SDN-Based Inter-ISP Caching

Author

Muhammad Talha Jahangir

205345

Supervisor

Dr. Muhammad Umar Farooq

Co-Supervisor:

Dr. Ali Hassan

DEPARTMENT OF COMPUTER & SOFTWARE ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

August, 2019

SDN-Based Inter-ISP Caching

Author

Muhammad Talha Jahangir

205345

A thesis submitted in partial fulfillment of the requirements for the degree of

MS Computer Engineering

Thesis Supervisor:

Dr. Muhammad Umar Farooq

Thesis Supervisor's Signature:____

DEPARTMENT OF COMPUTER & SOFTWARE ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY, ISLAMABAD

August, 2019

DECLARATION

I certify that this research work titled "SDN-Based Inter-ISP Caching" is own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

Signature of Student

Muhammad Talha Jahangir

2017-NUST-MS-CE-00000205345

LANGUAGE CORRECTNESS CERTIFICATE

This thesis has been read by an English expert and is free of typing, syntax, semantic, grammatical and spelling mistakes. The work is original contribution of the author and does not contain any plagiarism. Moreover, Thesis is also according to the format given by the university.

Signature of Student

Muhammad Talha Jahangir

2017-NUST-MS-CE-00000205345

Signature of Supervisor

DR. MUHAMMAD UMAR FAROOQ

ACKNOWLEDGEMENTS

I would be grateful to my Creator Allah SUBHAN-A-WATALA for guiding me in every phase of this job and every fresh thought setting in my mind to enhance it. Indeed, without your precious assistance and guidance, I could have achieved nothing. Whoever has helped me during my thesis, whether my parents or any other person has been your will, so no one is praiseworthy other than you.

I am deeply grateful to my beloved parents who raised me when I was unable to walk and kept supporting me in every department of my life.

I would also like to convey unique thanks to my supervisor Dr. Muhammad Umar Farooq for the assistance he has given me all across my thesis and for the lessons he has taught me on selected topics in computer networks. I can securely claim that I have not learned so profoundly any other topic of engineering than the ones he instructed. He came up with the solution every time I stuck in something. I couldn't have completed my thesis without his assistance. Throughout the thesis I enjoy his patience and guidance.

I would also like to thank my HOD and my Co-supervisor Dr. Ali Hassan in particular for his tremendous assistance and collaboration.

I also want to thank Dr. Farhan Hussain and Dr. Wasi Haider Butt for being on my thesis guidance and evaluation committee and express my special Thanks to these faculty members for his help.

Finally, I would like to convey my appreciation to all the people who have rendered valuable assistance to my study.

Dedicated to my exceptional parents and adored siblings whose tremendous support and cooperation led me to this wonderful accomplishment

ABSTRACT

Big data introduces many challenges from the service provider's perspective such as increase in latency, delay variance, congestion and network utilization etc. These days users require fast speed contents as well as high connectivity. Users are service aware and dislike network delays. In networking, bunch of effort has been made in throughput enhancement but significant efforts are needed to minimize latency delay and delay variance etc. New network framework turns into the attractive obligation of this technological age. Latest network Infrastructures such as content delivery network (CDN) and software defined network (SDN) assist to determine today's user's concerns. CDN provide Contented Caching and it is valuable practice that Internet Service providers (ISP) utilize at this period. Different cache server cooperates with each other and optimally offers the demanded content to the consumer. Challenges in cooperative Caching are where to place the cache server and optimally route the content to client using cache server. In proposed approach, SDN-based Inter-ISP Caching Technique is aspiring to resolve the difficulty of cache server placement which diminishes the data transmission latency and direction the cache content with optimized mode in the inter-ISP traffic. In proposed approach, assumed different ISPs which are present in same geographical location, communicate with each other. Specified approach, introduce light weight algorithm named as Row Based Floyd WARSHALL algorithm assist to put the cache server in optimized method. Cache Controller is as well pioneering to established the cooperative caching and competently way the cache content through the lessening of cache miss and recover network throughput with the incidence of cache hit. Ns3 which is advance event-based network simulator is being used for this toil. Simulation results demonstrate the efficiency of SDN-based Inter-ISP Caching beneath diverse network circumstances.

Keywords: Big data, Cooperative Caching, Floyd WARSHALL algorithm, Row based Floyd WARSHALL Algorithm, Software Defined Networking, ISP Networks

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ICN	Information centric Networking
SDN	Software Defined Networking
CDN	Content delivery network
ISP	Internet Service Provider
DNS	Domain Name Service
CSP	Content Service Provider
CS	Cache Server
CC	Cache Controller
HTTP	Hypertext Transfer Protocol
NAT	Network Address Translation
NC	Network Controller
QoE	Quality of Experience
RTP	Real-time Transport Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URL	Uniform Resource Locator
WAN	Wide Area Network
CBR	Constant Bit Rate
P2P	Peer to Peer

CHAPTER 1: INTRODUCTION

In this age almost everything is linked with internet. It is judging that, the rate of global IP traffic was 1.5 ZB per year in 2017 [1]. In 2021, It is assumed that global IP traffic will be cross 3.3 ZB per year, or 278 EB (one billion Gigabytes [GB]) per month [2]. Degree of video streaming and substance distribution is considerably rising in modern era. For example, Netflix traffic crafts 30% of the US traffic split in climax period [3]. Big data has been extremely dominant on today's system. The traffic is outstanding to swap huge sum of stuffing e.g. photos and videos. People hates latency and demand contents in very fast speed.

1.1 Background, Scope and Motivation

Volume of Internet traffic is increasing day by day due to increase in video streaming, applications encompassing big data, IoTs and cloud services etc., and rapid increase in network penetration. Users demand high quality data/video transmissions which require proportional bandwidth. In addition, Quality of Experience (QoE) and Quality of Service (QoS) necessitate guaranteed low latency transmissions over the Internet. Everything approaches towards big data. Big data based on bulk amount of data which contains high data rates. It is clear that data rate of file is directly proportional to the size of file. In internet, if the size of file is increase then data rate per second of file is also increase. Big data has been very influential on today's network. Similarly, multimedia transmissions over the internet have drastically increased. In history, we just necessitate the connectivity, computing reserve is fewer resourceful and bandwidth expenditure of connection is not fairly fewer. Inside that occasion, the structure of the internet was too prearranged in the similar method. The people's ambition was little and he used to be pleased on just connectivity. But at the present the diversion of internet is totally distorted and the cause following this is the survival of big data. In this age, people insist is rising with requirements, they should not just require connectivity; they want great amount of data so that they can do uploading and downloading. Internet should proportionally modify its structural design with today's order. The distributed architecture or nature of old network causes management and control issues [4]. People hates delays and requires fast speed network connection. To provide the high network speed, need to convert the traditional network infrastructure into efficient new network infrastructure which fulfills all the demands of current user. New rising expertise such as Content delivery network, Software defined system altering the image of internet.

1.2 What is Software Defined Networking

Software defined network is a framework which splits down the control plane and data plane. Software Defined Networking (SDN) is new-fangled structure and is fetching established day by day

In customary network intend data plane and control plane is in the similar network device such as in router. In preceding network structural design formation of routing and forwarding table is extremely expensive because it entails competent algorithm to decide finest trail for traffic. Whenever any altering happens in network then the data in tables also transforms and for synchronization each network machine updated table among each other. The procedure of distribution tables devours plenty of bandwidth which is not price effectual and as well sluggish along the business. New structures necessary which determine this difficulty. If we Compare the traditional network with SDN then SDN introduce new complexity in networking by facilitate management of network and provide the framework which support evolution [5–7]. SDN grants us a central controller which holds control plane whilst open flow switches hold data plane. The Central Controller which give orchestration and manage all the network devices in extremely competent way. Controller knows all the layout of topology, if there is any changing occur in network topology so all network devices across the network will not share information among themselves but they communicate with Controller and vice versa. Central controller efficiently updates desirable network device which is very cost effective and easily manageable. SDN orchestration is very beneficial for routing the traffic. SDN Controller can instruct the requested content to enter into specific cache server rather than the content traverse among several routers with remarkable delay and then find the destination.

1.3 SDN Architecture

SDN segregate the traditional control plane and data plane which allow or open many possibilities and opportunities in network. SDN Architecture based on two interface southbound interface and northbound interface. Control plane comprises of controller(s), responsible for control logic of the network. Data plane comprises of switches that make the forwarding decisions with help from the forwarding rules embedded by the controller. Separate control plane allows us to use any application as network or system administrator wants. While data plane comprises of cheap OpenFlow or similar switches, whose task is to forward the packets towards desired location. In these switches, flow table entry has three parts: 1) Rule of Matching, 2) actions to be performed 3) counters [53].

1.4 What is Content delivery Network?

Content delivery network contains cluster of servers which are connected in distributed fashion to provide the content in speedy way. Content Distribution Network (CDN) is the expertise that has modifies the financial system of Internet Service provider's (ISP's) and Internet Exchange Point's (IXP's).

In elderly days, one ISP converse with another ISP through internet. For instance, presume two ISP's within Pakistan unified via internet. Internet provider which attaches mutually ISP's is situated in London. If one ISP needs content through another ISP, then one is spread the routing from London and obtain the content to another ISP. In sight of this matter, the IXC was initiated. IXC is situated within the countries of confined ISP's. Content Distribution companies such as Google, YouTube, and Netflix etc. remain their content in IXC so that the latency can be reduced, in this age, the data that has been shapeless and has turn out to be as well big. We should not bind the CDN to the IXC merely, but we should competently organize cache servers in ISP's. The internet grips vast quantity of information and administrates serious traffic during the day; as a consequence, bandwidth jamming can be a concern on main networks.

1.5 Problem Statement

Currently we are turned into data cognizant and today's data varying its tendency. Data is mounting day by day as productively rehabilitated into big data. Big data is bottomed on five V's as Velocity, Volume, Value, Variety, and Veracity, when the volume of data augments then proportionally information velocity of data as well augments. Big data communiqué into wireless system has too turns into huge confront. Software Defined Network (SDN) [16], [17], [18] as a new networking framework which contains centralized control over the large number of network nodes. In projected approach we presume In-bound communiqué quite than out-bound statement. In bound communication, The Cache server admittance during communal system as shown in Figure. 7, whilst out-bound communiqué is similar to a mesh topology that every cache server is linked with all routers in the ISP.

Now here numerous challenges when we used cache server in multiple ISP by using SDN network. The challenges are: **1.** Placement of SDN Controller and Selection of border OPENFLOW switches in every ISP. **2.** When Controller sprints numeral of system submissions such as routing app, firewall i.e. then it will be congested so how we shun congestion of Controller **3.** Placement of cache server is an extremely gigantic difficulty. There are many parameters in network such as hop count, shortest distance, bandwidth consumption, traffic overhead, link delay, etc. to measure the placement of cache server. **4.** How to direct the cache memory in ISP's cache server and CACHE_CONTROLLER **5.** What will be the Cache placement Strategy? 6**.** How to optimally itinerary the demand of content in SDN based Network?

1.6 Thesis Contributions

We have thoroughly studied the SDN based architecture, CDN working and research contributions of the authors in this area. We have investigated the research challenges and key issues that are the clients face in different network conditions. Finally, we have developed a light weight algorithm named as row-based FLOYD WARSHALL algorithm, which provides the solution to place the cache server.

The proposed approach faces all the challenges and introduce light weight algorithm which successfully find the best optimal location of cache server. The main purpose of this thesis to find the best location of cache server in network. After placing cache server in best location, specified approach try to reduce the load of cache server by using cache controller. Cache controller main task is to provide the cooperative caching in network. The given approach also provides, mechanism of cache management and routing the content from user to server. We use network simulator (NS3) to design three large geographically co-located ISPs. The ISPs are interconnected by using Cache_Controller for coordination between cache servers. The cache servers use LRU policy for replacing cache contents. The aim of this work is to identify the effectiveness of cache server coordination and importance of placement location of cache servers by testing different network performance parameters by changing cache hit ratios and cache server location. Results reveals that the proposed approach which used Row based FLOYD

WARSHALL algorithm provides much better result. The results indicate that the cache server location is critical for networks performance Result also favor using of cache server increase user experience by reducing the latency, load on origin server and also by increasing the response time in our SDN based ISP network.

CHAPTER 2: LITERATURE REVIEW

Data is increasing day by day and successfully approaches towards big data. In the existence of old traditional network, it was becoming very problematic for internet service providers to manage users demands. We have routers, switches, firewall ETC as network devices. Routers is very important network device in ISPs, data center and IT to forward the traffic or packets into desired location. In non-SDN based network, router contains two planes such as control plane and data plane. Router comes with its vendor operating system which contains proprietary services. Users have limited privilege to alter and install the new services on it. Moreover, in previous network, quality of service management become very difficult. For example, ISP contains hundred routers and ISP wants to manage the traffic engineering on it. If some routers not able to install traffic engineering services such as quality of service (QoS), quality of experience then it became much difficult to control or manage the traffic. ISPs require or wants lots of changing in existing network. After seeing all issues in earlier network, SDN network come into existence.

Software defined network is a framework which broke down the two planes into two separate plane such as control plane and data plane. Control plane is like a controller which control everything in network while data plane is just used to move forward the network traffic. Separate control plane allows us to use any application as network or system administrator wants. While data plane is installed on cheap OPENFLOW switches which task is to forward the packet into desired location. SDN provide the service of orchestration which helps the administrator to manage the traffic engineering. Through orchestration, controller can easily add flows into switches and forward the traffic into desired route. SDN resolve the issue of traditional network architecture but, if user consume time or face latency in his response then administrators or engineers required such mechanism which increase the speed of response and decrease the network delays. Content caching is the best way to object such issue.

Large geographical distance between the origin server and end-user produces the hold up and packet loss. Latency and packet losses are necessary when videos are being viewed, or when real-time decisions are being made. It causes frames dropping or alternatively increased buffering delays in video streaming. Ultimately, these eventually effect the quality of service and lower the quality of experience for end users [34-39]. To avoid packet loss and obtain the benefits from reliable transmission, we should need cooperative content caching mechanism in our network. Content caching is a dedicated network server which store the web, intranet and internet content on it. Whenever user demands the content then ISPs should save the round-trip time and provide the content from its dedicated cache server. In this era, users are content conscious they need content very fast and single cache server face bulk number of requests, so single cache server is not sufficient because of congestion. Cooperative cache is required to fulfills the user needs. Cache server are distributed into several region and each cache server cooperate with others cache servers and divide the load of requests in very efficient way.

There are lots of vacuums are available in cooperative caching such as cache server placement is the big issue. If we placed the cache server in network using special algorithm then it produces much better result than randomly place the cache server in any location. Moreover, if multiple cache server is present in network then how to efficiently route the traffic among cache servers and manage the cache content of cache servers.

In this age, resource storage and caching technique is utilized in every area of network and communication technologies. In network, caching concept is very old. Its first popular use initiated from web-based caching, which even stretched to hierarchical caching [40-41]. Content caching and cooperation is already being offered as a service by many applications/businesses in the internet, e.g. CDN, P2P and ICN.

