Scientific Journal of Impact Factor (SJIF): 5.71

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 5, Issue 04, April -2018

ENHANCING THE LOAD BALANCING IN DATA FLOW TRANSMISSION FOR DATA CENTERS BY USING AN BINARY HEAP ROUTING PROTOCOL (BHRP)

Dr. M.Mohamed Sirajudeen[1], (Dr).D.Sathyanarayanan[2]

[1]Department of Information Technology, Faculty of Informatics, University of Gondar, Ethiopia. North East Africa,
[2]Department of Computer Science, Faculty of Informatics, University of Gondar, Ethiopia. North East Africa,

ABSTRACT:*Wireless communication is the emerging computing paradigm in the modern era. It makes the utilizationof computing resources in an effective and efficient manner. Different researchers endow with features of cloud services in their standpoint at each and every occasion. In general either a connection-oriented communication (wired) or Connection-less communication (wireless), it should emphasize the term "Traffic" in the packet flow among different service nodes. Due to the heavy traffic, entire transmission faces severe injury among the nodes. If the packets are waiting more in a particular transmission node, there is a chance for packet loss. It causes a serious situation and question marks for receiving end. At the same time, during the data transmission some of the access points are fully engaged (traffic) and few other existing access points are fully ideal, because of uneven data transmission. In order to avoid this imbalance workload among the access points as well as to engage all the access points with even load distribution by using the proposed algorithm in this research work named as "Orthogonal Intelligent Routing Protocol (OIRP)".*

Key Words*: Orthogonal, Routing, traffic, workload and transmission.*

I. INTRODUCTION

The data center is home to the computational power, storage, and applications necessary to support an enterprise business. The data center infrastructure is central to the IT architecture, from which all content is sourced or passes through. Proper planning of the data center infrastructure design is critical, and performance, resiliency, and scalability need to be carefully considered.Another important aspect of the data center design is flexibility in quickly deploying and supporting new services. Designing a flexible architecture that has the ability to support new applications in a short time frame can result in a significant competitive advantage. Such a design requires solid initial planning and thoughtful consideration in the areas of port density, access layer uplink bandwidth, true server capacity, and oversubscription, to name just a few.The data center network design is based on a proven layered approach, which has been tested and improved over the past several years in some of the largest data center implementations in the world. The layered approach is the basic foundation of the data center design that seeks to improve scalability, performance, flexibility, resiliency, and

maintenance. The following figure (1) shows the basic layered design.The layers of the data center design are the core, aggregation, and access layers. These layers are referred to extensively throughout this guide and are briefly described as follows: Core layer—provides the high-speed packet switching backplane for all flows going in and out of the data center. The core layer provides connectivity to multiple aggregation modules and provides a resilient Layer 3 routed fabric with no single point of failure. The core layer runs an interior routing protocol, such as OSPF or EIGRP, and load balances traffic between the campus core and aggregation layers using Cisco Express Forwarding-based hashing algorithms. Aggregation layer modules—Provide important functions, such as service module integration, Layer 2 domain definitions, spanning tree processing, and default gateway redundancy. Server-to-server multi-tier traffic flows through the aggregation layer and can use services, such as firewall and server load balancing, to optimize and secure applications. The smaller icons within the aggregation layer switch in figure (1) represent the integrated service modules. These modules provide services, such as content switching, firewall, SSL offload, intrusion detection, network analysis, and more.

