Interconnection Scheme for NOW (Network of Workstations)

Marcel O Odhiambo PhD.

Moi University
Department of Electrical and Communication Engineering
PO BOX 3900 Eldoret, Kenya
Tel. (254-321) 43292
Fax. (254-321) 43292
Email: dcse@net2000ke.com MTRKenya@maf.org

Abstract

Interconnection Scheme for NOW aims to provide a network of hosts (distributed computer system) in which distributed computing can be performed. The distributed system akin to the Internet, consists of a few or several hosts connected together by a logical interconnection network. Hosts on the network are grouped into domains, and the domains are interconnected together to form the network. Hosts in a domain are directly connected to every other host within that domain. Within a domain, a single host acts as a gateway host handling extra-domain messages. To provide the distributed computer system, I implemented Interconnection Scheme with two main functions: provide connectivity among the hosts in the network, and message routing capability between the hosts.

1 Introduction

The Network of Workstations (NOW) consists of several network hosts connected together by an Interconnection Scheme. The NOW is an irregular computer network akin to Internet, and provides a distributed computer system [Coulouris 1994] in which distributed computing can be investigated. The network hosts are grouped together into domains, and the domains are further connected together to form Network of Workstations (NOW). Each domain has a gateway host for routing extra-domain messages. Hosts are assigned host numbers (which translate to hostnames). The destination and source hosts are identified by the host number. Hosts communicate by message-passing. The Interconnection Scheme connects hosts on the network, identifies and marks gateway hosts and builds routing tables at each host. The routing table provide a look-up table for connections (routes) to destination addresses (host). Domain hosts route messages within the domains, however, messages crossing domain boundaries are routed to the gateway host.

2 Interconnection Scheme for NOW

The Interconnection Scheme relies on TCP/IP [Comer 1995] as the transport protocol, using socket stream in the Internet domain. The UNIX socket functions: connect, accept, listen, socket etc have extensively been used in the Interconnection Scheme. The Interconnection Scheme was implemented to connect hosts in HYDRA computer architecture [Marcel 1998]. HYDRA provides a distributed computer architecture for agents processing [Wayner 1995, Jennings 1996, Genesereth 1994]. When the Interconnection Scheme starts at each host, the first step is to get the local hostname, and then, read three system files necessary for starting and successful running of the Interconnection Scheme:

1. config_file - contains names of domain hosts (one config_file for each domain).
2. gateway_file - contains names of gateway hosts (one gateway_file for each domain).
3. network_file - contains names of all hosts in the network.

The Interconnection Scheme is best illustrated by the following list of procedures:

1. Get the local_hostname of the local host. The local_hostname is used by the local host to disqualify itself from identifying itself as a gateway host, and assigning itself a connection index to itself. And, read hostnames in both config_file and gateway_file and count the number of hosts in the two files.
2. Intialise the structure host_array which contains various elements necessary for storing host's information.

0-7803-5546-6/99/$10.00 © 1999 IEEE

```
struct host_info {
  char *host_name;        /* hostname in config_file and gateway_file */
  int  host_addrtype;     /* host address type */
  int  host_addr_length; /* length of host address */
  char *host_addr;        /* host address */
  char *gateway_host;     /* marks host as gateway */
  int  connection;        /* connection index to host */
  int  host_level;        /* host level */
};
host_info *host_arr;
```

The *host_name and connection are used at each host to build a routing table for routing messages to hosts on the network.

3. Create a listening socket by first creating a 'socket" and then, calling 'listen" function to await any event at the listening socket end-point. And, then send a connection requests to domain hosts or accepts connection requests from domain hosts.

4. Read network_file, and fill the elements in the structure host_data

```
struct machine_info {
    char *machine_names;
    int machine_num;
};
machine_info host_data;
```

The read hostnames from the network_file, counted, assigned numbers (machine_num) 1, 2, 3,.....n and *machine_names until all hosts have been assigned. The machine_num is unique for each host, thus, mapping unique host numbers to network hosts.