2.1 Related approach

Content distribution networks (CDNs) deliver major share of the internet's traffic. CDN is a framework which deploys a bulk amount of caching servers worldwide, in order to provide the content (usually) in edge routers of ISPs [42-43]. CDN is content oriented service, whose main task is to provide content to users and users are unaware about where the content is obtained from. If we deploy the cache server in each geographic location, it will not be able to achieve good performance. For instance, when user requests for same video content and CDN cache server is heavily loaded or congestion has occurred, some alternate path/server is needed [44].

Similarly, P2P networks [45] also provide the mechanisms for content caching, replication and distribution. The efficiency of P2P is based upon the users who share their network storage and resources of network [46]. However P2P approach lacks central control and also triggers security concerns.

In earlier web caching was introduced. Web caching aim to store website in its cache server. There is a bottleneck present in bandwidth and latency. Now-a-days web caching is not used due to high bandwidth consumption and the existence of https. In this era, network engineers and scientists working to minimize the latency. Now a days we are more concern about content caching rather than web caching. Due to the existence of big data, caching strategy reused again. For web caching, many proposals of cache replacement policies have been published [51]. In this article, least recently used (LRU) policy [52] is used as a cache replacement policy. This is because LRU is quite simple to implement.

Pervasive caching is providing a mechanism of storage on each router. GHODSI et al [48] and S.K. FAYAZBAKYSH et al [47] reveal that there are a lot of improvements required in pervasive caching. Similarly, information centric networking is based on content instead of host. ICN change the internet design of 1970s which is based on end to end model [50]. ICN assigns contents names that are not relate to their IP addresses [49]. In ICN approach, user is less worried about the actual host location but rather the request of their content. ICN is a good approach but there are lots of vacuum present in ICN architecture. Athanasios at el [50] explained the research challenges present in ICN based networks.

It observed in ICN, single cache server creates lots of problem, connect minimum one router of each shortest path between network end device and cache server. In such scenario, a bulk amount of request arrive at the single cache server may fill the memory of cache server. Single cache server is also unpractical in traditional network. Yong Cui et al [12] planned Cooperative cache aspires toward to optimize the network traffic by caching content of big data in systems. They projected SDN-based Cooperative Cache Network (SCCN) for ISP systems, which utilize SCCN Controller to manage, circulated SCCN Switches (i.e., cache nodes). In SCCN advance to increase the cache hit ratio, they used relaxation and heuristic algorithm. In this work, the OPENFLOW switch holds a particular storage space for content. The genuine reason of SDN is to divide the control plane and forward plane. In specified advance, OPENFLOW switch is congested as currently the chore of OPENFLOW switch is not immediately to ahead pour but as well direct and storage the content in the OPENFLOW switch.

Kim et al. [13] recommended a central cache server in its place of upholding cache at every router in the system for ICN. The advance essentially plants on two codes:

1. At slightest one router amid two rim routers of a domain should be linked to the cache server.

2. The cache-hit share at the cache server should be better than or equivalent to that in pervasive caching.

Limitation of this work is the OPENFLOW switch is just associated with cache server if the network mass acquires big, the solitary cache server will not execute fine. In addition, in pervasive cache, if a router is processing one request of user, other packs will stay and if those other packs are of actual time traffic, they will catch belated and might turn into ineffective. AUBRY et al. [14] projected a method that router send the request to controller when they not have the route. The controller calculates the course and mounts the way on the routers in overturn arrange. In addition, big network is alienated into a number of domains and every domain has its own controller as exposed in Figure. (1) [14]. The main drawback of AUBRY approach, if cache miss occurs then the network performance fall down instead to grow up.

This advance as well does not tackle the subject of cache server situation.

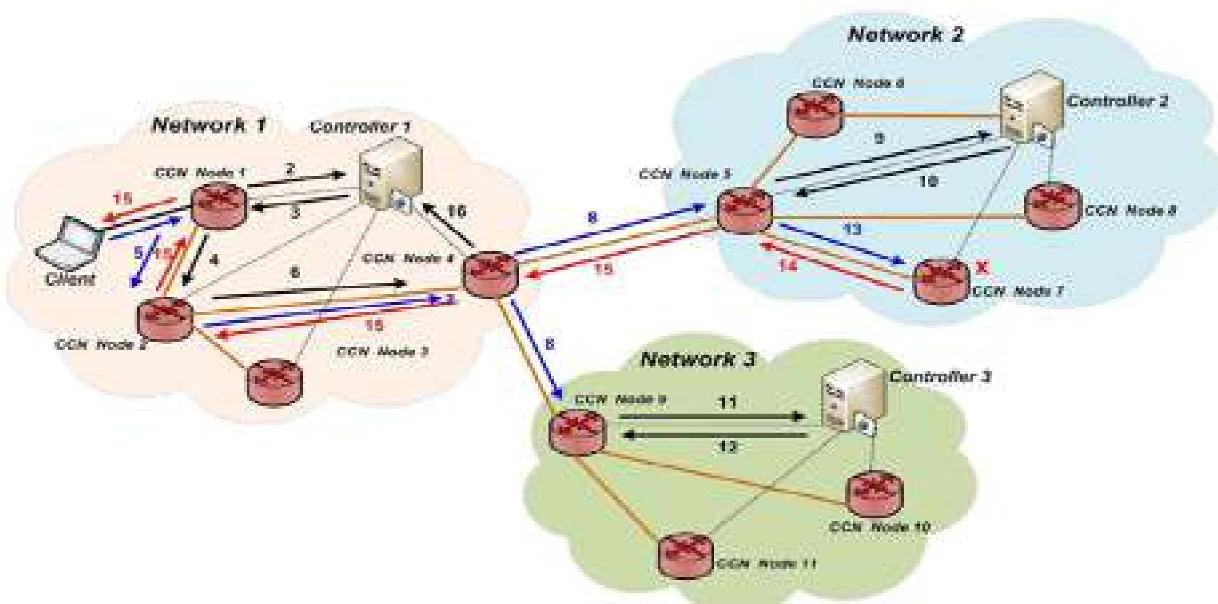


Figure 1: SRCS MODEL [14]

Jan Badshah at el [6] calculates the sites (the switches/routers through with cache servers are connected) and the integer of cache servers bottomed on the sdn based topology information in an offline mode. One curb of this advance, it is used in informative-centric networking which major hub to modify the internet obtainable structural design. Information Centric Networking - suggests a prospect Internet structural design that rotates about the contents being swapped quite than the communiqué of hosts as well as network device. Whilst, in host centric architecture, we transport a packet to a known destination address, communiqué is as of one host to a further host. Execution of ICN structural design is extremely luxurious and multifaceted. We deliberated OSI layer in lots of decades and all the network procedures, network function, is intended and expand for that reason. If we alter the internet structural design, we need altering the whole thing **i.e.** network functions, network procedure, and header file of every layer and planned novel set. ICN is fine advance but it needs batch of occasion to productively realize into present network system. Cache server placement also be studied as a commodity locality problem for better performance [54-55]

CHAPTER 3: PROPOSED METHODOLOGY

The proposed Approach calculates the position of cache server in entity ISP's and relate routing configuration to increase the hit ratio. Major reason is to put the cache server in best location and then contrast with every placement in particular topology. The proposed approach is supported on SDN based cache. SDN divided the control plane and forward plane and it offer an inner power that is extremely useful for orchestration. SDN is central controllers which holds all information about ISP and gifted to calculate optimize choices.

Content of different type is stored in each ISP cache server. The SCCN [12] approach contains the cache management in VSWITCH. It makes OPENFLOW switch overloaded and cause delay. To overcome this issue suggested approach used simple OPENFLOW switch and introduce specified cache server in each ISP. To overcome the limitation of single cache server in Kim et al [13], suggested approach gives three type of SDN based servers such as: 1. Local ISP Controller 2. Cache server 3. Cache Controller. Local ISPs server task is finding the placement of cache server in its local ISP and add the flows into OPENFLOW switches which helps the traffic to direct towards its cache server. Cache server used to store content in its cache space and also add rules in OPENFLOW switches to connect with cache controller. Cache controller helps ISPs to coordinate with each other's. In proposed approach, placement of cache server based on row-based Floyd WARSHALL algorithm. Floyd WARSHALL algorithm resolve the position of placement problem Floyd WARSHALL algorithm present us a matrix-based computation which can be competently procedure via multicore processor. Furthermore, every ISP holds huge figure of connectivity as it is linked with numerous structures such as switches, routers, servers, etc. If topologies of ISP can examine in the shape of diagram theory then ISP topologies is bottomed on dense graph somewhat than sparse graph. Dijkstra algorithm is also a way to find out the shortest distance between source and destination but it cannot give us a shortest distance among all pair of nodes. In proposed approach, Row Based FLOYD WARSHALL Algorithm is more effective than Dijkstra's Algorithm because it works best for dense graph problem while Dijkstra work is good in sparse graph problem, moreover it can also provide the shortest distance among all couple of nodes.

3.1 FLOYD WARSHALL Algorithm

Floyd WARSHALL is an algorithm which provide us a shortest distance among all nodes. To clarify Floyd WARSHALL and Row based Floyd WARSHALL we suppose a topology which is specified in Figure. (2). In Figure. (2) we have five nodes such as {R1, R2, R3, R4, and R5} with edges value.

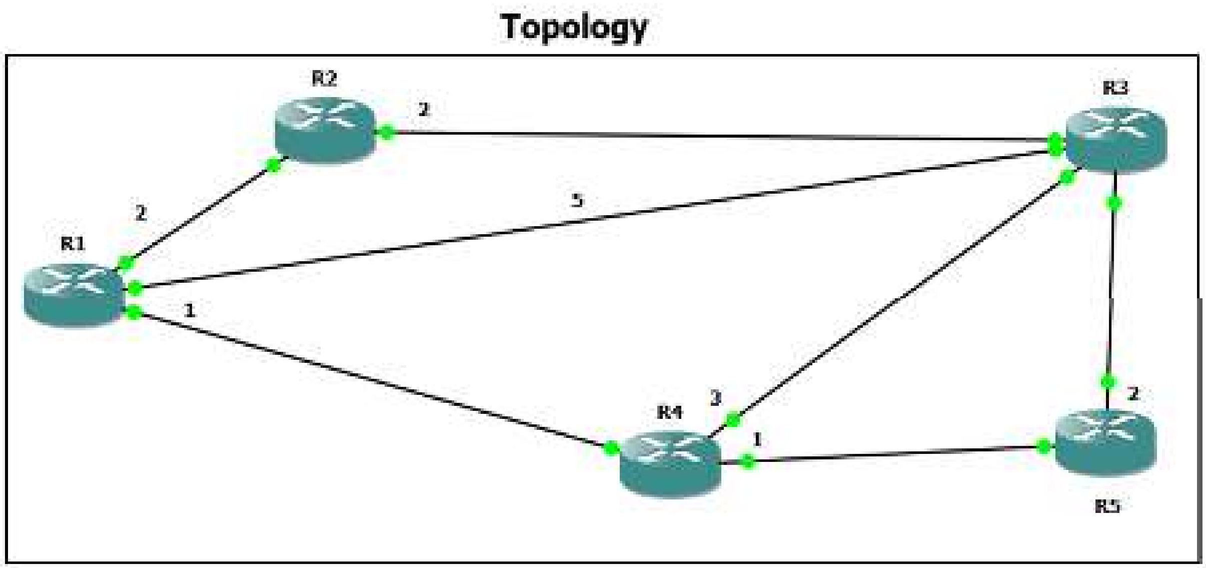


Figure 2: Simple Network Topology

Concerning FLOYD WARSHALL Algorithm on above topology and obtain ensuing Matrix Value:

$$\begin{Vmatrix} 0 & 2 & 4 & 1 & 2 \\ 2 & 0 & 2 & 3 & 4 \\ 4 & 2 & 0 & 3 & 2 \\ 1 & 3 & 3 & 0 & 1 \\ 2 & 4 & 2 & 1 & 0 \end{Vmatrix}$$

Applying Row- based FLOYD WARSHALL Algorithm on above Matrix:

First row Summation=9

Second row Summation=11

Third row Summation=11

Fourth row Summation=8

Fifth row Summation=9

Results: Row based FLOYD WARSHALL algorithm proof that 4a is the location where cache server should place. Hence 4a is the ideal place for the placement of cache server and novel topology with solving of placement problem is shown in Figure. 3.

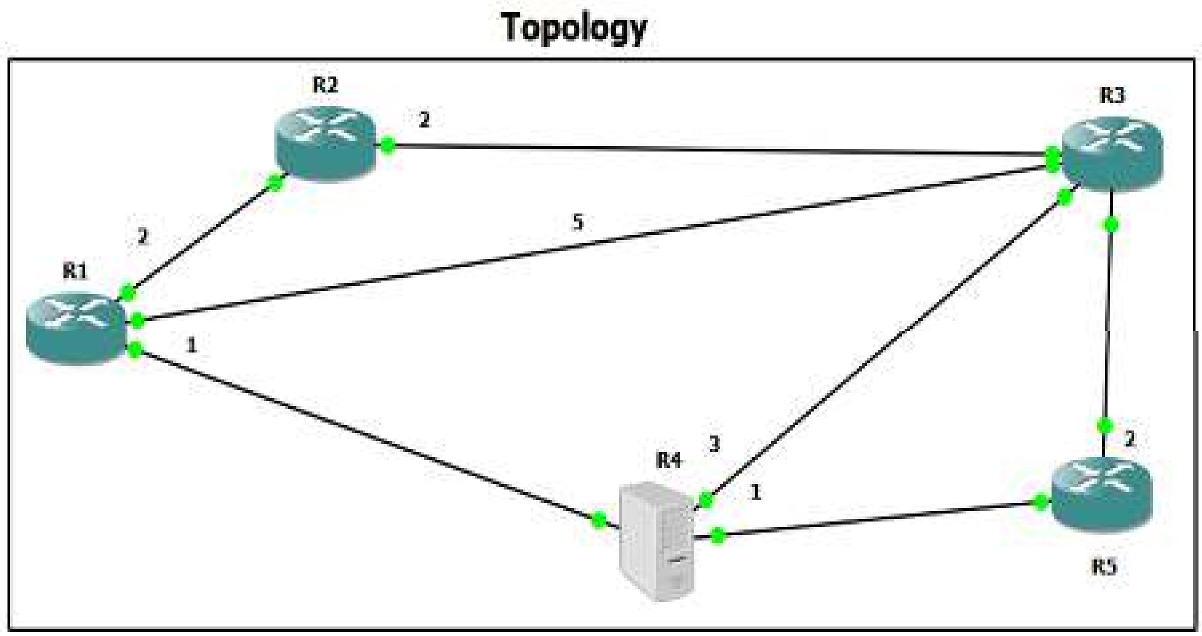


Figure 3: Placement of server in topology

3.2 Placement of cache server

The Proposed Strategy is bottomed on FLOYD WARSHALL algorithm and Row based FLOYD WARSHALL algorithm. Floyd-WARSHALL algorithm is determine the shortest distance of each node with other node. FLOYD WARSHALL algorithm is relating on projected design as specified in Figure. 4. We suppose that three ISP's in same geographical location and every ISP holds one cache Server. FLOYD WARSHALL algorithm offers a matrix (**D_{ij}^{k}**) which gauges the shortest delay of every node of additional node so we obtain three matrixes. Every Matrix belongs to all ISP. Main aim is less computational time to process the packet and decrease the factor of latency. Multi-core processor can carry out matrix-based computation extremely quick and this information will create burly our future approach.