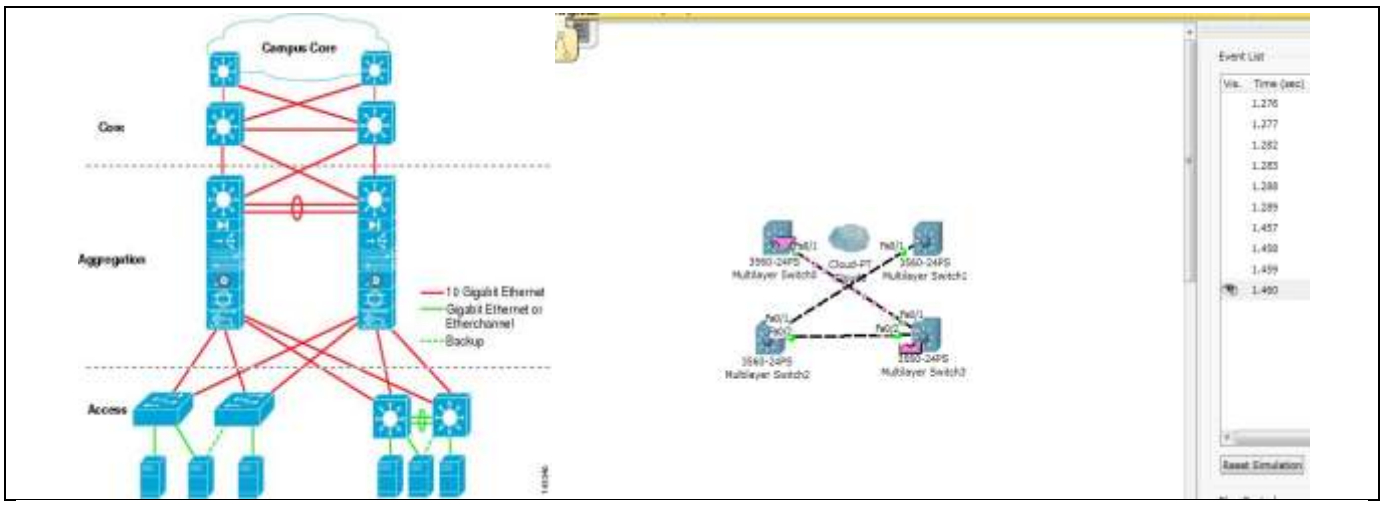

Figure 1-1 Basic Layered Design

Access layer—where the servers physically attach to the network. The server components consist of 1RU servers, blade servers with integral switches, blade servers with pass-through cabling, clustered servers, and mainframes with OSA adapters. The access layer network infrastructure consists of modular switches, fixed configuration 1 or 2RU switches, and integral blade server switches. Switches provide both Layer 2 and Layer 3 topologies, fulfilling the various servers broadcast domain or administrative requirements.

The multi-tier model is the most common design in the enterprise. It is based on the web, application, and database layered design supporting commerce and enterprise business ERP and CRM solutions. This type of design supports many web service architectures, such as those based on Microsoft .NET or Java 2 Enterprise Edition. These web service application environments are used by ERP and CRM solutions from Siebel and Oracle, to name a few.

The multi-tier model relies on security and application optimization services to be provided in the network.The server cluster model has grown out of the university and scientific community to emerge across enterprise business verticals including financial, manufacturing, and entertainment. The server cluster model is most commonly associated with high-performance computing (HPC), parallel computing, and high-throughput computing (HTC) environments, but can also be associated with grid/utility computing. These designs are typically based on customized, and sometimes proprietary, application architectures that are built to serve particular business objectives.

The multi-tier data center model is dominated by HTTP-based applications in a multi-tier approach. The multi-tier approach includes web, application, and database tiers of servers. Today, most web-based applications are built as multi-tier applications. The multi-tier model uses software that runs as separate processes on the same machine using interprocess communication (IPC) or on different machines with communications over the network. Typically, the following three tiers are used:Web-server,Application and Database. Multi-tier server farms built with processes running on separate machines can provide improved resiliency and security. Resiliency is improved because a server can be taken out of service while the same function is still provided by another server belonging to the same application tier.

Security is improved because an attacker can compromise a web server without gaining access to the application or database servers. Web and application servers can coexist on a common physical server; the database typically remains separate.Resiliency is achieved by load balancing the network traffic between the tiers, and security is achieved by placing firewalls between the tiers. You can achieve segregation between the tiers by deploying a separate infrastructure composed of aggregation and access switches, or by using VLANs show by the following figure 2.