3 Connection Algorithm

The connection algorithm was developed based on the Interconnection Scheme. The connection algorithm is divided into three parts:

1. Host count - each host gets its local_hostname, and counts the number of hosts in both config and gateway files, this part of the algorithm is represented in Figure 1. The host_count returned is the total number of hosts counted in the two files, and is used during hosts' connection phase. Every host establishes host_count - 2^1 connections to remote hosts.

2. Hosts initialisation - each host reads the the hostnames in both config and gateway files, and fills in the elements of the host_arr structure Figure 2 and Figure 3. The information in host_arr structure is used to build the routing table, to obtain connection index and to establish connection to remote hosts.

3. Connect hosts - reads hostnames from the host_arr structure, and establishes connections to each and every host listed in the host_arr structure. When a connection to a remote has been established, the local host, fills 'connection' in the host_arr structure for the remote host. 'connection' is the descriptor returned after a successful connection to a remote host has been established Figure 4 and Figure 5.

After the hosts for each domain have been successfully connected, each host opens the network_file, reads hostnames (network hosts) from the network_file, and maps hostnames to host numbers starting with PM number 1 (note: host number 0 is not assigned at this stage). The Interconnection Scheme creates a connection for the Access Point (user access or PM 0) at each host. When the user connects to the network, the host to which the user is connected broadcasts its local_machine_num[2] to every host in the network. The hosts in the network store this number as the identity of the host to which the user messages should be sent during a task execution or when sending the results obtained after termination of a task execution to the user.

[1]The host_count returned is 1 more than the total number of hosts in the two files. In addition, a host does not establish a connection to itself

[2]During the mapping of hostnames to PM numbers, if the hostname from the network_file is the same as the local_hostname then, the respective count number is assigned to local_machine_num

0-7803-5546-6/99/$10.00 © 1999 IEEE

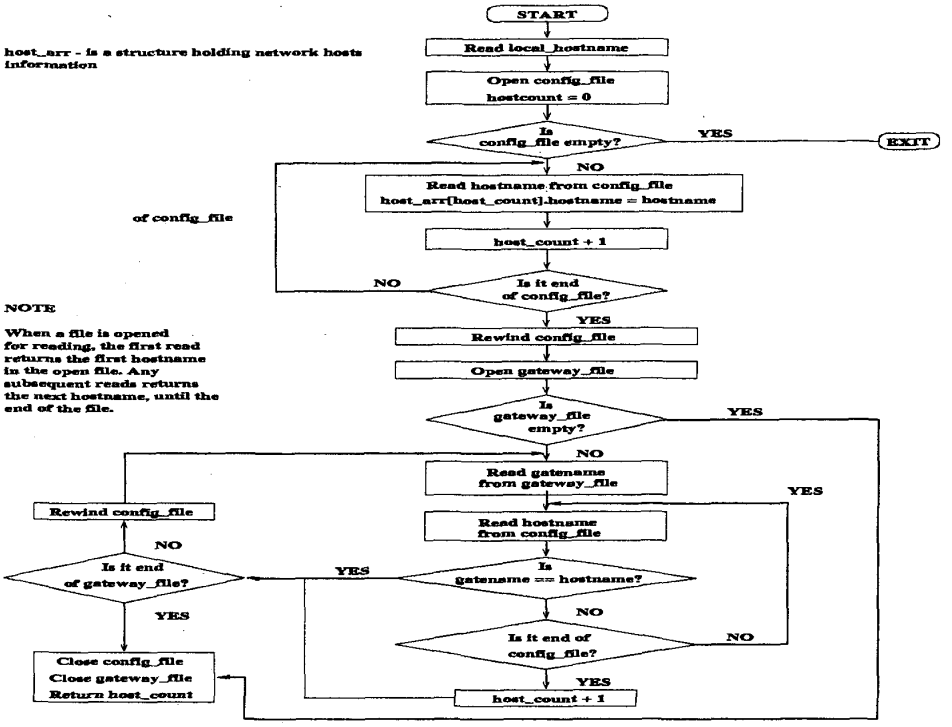

Figure 1: Interconnection algorithm - host count

4 Simulation Test and Results

Interconnection Scheme was used to connect the network Figure 6. In Figure 6(a), contains three hosts in one domain, while Figure 6(b) contains six hosts in two domains. Hosts mearas and cad3 act as gateway hosts for their respective domains. The network in Figure 6(b) was used to implement a distributed computer system consisting of network of hosts, on which successful concurrent agent execution was simulated. Hosts in Figure 6(a) were distributed as follows:

config file	gateway file
mearas.ee.surrey.ac.uk	mearas.ee.surrey.ac.uk
leod.ee.surrey.ac.uk	
felarof.ee.surrey.ac.uk	

Hosts in Figure 6(b) were distributed as follows:

config file (1)	gateway file
mearas.ee.surrey.ac.uk	mearas.ee.surrey.ac.uk
leod.ee.surrey.ac.uk	cad3.ee.surrey.ac.uk
felarof.ee.surrey.ac.uk	
config file (2)	**gateway file**
cad1.ee.surrey.ac.uk	mearas.ee.surrey.ac.uk
cad3.ee.surrey.ac.uk	cad3.ee.surrey.ac.uk
cad4.ee.surrey.ac.uk	

The user connected to host felarof in Figure 6(b) was able to execute commands in the hosts in domain 2. This is only possible if the hosts mearas and cad3 acted as gateway hosts for their respective domains in routing extra-domain messages. In addition to Interconnection Scheme connecting the hosts in Figure 6(b), messages (agents) were correctly routed to the hosts in the network. The obtained test Figure 7 show that the Interconnection Scheme was able to connect the hosts in Figure 6(b), and to route agents between the hosts. The Interconnection Scheme is implemented as Network Router. the Routing Module only starts executing after successful interconnection domain hosts.