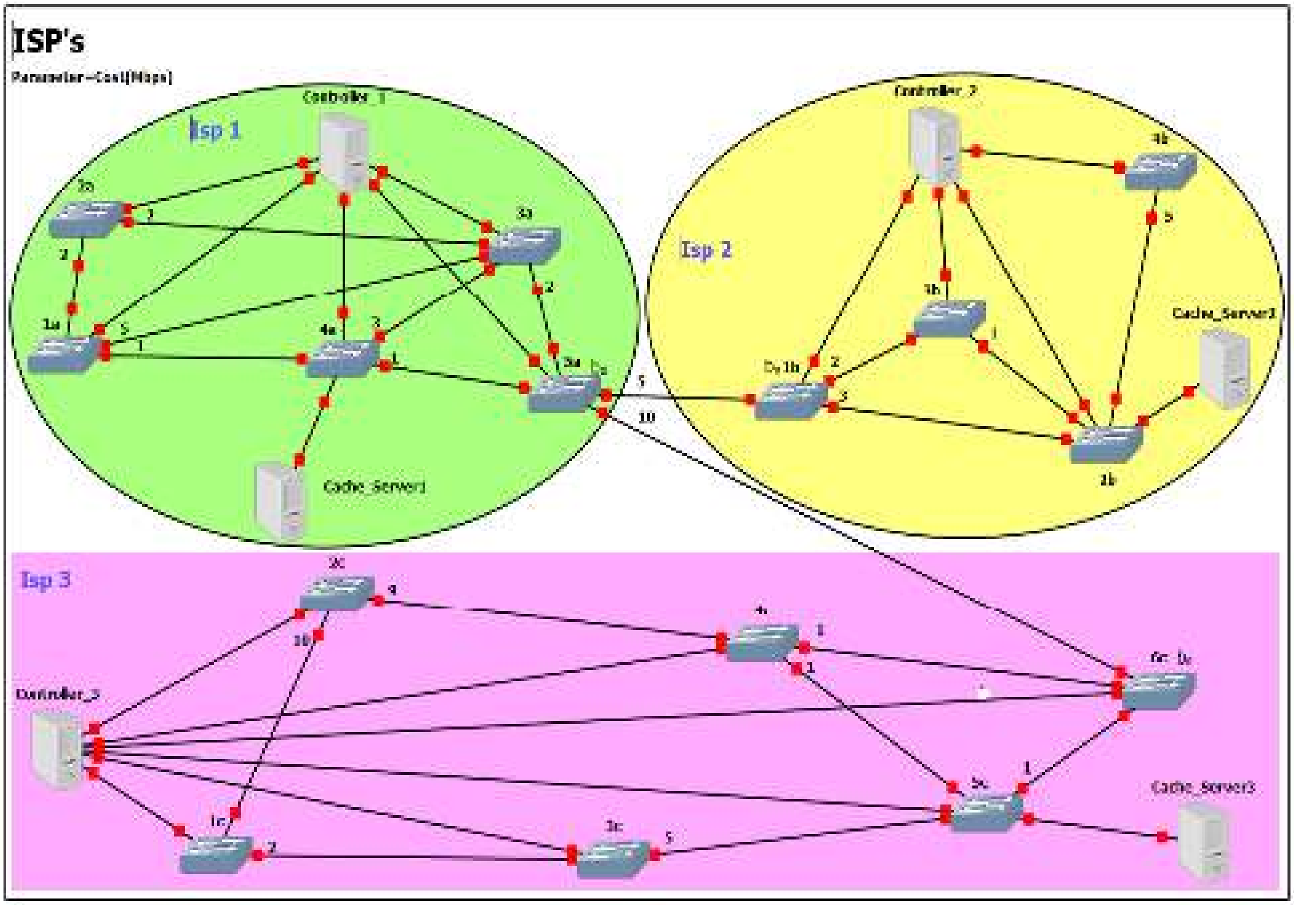


Figure 4: Placement of Cache Server on each ISP

Row based FLOYD WARSHALL algorithm assists us to decide which vertex or node is decided as a cache server. Regard as a network where a node can be a controller (ɵ), switch(S), a client, or a cache server (C_s). Users issue content requests (R_{eq}) to the OpenFlow switch(S). Every user can admittance some ISP switch(S). We planned ISP topology keen on graph (G= {V, E}). Here **"V"** feel right to network nodes and **"E"** is the edges. Our main purpose to lessen the delay. It assumed that's bandwidth is same among all nodes and take delay as parameter(μ). Relating Row supported FLOYD WARSHALL algorithm on every topology and located cache server on intended site by as prearranged in Figure. 4. Row based FLOYD WARSHALL algorithm is sprint on every ISP's controller to gauge the top trail to put the cache server in preferred ISP. ISP Controller also add the flow in each OPENFLOW switch which tells the path of cache server in each ISP. Specified subsequent information as revealed in Table. 1**.**

Table 1: Symbols and Description of proposed Approach

Symbol	**Description**
ɵ	Controller
S	Open flow switch
D_{ij}^{k}	Matrix of “i” row and “j” column which give shortest distance among each node or vertex.
Ω	Shortest Delay
€	Row based vertex
ḉ	Content
C_s	Cache Server
b_o	Border_openflow switch
R_{eq}	Request of content
R_{es}	Response of content
F_T	Flow Table in switch
µ	Link State Information (delay metric)
Dc	Database of Controller
D_{cs}	Database of Cache server
cc	Cache Controller
U	Client who request for content
I	Internet

Th	Threshold_value
tr	Timer_value

Placement of cache server in proposed network model is big issue. Two Algorithm FLOYD WARSHALL, row-based FLOYD WARSHALL algorithm resolving the matter of position of cache server. The Algorithms of Proposed Approach for post of cache server is specified in Algorithm1 [19] and Algorithm2 as shown in Table.2 and Table.3.

Table 2: FLOYD WARSHALL ALGORITHM

Algorithm1. Floyd-warshall algorithm

Input: n=number of vertices {v1, v2, ….., vn}, D=Initial Matrix of each vertices with edges value, Parameter ← Bandwidth or link.

Output: FLOYD WARSHALL MATRIX (Dijk)

1. Start

2. Apply Loop

3.for k ←1 to n

4.do for i←1 to n

5.do for j ←1 to n

6.D_{ij}^{k}←min (D_{ij}^{k-1}, D_{ik}^{k-1}*D_{kj}^{k-1})

7.Diagonal element of D_{ij}^{k} is zero

8. Inner Loop close

9. Inner Loop close

10. Outer Loop close

11. End

Table 3: Row based FLOYD WARSHALL Algorithm

Algorithm2. Row based FLOYD WARSHALL algorithm

Input: n=number of vertices {v1, v2, ….., vn}, D=Initial Matrix of each vertices with edges value, Parameter ← Bandwidth or link.

Output: BEST LOCATION FOR THE PLACEMENT OF CACHE SERVER

1. Start

2. Apply Loop

3. do for i←1 to n

4. do for j ←1 to n

5. row[i] ← sum(d_{ij}^{k})

6. **Ω** =min (row [1] to row[n])

7. Inner loop close

8. Outer loop close

9. Apply loop for i← 1 to n

10. If (shortest_delay **(Ω)**==row[i])

11. Select row[i] FOR THE OPTIMAL PLACEMENT OF CACHE SERVER

12. Else

13. i+1

14. End ifelse

15. End Loop

16. END

Flow chart of Row based FLOYD WARSHALL ALGORITHM is shown in Figure. 5.

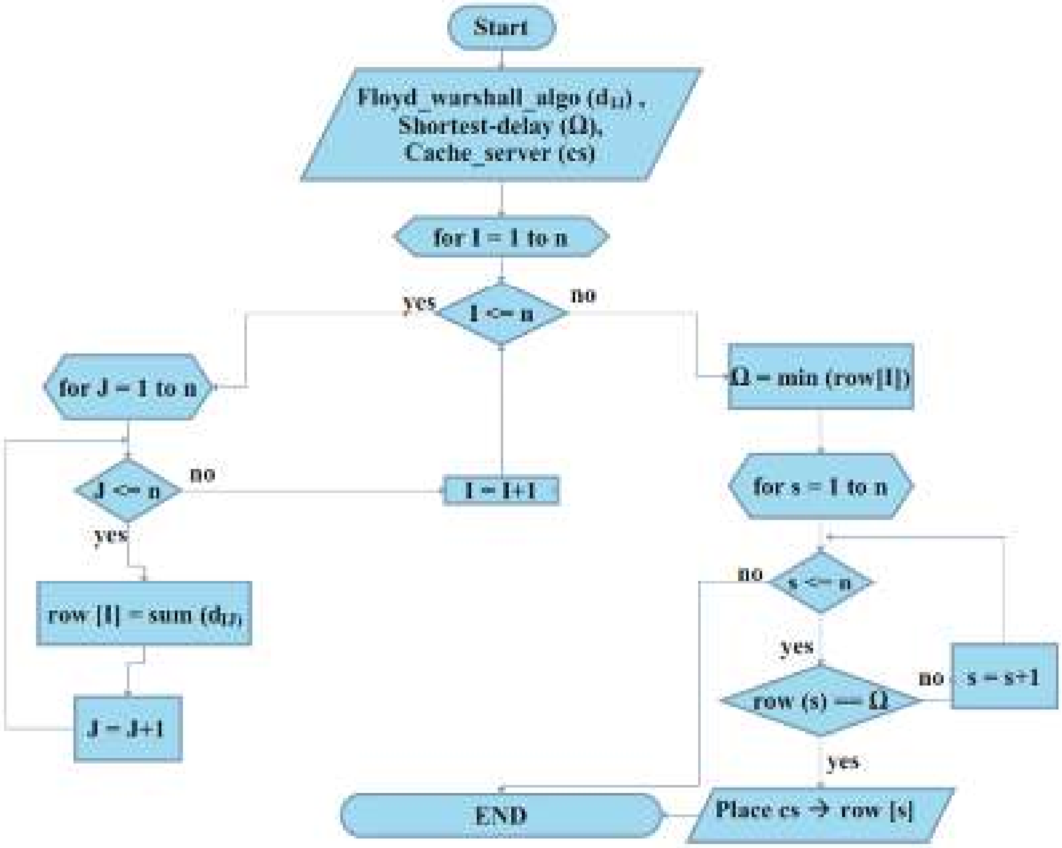


Figure 5: FLOW CHART OF ROW BASED FLOYD WARSHALL ALGORITHM

We assume so as to Row-based FLOYD WARSHALL algorithm based on two limitations:

1. Number of Nodes.
2. Metric value (delay) among Nodes.

Every ISP Controller run the row-based FLOYD WARSHALL algorithm as well as located the cache server in its ISP. It ought to be renowned if u modifies the topology and metric value (delays value) then cache server placement will as well vary.

3.3 Placement of ISP controller

Placement of cache server is not sufficient, cache memory management and traffic flow or traffic route is also the tasks explained in proposed approach. The Isp's_Controller task is just to find the shortest path of cache server in each ISP and configure the flow entries of ISP cache server in OPENFLOW switch. Furthermore, if a few consumers approach in ISP and ask for to content during such OPENFLOW switch which is not directly linked with controller then we countenance latency. To resolve the above issue, controller and OPENFLOW switch is connected though Mesh Topology as shown in Figure. 6.

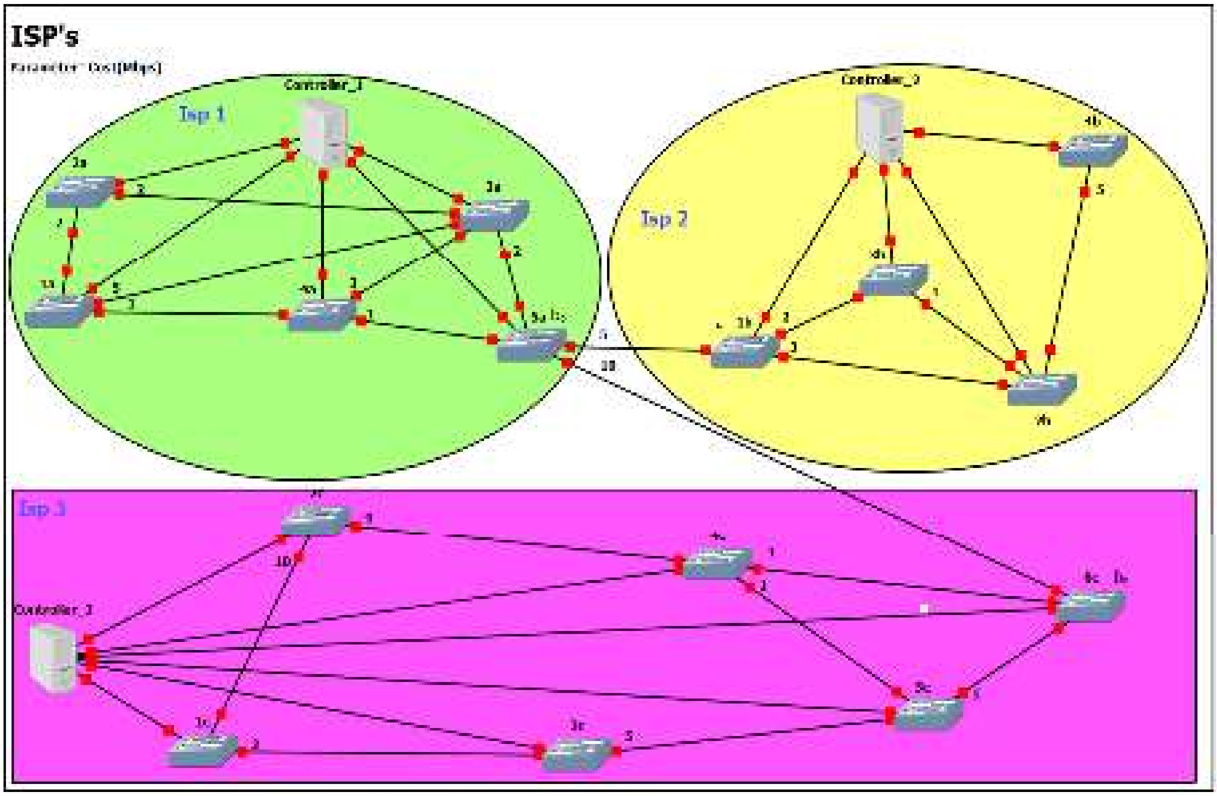


Figure 6: SDN Controller Placement in each ISP

3.4 Remove the load on ISP Controller

Placement of cache server is not enough, cache memory organization and traffic flow or traffic route is as well the errands clarify in projected work. JUNBI approach [20] gives three main reasons why the loads of controllers must be taken into concentration whenever designing a placement policy: 1. Server capacity limitation 2. The delay of message processing in controller 3. Load balancing or Impact of failure of one controller to other controllers in network. It must be noted that, If the load of a controller approaches towards a threshold value, the message processing delay on the controller will rise largely depend upon result in [27]. Moreover, in some scenarios, the failure of largely-loaded controller may affect other controllers [28]. In proposed topology introduce the concept of Cache_controller located in IXC. The ISPs_Controller job is now to discover the best possible trail of cache server in every ISP and arrange the flow entries of ISP cache server into OPENFLOW switch. ISP's Controller turns out to be overfull if it as well allocates the job of routing and running the cache memory. ISPs which is collocated in the same geographical location such as ISPs in ISLAMABAD are JAZZ, PTCL, UFONE etc. Each ISP have HTTPs privilege to store specific type of data. We should need some mechanism which is trustworthy (each ISPs trust on it) and help to perform efficient cooperative caching. To resolve this difficulty, Cache Controller is initiated. Today each user wants less latency with fast internet connection. Cache controller can reduce the weight on ISP controller and as well turn into the basis of cooperative caching as it is communal amongst every ISP's cache server as well as holds every sequence of all cache server content keen on its database. Cache Controller to insert the flow into cache server as exposed in Figure. 7. It is suggested that every ISP's cache server as well as cache controller friendly through internet.