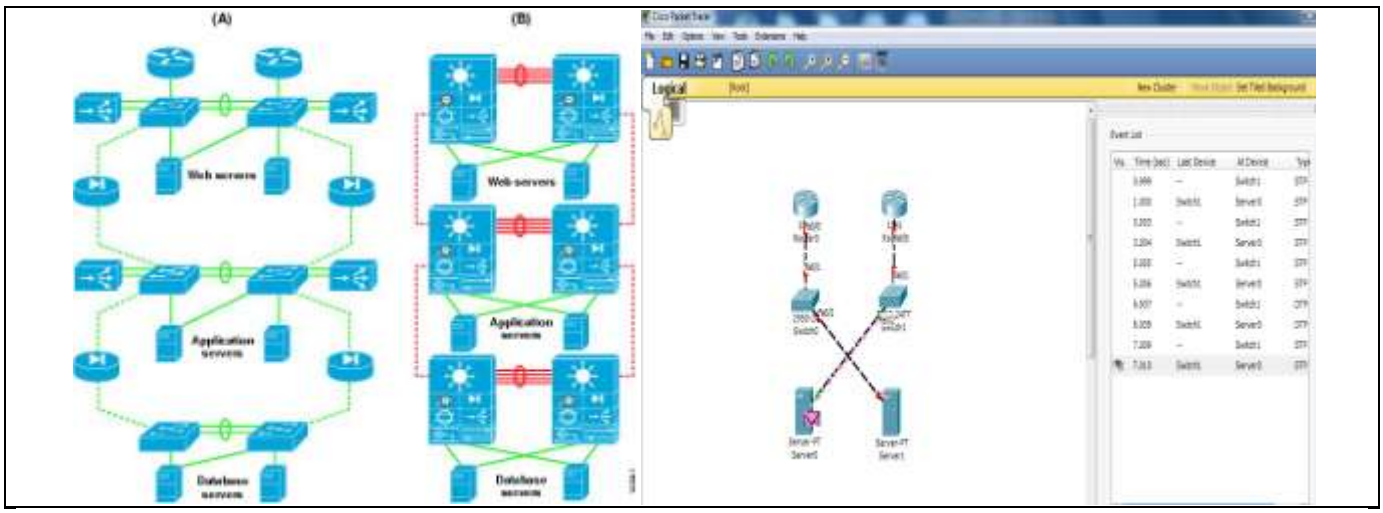

Figure 2 Physical Segregation in a Server Farm with Appliances (A) and Service Modules (B)

The design shown in figure (2) uses VLANs to segregate the server farms. The left side of the illustration (A) shows the physical topology, and the right side (B) shows the VLAN allocation across the service modules, firewall, load balancer, and switch. The firewall and load balancer, which are VLAN-aware, enforce the VLAN segregation between the server farms. Note that not all of the VLANs require load balancing. For example, the database in the example sends traffic directly to the firewall.

II. RELATED WORK

In the modern data center environment, clusters of servers are used for many purposes, including high availability, load balancing, and increased computational power. This guide focuses on the high performance form of clusters, which includes many forms. All clusters have the common goal of combining multiple CPUs to appear as a unified high performance system using special software and high-speed network interconnects. Server clusters have historically been associated with university research, scientific laboratories, and military research for unique applications, such as the following:Meteorology (weather simulation),Seismology (seismic analysis) and Military research (weapons, warfare) .Server clusters are now in the enterprise because the benefits of clustering technology are now being applied to a broader range of applications. The following applications in the enterprise are driving this requirement:Financial trending analysis—Real-time bond price analysis and historical trending, Film animation—Rendering of artist multi-gigabyte files,Manufacturing—Automotive design modeling and aerodynamics, Search engines—Quick parallel lookup plus content insertion. In the enterprise, developers are increasingly requesting higher bandwidth and lower latency for a growing number of applications. The time-to-market implications related to these applications can result in a tremendous competitive advantage. For example, the cluster performance can directly affect getting a film to market for the holiday season or providing financial management customers with historical trending information during a market shift.

In the high performance computing landscape, various High Performance Clusters (HPC) types exist and various interconnect technologies are used. The majority of interconnect technologies used today are based on Fast Ethernet and Gigabit Ethernet, but a growing number of specialty interconnects exist, for example including Infiniband and Myrinet. Specialty interconnects such as Infiniband have very low latency and high bandwidth switching characteristics when compared to traditional Ethernet, and leverage built-in support for Remote Direct Memory Access (RDMA). 10GE NICs have also recently emerged that introduce TCP/IP offload engines that provide similar performance to Infiniband.Although high performance clusters (HPCs) come in various types and sizes, the following categorizes three main types that exist in the enterprise environment:HPC type 1—Parallel message passing (also known as tightly coupled) and Applications run on all compute nodes simultaneously in parallel.A master node determines input processing for each compute node. The large or small cluster, broken down into hives and (for example, 1000 servers over 20 hives) with IPC communication between compute nodes/hives. HPC type 2—Distributed I/O processing (for example, search engines), the client request is balanced across master nodes, and then sprayed to compute nodes for parallel processing (typically unicast at present, with a move towards multicast).This type obtains the quickest response, applies content insertion (advertising), and sends to the client.HPC Type 3—parallelfiles processing (also known as loosely coupled). The source data file is divided up and distributed across the compute pool for manipulation in parallel. Processed components are rejoined after completion and written to storage.Middleware controls the job management process (for example, platform linear file system [LFS]).The traditional

