International Journal of Electrical and Computer Engineering (IJECE)
Vol. 6, No. 4, August 2016, pp. 1560~1569
ISSN: 2088-8708, DOI: 10.11591/ijece.v6i4.10676

Building Fault Tolerance within Clouds at Network Level

DBK Kamesh, JKR Sastry, Ch. Devi Anusha, P. Padmini, G. Siva Anjaneyulu
Department of Electronics and Computers Engineering, KL University, Vaddesewaram, India

Article Info

Article history:

Received Mar 29, 2016
Revised May 18, 2016
Accepted May 30, 2016

Keyword:

Butterfly topology
Cloud computing
Fault tree
Reliability

ABSTRACT

Cloud computing technologies and infrastructure facilities are coming up in a big way making it cost effective for the users to implement their IT based solutions to run business in most economical way. Many intricate issues however, have cropped-up which must be addressed to be able to use clouds the purpose for which they are designed and implemented. Among all, fault tolerance and securing the data stored on the clouds takes most of the importance. Continuous availability of the services is dependent on many factors. Faults bound to happen within a network, software, and platform or within the infrastructure which are all used for establishing the cloud. The network that connects various servers, devices, peripherals etc., have to be fault tolerant to start-with so that intended and un-interrupted services to the user can be made available. A novel network design method that leads to achieve high availability of the network and thereby the cloud itself has been presented in this paper.

Corresponding Author:

JKR Sastry,
Departement of Electronics and Computer Engineering,
KL University,
Vaddeswaram, Guntur District, India
Email: drsastry@kluniversity.in

1. INTRODUCTION

Cloud has been a major platform hosting different kind of services for facilitating information processing. It is being used quite extensively even when compared to grid computing, a type of computing where unused processing cycles of all computers in a network are harnessed to solve problems. In cloud computing, the word “cloud” is used as a metaphor for *Internet* and in a way cloud computing can be termed as at type of internet based computing. Cloud computing provides different types of services such as such as servers, storage and applications which are delivered to computer and devices with which the users interacts.

The goal of cloud computing is to apply traditional supercomputing, or high-performance computing power, normally used by military and research facilities, to perform tens of trillions of computations per second, in consumer-oriented applications such as financial portfolios, to deliver personalized information, to provide data storage or to power large, immersive online computer games. To do this, cloud computing uses network of large group of servers typically running low-cost consumer PC technology with specialized connections to spread data-processing chores across them. This shared IT infrastructure contains large pools of systems that are linked together. Often, virtualization techniques are used to maximize the power of cloud computing.

The network plays a prominent role in cloud infrastructures. Cloud computing uses the network to gain on-demand access to computing resources, and the network becomes the conduit for enormous computing capability. This critical role of the network in cloud computing demands that network is right and the network must be configured to achieve the desired level of performance, security, availability, responsiveness, and manageability.

Faults can occur within the networks and as a result the network fails. The failure of a network connecting the cloud related infrastructure leads to disruption in the services provided to the user which may ultimately affect the businesses being conducted by the users. The 100% availability of the network is one of the most important issues that must be handled for providing highly reliable and continued services to the user.The probability of failure of a network during the delivery of the services is normally more especially when the number of users increases. It nearly impossible to prevent failures and that happen at run-time. Since it is inevitable that faults can occur, it becomes necessary to make Hardware, Network and Software that form a cloud to be made fault tolerant. Fault tolerance has become a major task for computer engineers and software developers because the occurrence of faults increases the cost of using resources. Also the problem arise when a server is overloaded, a server or a node failed etc. The Faults must be handled first so that the normal operation of cloud is not effected and then are rectified to bring the system to normal operation. During the time when the faults are handled, it is possible the system be operation with fewer throughputs, performance, response time etc., till such time the fault are fully rectified and brig the system to normal operation yielding the designed characteristic to fullest extent.

Thus the most important thing is to make the network fault tolerant in the first place. The network connectivity, topology, use of protocols, protocol conversions, routing, congestion control must be done in such a way that alternate paths exists for data transmission between the user and the cloud and vice versa. The communication can then be made to move in shortest path possible. Once the path that has failed is rectified, the communication can be made to move in the scheme of original design. In this paper, a method of making the network fault tolerant by implementing Butterfly topology has been presented and it has been shown how the network has been made to be fault tolerant.

2. RELATED WORKS

F. Thomson Leighton, and Bruce M. Baggsc [1] has described basic deterministic algorithms for routing These algorithms are vigorous against faults even in worst case and are effective from practical point of view. They found that multi-butter-fly is an excellent candidate foracheiving high bandwidth, low diameter switching network etc.

W. Shi, P. K. Srimanic [2] has expressed the use of butterfly network in VLSI point of view and according to them, interconnected networks can have only fixed number of inputs and outputs. They have explained the drawbacks of such a network especially the inability to achieve the desired level of fault tolerance based on the number of nodes contained in the network. For large networks the designing of the nodes is problematic. To overcome the drawbacks seen in butter-fly networks, they have implemented hypercube network to reduce the faults.