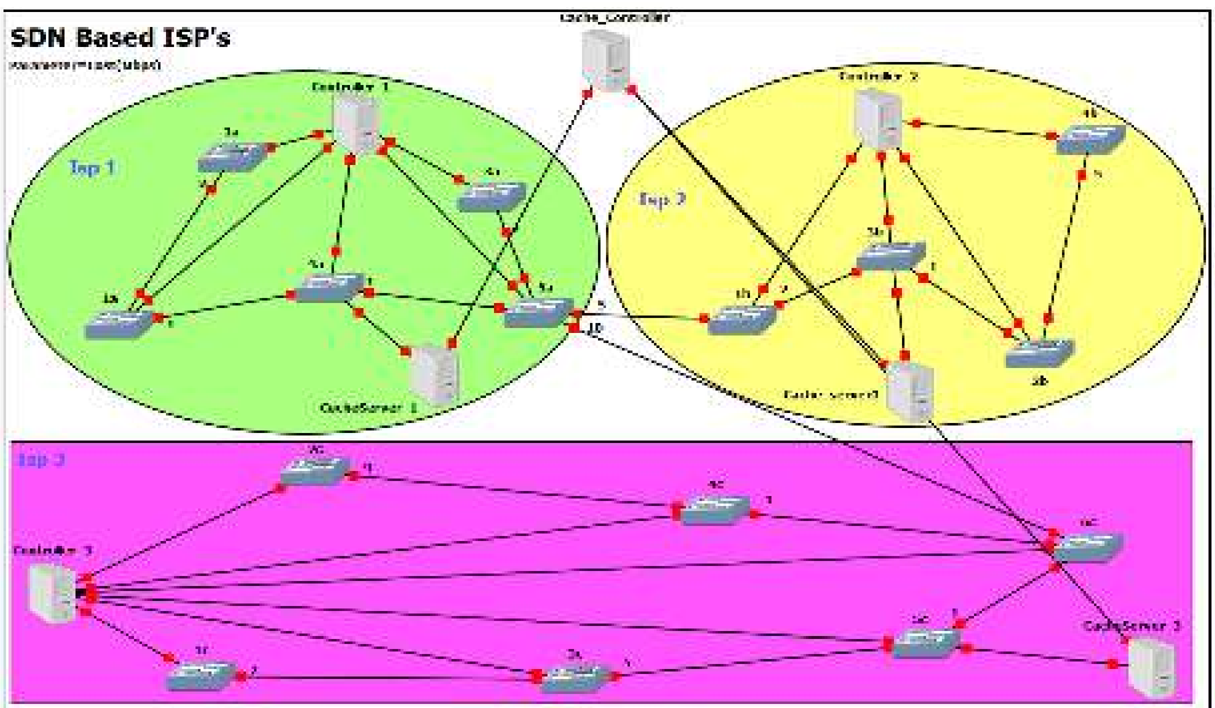


Figure 7: Introduction of Cache Controller in Proposed Topology

3.5 Managing the Cache Content

Placement of cache server is not enough but requires such instrument which competently itinerary the demanded transfer from side to side cache servers. Of course, it needs two main segments **(1)** organization of the cache Server memory **(2)** Optimally direct the demand of consumer. Currently primary argue is the administration of cache memory. Cache memory administration should believe numerous features such as:

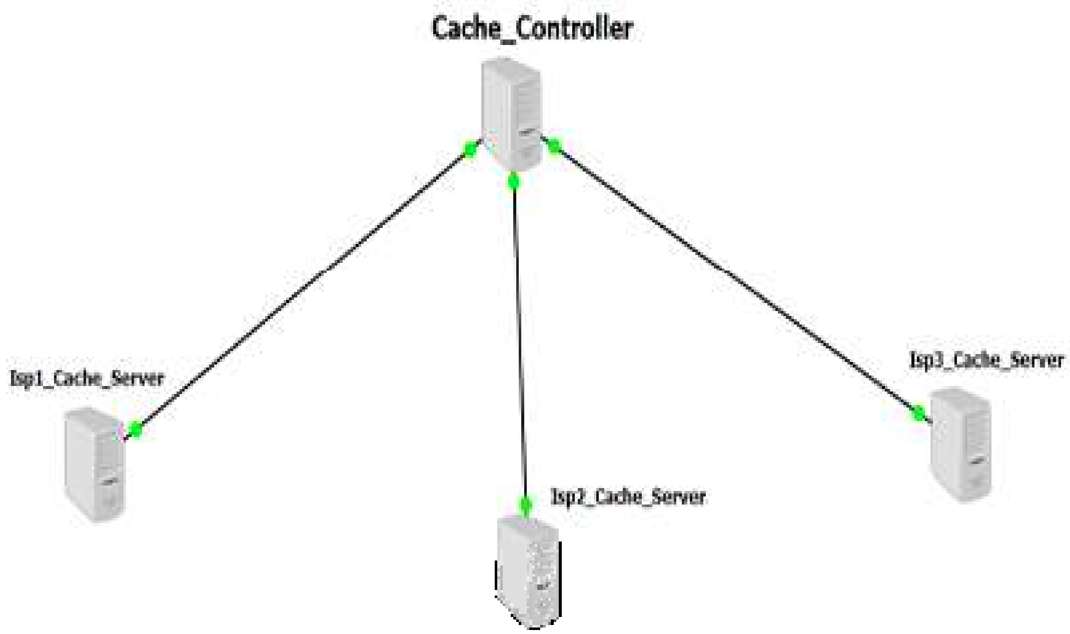
1. What should be the storage capacity of cache server and cache controller?
2. In which situation cache server competently put the content in cache server as well as cache controller?
3. If the cache server and cache controller recollection (amount) turn out to be filled then which cache placement strategy will be followed?

In given case, it considers that cache server and cache controller can handle bulk of requests, contain very fast memory to save the big data and there is no congestion, security and network bandwidth consumption issue in proposed topology. In proposed approach it is also suggest that cache server and cache controller also installed internet service. In given approach, each ISP contains cache server and each cache server coordinates with each other. Cache server placed in distributed fashion. Various distributed cache systems were proposed in [21-22], Multi-cache infrastructure mainly consists of hierarchical cache, distributed cache and hybrid cache. Hierarchical Web cache mention in the Harvest project [23]. Similar content placement schemes are explained in [24], [25], [26], which purpose to reduce the average access costs. In proposed approach, at initiate cache server is unfilled and whenever any client asks for contented then cache server demand the similar content from internet. On response cache server send the desirable content to its user and updates its information table without saving the content in its cache server. ISP cache server and cache controller obtain needs from consumer and uphold its database with information table as revealed in Figure. 8. Information table of ISP's cache server holds six field such as hash value of URL content, name of cache server, type of content, content state, counter and timer value. Hash value of URL content can help to search the content. Each content is divided into several chunks and each cache server can hold full content or several chunks of each content in distributed way. Type of content is placed in information database. Type of content contains the info of the content or specific part of content called chunk. Another field is ISPs cache server name. Multiple ISPs cache server send the content and each ISPs cache server have unique name so this field is very helpful to find the content of specific cache server. Other field is content state that whether it is cached content or non-cached content. Another field of information table called counter contains the frequency of request. At any time any consumer demand for some kind of content, then the occurrence of ask for similar content located in

Counter, for instance, in ISP1 consumer demand software content such as (video player) then cache server modifies its table with the IP address of demanded consumer, IP address of its cache server, sort of content which is about video player and allocate counter with 1. After a little instance, a few additional consumers as well needs for the similar video player then cache server immediately update the information table by altering the defy worth from one to two. Cache controller too holds the similar in turn table similar to cache server but with little altering. In cache controller in turn table holds IP address of cache server (who propel demand to cache controller rather than consumer IP address).

It has to be renowned that cache server and cache controller have threshold value which contrasted by counter. Timer is as well-worn in cache server and cache controller which informs the period in which servers recognizes requirements from consumer and following conclusion of timer if threshold will be activated then cache server and cache controller carry out exact kind of deed. Presume when cache server or cache controller begin, timer is as well initialized with zero signify it is as well establish. Cache server and Cache controller limited the period of timer. For instance, cache server puts timer extent corresponding to two days and threshold value is identical to hundred. Inside two days, cache server just collects demand of exact category of content allows “Type A” content as well as revise its in information table with its admission. After completion of duration of timer, if the counter value triggered the threshold, mean if counter value will become equal to 100 or number of requests approaches to hundred then cache server saves the content in its database and update its counter value equal to zero. If the counter value is less than 100 then cache server send the information of content to cache controller and initialize its counter value equal to zero.

Likewise, cache controller too puts its threshold value and counter value. Cache controller organizes with ISP’s cache server and obtain in information table evidence from ISP’s cache server. For instance, cache controller put a timer worth equivalent to five days and threshold equivalent to two hundred. Inside five days, a cache controller just obtains record of exact kind of content from three ISP’s cache server. After five day when timer worth restrictions finish, then cache controller examines its database information table. Information table holds entries similar to cache server IP address, cache controller IP address, kind of content and its counter. Presume three consumers from dissimilar ISP needs similar sort of content “Type B”. User A from Isp1 propels 40 requirements for TYPEA content, user B from ISP2 propels 200 demands and user C from ISP3 propels 350 requirements. It should be renowned that, in ISP some consumer propel demand to cache controller on the behalf of cache server. Cache controller just count the number of requests of specific type of content and match its value with its threshold value. In this example only ISP2 and ISP3 users meets the threshold value of cache controller. Now, cache controller sets its timer value zero and cache server2 and cache server3 content entry delete from its database. Cache controller automatically request the content from internet and sends content to desirable ISPs cache server to save it. Next time any user request for "TYPEB" content then cache server not need to request it from internet because cache controller already provides "TYPEB" content to its cache server. Cache controller become very useful because it saves bandwidth, increase throughput, increase page load time and most important decrease end to end delay (latency).



Hash Value of Content	ISPs Cache Server Name	Type of Content	Counter	Timer Value	Content State

Figure 8: Information Table record Entry

3.5.1 Cache Replacement Strategy

In Yong Cui Approach [19] design an increment recording approach based on (LRU) policy [29]. In given approach, if the cache server is filled, then employ least recently used (LRU) rule for content substitute.

Algorithm of Optimal cache management of Cache server and cache controller is given in Table. 4.

Table 4: Algorithm of memory management of cache server & controller

Algorithm3. Memory management of cache server algorithm

Input: Req=Request of content from user(U), Internet (I), threshold value (th), Timer (Tr), Information table (It)
Output: Optimal cache the content in cache server

1. Start

2. If (Cs memory → Full)

3. Implement LRU replacement strategy to vacant space in Cs.

4. End If

5. Start Ifelse

6. If (cache_controller(ģ) is recieved)

7. Save ģ → Cs

8. counter=0

9. content_state == cached

10. update(it)

11. Send(it) → cc

12. End If

13. if (Req → Cs)

14. Inner ifelse

15. If (Cs counter_value== triggered && tr==threshold_value)

16. Save ç → Cs

17. Cs Counter_value == zero

18. Cs content_state==cached

19. Share Updated It info → cc

20. Else

21. Counter_value++

22. Share Update It → cc

23. End inner ifelse

24. else

25. send(it) →cc

29. end outer ifelse

30. End

Algorithm4. Memory management of cache controller algorithm

Input: Req=Request of content from user(U), Internet (I), threshold value (th), Timer (Tr), Information table (It)
Output: Optimal cache the content in cache controller

1. Start

2. Received(it) from Cs

3. If (cc memory → Full)

4. Implement LRU replacement strategy to vacant space in cc.

5. End If

6. If (cc counter_value== triggeerd && tr==threshold_value)

7. Req(c) → I

8. Res(c) from I

9. send (c) → ISPs_Cs

10. Update_It record info

11. else

12. Update (it)

13. End Ifelse

14. End

Flow chart of Managing memory of cache_server and cache_controller is shown in Figure. 9 and 10.

