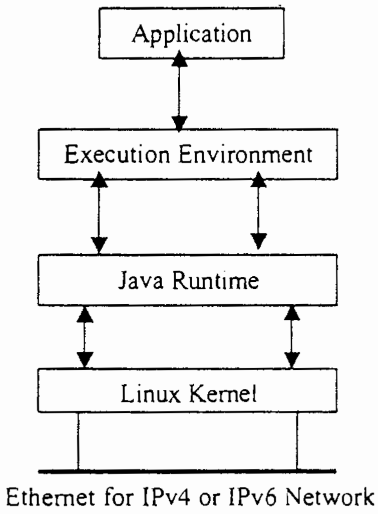


Figure 2: Java Architecture and EE

We prefer for our active network platform that the packet carry only identifiers or references to the services that reside in the node due to the following reasons:

1. It maintains the packet size to minimum.
2. The idea is to optimise the resources available on every machine connected to a LAN. Code can be stored in non-exhaustive machines.
3. The packet can be designed in conformity with the parallel processing architecture of interest.

3 Parallel Virtual Machine (PVM) For Resource Usage Optimisation

The essence of using the PVM in our experiments here is to avoid making a particular node as a computation hot-spot and to ensure that computing resources are optimally utilised, as much as possible. By Amdahl's law, processing speed can be increased by exploiting parallelism.

$$\begin{aligned} S(n) &= \text{Execution time using one processor} \, / \\ &\quad\ \text{Execution time using a multiprocessor with } n \text{ processors.} \\ &= t_s / t_p \end{aligned}$$

$$\begin{aligned} \text{Maximum speed up,}\quad S(n) &= t_s / \; [f t_s + (1 - f)\, t_s / n] \\ &= n / [\, 1 + (n-1)\, f \,] \end{aligned}$$

$$\begin{aligned} \text{Efficiency} &= \text{Execution time of one processor} \, / \\ &\quad\ [\text{Execution time using a multiprocessor * number of processors}] \end{aligned}$$

A major bottleneck that hinders effective exploitation of parallelism is (data) dependency among the various instructions in a given code sequence. If dependency among the instructions is high then exploiting parallelism may not yield any positive dividends. Therefore in Figure 3, better parallel behaviour will be obtained when the value of the sequential section, *f*, is low.

Normally one type of operation is performed, like file transfer using FTP, in an active networks session. Once a router learns the processing steps of a session from analysing a packet during the session, it can decide whether or not the exploitation of parallelism is worthwhile in the session. Such router decision, based on single-packet processing in a session, is, most of the time accurate and makes worthwhile the exploitation of parallelism.

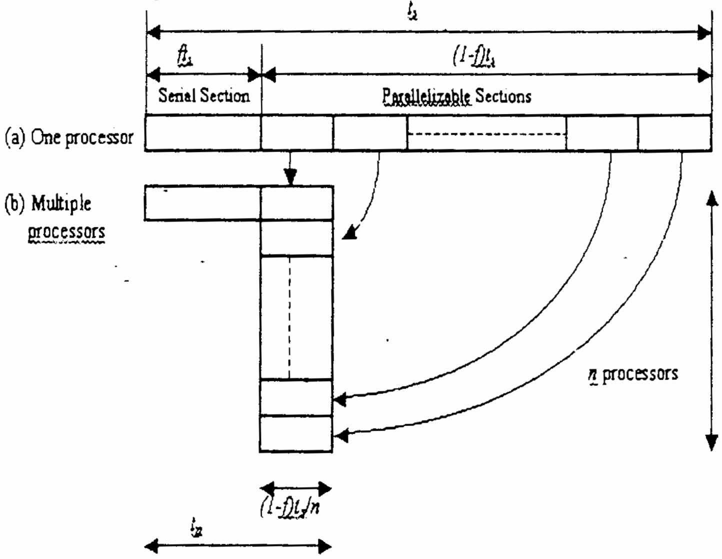


Figure 3 : Amdahl's Law [8]

An important factor in the use of PVM to exploit parallelism is the physical separation of the networked computers due to communications costs. Our initial implementation considers an active networks system within a LAN setting.