high performance computing cluster that emerged out of the university and military environments was based on the type 1 cluster. The new enterprise HPC applications are more aligned with HPC types 3 and 4, supporting the entertainment, financial, and a growing number of other vertical industries.

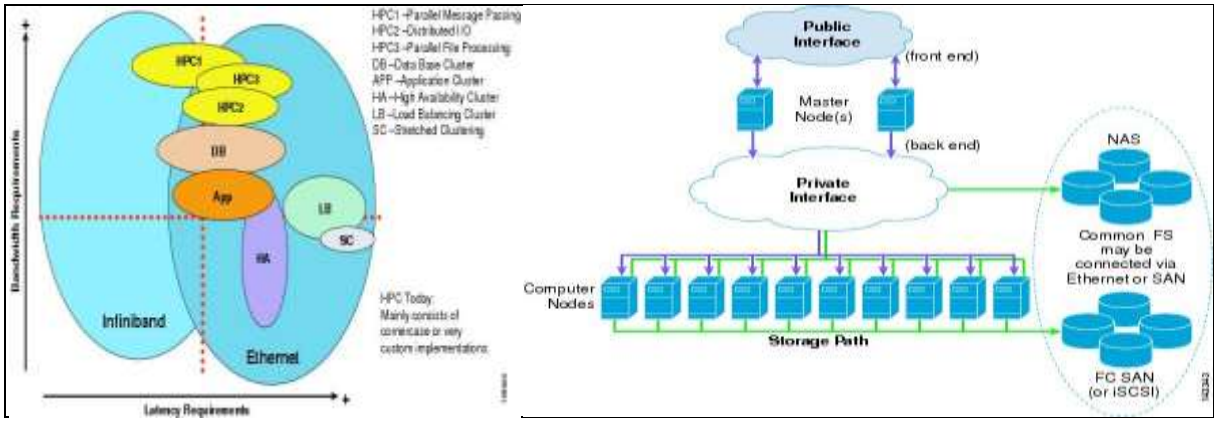

Figure 3server cluster and Logical View

The components of the server cluster are as follows:Front end—these interfaces are used for external access to the cluster, which can be accessed by application servers or users that are submitting jobs or retrieving job results from the cluster. An example is an artist who is submitting a file for rendering or retrieving an already rendered result. This is typically an Ethernet IP interface connected into the access layer of the existing server farm infrastructure.Master nodes (also known as head node)—the master nodes are responsible for managing the compute nodes in the cluster and optimizing the overall compute capacity. Usually, the master node is the only node that communicates with the outside world. Clustering middleware running on the master nodes provides the tools for resource management, job scheduling, and node state monitoring of the computer nodes in the cluster.

Master nodes are typically deployed in a redundant fashion and are usually a higher performing server than the compute nodes.Back-end high-speed fabric—this high-speed fabric is the primary medium for master node to compute node and inter-compute node communications. Typical requirements include low latency and high bandwidth and can also include jumbo frame and 10 GigE support. Gigabit Ethernet is the most popular fabric technology in use today for server cluster implementations, but other technologies show promise, particularly Infiniband.Compute nodes—the compute node runs an optimized or full OS kernel and is primarily responsible for CPU-intense operations such as number crunching, rendering, compiling, or other file manipulation.Storage path—the storage path can use Ethernet or Fiber Channel interfaces. Fiber Channel interfaces consist of 1/2/4G interfaces and usually connect into a SAN switch such as a Cisco MDS platform. The back-end high-speed fabric and storage path can also be a common transport medium when IP over Ethernet is used to access storage. Typically, this is for NFS or iSCSI protocols to a NAS or SAN gateway, such as the IPS module on a Cisco MDS platform.Common file system—The server cluster uses a common parallel file system that allows high performance access to all compute nodes. The file system types vary by operating system (for example, PVFS or Luster).