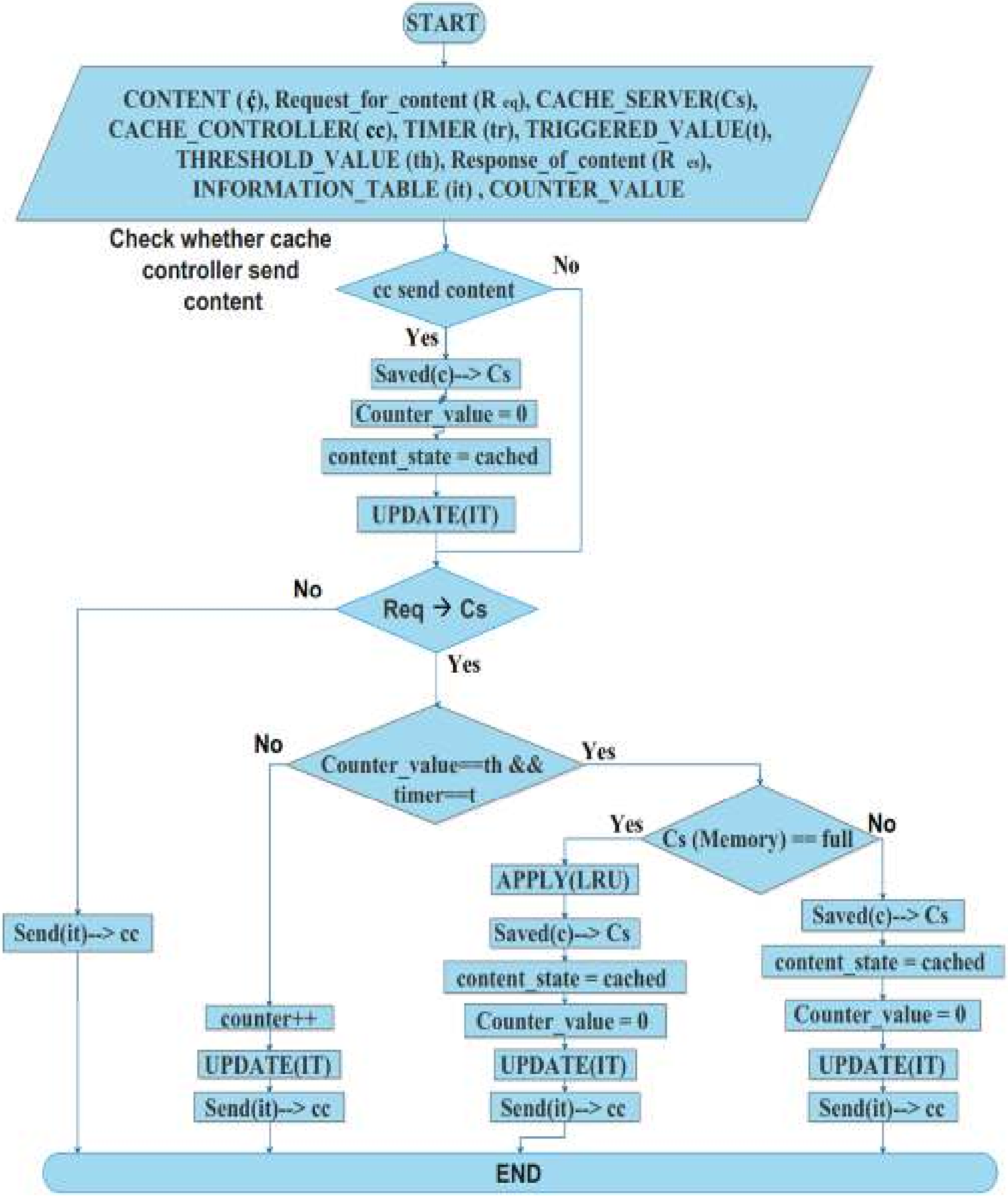


Figure 9: Flow Chart of Cache_server memory management

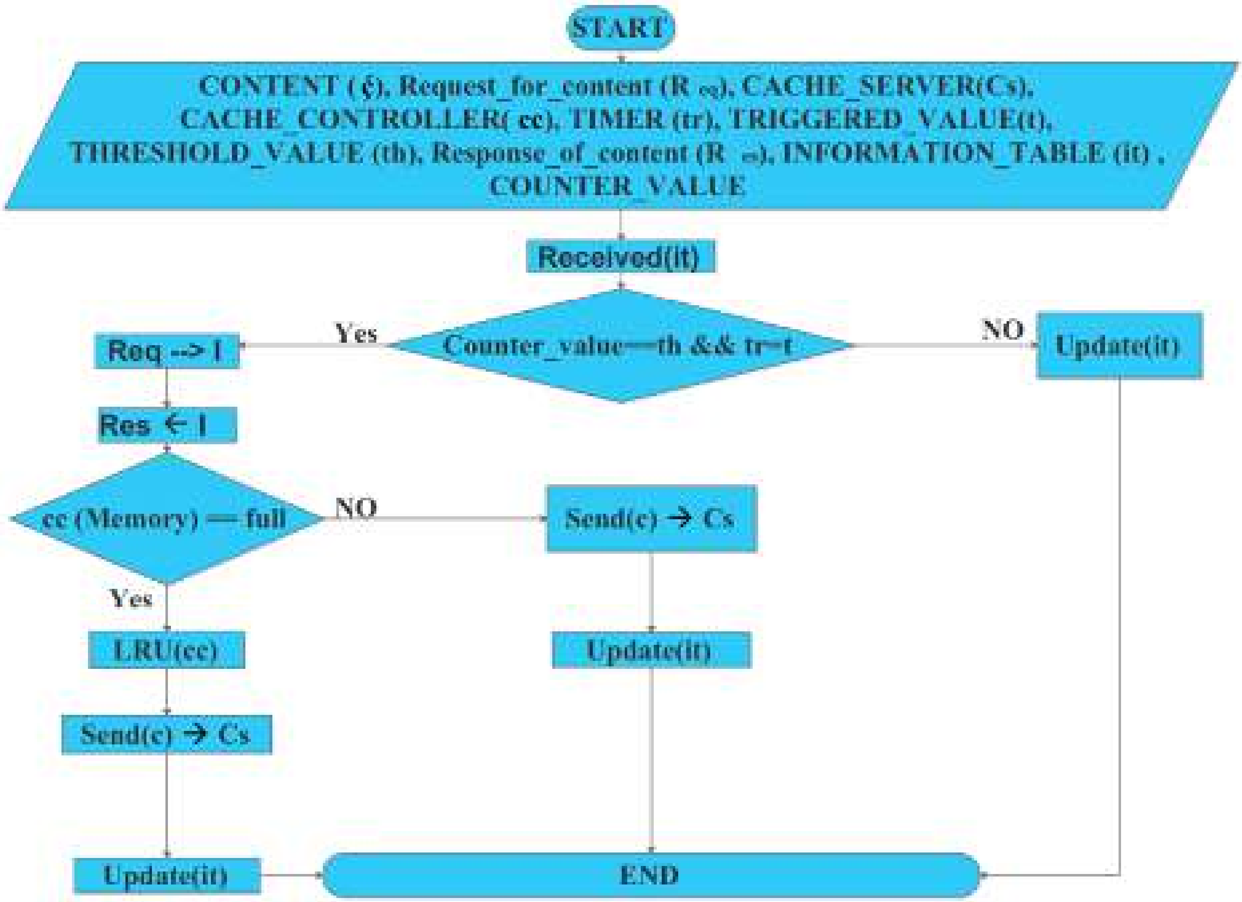


Figure 10: Flow Chart of Cache_controller memory management

3.6 Routing the Cache Content

How traffic route in proposed approach is mention in this section. Whenever cache server gets any request from anywhere it must share information with cache controller. At this stage it assumed that cache is managed properly and system is ready to successfully route the traffic. At start cache server will empty and start filling cache of the cache servers with the consumption of time. Each ISP cache server is synchronized with Cache Controller. When cache server gets any request from user it also shares the information of content with shared cache controller**(cc),** cache controller updates the information in its database. Cache controller measure the frequency or counter of requested content. When cache controller meets its threshold value of requested counter, it automatically requests the content using internet. On response from internet, Cache controller automatically send content to desire able cache server. Cache server store copy of content and send the content to requested client through switch.

For instance, client **(u)** such as **ISP1_User** from any isp1 demand for content **(R_{eq})**, it propels demand for its ISP switch such as **1a** as shown in Figure. 11. After getting demand, the switch seems up its onward table. The flow table of the switch **(F_T)** will have the flow regulation merely of its obtainable ISP cache server. Cache Server holds the data as well as database in sequence of content. Switch 1 ahead the demand to its Cache Server. Now at this phase multiple situation survives which are clarified in below:

1. Cache Server check whether content is available in its cache server or not.
2. If content is present in cache server then it transmits the content**(ç)** to client with response (**R_{es}**) as shown in Figure. (12).
3. If Content is present in other's ISP, cache server first checks the content in cache controller. Cache controller which provides the cooperative caching and hold the details of content state of each ISPs cache server. if it obtained the attribute of content state equivalent to cached then cache server not send the request to internet. Whilst it gets the content from cache controller save in its memory and response it to user as shown in Figure. (13).
4. If the content is not present in Local cache server and content is not present in cache controller then two cases occur.
5. if user request for content and threshold value is triggered then cache server forward the request of user to internet. Internet communicate with origin server and reply back to cache server. Cache server save the content and reply back to user as shown in Figure. 14.
6. if user request for content and threshold value is not triggered then cache server forward the request of user to internet. Internet communicate with origin server. Instead of replying back to cache server, internet directly send the content to user as shown in Figure. 15.