3.1. Parallel processing part

As highlighted earlier, we use PVM in our active networks system to exploit parallelism. The PVM system runs on both Windows as well as Linux platforms. Although our system is heterogeneous, it works well as a homogeneous system.

The router or the switch is used to run the PVM software and the other connected machines run the PVM Daemon. Once the router receives a job to be processed, it distributes the job and executes it in parallel with the help of other computers that run the PVM daemon.

3.2. Pipelining

Pipelining is the means by which more than one unit (routers in our case) can operate on an input incrementally and in parallel. If we are using a connection-oriented data transfer method using TCP sockets in Java, then we can use the concept of pipelining to exploit parallelism. This can be done by making the routers handle different set of jobs, by sharing their respective jobs among their PVM Daemons.

4 Experimental Setup

The set-up of our experiments consists of a 16-node network designed and simulated using OPNET, a powerful tool for designing and simulating computer networks. Our network model shown in Figure 4 consists of 4 subnets each comprising of 4 nodes. We made use of OPNET's random packet generation algorithm at each of the nodes. We implemented a static routing algorithm that fixes the next router through which packets can be forwarded from the current one based upon the destination subnet and the host address present in the packet. In order to maintain a backup route for various routes, we assign secondary routes too. The node part of the host and the hub are shown in Figure 5.