Server cluster designs can vary significantly from one to another, but certain items are common, such as the following:Commodity off the Shelf (CotS) server hardware—the majority of server cluster implementations are based on 1RU Intel- or AMD-based servers with single/dual processors. The spiraling cost of these high performing 32/64-bit low density servers has contributed to the recent enterprise adoption of cluster technology.GigE or 10 GigE NIC cards—the applications in a server cluster can be bandwidth intensive and have the capability to burst at a high rate when necessary. The PCI-X or PCI-Express NIC cards provide a high-speed transfer bus speed and use large amounts of memory. TCP/IP offload and RDMA technologies are also used to increase performance while reducing CPU utilization.Low latency hardware—usually a primary concern of developers is related to the message-passing interface delay affecting the overall cluster/application performance. This is not always the case because some clusters are more focused on high throughput, and latency does not significantly impact the applications. The Cisco Catalyst 6500 with distributed forwarding and the Catalyst 4948-10G provide consistent latency values necessary for server cluster environments.Non-blocking or low-over-subscribed switch fabric—Many HPC applications are bandwidth-intensive with large quantities of data transfer and interprocess communications between compute nodes. GE attached server oversubscription ratios of 2.5:1 (500 Mbps) up to 8:1(125 Mbps) are common in large server cluster designs.Mesh/partial mesh connectivity—Server cluster designs usually require a mesh or partial mesh fabric to permit communication between all nodes in the cluster. This mesh fabric is used to share state, data, and other information between master-to-compute and compute-to-compute servers in the cluster.Jumbo frame

supports—Many HPC applications use large frame sizes that exceed the 1500 byte Ethernet standard. The ability to send large frames (called jumbos) that are up to 9K in size provides advantages in the areas of server CPU overhead, transmission overhead, and files transfer time.

III. PROPOSED WORK

In computer science, a binary tree is a tree data structure in which each node has at most two children, which are referred to as the left child and the right child. A recursive definition using just set theory notions is that a (non-empty) binary tree is a tuple (L, S, R), where L and R are binary trees or the empty set and S is a singleton set.Some authors allow the binary tree to be the empty set as well shown by the following figure 4.

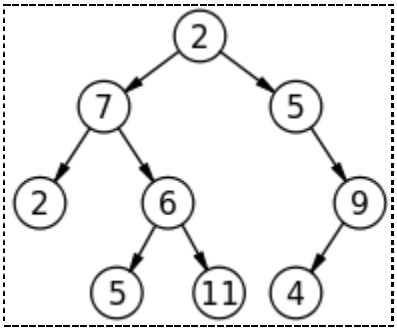

Figure 4 Access point Layout with Nodes

A labeled binary tree of size 9 and height 3, with a root node whose value is 2. The above tree is unbalanced and not sorted.A binary tree may which appears in some computer science terminology prevailed. It is also possible to interpret a binary tree as an undirected, rather than a directed graph, in which case a binary tree is an ordered, rooted tree.A binary tree is a special case of an ordered K-array tree, where k is 2.In computing, binary trees are used solely for their structure. Much more typical is to define a labeling function on the nodes, which associates some value to each node. Binary trees labelled this way are used to implement binary search trees and binary heaps, and are used for efficient searching and sorting. The designation of non-root nodes as left or right child even when there is only one child present matters in some of these applications; in particular it is significant in binary search trees.In mathematics, what is termed binary tree can vary significantly from author to author. Some use the definition commonly used in computer science, but others define it as every non-leaf having exactly two children and don't necessarily order (as left/right) the children either.Another way of imagining this construction (and understanding the terminology) is to consider instead of the empty set a different type of node—for instance square nodes if the regular ones are circles.