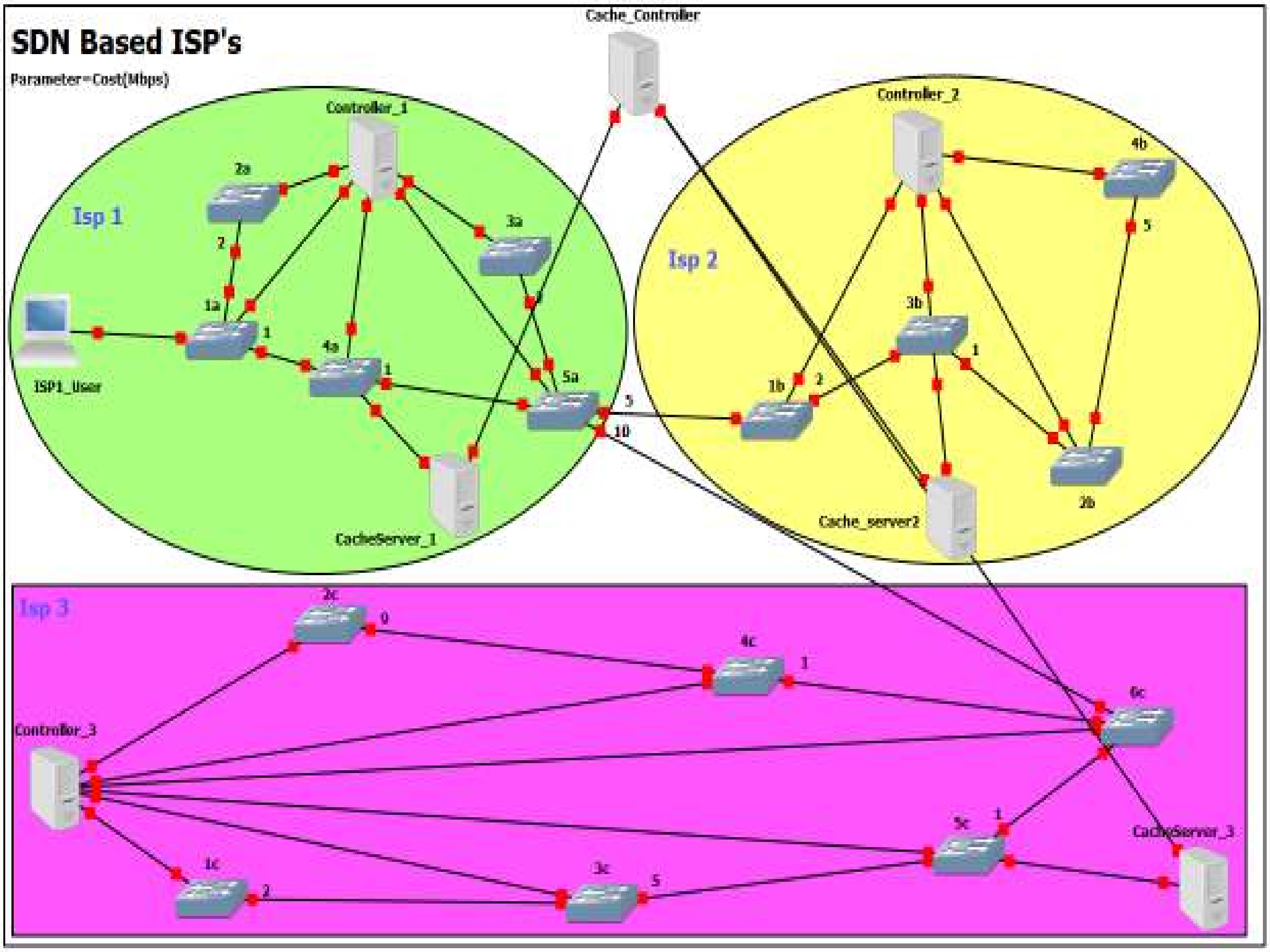


Figure 11: ISP1_User Request content in Proposed Topology

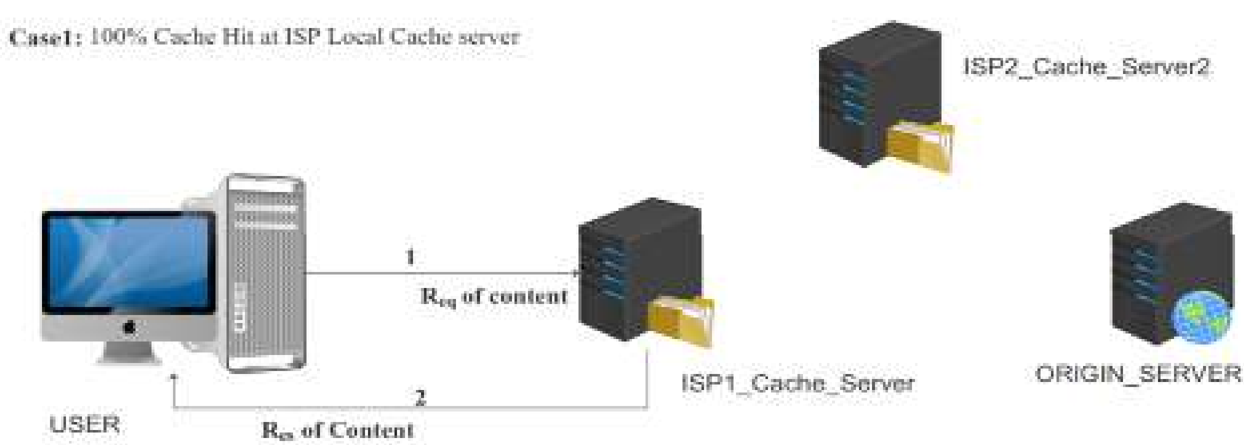


Figure 12: Proposed Scenario at 100% Cache Hi

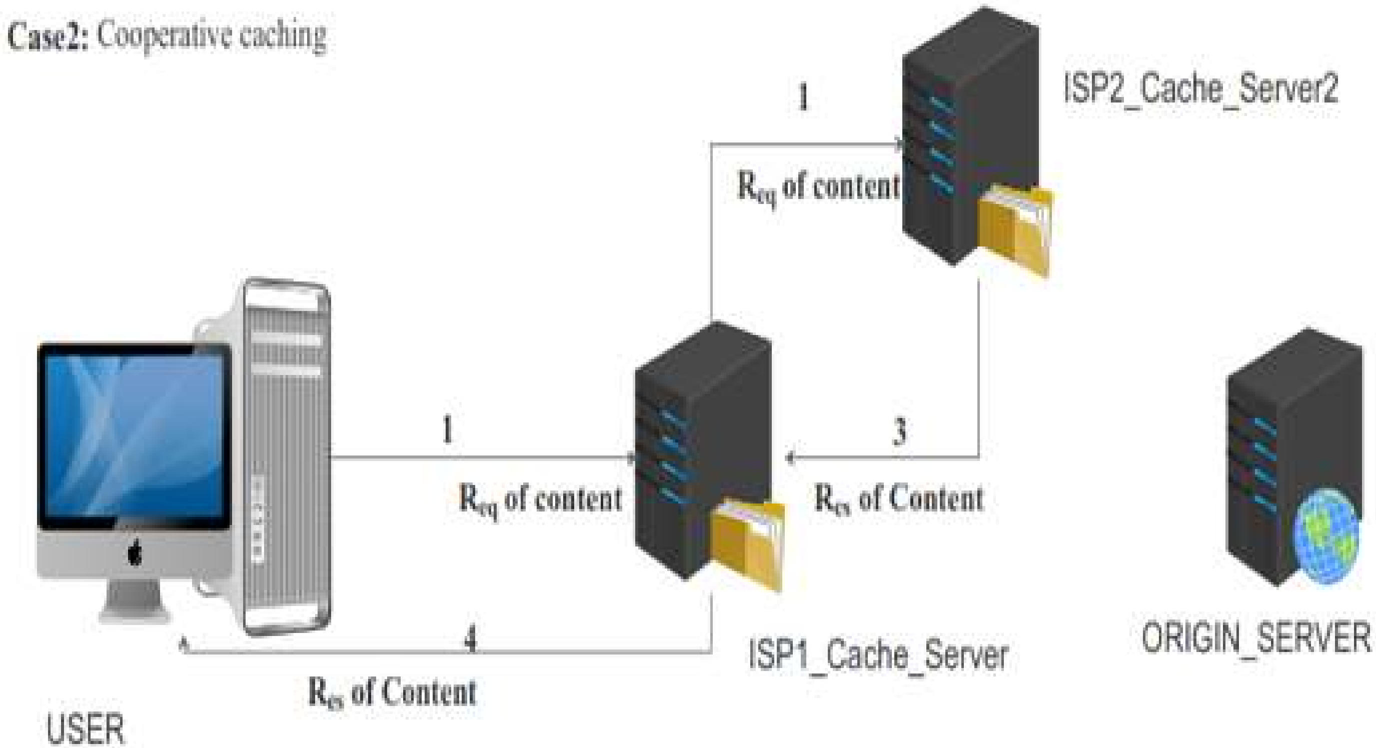


Figure 13: Cooperative Caching

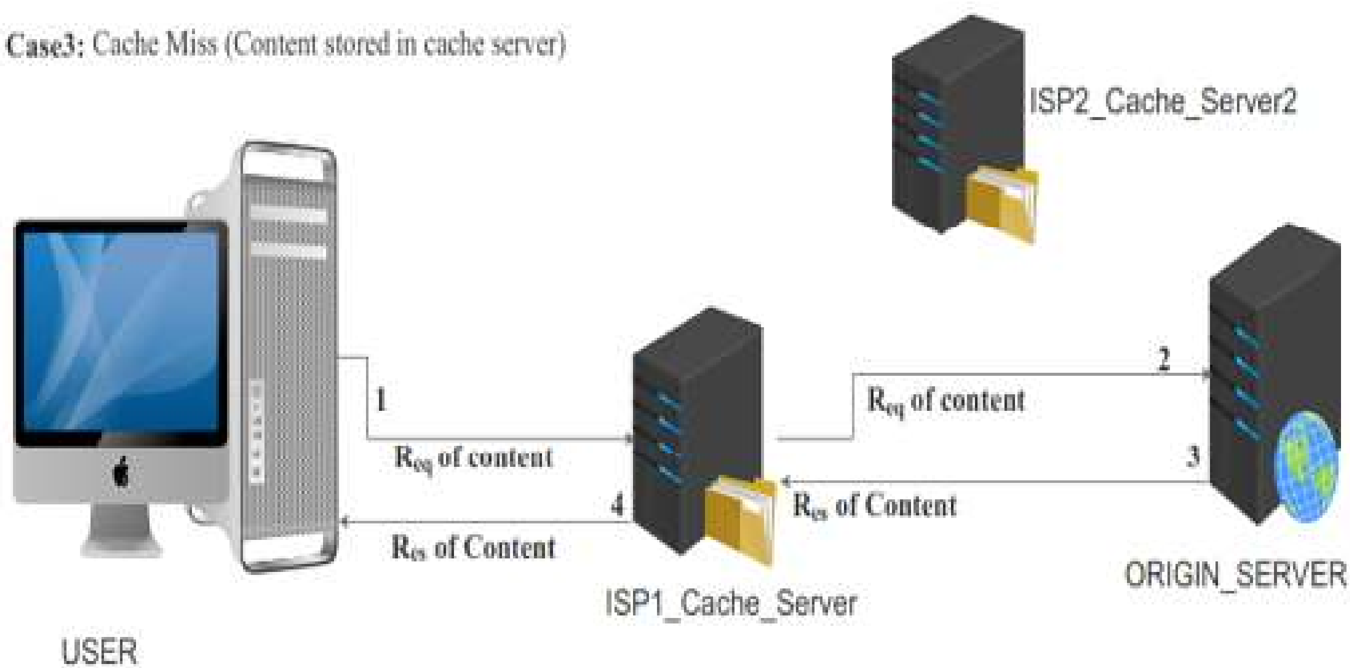


Figure 14: Cache miss when content stored in cache server

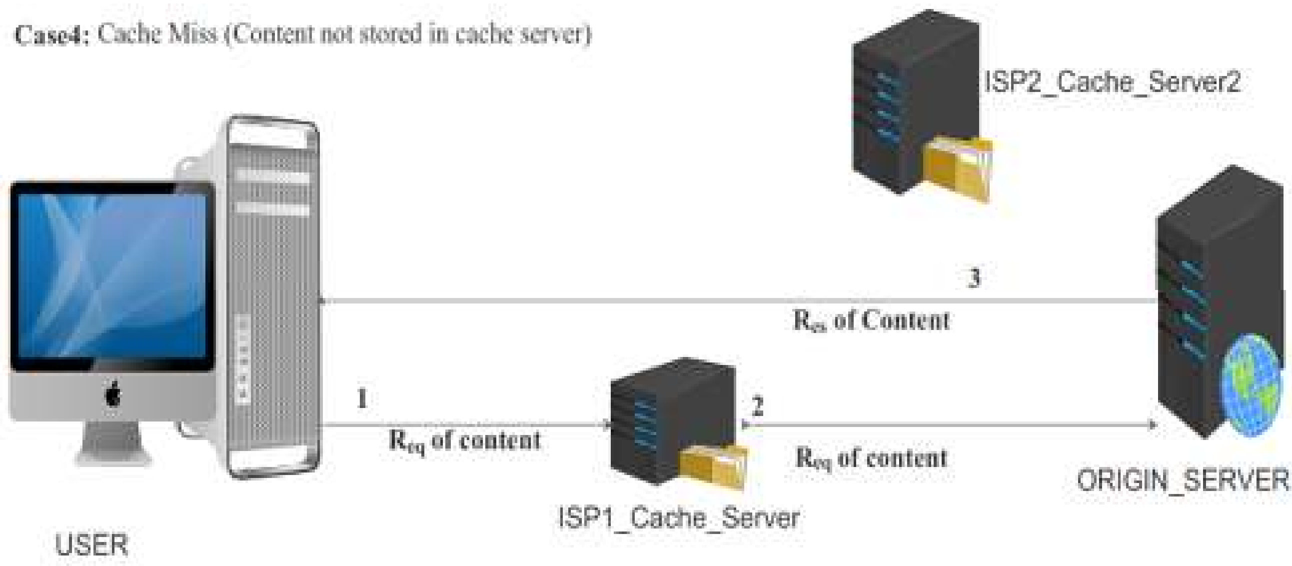


Figure 15: Cache miss when content not stored in cache server

Algorithm of optimal route the traffic of user to fulfills his demand is given in Table 5.

Table 5: Algorithm to Optimal route the content

Algorithm5. Algorithm to Optimal route the content

Input: Req=Request of content from user(U), Internet (I), threshold value (th), Timer (Tr)
Output: Optimal path to route the traffic of user to fulfills his demand.
1. Start

2. Req (ç) → Cs

3. Start Ifelseif

4. If (Cs cached_content== cached)

5. Cs gives ç → U

6. Else If (Cs cache_content != cached && cc cached_content ==cached)

7. cache_controller request content from internet

8. Internet give content to controller

9. cc gives ç → Cs

10. Cs gives ḉ → U

11. Else if (Cs cache_content != cached && cc cached_content !=cached)

12. Two cases occur

13. If (th== triggerd && tr ==ended)

14. Cs Req(ḉ) → I

15. I gives ḉ → Cs

16. Cs gives ḉ → U

17. else

18. Cs Req(ḉ) → I

19. I give ḉ → U

20. End if else

21. End ifelseif

22. End

Flow chart of Managing the route of content is shown in Figure. 16.

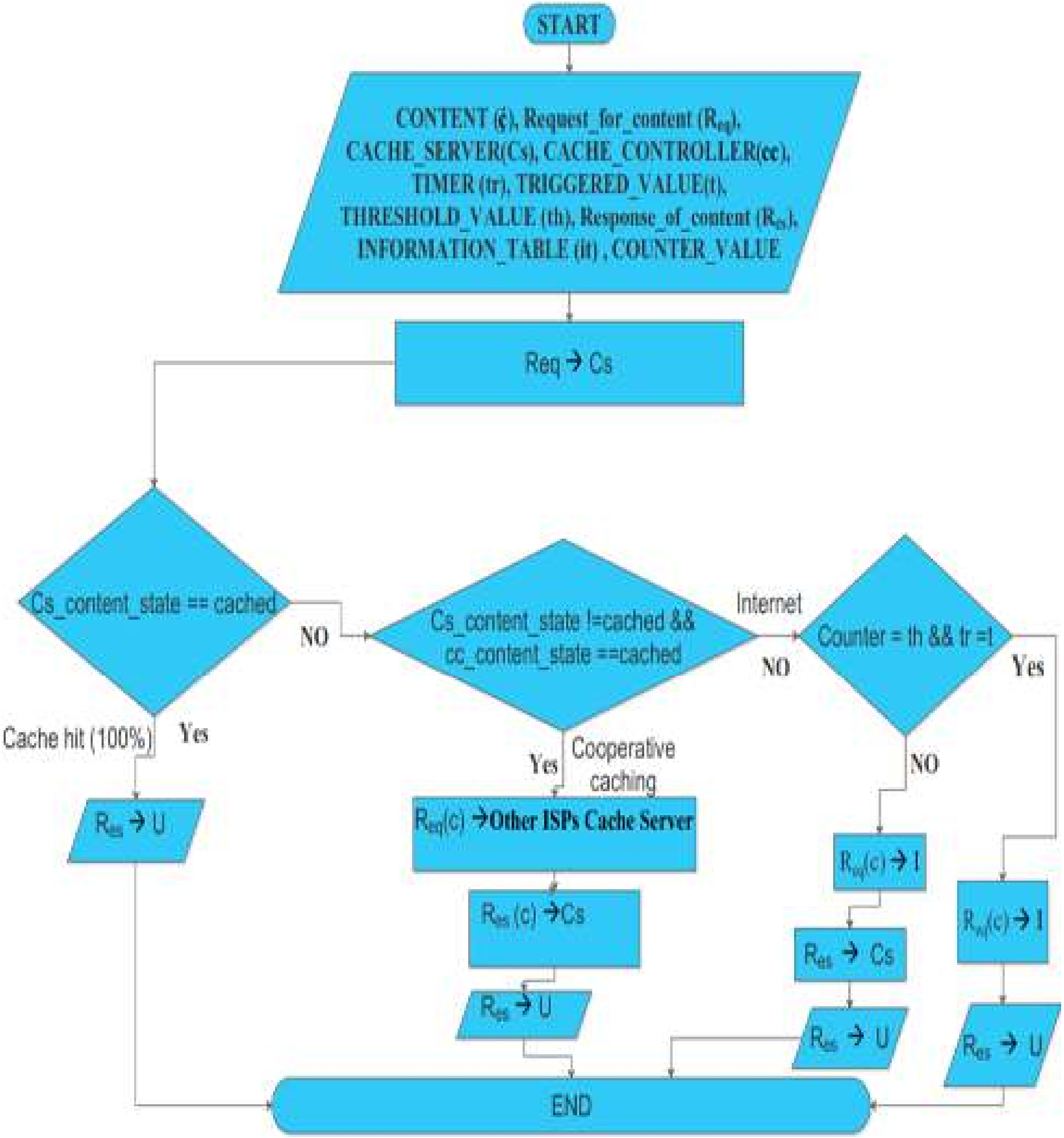


Figure 16: Flow chart of Managing the route of content

CHAPTER 4: EXPERIMENTAL SETUP AND RESULTS

4.1 Experimental Setup

For implementation and experimental verification of our proposed algorithm, used Advanced Network Simulator tool ns-3 [30]. This tool helps us to test our algorithm by using state of the art protocols and network infrastructure. Ns3 is a time driven event-based simulator which gives result at each event. In Ns3, used two important module such as OPENFLOW 1.3 [31] for SDN support and flow monitor [32] for deduct results. OPENFLOW switch contains one or more DROPTAIL Queue in each port. OPENFLOW 1.3 module give the OFSwitch13LearningController class which used "learning bridge controller" This learning controller guides the switches to forward the flow from one port to the desired output port [31]. If users want to add the flow in OpenFlow switches they can used "DPCTL" command. By using DPTCL command each ISPs SDN controller add entry in his OpenFlow switches to communicate with other ISPs. Ns3 can set OPENFLOW switch attributes such as FLOWSIZE, GROUPSIZE, METERSIZE, PIPELOADS etc. In Simulated scenario, it considers that FLOWSIZE=30, GROUPSIZE=30, METERSIZE=30, PIPELOADS having data rate equivalent to 1Mbps. It must be noted that if packet exceeding the pipeline capacity of switch then packet will be dropped. By default, the connection between OPENFLOW switches and controller are using CSMA channel and used IP address of 10.100.0.0 network with subnet mask of 255.255.255.0. In OPENFLOW 1.3 module, SDN controller used 65536 port to listen incoming packets. OPENFLOW switch contains one or more DROPTAIL Queue in each port.

In projected situation, it is assumed that, cache server send content at the rate of 100kbits/sec and each pack consists of 1000Byte or 1MTU=1000 so 12.5 pack will be sent at every second.

4.2 Experimental Scenario

In imitation, believe, consumer demand of content from switch (1a) to cache server and cache server is located according to row based FLOYD WARSHALL algorithms show in Figure (17). Table. 6 also elaborate the experimental metrics.

Table 6: Experiment Metrics

Bandwidth	Traffic Type	MTU	Data rate	Simulation time
2Mbps	Constant Bit Rate	1000 byte	100Kb/s	3 Minute

Here we take the ISP 1 scenario and measure the average end to end delay under all cases as explain in previous section shown in Figure [12-15].