Jin-Fu Li *et*., al. [3], expressed a very large scale integration technology known as fast fourier transform network (FFT) into a single chip. Actually this chip is very big. To get effectiveness of the chip they have recommended using fault tolerance network. A network “C-testable FFT” has been designed which show higher reliability and the need for lower hardware. The combination of single cell fault interconnected network depends on the size of the testing pattern. It has been shown that a faulty row in multiple subtracts and adds can be repaired by 3-bit level cell.

Richard L. Graham, et. al. [4], convey that the LA-MPI (Los Alamos message passing interface) is a peer to peer network fault tolerant system designed specifically for the Tera scale clusters. They have presented a system which is highly tolerant to error related to networks, network transmission errors and wired-network errors. LA-MPI, supported multi layered network interfaces. LA-MPI’s main feature is that it can transmit messages in a reliable way through multiple network paths.

Chuanxiong Guo et. al. [5], expressed that the basic challenge in networking is how to interconnect exponentially increasing number of servers and clients very efficiently. They have presented DCell, a novel network structure that has the required features for networking. DCell is a recursively defined structure, in which a high-level DCell is constructed from many low-level Dcells. DCell scales doubly exponentially as the node degree increases. DCell is fault tolerant, since it does not have single point of failure and its distributed fault-tolerant routing protocol performs near shortest-path routing even in the presence of severe link or node failures. DCell also provides higher network capacity than the traditional tree-based structure for various types of services. Furthermore, DCell can be expanded. Results from theoretical analysis, simulations, show that Dcell is a very reliable interconnection structure for data centres.

Vincent Liu et. al. [6], expressed that, in cloud computing, the data centre networking is evolving into highly costly, reliable, and high performance computing. Even though multi-tree topologies can provide scalable bandwidth and traditional routing algorithms can provide eventual fault tolerance, the recovery speed can be very high with combination and designing of various network topologies, routing algorithm and fault detector. They have presented a novel network topology that has all the desirable characteristics of a fault tolerant topology and has very high speed recovery by implementing a failover protocol which help in connecting to the network instantly even after occurrence of a failure and also help in load balancing on the network.

Huasha Zhao, and John Canny [7] has referred incremental models that are updated many times using small subsets of training data. The model presented by them supports both stochastic gradient and MCMC (Marko chain mante corlo) messaging and helps in fast sequential performance but it cannot handle parallel or cluster settings. They have expressed that butterfly mixing approaches leads to interleaved communication and computation. They have evaluated butterfly mixing on stochastic gradient algorithms to get logistic regression. It has been shown that butterfly mixed steps are fast and failure tolerant. 3.3x speeds which are more than a full mix on an Amazon EC2 cluster have been achieved.

Ravi Jhawar et. al. [8], proposed a comprehensive approach for implementation of high level techniques for fault tolerance. In the approach presented by them the users need not know the fault tolerance requirements of their application, they would like to just know how the fault tolerance has been implemented.

Mohamed Abu Sharkh et. al. [9], has expressed that cloud computing is a utility processing paradigm that has turned into a strong base for wide exhibit of end-client applications. Providers' offten change portfolios of the employees which require different kind of services. An excellent resource allotment model is the key to any cloud computing system. Any asset allocation model needs to consider computational assets as well as system assets to precisely make the people adher to their responsibilities.

S. Giriesh et. al. [10], have presented various types of failures that can happen while enforcing the cloud computing and maintaining the system. Failuers can happen at different levels which include component failures, network failures and security failures which all must be considered and dealt with to make the entire system of cloud computing effective and fail safe. To reduce the faults, a new fault tollrence mechnasim called collaborative fault tolerance mechanism (CFTM) has been introduced. In this mechanism, the data is uploaded into the cloud under authentication by the users. There is less chance of loss at the server end by fragmenting and replicating into the virtual storages. The main key features of CFTM are authentication and data recovery.

P. Padmakumari and A. Umamakeswari [11] have described that the cloud computing could be made feasible by using fault tolerance and monitoring services. They have described two measures which are proactive and reactive that takes place with-in cloud. For cloud provider and cloud customers, fault tolerance is important as they are pre-requisites for providing continuous services. In their findings they has described that the reliability of the cloud by different or diverse fault tolerance methods could be increased.

3. ASSESSING FAILURE RATE OF AN EXISTING CLOUD

An existing cloud has been considered for effecting fault tolerance and increase the reliability of the same. The more the reliability of a network connecting a cloud, the more continued services can be provided to the end-users. The networking diagram which follows a tree like topology and connects a university cloud is shown in the Figure 1.