A binary tree is a rooted tree that is also an ordered tree in which every node has at most two children. A rooted tree naturally imparts a notion of levels (distance from the root), thus for every node a notion of children may be defined as the nodes connected to it a level below. Ordering of these children makes possible to distinguish left child from right child.[13] But this still doesn't distinguish between a node with left but not a right child from a one with right but no left child.The necessary distinction can be made by first partitioning the edges, i.e., defining the binary tree as triplet (V, E_1, E_2), where $(V, E_1 \cup E_2)$ is a rooted tree (equivalently arborescence) and $E_1 \cap E_2$ is empty, and also requiring that for all $j \in \{ 1, 2 \}$ every node has at most one E_j child.In the infinite complete binary tree, every node has two children (and so the set of levels is countably infinite). The set of all nodes is countably infinite, but the set of all infinite paths from the root is uncountable, having the cardinality of the continuum. A balanced binary tree has the minimum possible maximum height (a.k.a. depth) for the leaf nodes because, for any given number of leaf nodes, the leaf nodes are placed at the greatest height possible,

H Balanced Unbalanced, h = (n + 1)/2 - 1
0: ABCDE ABCDE
/ \ / \
1: ABCD E ABCD E
/ \ / \
2: AB CD ABC D
/ \ / \ / \
3: A B C D AB C
/ \
4: A B

Figure 5 Binary heap Routing Mechanism

One common balanced tree structure is a binary tree structure in which the left and right subtrees of every node differ in height by no more than 1 (from the figure 5). Nodes can be inserted into binary trees in between two other nodes or added after a leaf node. In binary trees, a node that is inserted is specified as to which child it is.To add a new node after leaf node A, A assigns the new node as one of its children and the new node assigns node A as its parent and in internal nodes (from the figure 6).

Insertion on internal nodes is slightly more complex than on leaf nodes. Say that the internal node is node A and that node B is the child of A. (If the insertion is to insert a right child, then B is the right child of A, and similarly with a left child insertion).

An assigns its child to the new node and the new node assigns its parent to A .Then the new node assigns its child to B and B assigns its parent as the new node. Deletion is the process whereby a node is removed from the tree. Only certain nodes in a binary tree can be removed unambiguously.

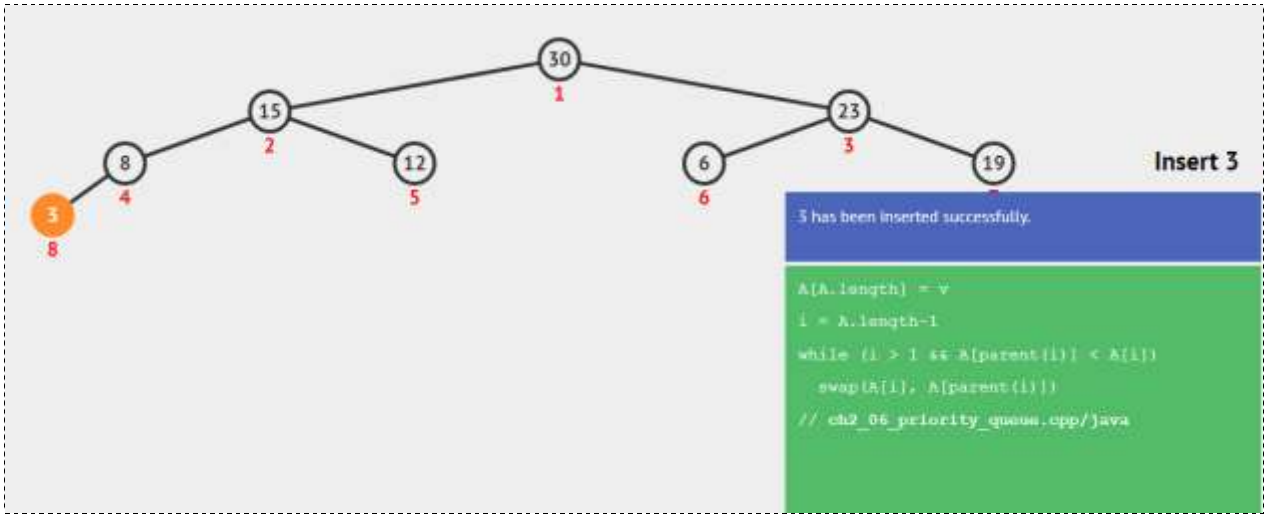

Figure 6. Access Points insertion Layout with BHRP

Suppose that the node to delete is node A. If A has no children, deletion is accomplished by setting the child of A's parent to null. If A has one child, set the parent of A's child to A's parent and set the child of A's parent to A's child. Let us consider to Crate (A) –O(N log N) :23,12,6,8,15,19,45,30,