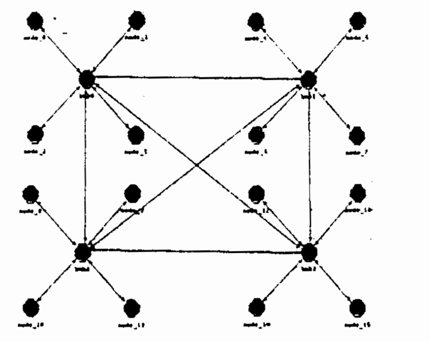


Figure 4. Our Network Architecture

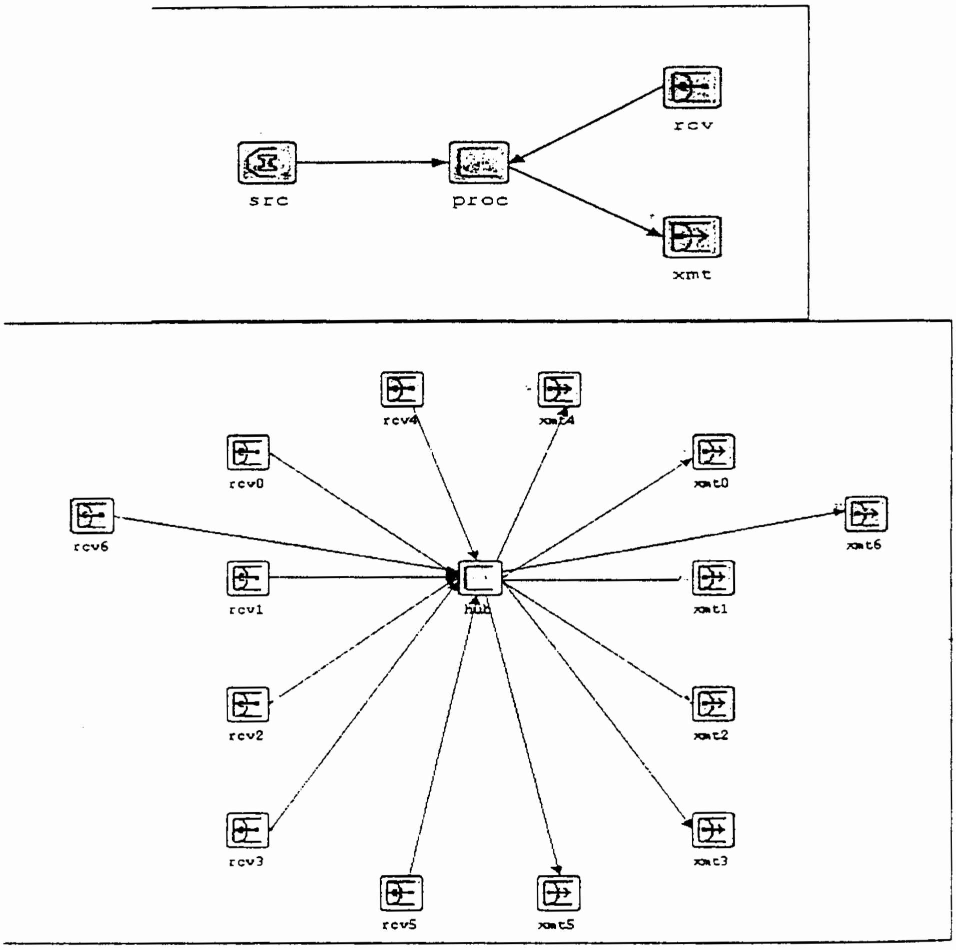


Figure 5: Node diagram of the host and the hub

4.1. Packet Format

Source Address	Destination Address	Protocol	Packet Type	Timestamp	File Name	Data

The packet format for our capsule-based active networks system is as follows:
The source, destination addresses and the protocol are used for the access control permit or denial of the packet. The packet type is used to identify whether the packet is a data packet or an IP/SQ packet. The timestamp is used to calculate whether there is a need to generate Source Quench packet. The file name and data contains the name of the file and the data, which is carried by the capsule.

4.2. Processing part of host and hub

The processing portion of the host and the hub are shown in Figure 6. The host's "init" part is used to make the packet destination address generation to fall between 0 and 15. 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 address specified.

The destination address is checked at the hub and the packet is forwarded in the respective transmitter stream. Once the packet reaches the destination, the packet is now processed to identify the time it has taken to reach the destination from the source. Depending upon the time duration taken the source quench message is generated to the source. Being a simulated system, the whole setup is executed on one machine. The time at the source and destination are synchronized. But for the real-time systems, we use the Round Trip Time [RTT – Time taken for the capsule to travel from source to destination and back] as the means to calculate the time duration.

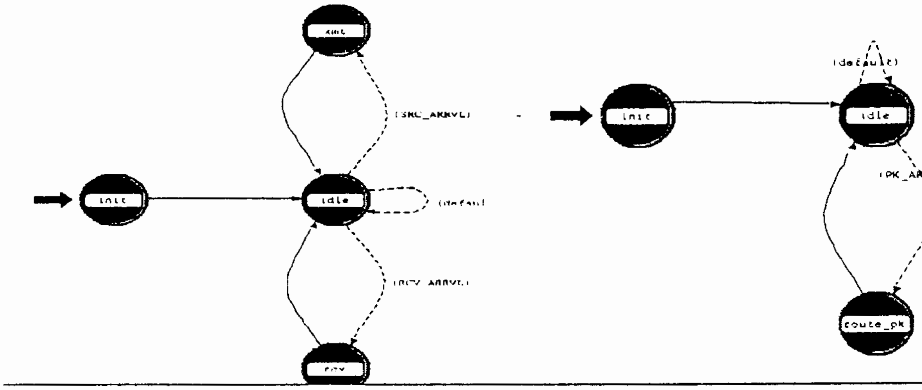


Figure 6: Process module of the host and the hub

As an example, let us suppose that the result of some computation is to be transmitted from node_0 to node_15. The data can pass either through hubs [0, 2 and 3] or [0,1 and 3] (Figure 4). PVM resides in active routers (hubs) and also in all the nodes. When the data is passing through the first router, the router will partition the computation to be performed on the data into many modules based on availability of nodes. Each module along with its associated data is sent to one PVM process, which is virtually connected and running in parallel at different nodes connected to the router. The nodes will process the data and send the result to the router. Then the router will forward the accumulated results to the next router, which also forwards the result to the destination [node_15]. Congestion is controlled in the router itself, since the router is active and it can run congestion control algorithms.

In our experiments reported in this paper, we simulated the throughput and utilization of our network in order to demonstrate the potential practical usefulness of our ideas. Throughput, utilization and ETE delay are defined as:

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

***Utilization**: represents the percentage of the consumption of data of an available channel bandwidth, where a value of 100.0 would indicate full usage.*

***ETE delay**: End-to-End delay is the time taken by the packet to reach the destination from the source (one end to other end). The packet creation time is calculated.*

Figures 7 and 8 respectively depict results of our simulated network's ETE delay between source to hub, hub to hub and hub to destination and ETE delay between source and destination. Figures 7 and 8 each has two component graphs; the graph in the upper part shows the result obtained when the active version of the network is simulated and that in the lower part shows the result for the corresponding non-active network. Apart from the active and the non-active differences in the models, the data used in the experiments is the same and all experiments were conducted under the same (as much as possible) network conditions using OPNET kernel procedures. The simulation of both active and non-active system is done using the simulation parameter that is set before simulation of the network.