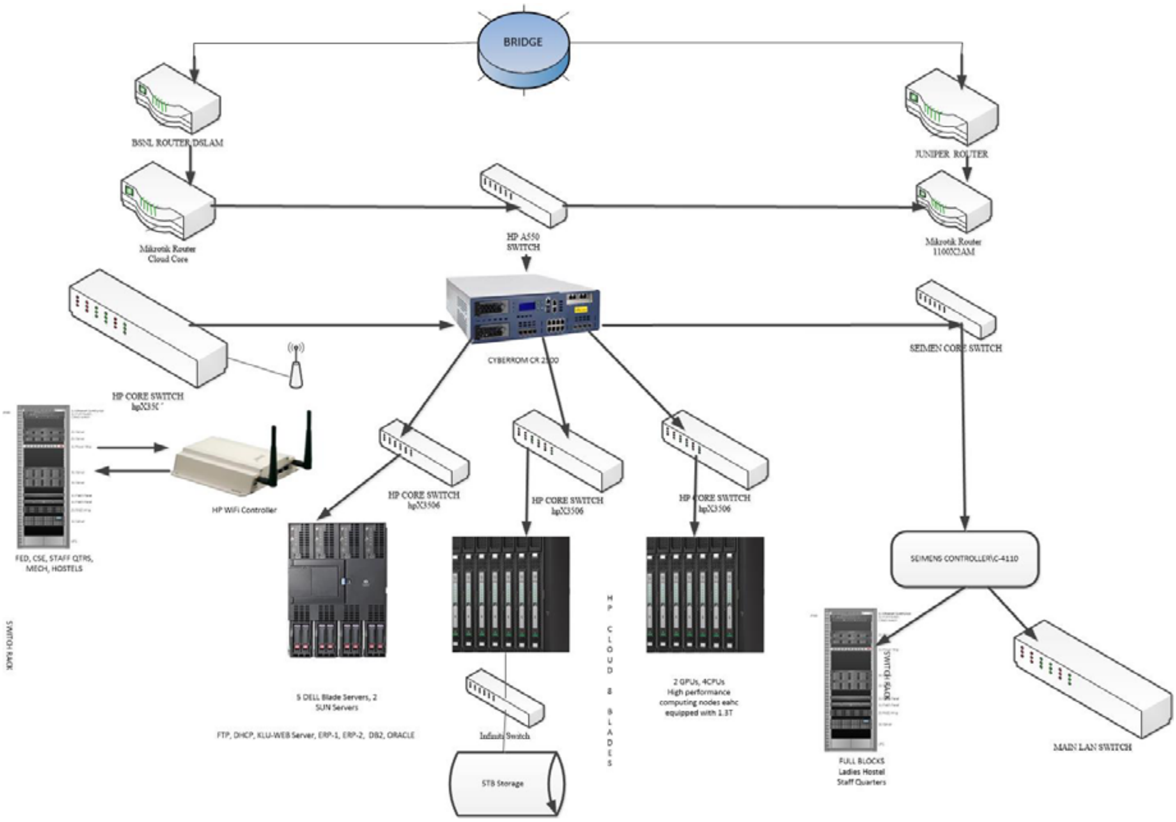

Figure 1. Topology of a university cloud

The users of the cloud are connected through internet connections that are provided by BSNL and NKN. The bandwidth from both the networks is routed through a common switch and CYBERCAM switch. The bandwidth from this point is distributed in 4 channels. In one of the channels 6 HP Blade servers and 5TB disk storage connected through an INFINITE switch has been connected to form into cloud. One of the blades has been used as the Middleware server while other servers are used as windows-Oracle server; UNIX based server, windows-SQL server and other application servers. The performance of the entire cloud computing platform is based on the proper functioning of the CYBERCAM switch. A fault tree has been constructed for the networking diagram shown in Figure 1, based on which reliability of the network has been computed. The fault tree is shown in Figure 2.

