Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 4, April 2015, pg.195 – 199

RESEARCH ARTICLE

Periodic LSA Broadcasting for Monitoring Resource Availability and Sharing in a Network of Hosts

M Priyadharsini[1], M S Pratheepa[2], G Shree Sharumathi[3], N Srividya[4]

[1]Assistant Professor [Sr.Gr], Information Technology, Sri Ramakrishna Engineering College, Coimbatore, India

[2]Information Technology, Sri Ramakrishna Engineering College, Coimbatore, India

[3]Information Technology, Sri Ramakrishna Engineering College, Coimbatore, India

[4]Information Technology, Sri Ramakrishna Engineering College, Coimbatore, India

[1] priya.murugesan@srec.ac.in; [2] pratheepa93@gmail.com; [3] sharugopal17@gmail.com; [4] jabbu1993@gmail.com

Abstract— Hosting of computing infrastructure in the internet provides scalable and on demand resources. Similarly adding each hosts resource in the network of hosts provides better quality of services. A Resource Monitor is created in the network that contains the available resource information of all hosts. Each host periodically broadcasts the resource information using Link State Advertisement (LSA) packets that are encapsulated within UDP packets. LSA packet contains routing information and the hosts' current resource information (CPU, RAM). These LSA packets are broadcasted periodically using timers when the resources are available. When the hosts resources are low it sends request message using a TCP packets to the Resource Monitor asking for the required amount of resources. Then Resource Monitor sends response message to the host using a TCP packet if the required amount of resource is available in some other host. This way Resource Monitor ensures that it has up-to-date information of resources in the host and supports sharing of resources.

Keywords— Resource Monitor, host's resources, broadcasting, resource sharing

I. INTRODUCTION

The current scenarios in networking require optimized use of the available resources. A resource optimisation scheme involves resource monitoring and resource sharing. Many such resource optimisation schemes are available. Simple schemes for tracking availability of resources by employing a head node and analysing the policy that govern resource sharing among hosts are some of the resource optimization schemes that are in existence. In this paper, we have proposed an efficient scheme for resource monitoring and sharing. Any host in the network can act as a Resource Monitor. The jobs of the Resource Monitor are maintaining the current resource availability details of all hosts it is destined to monitor, handling request messages for resources and providing appropriate response messages for resource sharing. In this way, the resource monitor is solely responsible for all the activities in the network.

II. RELATED WORK

There are many existing resource monitoring schemes like creation of a network embedded cloud also called as a carrier cloud. A network embedded cloud is a distributed cloud architecture in which the network providers add computing and storage resources to their existing network infrastructure in order to offer cloud services and resource sharing [1]. Sharing of resources across multiple sites or enterprises brings up several important problems. In order to preserve autonomy of each site, it is necessary that all decisions about the local resources should be made by site manager. An analysis tool is proposed to determine the various policies (when or how can a resource be borrowed or donated and what resources can be borrowed or donated) that govern sharing of resources across multiple enterprises [2].

In some other monitoring schemes, a control method is proposed for optimizing CPU usage under changing workloads in a virtualized server environment. Based on the reserved CPU share, request arrival time and processing time of an application, the available CPU capacity is shared among applications while taking Service Level Agreement (SLA) of the application into consideration [3].

For quickly locating the available resources a head node is created to non-intrusively identify nodes that have resources in a cluster of nodes. The head node knows memory utilization of all nodes connected to it. This technique allows establishing a correlation between the memory load and timely response of network traffic from a node [4].

We propose an efficient and simpler method for resource monitoring and sharing by employing TCP and UDP packets. The entire implementation is done using C programming language.

III.EMULATING AND CONFIGURING NETWORK OF HOSTS

Emulation of the network is done using Mininet network emulator. It runs a collection of hosts, switches, routers and links on a single Linux kernel. It uses light weight virtualization to make a single system look like a complete network. Mininet's virtual hosts, switches, links and controllers are real things that are created using software than hardware. The hosts in the network are configured with virtual RAM resources, IP addresses, default gateways and the routers are configured with OSPF (Open Shortest Path Routing) protocol using a Quagga. Quagga is a network routing software suite for implementation of many routing protocols like OSPF.OSPF is link state routing protocol of the interior gateway routing protocol type.