The horizontal axes in the figures 7 and 8 represent the simulation time in seconds. The vertical axes in these figures represent transmission time.

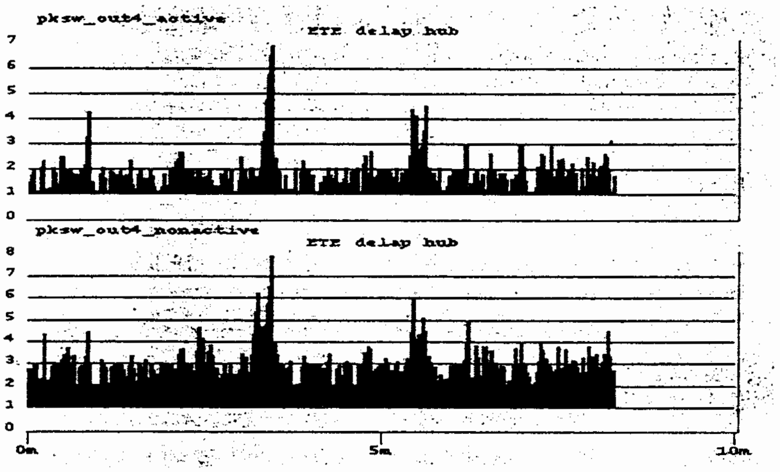


Figure 7: Comparative ETE Delay of the hub of the Active and Non-active Systems

Figure 7 reveals that the ETE delay in the hub is smaller for each packet in the case of the active system. This is because after doing computations on the data in the routers, only results of the computations are passed to the destination where the resultant data is typically smaller in size compared to the original data sent by the source. Here, packets destruction is also done based on the following conditions.

1. The number of hops the packet has passed-by has reached the maximum limit. This is recorded in the TCP header itself.
2. The packet is destroyed if it is known that the packet may not reach the destination.
3. The network traffic is high. This can be known by using the ETE delay time.
4. If *n* packets are sent and *k*<*n* packets are sufficient to transmit the results of processing the *n* packets at the routers, then the excess *n*-*k* packets are destroyed.

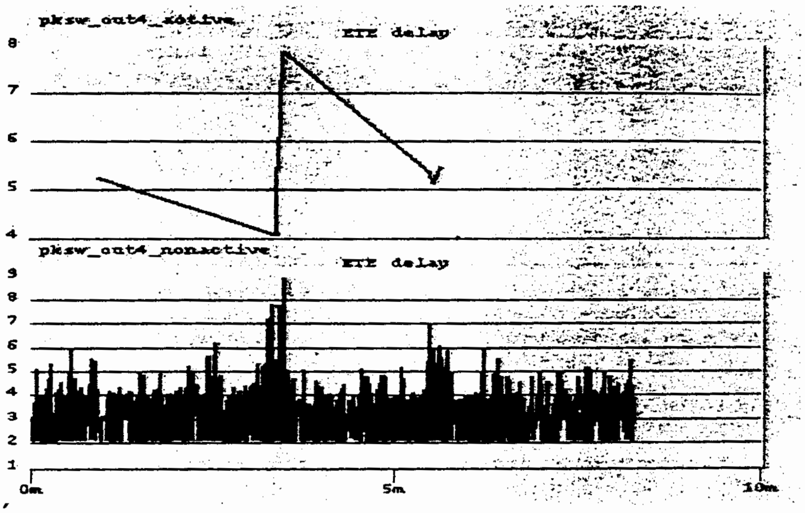


Figure 8: Comparative ETE Delay between the source and the destination of the Active and Non-active Systems

The ETE delay between source and destination is also smaller [Figure 8] in active system because the result of computation is transmitted from the router to the destination. Since in our case the size of the resultant data is smaller the ETE delay is also smaller. The opposite will be true if the result is larger than the raw data. The simulation time also decreases when compared to the non-active system, because the computations are done in parallel. Throughput of the active and non-active systems is the same. In the above graph, the density of the peaks in non-active part is more than the active part because in the active system, fewer numbers of packets are transmitted when compared to non-active system.