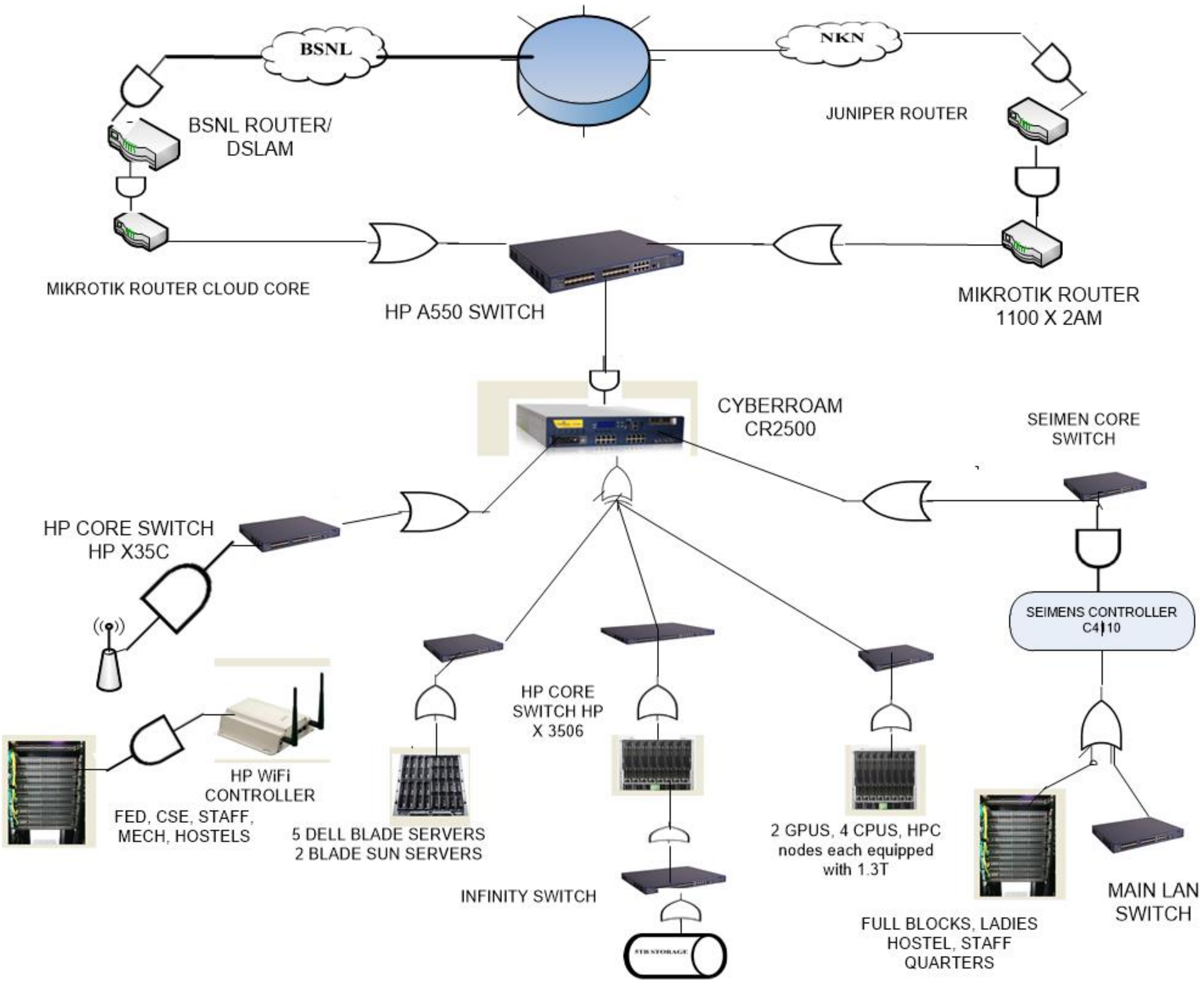
BSNL
NKN
BSNL ROUTER/ DSLAM
JUNIPER ROUTER
MIKROTIK ROUTER CLOUD CORE
HP A550 SWITCH
MIKROTIK ROUTER 1100 X 2AM
CYBERROAM CR2500
SEIMEN CORE SWITCH
HP CORE SWITCH HP X35C
SEIMENS CONTROLLER C4110
HP CORE SWITCH HP X 3506
HP WiFi CONTROLLER
FED, CSE, STAFF, MECH, HOSTELS
5 DELL BLADE SERVERS 2 BLADE SUN SERVERS
INFINITY SWITCH
2 GPUS, 4 CPUS, HPC nodes each equipped with 1.3T
FULL BLOCKS, LADIES HOSTEL, STAFF QUARTERS
MAIN LAN SWITCH

Figure 2. Fault tree diagram for the University Network

Using the Fault tree diagram reliability of the network has been computed. The reliability computation results are shown in the Table 1. The proceeding devices for every device have been connected using an appropriate gate based on the paths available for safe working of the designated device. Either OR gate or AND gate is used for estimating the FTA. In the case of connections established through OR gate, highest failure rate of incoming devices has been considered to be the failure rate of the connected device whereas when an AND gate is used, the combined failure rate has been considered as the failure rate of the connected device. The failure rate of the University cloud network is estimated to be the failure rate of the APEX node which is top gateway. The success rate of the entire network has been estimated to be 0.222 as can be seen from the Table 1.

4. IMPLEMENTING BUTTERFLY NETWORK TOPOLOGY FOR THE UNIVERSITY NETWORK

Multi-stage networks are commonly used to connect a set of inputs to a set of outputs; the concept as such is similar to cloud computing. The connectivity is sued through links between the computing / switching systems. These networks use 2 X 2 switches. Each switch takes two inputs and produces 2 outputs via different connections (Straight, cross, upper broadcast and the lower broadcast. A butter fly network is a multi-stage networks. Number of stages used depends on the kind of connectivity required. A butterfly topology which uses an 8 stage network has been considered and the same is used to fit into the University network.