IV. RESOURCE MONITORING AND PERIODIC LSA BROADCASTING

The resource to be monitored here is host's RAM. Every time a request for some application is made (examples: start browser, start ftp) the memory required for processing these applications is deducted from the available memory in the host. Until the host resources are more than the memory required for processing application user can request for applications. Each time an application is processed a UDP packet is broadcast periodically that contains the LSA packet encapsulated in the payload portion (Fig. 1). The LSA packet contains information like the advertising hosts IP address and the current resource information. Since the UDP packets are broadcasted, all the hosts in the network receive the current resource information of host as long as the host has sufficient resource. When the broadcast UDP packets are sent, the current resource information and IP address of host are written into a simple log file. The log file contains recent resource information as broadcasted by the UDP sender.

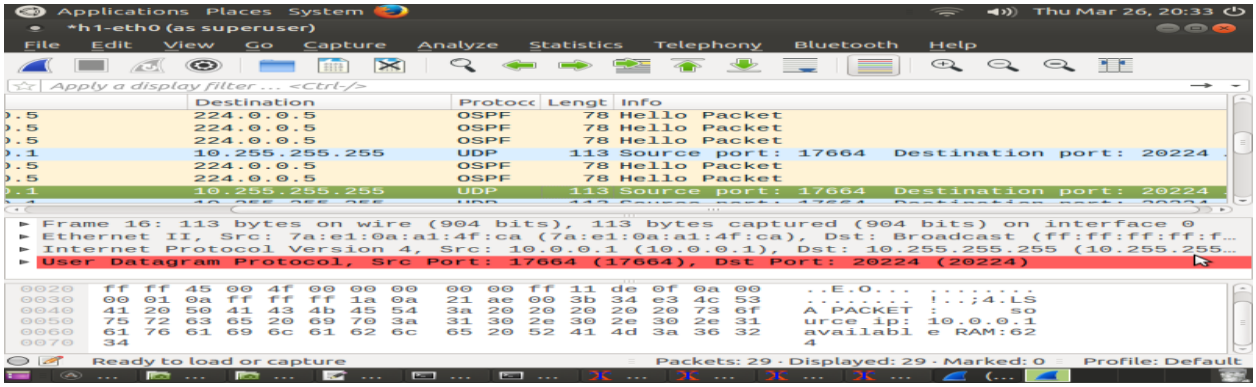

Fig. 1. Broadcasted UDP packets

V. RESOURCE SHARING AMONG HOSTS

The UDP packets are sent as long as the required resources are less than the available resource in host. As soon as the resources are not sufficient a request for the required resource is sent to the Resource Monitor in the network. For this purpose, a TCP packet is created with the amount of resource requested which is encapsulated in the payload portion of the TCP packet (Fig. 2). The Resource Monitor uses the log file created while sending the UDP packets to look up for a host with the required amount of resource. If it finds some other host has the required amount of the resource it sends a response message that contains IP address of the host with the required resource. For this purpose, again a TCP packet is created with the payload containing the details of available host resources. In this way, resources availability information is continuously updated using UDP broadcast and request for resource is carried out through TCP packets.