5 Limitations Of Our Simulation System Experiment

First we note that although our system is simulated, real computations are performed on the host computer in a separate process from the OPNET simulation, typically executing C programs and making system calls, in order to respond to the OPNET simulation tool. In a concrete (non-simulated) system the parallel computations will be performed on more than one machine.

6 Experimental Analysis Using Real-Time Application System

We have set-up an active network platform with three computers. Two computers each act as an end system and the other computer is treated as a router. The configuration is shown in Figure 9.

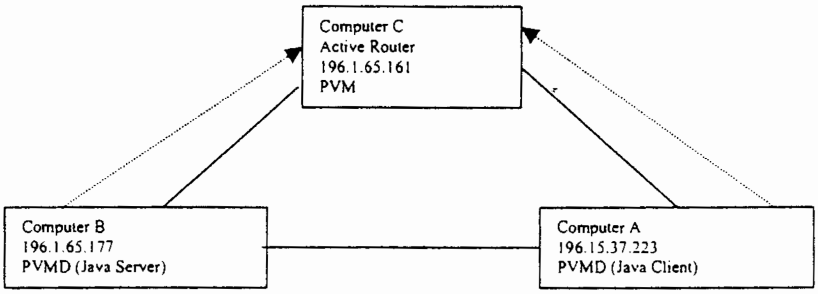


Figure 9: The Configuration Scenario

Although the machines are physically connected through the Ethernet, we arranged that communication be made through the router, which is a Personal Computer (Computer C). PVM resides in the active router and Computer A and Computer B act as PVM daemon.

The current implementation is written in Java and runs as a user-level process under Linux, Windows 98 and Windows NT. We choose Java because of its support for safety and mobility. Other reasons are its flexibility as a high-level language and support of dynamic linking/loading and multi-threading.

The user interface developed is run at both the client and the server side. The server side executes the appropriate protocol (TCP or UDP). The client part finds out the source address by using the Java program. This is done so as to make sure that the user doesn't enter any invalid IP address or false IP address. The user is then asked to supply a destination address, a transmission protocol, a flag setting whether the packet is active or non-active, and if it is an active packet, the application can be run at the router. Then, the packet is formed as shown in Figure 10.

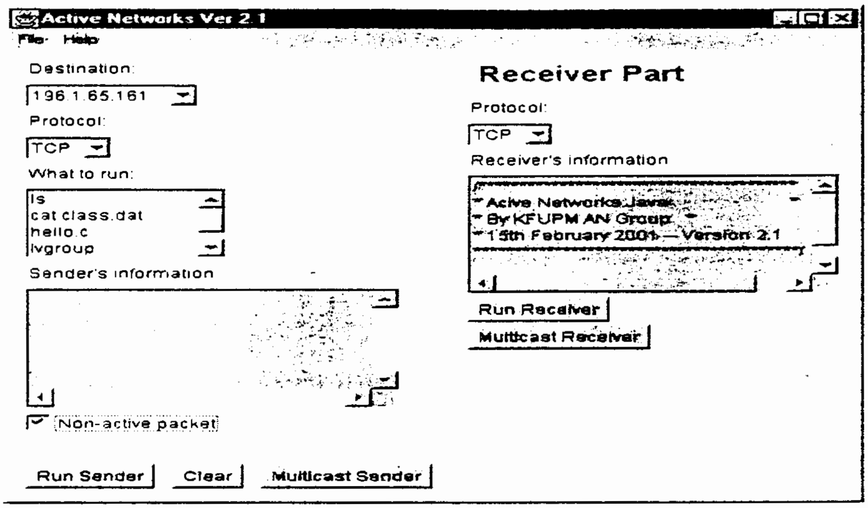
Active / non-active flag	IP address Destination subnet	IP address destination	File Name	File contents

Figure 10: Packet Format

After the user provides the information for the packet, the access control checking is performed. The access control list (ACL) specifies whether a user from the specified source is allowed to access a machine with the respective protocol. To do access control, the source and destination IP addresses along with the protocol [Extended Access Control] are used and verified for the permission or denial of access. Updating the ACL is done by the administrator because a password is required. After access control check, the packet is checked for the next hop using the routing information. The packet is then sent to the next hop using socket application.