Table 1. Reliability computational results for a University network

Sl. no	Device	Success Rate	Gates used For Connection	Preceding Devices					Combined Success Rate
				Device name D1 Success Rate S1	Device name D2 Success Rate S2	Device name D3 Success Rate S3	Device name D4 Success Rate S4	Device name D5 Success Rate S5	
1	Storage	0.8							0.8
2	Infinite Switch	0.8	OR	Storage 0.8					0.8
3	HP Blade	0.9	OR	Infinite Switch 0.8					0.8
4	5 Dell Blade	0.9							0.9
5	Multi Processors Architecture (2 GPUs, 4 CPUs)	0.9							0.9
6	SWITCH Rack (All Blockds, Ladies Hostel	0.8							0.8
7	Main LAN Switch	0.8							0.8
8	Siemens Control	0.8	OR	Switch Rack 0.8	Main Lan Switch 0.8				0.8
9	HP Core Switch HPX 3506 S1	0.7	OR	5 DELL Blades 0.9					0.7
10	HP Core Switch HPX 3506 S2	0.7	OR	HP Blade 0.9					0.7
11	HP Core Switch HPX 3506 S3	0.7	OR	2 GPUs, 4CPUs 0.9					0.9

A switch box in stage-I is connected with the links that are at a distance of 2^i apart. The 8 X 8 butterfly network is achieved through two 4 X 4 networks and further four number of 2 X 2 networks. The probability that one of the paths exists for connecting a cloud computing processor can be computed as

$$A_{c\,=}\,2^{k\,\rho l}\;\Phi(k) \qquad (1)$$

where k = Number of stages, ρ = probability that a node fails and Φ (k) is the probability that that a switch box in the stage K can fail. Φ (k) can be computed using the equation (2).

$$\Phi\,(k) = 1 - (\,1\text{-}\,\rho_l\,\Phi(k\text{-}1)^2 \qquad (2)$$

The butterfly network connected for fitting university cloud has been shown in Figure 3. The butter fly network has been established using 8 X 8 network containing 8 stages. The 8 X 8 network has become necessary due to the availability of 8 elementary levels of inputs and 8 different types of outputs required to make the network reliable and available. Additional switches have been added to make it possible to connect the University cloud into a butefy network.

Using the equation (1) and equation (2) the probability of success that at least one path exists from input point to an output has been computed as 0.28.

5. MODIFIED UNIVERSITY NETWORK TO FIT BUTTERFLY TOPOLOGY

Considering the butterfly network shown in the Figure 3 developed for University network, the extra switches required have been identified and the same are considered for including in the tree like structured of the university cloud. The revised KLU network that has been fitted with butterfly topology is shown in Figure 4. It is seen from the figure that 3 extra switches have been added and the internet bandwidth is literally broken into two halfs; each half working as a backup to the other.

Fault tree has been constructed for the modified University network diagram and the same is shown in Figure 5. The connectivity is achieved through OR and AND gates as described to produce an FTA for the original University network. The probability of the success of the revised network is once again computed and the computational results are shown in Table 2. From the Table, it can be seen that the success rate of the revised University cloud has been increased from 0.222 to 0.2764.