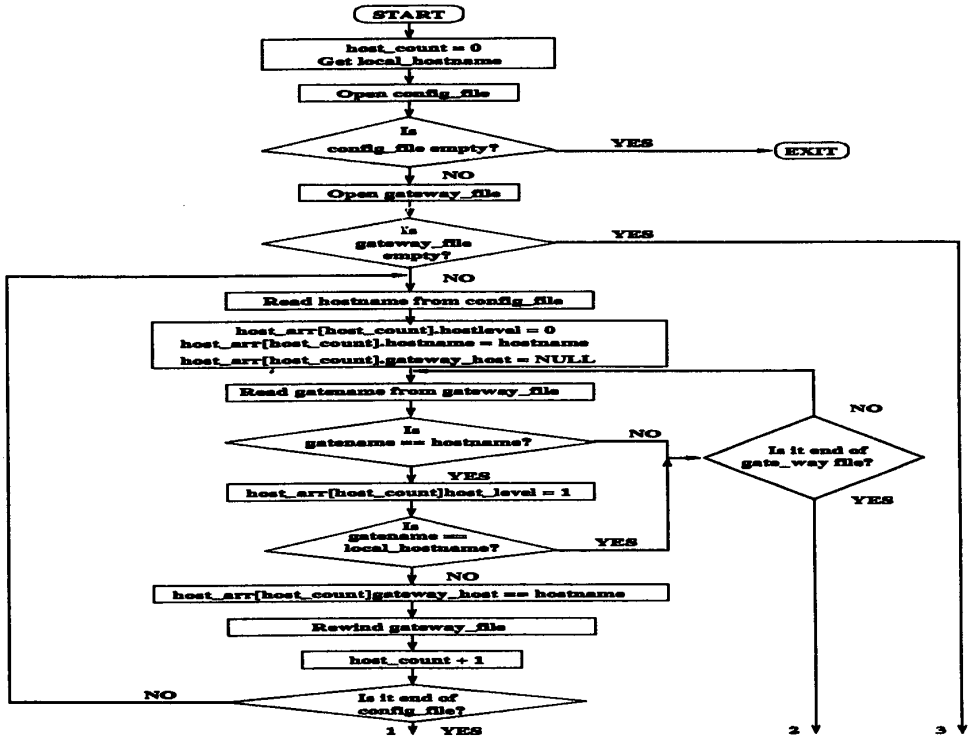

Figure 2: Interconnection algorithm - host initialisation

5 Conclusion

The simulation test results obtained indicate that the Interconnection Scheme can be used to connect hosts into a network, and to route messages (agents) between the hosts. Thus, providing a distributed computer system to support distributed computing. The Interconnection Scheme can therefore be used to implement a low cost distributed computer system consisting of heterogeneous hosts to provide distributed computing facilities. In distibuted systems, it is difficult to talk about message latency measures because hosts in the network can be up or down without the user's knowledge. The Interconnection Scheme was therefore designed without paying attention to message latency measures. The Interconnection Scheme relied on the transport protocol TCP/IP to provide acceptable message latency. However, hosts interconnection and routing messages to correct destination (hosts) was considered to be crucial in distributed systems. The simulation results show that Interconnection Scheme achieved the aim for which it was designed.

0-7803-5546-6/99/$10.00 © 1999 IEEE

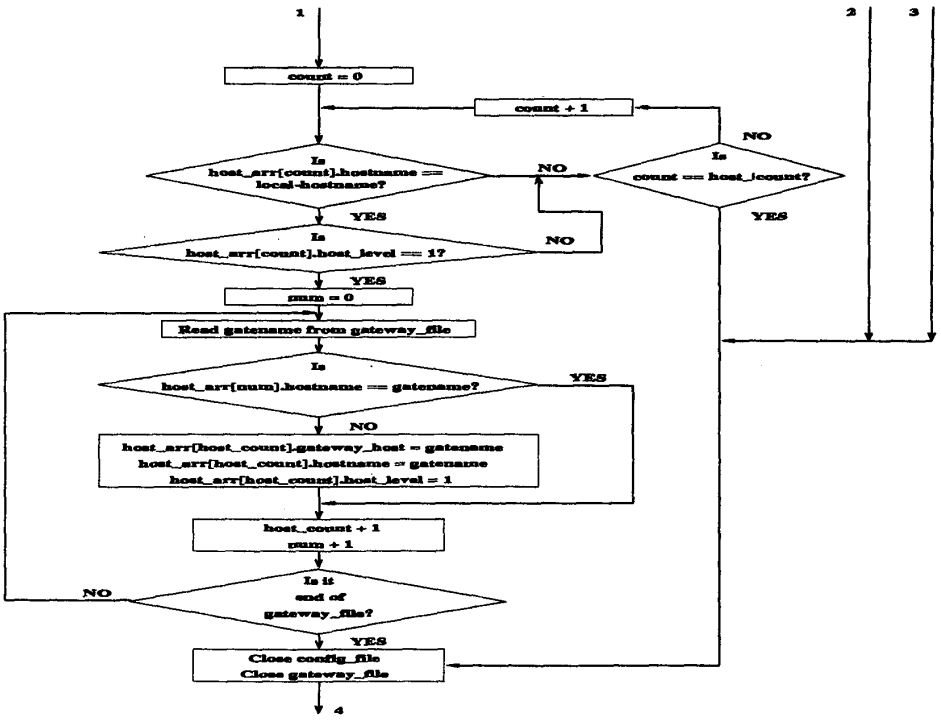

Figure 3: Interconnection algorithm - host initialisation cont.