Upon arrival of the packet, the router processes the packet and identifies any processing to be done. If there is any processing, it is performed at the router. The main advantage here is that the process currently existing can do both PVM oriented parallel job as well as Matlab applications. Thus a wide variety of applications can be done at the router itself. The processing done is mainly based on the content of the packet. The packet may also contain certain code segments that are copied from the source. These segments could also be processed at the router level. The code pieces that are executed at the router level can be C or Matlab programs or both. The processing of C from Java is done using Java Native Interface.

Our FANS (Friendly Active Network System) user interface consists of a sender part and a receiver part. The initial appearance of the user interface is depicted in Figure 11.



Destination: 196.1.65.161

Protocol: TCP

What to run:
ls
cat class.dat
hello.c
lvgroup

☑ Non-active packet

Protocol: TCP

Receiver's information
* Acive Networks Java
* By KFUPM AN Group
* 15th February 2001 — Version 2.1

Figure 11: FANS User Interface

The active scenario is designed as follows. A user in Computer A wants to do some computation on some data by using an application (for example, Matlab) which does not exist on that machine. The user wants to send the result of the application to another Computer B. In addition, the machine A has neither room to run that application nor to store the result. However, there is a similar application stored in a router, and there is enough room to store the result in Computer B. To accomplish the task, Computer A sends an active packet to the router to activate PVM. The PVM then sets up a virtual connection between Computer A and Computer B. The Matlab runs in parallel at the router and at Computer B. The result is then stored in Computer B. After the result is obtained and stored, the PVM is then deactivated and all the virtual connections are released.

7 Advantages

From our scenario and implementation, we have demonstrated that the combination of Active technology and Parallel processing offers several advantages:

1. It makes optimal use of the resources in a LAN system in a distributed manner.
2. With the help of active networks technology, the machine that has no room to run an application can do so by sending active packet, with an identifier to the requested application, to one of the other machines that has that application and has also enough memory to run the application. In addition, it is even possible to save the result to another machine with huge memory capacity.
3. The application can run faster as it is processed in parallel.
4. The application code can be injected and run in the network only if necessary and therefore efficiently use the resources.
5. Many applications may run simultaneously given that new applications may be released on-the-fly.

8 Conclusion And Further Work

There is increasing demand by users to perform custom computations or to support user-driven services in traditional networks. Such demands are typically realized through implementations in firewalls in an ad hoc manner. This paper proposes the use of active networks to handle user-driven services in a systematic way.

We designed and simulated a 16-node network using OPNET, a repertoire of tools for modelling and simulating computer networks. Our experimental results show a great potential of the use of active networks for congestion control. A concrete system was also built as a follow-up on the simulation. We intend to develop this system further.

A next major research direction is for us to address security issues in both the simulated and the real systems. The issue of security in active networks is paramount especially for the fact that packets can carry code to be executed on the routers.

9 References

[1] Calvert, K.L, Architectural Framework for Active Network version 1.0, Active Network Working Group RFC, University of Kentucky, July 27, 1999

[2] Bhattacharjee, S; Calvert, K.L.; Zegura, E.W., Implementation of an Active Networking Architecture, Technical Report, Georgia Institute of Technology, July 1996

[3] Bhattacharjee, S; Calvert, K.L.; Zegura, E.W., An Architecture for Active Networking, Proc. IEEE INFOCOM '97, April 1997

[4] Wetherall, David J; Guttag, J. V.; Tennenhouse, David L, ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols, Proc. IEEE OPENARCH '98, April 1998

[5] Schwartz, Beverly; Zhou, Wen Yi; Jackson, Alden.W, SmartPackets for Active Networks, BBN Technologies, Jan. 1998

[6] Alexander, Scott D., *et al.*, Active Network Encapsulation Protocol(ANEP), RFC Draft, Juli, 1997

[7] Banch, Albert; *et al*, Multicasting Multimedia Stream with Active Networks, ICSI Technical Report 97-050

[8] Barry Wilkinson and Michael Allen, Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers, Prentice Hall, 1999.