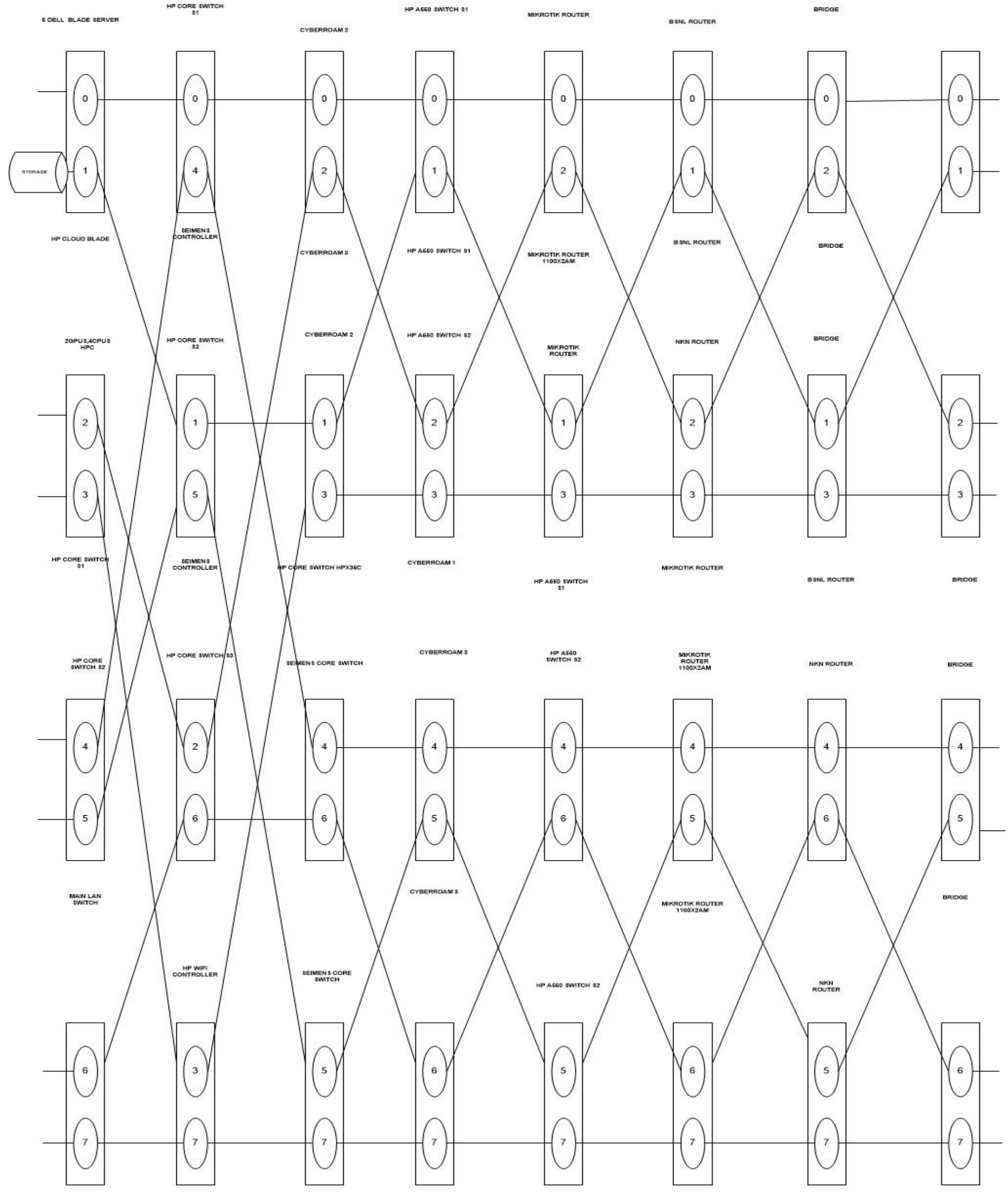

Figure 3. Butterfly network for the University Cloud

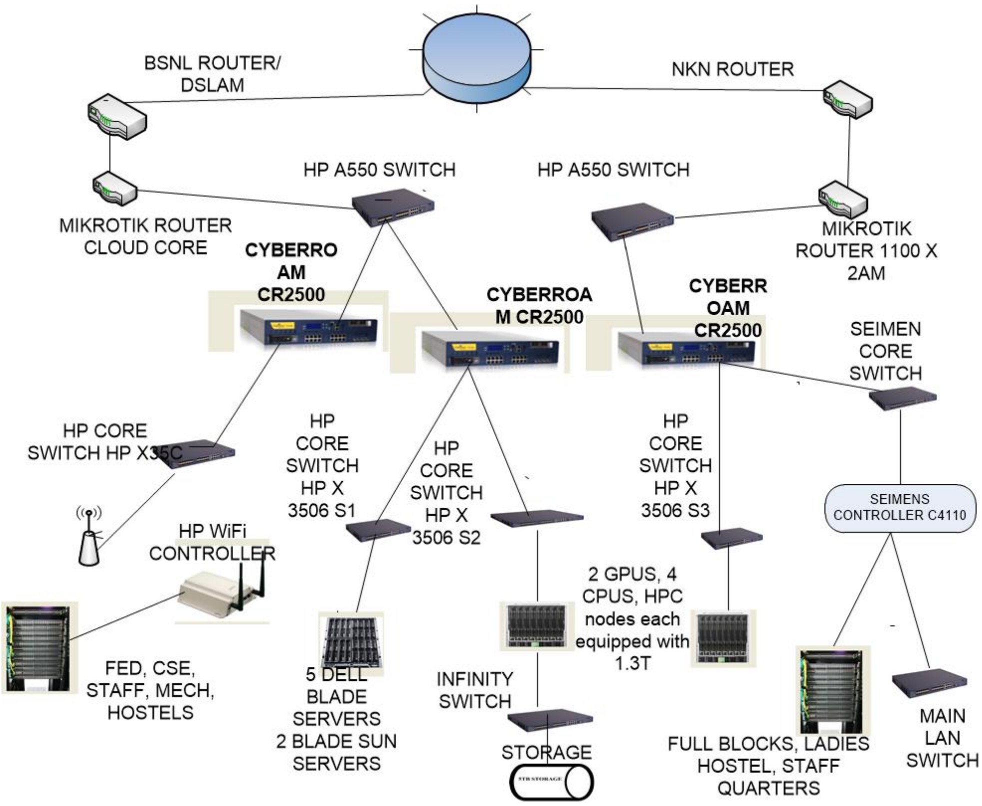
BSNL ROUTER/ DSLAM
NKN ROUTER
HP A550 SWITCH
HP A550 SWITCH
MIKROTIK ROUTER CLOUD CORE
MIKROTIK ROUTER 1100 X 2AM
CYBERRO AM CR2500
CYBERROA M CR2500
CYBERR OAM CR2500
SEIMEN CORE SWITCH
HP CORE SWITCH HP X35C
HP CORE SWITCH HP X 3506 S1
HP CORE SWITCH HP X 3506 S2
HP CORE SWITCH HP X 3506 S3
SEIMENS CONTROLLER C4110
HP WiFi CONTROLLER
2 GPUS, 4 CPUS, HPC nodes each equipped with 1.3T
FED, CSE, STAFF, MECH, HOSTELS
5 DELL BLADE SERVERS 2 BLADE SUN SERVERS
INFINITY SWITCH
STORAGE
FULL BLOCKS, LADIES HOSTEL, STAFF QUARTERS
MAIN LAN SWITCH