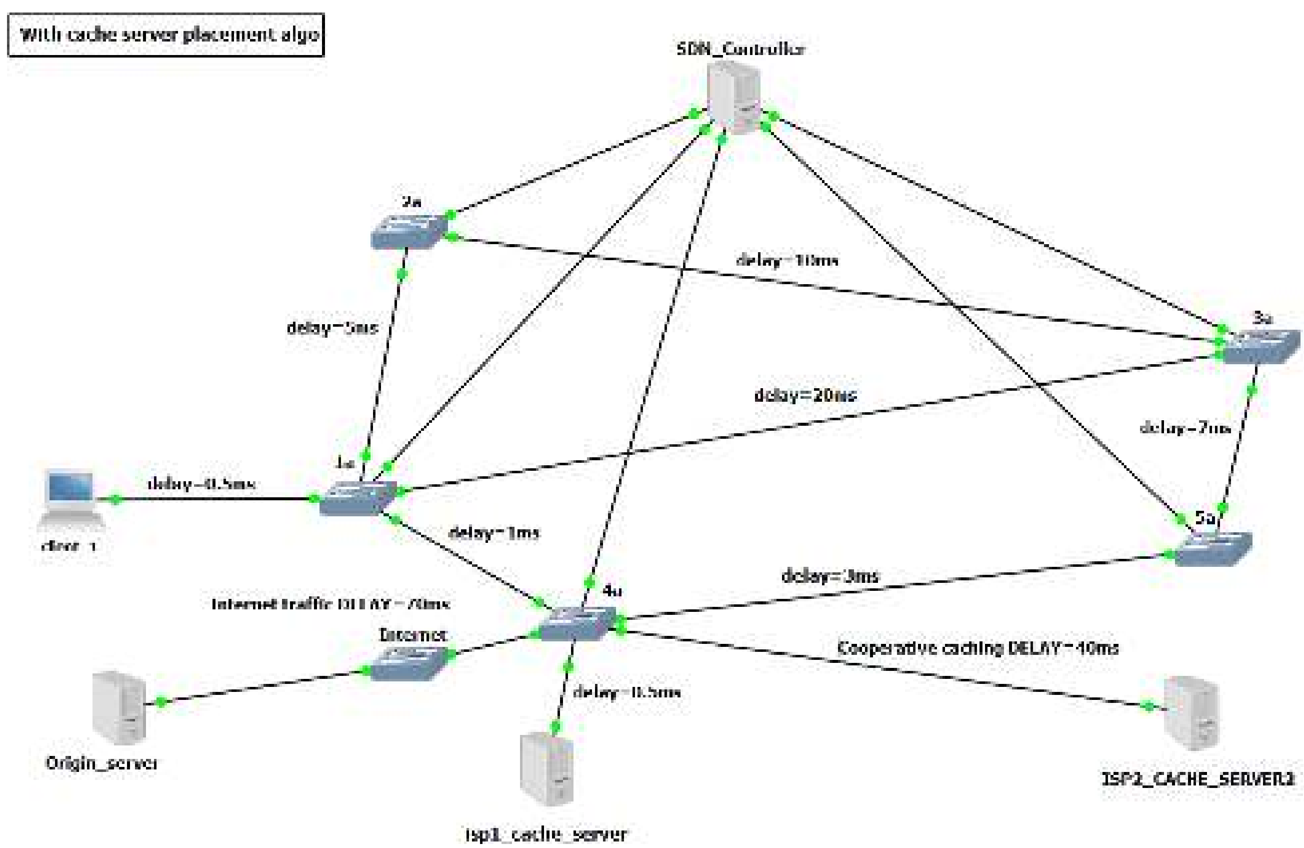


Figure 17: Simulated Topology

4.2.1 Row based Floyd WARSHALL Calculation

Row based FLOYD WARSHALL Matrix results shows in Table. 7.

Table 7: Matrix of Algorithm

Row based FLOYD WARSHALL Matrix	1a (delay)	2a (delay)	3a (delay)	4a (delay)	5a (delay)	Total Latency

OPENFLOW SWITCH (1a)	0	5ms	11ms	1ms	4ms	1a= 21ms
OPENFLOW SWITCH (2a)	5ms	0	10ms	6ms	9ms	2a= 30ms
OPENFLOW SWITCH (3a)	11ms	10ms	0	10ms	7ms	3a= 38ms
OPENFLOW SWITCH (4a)	1ms	6ms	10ms	0	3ms	4a= 20ms
OPENFLOW SWITCH (5a)	4ms	9ms	7ms	3ms	0	5a= 23ms

After Adding cache server and client link delay in Row based Floyd WARSHALL matrix then matrix value is updated is shown in Table. 8.

Table 8: Matrix of algorithm with client link delay

Row-based FLOYD WARSHALL Algorithm	1a (delay)	2a (delay)	3a (delay)	4a (delay)	5a (delay)	Total Latency
OPENFLOW SWITCH (1a)	1ms	6ms	12ms	2ms	5ms	1a= 26ms
OPENFLOW SWITCH (2a)	6ms	1ms	11ms	7ms	10ms	2a= 35ms
OPENFLOW SWITCH (3a)	12ms	11ms	1ms	11ms	8ms	3a= 43ms
OPENFLOW SWITCH (4a)	2ms	7ms	11ms	1ms	4ms	4a= 25ms
OPENFLOW SWITCH (5a)	5ms	10ms	8ms	4ms	1ms	5a= 28ms

4a<1a<5a<2a<5a. Hence according to row-based FLOYD WARSHALL, cache server should be placed at OPENFLOW switch (4a).

4.3 Experimental Parameters

It is worth mentioning that the results are based on two factors

1. Measure the average end to end delay in SDN based ISP network with Row-based FLOYD WARSHALL placement algorithm.

2. Measure the average end to end delay in SDN based ISP network without optimal placement of cache server.

In simulation, results of average end to end delay based on, clients request for content from each switch to cache server. Results are divided into two phases. In one phase, consequence is based on placement of cache server using row-based FLOYD WARSHALL algorithm. Whilst, in second phase results show the effect of end to end delay in proposed SDN based ISP network with the placement of cache server without using row-based FLOYD WARSHALL algorithm.

It is not probable to attain 100% cache hit. It should be renowned that, in the majority cases cache miss is familiarity in usual system so cache miss can generate the additional holdup. By Using Cache placement algorithm, the results in Figure. [18-22] show that the end to end delay of our proposed approach is comparable to the without cache placement approach in all scenarios.

4.3.1 SDN based Controller and OPENFLOW switch 1.3 latency

SDN Controller and OPENFLOW switch communicate with each other because controller add flow to switch which help switch to route the user traffic. SDN Controller and OPENFLOW communication create constant latency**(4.2µs).**

4.4 Results

To demonstrate the performance of cache server at every location in ISP1, we used different percentages of cache hits as shown in Figure. 21, and 22. First, we test the SDN based isp1 network performance in 100% cache hit as shown in Figure. 18. Result at 100% cache hit produce much less delay which favors towards the good performance of SDN based ISP network. In order to observe the effect of cache miss in SDN based ISP network, we traverse the content request to internet. In this scenario two cases occur 1. Figure. 20 show the effectiveness of cooperative caching. When some content is present in cache server and some content request from internet, cache server store the copy of content and forward the request to user such results is shown in Figure. 19. Whilst in second case, when cache server not saved the content in its cache server and internet direct forward the request to user Figure. 21. represent the effect of cooperative caching in end to end delay. In more rational way, different cache hit ratio represent the effectiveness of proposed methodology in Fig. [18-22]. The results successfully reveal that cache server is a good approach toward big data in SDN based ISP network.

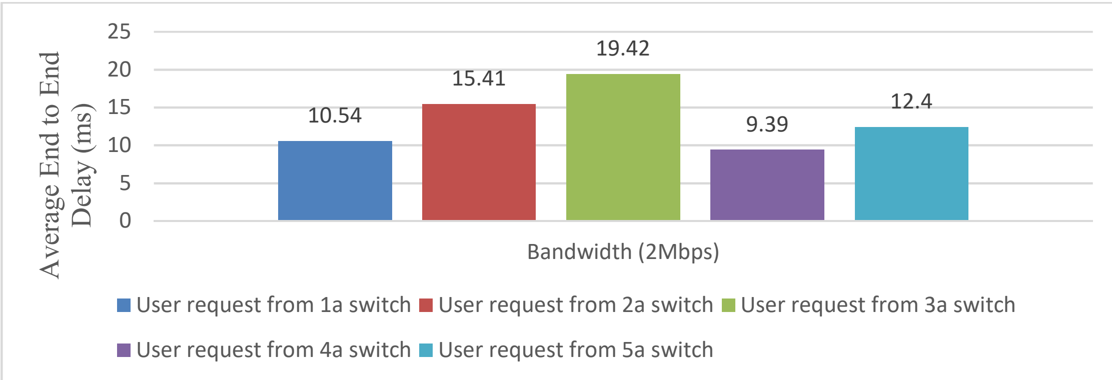


Figure 18: Cache server placed at 4a switch (100% cache hit)

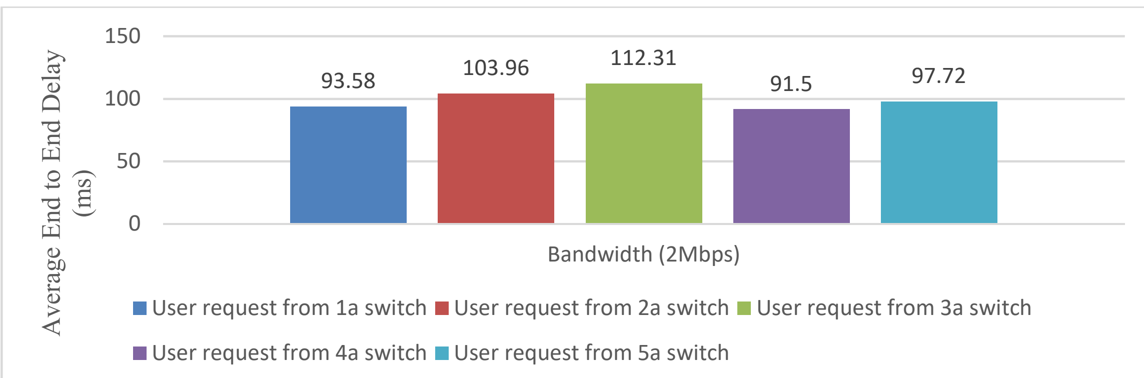


Figure 19: Cache server placed at 4a switch (0% cache hit content from internet)

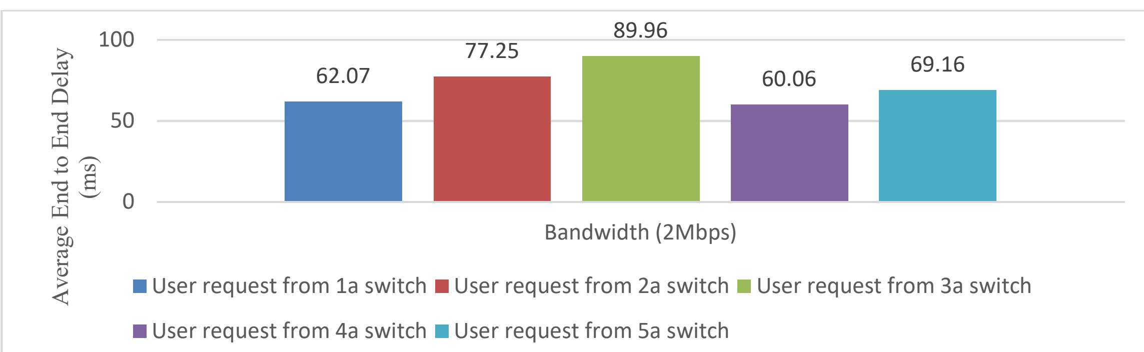


Figure 20: Cache server placed at 4a switch (cooperative caching)

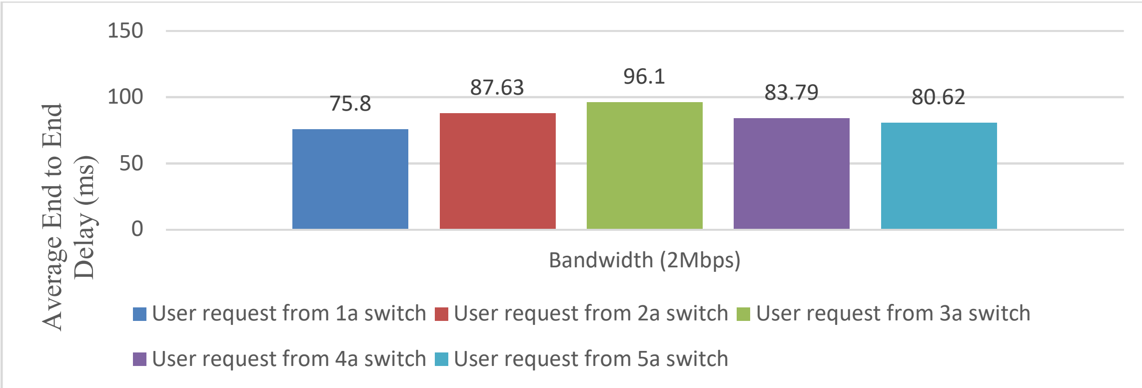


Figure 21: 10% Cache_Server1, 30% Cache Hit Cache_Server2, 60% From Internet

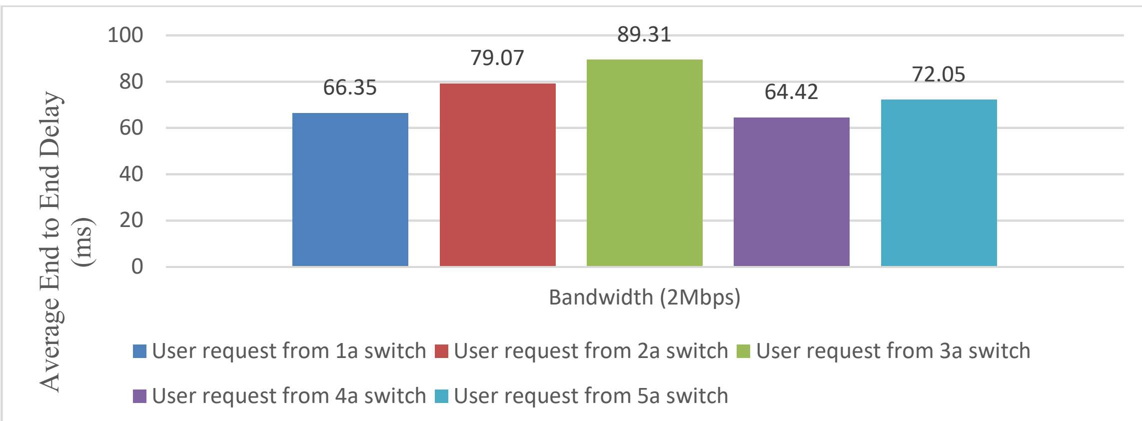


Figure 22: 10% Cache_Server1, 60% Cache Hit Cache_Server2, 30% From Internet

In order to compare and elaborate the cache placement algorithm we placed the cache server at all the location in ISP1 as shown in Table [9-11].