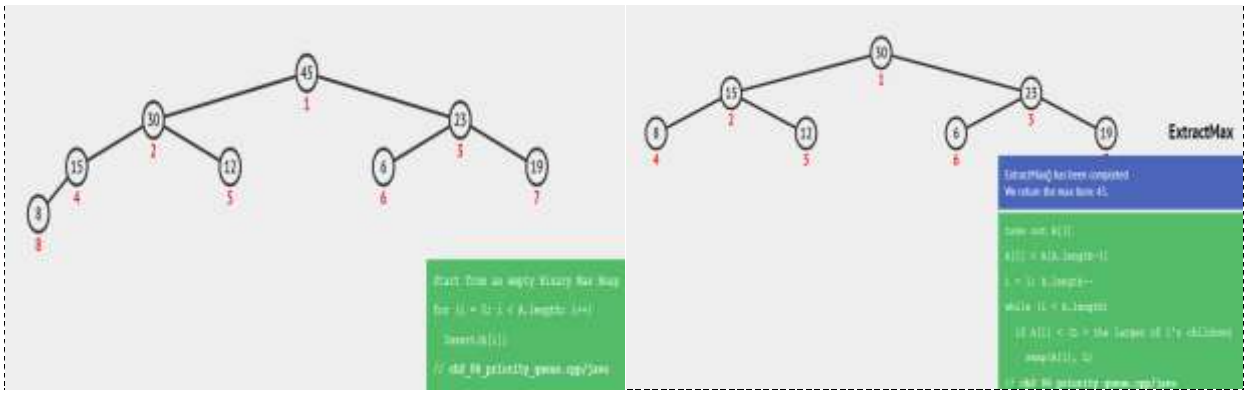

Figure 7. BHRP –Access point Creation and Extract Maximum Load Level

The major working principle of the BHRP is to divide the existing access points as a High Performance clusters (HPC) with the help of Binary search mechanism. The divided groups then compared with load from ascending order and make it the new connection assignment to the minimal node.

IV. CONCLUSION AND FUTURE WORK

This research work is focus on to make a balanced data transaction among the existing access points (AP) in the wireless communication environment at University of Gondar ICT center in Ethiopia. The University campus zone has many academic boundaries with many access points in order to provide a Quality of Service to their wireless users. But, even though most of the access points are fully loaded and some of them are ideal due to the uneven load assignment. In order to avoid such unbalanced load access, this work provides a road map to make a even load assignment with the help of a modern approach "Binary Heap Routing Protocol (BHRP)". In this section, to discuss only the logical implementation of BHRP and the continuation of this research work will provide a complete scenario about protocol architecture, method of load assignment and performance evaluation for each and every access points involving the data transfer over the communication channel at University of Gondar –ICT data Center.

REFERENCES

[1] V. Aho, J. E. Hopcroft, and J. D. Ullman. The Design and Analysis of Computer Algorithms, Addison-Wesley, New York, 1974.
[2] S. Baase and A.V. Gelder. Computer Algorithms, 3rd ed., Addison Wesley Longman, 2000.

[3] S. Carlsson. A variant of heap sort with almost optimal number of comparisons, Information Processing. Letters, 24(4), 247-250, 1987.
[4] R. A. Chowdhury, S. K. Nath, and M. Kaykobad. The heap-merge sort, Computers and Mathematics with Applications, 39, 193-197, 2000.
[5] R. Cole. An optimally efficient selection algorithm, Information Processing. Letters, 26, 295-299, 1988.
[6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Introduction to Algorithms, 2nd ed., MIT Press, 2001.
[7] Paul F. Dietz and Rajeev Raman. Small-rank selection in parallel, with applications to heap construction, Journal of Algorithms, 30, 33-51 1999.
[8] R.W. Floyd. Algorithm 245 - Treesort3, Communications of the ACM, 7 (12), p.701, 1964.
[9] D. W. Jones. An empirical comparison of priority queue and event-set implementations, Communications of the ACM, 29 (4), 300-311, 1986.
[10] D. E. Knuth. The Art of Computer Programming, Vol. III: Sorting and Searching, Addison-Wesley, Reading, MA. 1973.
[11] M. A. Weiss. Data structures and algorithm analysis in C. 2nd edition, Addison-Wesley, 1997.
[12] J. W. J. Williams. Algorithm 232 – Heap sort, Communications of the ACM, 7 (12), 347-348, 1964.