Figure 4. Modified University Cloud that fits into it the Butterfly topology

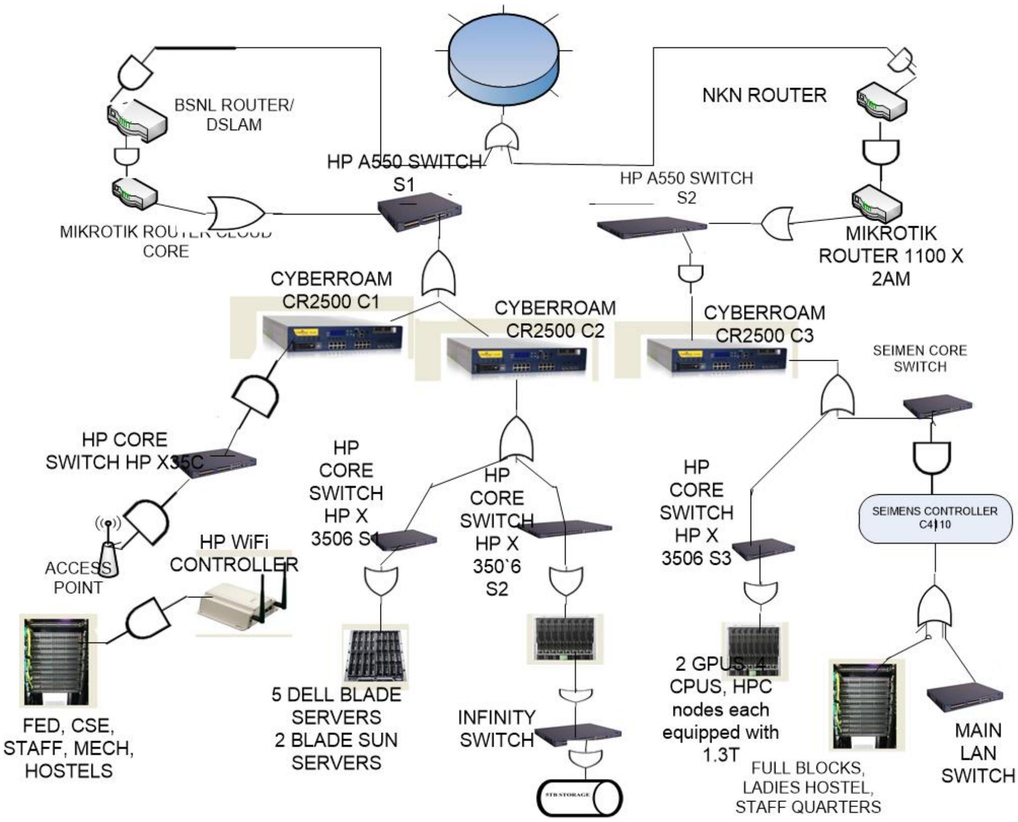
BSNL ROUTER/ DSLAM
NKN ROUTER
HP A550 SWITCH S1
HP A550 SWITCH S2
MIKROTIK ROUTER CLOUD CORE
MIKROTIK ROUTER 1100 X 2AM
CYBERROAM CR2500 C1
CYBERROAM CR2500 C2
CYBERROAM CR2500 C3
SEIMEN CORE SWITCH
HP CORE SWITCH HP X35C
HP CORE SWITCH HP X 3506 S1
HP CORE SWITCH HP X 350`6 S2
HP CORE SWITCH HP X 3506 S3
SEIMENS CONTROLLER C4110
ACCESS POINT
HP WiFi CONTROLLER
FED, CSE, STAFF, MECH, HOSTELS
5 DELL BLADE SERVERS 2 BLADE SUN SERVERS
INFINITY SWITCH
2 GPUS, 4 CPUS, HPC nodes each equipped with 1.3T
FULL BLOCKS, LADIES HOSTEL, STAFF QUARTERS
MAIN LAN SWITCH

Figure 5. Modified University Cloud in terms of Butterfly topology

Table 2. Fault Rate calculations for the Modified University Cloud network that fits a Butterfly topology

Sl.no	Device	Success Rate	Gates used For Connection	Preceding Devices		Combined Success Rate
				Device name D1 Success Rate S1	Device name D2 Success Rate S2	
1	SWITCH RACK(FED,CSE)	0.9				0.9
2	5 DELL BLADE	0.9				0.9
3	STORAGE	0.8				0.8
4	INFINITY SWITCH	0.8	OR	STORAGE 0.8		0.8
5	HP CLOUD	0.9	OR	INFINITY SWITCH 0.8		0.8
6	2GPUS,4CPUS HPC	0.9				0.9
7	HP CORE SWITCH S1	0.7	OR	5 DELL BLADE 0.9		0.7
8	HP CORE SWITCH S2	0.7	OR	HP CLOUD 0.8		0.7
9	HO CORE SWITCH S3	0.7	OR	2GPUS,4CPUS HPC 0.9		0.7
10	SWITCH RACK (FULLBLOCK)	0.8				0.8
11	MAIN LAN SWITCH	0.8				0.8
12	SEIMENS CONTROLLER	0.8	OR	MAIN LAN SWITCH 0.8	SWITCH RACK(FULL) 0.8	0.8
13	SEIMENS CORE SWITCH	0.8	AND	SEIMENTS CONTROLLER 0.8		0.64
14	HP WIFI CONTOLLER	0.75	AND	SWITCH RACK(FED) 0.9		0.675