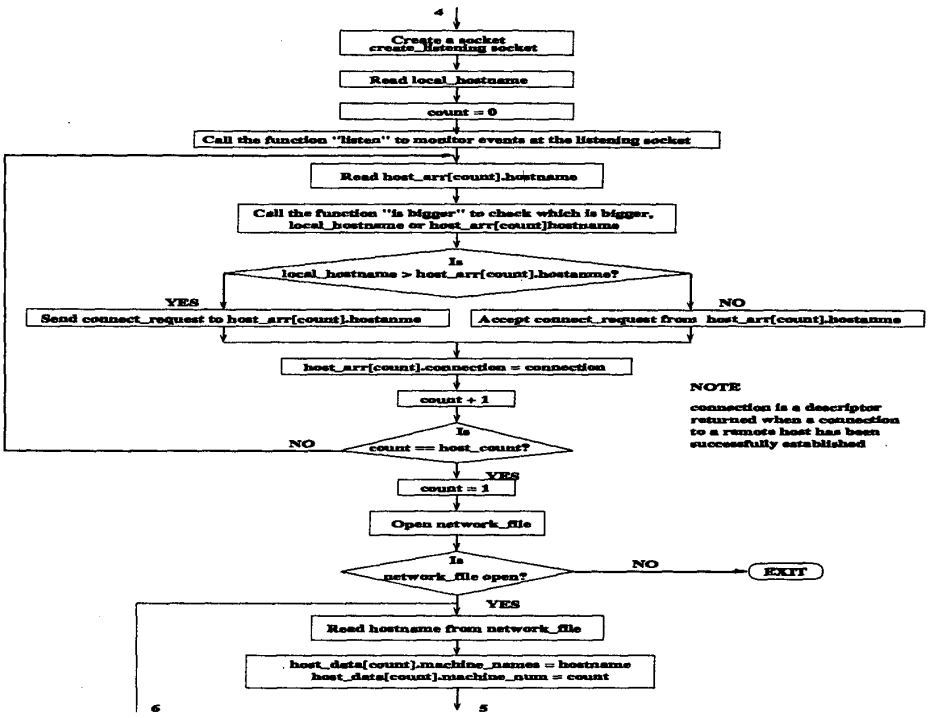

Figure 4: Interconnection algorithm - host connect

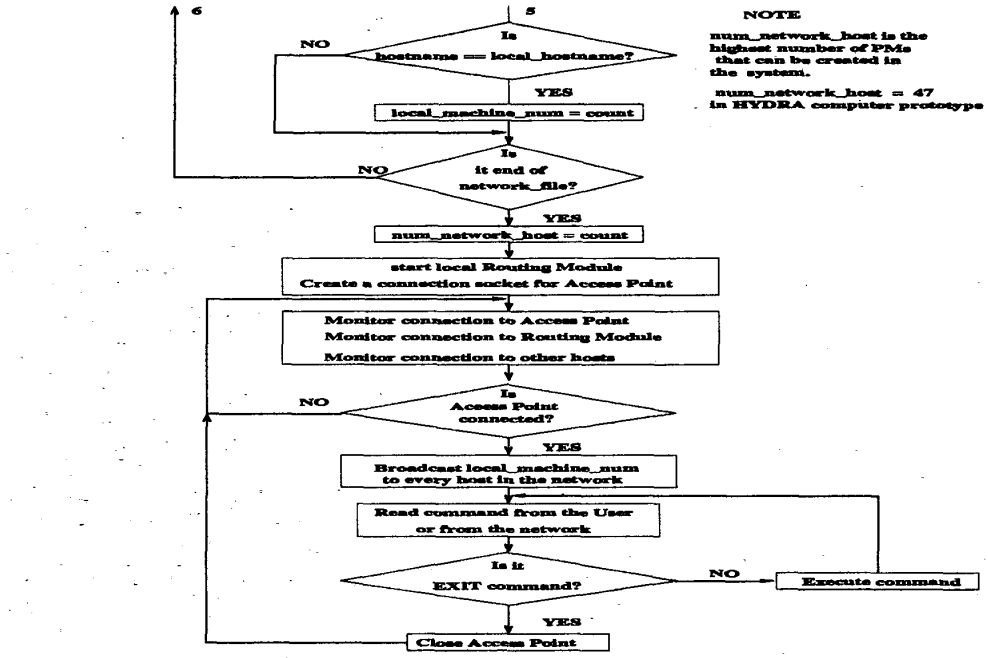

Figure 5: Interconnection algorithm - host connect cont.