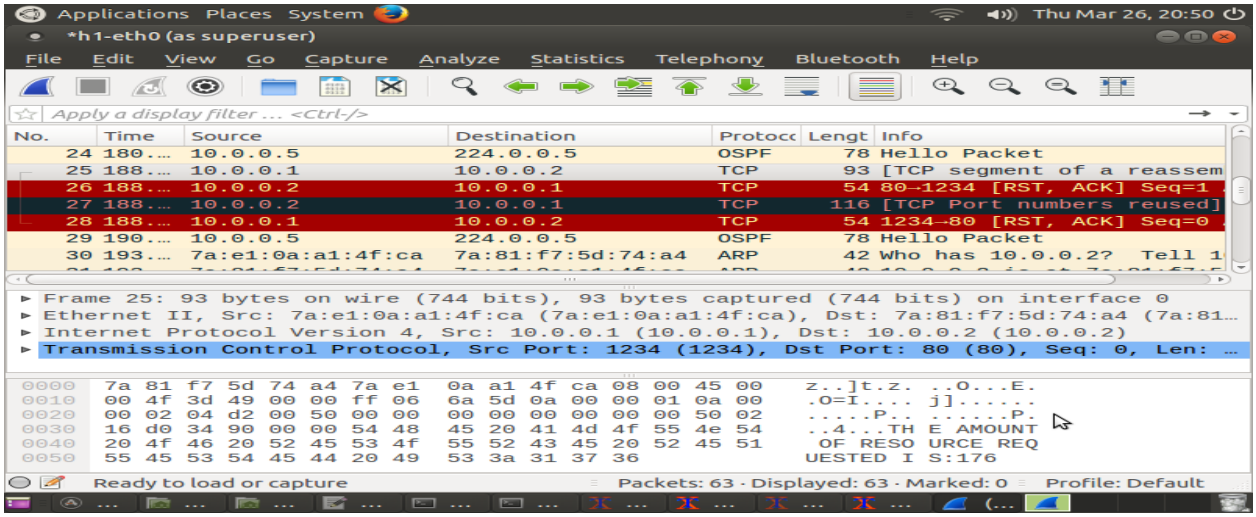

Fig. 2. TCP Request Message

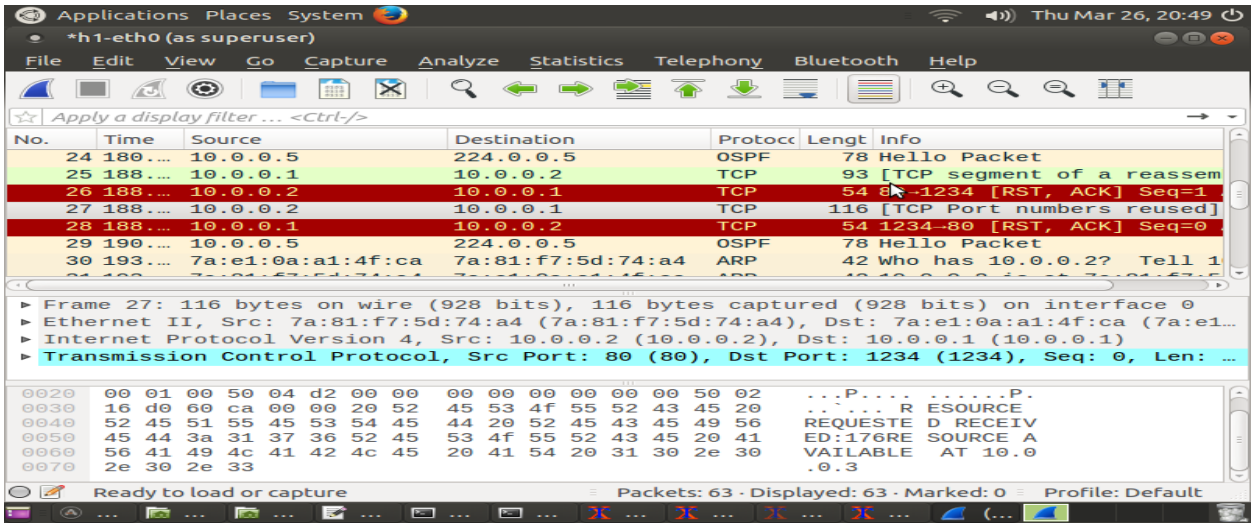

Fig. 3. TCP Response Message

VI. CONCLUSION

An efficient mechanism for resource monitoring and sharing of resources among the hosts is proposed. The Resource Monitor has current information of all the available resources in the network of hosts. A simple log file is used to maintain the record of available resources. Similarly, when a host needs additional resource it sends request to the Resource Monitor. The Resource Monitor examines the log file to send response with the available resource location to the requesting host. In this way, all the unused resource in the network are being optimized and shared when needed. By managing resources effectively in the network quality of service is improved.

The mechanism for resource monitoring and sharing has been carried out in a network. To make this very efficient in a large network, the log files containing the current resource information can be hosted in a cloud. When details are hosted in cloud better resource management and sharing can be done globally. Numerous hosts will be available to share their resource and there will not be any memory constraints problem. Global resource optimization can be done with less cost spent in local monitoring of individual hosts and unused memory can be used efficiently.

REFERENCES

[1] Amir Roozbeh and Azimeh Sefidcon, "Resource Monitoring in a Network Embedded Cloud: An extension to OSPF-TE," Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing, 2013.

[2] Mudhakar Srivatsa, Nithya Rajamani, Murthy Devarakonda," A Policy Evaluation Tool for Multisite Resource Management," IEEE Transactions on Parallel and Distributed Systems, Vol. 19, No. 10, October 2008.

[3] Martin Ansbjerg Kjaer, Maria Kihl, and Anders Robertsson, "Resource Allocation and Disturbance Rejection in Web Servers using SLAs and Virtualized Servers," IEEE Transactions on Network Service Management, Vol. 6,No. 4,December 2009.

[4] Lanier Watkins, Robinson, Raheem Beyah, "A Passive Solution to the Memory Resource Discovery Problem in Computational Clusters," IEEE Transactions on Network and Service Management, Vol. 7, No. 4, December 2010.

[5] A. V. Aho and D. Lee, "Hierarchical networks and the LSA Nsquared problem in OSPF routing," in IEEE Global Telecommunications Conference, 2000. GLOBECOM '00, vol. 1, 2000, pp. 397–404