6. COMPARATIVE ANALYSIS OF RELIABILITY EVALUATION OF THE CLOUD COMPUTING NETWORKS

The computation of success rates of different topologies used to develop the University cloud related network is shown in the Table 3. It can be seen from the table that Butterfly topologies when incorporated into University cloud related network has increased the success rate making available more continuity of the services as required by the user.

Table 3. Comparison of success rates of Cloud related Network when designed with different topologies

Topology Serial	Topology	Success Rate
[1]	Tree topology – Original University cloud related Network	0.227
[2]	Butterfly topology built-in with extra switches	0.280
[3]	Tree Topology enhanced with identified redundancies included into butterfly network	0.277

7. CONCLUSIONS

Networks that connects various resources that form a cloud plays a major role in providing continued services to the users of the clouds which leads to heavy user satisfaction. However if the network is either faulty or when different faults occurs while system is running, the services to users will be disrupted till the time the network is made operational. Making available continuous services to the user is most important pre-requisite of implementing cloud based services to the user. The network used to host the cloud computing based services must be reliable and the network must be designed to be fault tolerant so that the services to the users will be provided continuously even in the event of occurrence of the faults at network level. Alternative paths of communication between the user and the clouds are to be established to make available un-interrupted service to the users. Multi stage networks helps in improving the reliability many fold. Butterfly networking topology supports multistage networks through 2 X 2 switches which provide 4 alternative paths of switching and many intermittent switches multiplies many other paths. Butterfly networking topologies increases the fault tolerance capability with lease cost which could be the cost of few switches.

REFERENCES

[1] F. T. Leighton and B. M. Baggs, “Fast Algorithms for Routing Around Faults in Multibuffer flies and Randomly - Wired Splitter Networks,” *IEEE TRANSACTIONS ON COMPUTER*, vol. 41, pp. 578-587, 1992.

[2] W. Shi and P. K. Srimani, “Hyper-butterfly network: a scalable optimally fault tolerant architecture,” *IEEE TRANSACTION ON Parallel Processing Symposium*, 1998. IPPS/SPDP 1998. Proceedings of the First Merged International and Symposium on Parallel and Distributed Processing 1998, pp. 732-736, ISSN: 1063-7133, 1998.

[3] J. F. Li, *et al.*, “Easily Testable and Fault-Tolerant FFT Butterfly Networks,” *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: ANALOG AND DIGITAL SIGNAL PROCESSING*, vol. 47, pp. 919-929, 2000.

[4] R. L. Graham, *et al.*, “A Network-Failure-Tolerant Message-Passing System for Terascale Clusters,” *International Journal of Parallel Programming*, vol/issue: 31(4), pp. 285-303, 2003.

[5] C. Guo, *et al.*, “DCell: A Scalable and Fault-Tolerant Network Structure for Data Centres,” *SIGCOMM '08 Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, pp. 75-86, 2008. ISBN: 978-1-60558-175-0.

[6] V. Liu, *et al.*, “F10: A Fault-Tolerant Engineered Network,” *10th USENIX Symposium on Networked Systems Design and Implementation*, pp. 339-412, 2013.

[7] H. Zhao and J. Canny, “Butterfly Mixing: Accelerating Incremental-Update Algorithms on Clusters,” *SIAM International Conference on Data Mining*, 2013. ISBN: 978-1-61197-262-7.

[8] R. Jhawar, *et al.*, “Fault Tolerance Management in Cloud Computing: A System-Level Perspective,” *IEEE SYSTEMS JOURNAL*, vol/issue: 7(2), pp. 288-297, 2013.

[9] M. A. Sharkh, *et al.*, “Resource Allocation in a Network-Based Cloud Computing Environment: Design Challenges,” *IEEE Communications Magazine*, pp. 46-52, 2013. 0163-6804/13.

[10] S. Giriesh, *et al.*, “Dynamic Data Fault Tolerance Mechanism to Enhance Reliability and Availability in Cloud,” *Indian Journal of Science and Technology,* vol/issue: 8(S9), pp. 300–305, 2015.

[11] P. Padmakumari and A. Umamakeswari, “Methodical Review on Various Fault Tolerant and Monitoring Mechanisms to improve Reliability on Cloud Environment,” *Indian Journal of Science and Technology*, vol/issue: 8(35), 2015. DOI: 10.17485/IJSTt/2015/v8i35/80130.