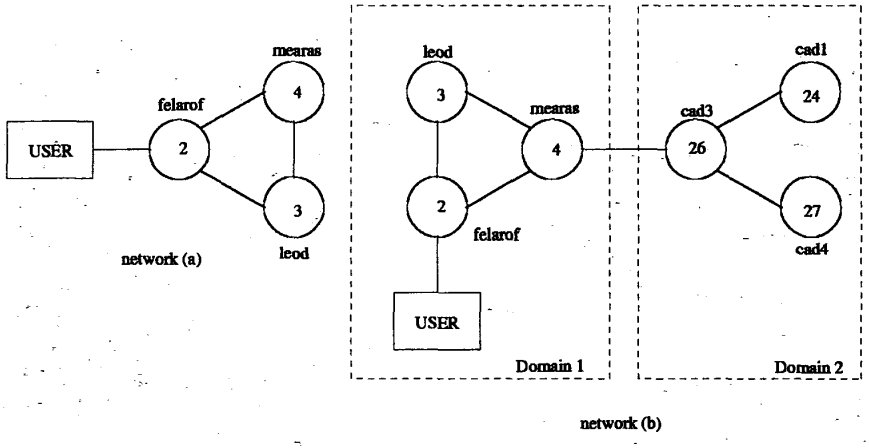

Figure 6: Hosts connectivity in the network

router.log

Network Router started on host number 26
Sun May 25 21:54:54 1997
Network Router started on host number 24
Sun May 25 21:54:55 1997
Network Router started on host number 27
Sun May 25 21:54:55 1997
Network Router started on host number 3
Sun May 25 21:54:55 1997
Network Router started on host number 2
Sun May 25 21:54:56 1997
Network Router started on host number 4
Sun May 25 21:59:26 1997
Routing Module started on host number 26
Sun May 25 21:54:55 1997
Routing Module started on host number 24
Sun May 25 21:54:56 1997
Routing Module started on host number 27
Sun May 25 21:54:55 1997

Routing Module started on host number 2
Sun May 25 22:03:56 1997
Routing Module started on host number 3
Sun May 25 22:03:57 1997
Routing Module started on host number 4
Sun May 25 22:04:32 1997
Successful data transfer to PM: 2
Sun May 25 22:09:19 1997
Successful data transfer to PM: 3
Sun May 25 22:09:27 1997
Successful data transfer to PM: 4
Sun May 25 22:09:38 1997
Successful data transfer to PM: 24
Sun May 25 22:10:34 1997
Successful data transfer to PM: 26
Sun May 25 22:10:49 1997
Successful data transfer to PM: 27
Sun May 25 22:10:59 1997

Figure 7: Network Router and Routing Module log files (6 hosts)

References

[Comer 1995] COMER DOUGLAS, E. *Internetworking with TCP/IP, Vol. 1, Principles, 3rd editionProtocols and Architecture.* Prentice-Hall, Inc. Englewood Cliffs, New Jersey 07632, 1995.

[Coulouris 1994] COULOURIS, G. .F., DOLLIMORE, J., KINDBERG, K. *Distributed Systems: Concepts and Design.* Addison-Wesley Publishing Company, 1994.

[Genesereth 1994] GENESERETH, MICHAEL R., and KETCHPE, STEVEN P. Software Agents. *Comms. of the ACM, July 1994, Vol.37, No.7, pp. 48—54.* 1994.

[Jennings 1996] JENNINGS, NICK and WOOLDRIDGE, MICHAEL. Software Agents. *IEE Review, January 1996, pp. 17—20*

[Marcel 1998] ODHIAMBO MARCEL O. Parallel and Distributed Swarm Computer Architecture. *A Dissertation for the degree of Doctorate of Philosophy, School of Electronic Engineering, Information Technology and Mathematics, University of Surrey. May 26, 1998.*

[Wayner 1995] WAYNER, PETER. *AGENTS UNLEASHED A Public Domain Look at Agent Technology.* AP PROFESSIONAL 1995.