Table 9: Average end to end delay at 0% cache hit using 2Mbps

Cache server Location	User request from 1a	User request from 2a	User request from 3a	User request from 4a	User request from 5a
1a	91.54ms	101.95ms	114.43ms	93.64ms	99.88ms
2a	101.95ms	91.55ms	112.36ms	104.05ms	110.28ms
3a	114.43ms	112.36ms	91.55ms	112.36	106.12ms
4a	93.58ms	103.96ms	112.31ms	91.5ms	97.72ms
5a	99.88ms	110.28ms	106.12ms	97.8ms	91.55ms

Table 10: Average end to end delay at 50% cache hit using 2Mbps

Cache server Location	User request from 1a	User request from 2a	User request from 3a	User request from 4a	User request from 5a
1a	50.47ms	58.18ms	67.42ms	52.02ms	56.64ms
2a	58.18ms	50.47ms	65.89ms	59.73ms	64.35ms
3a	67.42ms	65.89ms	50.47ms	65.89ms	61.26ms
4a	52.06ms	59.68ms	65.86ms	50.44ms	55.06ms
5a	56.64ms	64.35ms	61.26ms	55.1ms	50.47ms

Table 11: Average end to end delay at 100% cache hit using 2Mbps

Cache server Location	User request from 1a	User request from 2a	User request from 3a	User request from 4a	User request from 5a
1a	9.39ms	14.41ms	20.42ms	10.4ms	13.4ms
2a	14.41ms	9.39ms	19.42ms	15.41ms	18.42ms
3a	20.42ms	19.42ms	9.39ms	19.42ms	16.41ms
4a	10.54ms	15.41ms	19.42ms	9.39ms	12.4ms
5a	13.4ms	18.42ms	16.41ms	12.4ms	9.39ms

Cache server was put in almost every location and measured it on all three cache hit ratios such as 0%, 50% and 100% but the results revealed that were taken against Row Based FLOYD WARSHALL Algorithm were better than any other option in this regard like without any cache server placement algorithm.

4.5 Analysis

After analyzing the results various aspects reveals. Whenever 100% cache hit occur it produce best result. But in rational environment, the chance of occurrence of 100% cache hit is very rare. Results in Figure. 20 show the cooperative caching, which show that less end to end delay occurs than the traffic get data from internet as shown in Figure. 19. It must be noted that paper based on big data. Cooperative caching performs much better in case of big data rather than web-based caching. It is already mention ISPs are connected through access link. Whenever network traffic travel from one ISP to other ISP then other ISP charges the cost. Here the bottleneck is existing between latency and ISP link cost. If we just open the web page then it is not a good practice to perform cooperative caching whilst in case of big data it can be a good practice. Moreover, the link between ISP is shared so it remains very congested. When your ISPs cache server found that content is present in other ISP cache server, then it sends the request of content to another ISPs cache server. Due to link delay user request will waits. If at the same time other ISPs cache_server2 performs LRU replacement policy and delete the requested content then user experience much end to end delay and cooperative caching degrade the performance of proposed network. This issue will also degrade the performance live video streaming. To handle such issue, efficient mechanism of cooperative cache is introduced and result show that by introducing the cache controller which provides the efficient mechanism of cooperative caching. It is also experience cache server placement in ISP effects the consequences of average end to end delay. In short, results exposed that cache server in SDN based ISP network's increase the performance of network by minimizing the average end to end delay, improve user experience and elaborate the challenges in SDN based cooperative caching ISPs network

CHAPTER 5: CONCLUSION AND FUTURE WORK

5.1 Conclusion

Big data holds enormous quantity of information which utilize additional bandwidth and if user has little bandwidth then big data will generate the latency to client. In traditional network, it was very difficult to perform traffic engineering. Hence, services like coordinating autonomous cache servers could only be offered at application level. SDNs allows offering such services by simply embedding the forwarding rules at flow switches. However, the benefits of cooperative caching can be best reaped by placing the cache server at optimal locations, accessible to all users with minimum latency and no bottlenecks. With optimally placed cache server, SDN controllers, and coordination mechanism among cache servers, cooperative Caching can significantly reduce average latency, increase network throughput, while still conforming to network security requirements. Cache server and new structural design named SDN resolve the plenty of challenges of big data. Cache server can reduce the weight on real server and add to throughput by lessening the end to end holdup (latency).

No doubt Cache server increase the performance of SDN based ISP network but if we design the algorithm which optimally directs where to put the cache server in topology, then result will be even better. In proposed work, row-based FLOYD WARSHALL algorithm is set up to resolve the placement of cache server. Moreover, projected approach introduced the way of content management in cache server and cache controller for cooperative caching. Projected work is trialed beneath dissimilar cache hit fraction. Proposed approach is experimented at 100%, 50%, 0% cache hit ratio. The imitation consequences shaped diagrams which demonstrate that the projected advance which via row-based FLOYD WARSHALL algorithm to put the cache server, augments the throughput and reduce the latency (end to end holdup) in network. Cooperative caching become very effective in SDN based ISPs by lessening the traffic load onto the access link, by minimizing average response time of content, and by increasing average throughput of the network.

5.2 Future Work

Increased network traffic over the internet has given rise to new research challenges. Video in network is a type of big data. Video content over the internet consumes up to three quarters of the total bandwidth of the network. A bulk number of users in a same network watch videos on different heterogeneous platforms. It increases unfairness among the clients and results in congestion in the network. Therefore, an intelligent system is need, which forecast the future network condition and behave accordingly before the performance of the network degrades. Secondly, as a prospect effort, would similar to examine several heuristics to additional decrease the end to end holdup (latency) as well as augment the throughput of the projected approach. Information-centric networking (ICN) will as well be worn in assumed work. Furthermore, also plan to experiment the results of specified approach on real physical environment.

REFERENCES

[1]. “Cisco Visual Networking Index: Forecast and Trends, 2017–2022 White Paper,” Cisco, 2019.

[2]. “Cisco Visual Networking Index: Forecast and Methodology, 2016–2021 White paper,” Cisco, 2017.

[3]. Amin, R., Reisslein, M., Shah, N.: Hybrid SDN networks: a survey of existing approaches, In: IEEE Communications Surveys and Tutorials. https://doi.org/10.1109/COMST.2018.2837161 (2018).

[4]. D. Kreutz, F. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” Proc. IEEE, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[5]. W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, “A survey on software-defined networking,” IEEE Commun. Surveys Tuts., vol. 17, no. 1, pp. 27–51, 1st Quart., 2014.

[6]. B. Wang, Y. Zheng, W. Lou, and Y. T. Hou, “DDoS attack protection in the era of cloud computing and software-defined networking,” in Proc. IEEE 22nd Int. Conf. Netw. Protocols, Oct. 2014, pp. 624–629.

[7]. F. Hu, Q. Hao, and K. Bao, “A survey on software-defined network and OpenFlow: From concept to implementation,” IEEE Community. Surveys Tuts., vol. 16, no. 4, pp. 2181–2206, 4th Quart., 2014.

[8]. V. Pacifici, F. Lehrieder, and G. Dan, “Cache capacity allocation for BitTorrent-like systems to minimize inter-ISP traffic,” in Proc. IEEE INFOCOM, 2012, pp. 1512–1520

[9]. Y. Huang, T. Z. Fu, D.-M. Chiu, J. Lui, and C. Huang, “Challenges, design and analysis of a large-scale P2P-VoD system,” in Proc. ACM SIGCOMM Conf. Data Commun., 2008, pp. 375–388.

[10]. S. Borst, V. Gupta, and A. Walid, “Distributed caching algorithm for content distribution networks,” in Proc. IEEE INFOCOM, 2010, pp. 1–9.

[11]. M. Qian, Y. Wang, Y. Zhou, L. Tian, and J. Shi, “A super base station based centralized network architecture for 5G mobile communication systems,” Digit. Commun. Netw., vol. 1, no. 2, pp. 152- 159, 2015.

[12]. Yong Cui , Jian Song, Minming Li , Qingmei Ren, Yangjun Zhang, and Xuejun Cai, “SDN-Based Big Data Caching in ISP Networks”,pp. 1-3,2018.

[13]. Kim, D., Kim, Y.: Enhancing NDN feasibility via dedicated routing and caching. Comput. Netw. 126, 218–228 (2017)

[14]. Aubry, E., Silverston, T., Chrisment, I.: SRSC: SDN-based routing scheme for CCN. In: 2015 1st IEEE Conference on Network Softwarization (NetSoft), pp. 1–5. IEEE (2015)

[15]. Qingxia Chen, F. Richard Yu, Fellow, IEEE, Tao Huang, Renchao Xie, Jiang Liu and Yunjie Liu, “Joint Resource Allocation for Software-Defined Networking, Caching, and Computing”, 2018.

[16]. N. McKeown, et al., “OpenFlow: Enabling innovation in campus networks,” ACM SIGCOMM Comput. Commun. Rev., vol. 38, pp. 69–74, 2008.

[17]. D. Levin, A. Wundsam, B. Heller, N. Handigol, and A. Feldmann, “Logically centralized?: State distribution trade-offs in software defined networks,” in Proc. 1st Workshop Hot Topics Softw. Defined Netw., 2012, pp. 1–6.

[18]. Q. Duan, “Modeling and performance analysis for composite network–compute service provisioning in software-defined cloud 8 CONCLUSION environments,” Digit. Commun. Netw., vol. 1, no. 3, pp. 181–190, 2015.

[19]. Stefan Hougardy “The Floyd–Warshall algorithm on graphs with negative cycles”, pp. 1-2,2010.

[20]. JunBi, “On the Capacitated Controller Placement Problem in Software Defined Networks”, pp. 2-2, 2014.

[21]. S. Sanadhya, R. Sivakumar, K.-H. Kim, P. Congdon, S. Lakshmanan, and J. P. Singh, “Asymmetric caching: Improved network deduplication for mobile devices,” in Proc. 18th Annu. Int. Conf. Mobile Comput. Netw., 2012, pp. 161–172.

[22]. S.-H. Shen and A. Akella, “An information-aware QoE-centric mobile video cache,” in Proc. 19th Annu. Int. Conf. Mobile Comput. Netw., 2013, pp. 401–412.

[23]. A. Chankhunthod, P. B. Danzig, C. Neerdaels, M. F. Schwartz, and K. J. Worrell, “A hierarchical internet object cache,” Proc. Annu. Conf. USENIX Annu. Tech. Conf., 1996, pp. 13–13.

[24]. W. Li, E. Chan, G. Feng, D. Chen, and S. Lu, “Analysis and performance study for coordinated hierarchical cache placement strategies,” Comput. Commun., vol. 33, no. 15, pp. 1834–1842, 2010.

[25]. Y. Kim and I. Yeom, “Performance analysis of in-network caching for content-centric networking,” Comput. Netw., vol. 57, no. 13, pp. 2465–2482, 2013.

[26]. M. Mangili, F. Martignon, and A. Capone, “A comparative study of content-centric and content-distribution networks: Performance and bounds,” in Proc. IEEE Global Commun. Conf., 2013, pp. 1403– 1409.

[27]. A. Tootoonchian, S. Gorbunov, Y. Ganjali, M. Casado, and R. Sherwood, “On controller performance in software-defined networks,” in Proc. HotICE, 2012, pp. 7–10.

[28]. G. Yao, J. Bi, and L. Guo, “On the cascading failures of multi-controllers in software defined networks,” in Proc. IEEE ICNP, 2013, pp. 1–2.

[29]. S. Podlipnig and L. B€osz€ormenyi, “A survey of Web cache replacement strategies,” Comput. Surveys, vol. 35, no. 4, pp. 374– 398, 2003.

[30]. (March, 2018). *NS-3 (ns-3.28 ed.)*. Available: https://www.nsnam.org

[31]. Luciano Jerez Chaves, Islene Calciolari Garcia, and Edmundo Roberto Mauro Madeira,"OFSWITCH13: Enhancing ns-3 with Openflow 1.3 Support," 2016.

[32]. Pedro Fortuna and Manuel Ricardo, "FlowMonitor - a network monitoring framework for the Network Simulator 3 (NS-3)" 2009.

[33]. Jan Badshah, Muhammad Kamran, Nadir Shah and Shahbaz Akhtar Abid, "An Improved Method to Deploy Cache Servers in Software Defined Network-based Information Centric Networking for Big Data", pp 1-23, 2019.

[34]. S. S. Krishnan and R. K. Sitaraman, “Video Stream Quality Impacts Viewer Behavior: Inferring Causality Using Quasi- experimental Designs,” in ACM SIGCOMM IMC 2012, 2012, pp. 211–224.

[35]. F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang, “Understanding the Impact of Video Quality on User Engagement,” in ACM SIGCOMM 2011, 2011, pp. 362–373.

[36]. X. Liu, F. Dobrian, H. Milner, J. Jiang, V. Sekar, I. Stoica, and H. Zhang, “A Case for a Coordinated Internet Video Control Plane,” in ACM SIGCOMM 2012, 2012, pp. 359–370.

[37]. S. Sen, J. Rexford, and D. Towsley, “Proxy Prefix Caching for Multimedia Streams,” in 19th IEEE INFOCOM 1999, vol. 3, 1999, pp. 1310–1319 vol.3.

[38]. J. V. D. Merwe, S. Sen, and C. Kalmanek, “Streaming Video Traffic: Characterization and Network Impact,” in Int. Web Content Caching and Distribution Workshop, 2002.

[39]. P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, “Towards Network-wide QoE Fairness Using Openflow-assisted Adaptive Video Streaming,” in ACM SIG- COMM 2013 Workshop on Future Human-centric Multimedia Networking (FhMN), 2013, pp. 15–20.

[40]. A. Chankhunthod at el, “A Hierarchical Internet object cache” 1996 USENIX Technical conference, pp. 153-163, San Diego, USA, Jan. 2006.

[41]. P. Rodriguez at el “Analysis of web cache architectures: Hierarchical and distributed caching,” IEEE/ACM Trans. Netw., vol.9, no.4, pp.440-418, Aug. 2001.

[42]. E. Nygren, R. K. Sitaraman, and J. Sun, “The Akamai Net- work: A Platform for High-performance Internet Applications,” SIGOPS OS Rev., vol. 44, no. 3, pp. 2–19, Aug. 2010.

[43]. M. Zhao, P. Aditya, A. Chen, Y. Lin, A. Haeberlen, P. Druschel, B. Maggs, B. Wishon, and M. Ponec, “Peer-assisted Content Distribution in Akamai Netsession,” in 13th ACM SIGCOMM IMC 2013, 2013, pp. 31–42.

[44]. Netflix Open Connect Platform, http://www.netflix.com/ openconnect.

[45]. E.k. Lua at el “A survey and comparison of peer-to-peer overlay network scheme” IEEE Common. Surv. Tutorials, vol.7, no.2,pp.77-93, 2005.

[46]. J. Pouwelse, J. Taal, R. Lagendijk, D. H. J. Epema, and H. Sips, “Real-time Video Delivery using Peer-to-Peer Barter-ing Networks and Multiple Description Coding,” in IEEE Int. Conference on Systems, Man and Cybernetics 2004, vol. 5, 2004, pp. 4599–4605 vol.5.

[47]. S.K Fayazbaksh at el “Less pain most of the gain: Incrementally deployable ICN,” proc. ACM SIGCOMM 2013 conferrence on SIGCOMM, SIGCOMM’13, pp 147-158,2013.

[48]. A. Ghodsi at el “Information Centeric Networking: Seeing the forest for trees, ” Proc. 10th ACM Workshop on hot topics in networks, HotNets”11, pp.1-6, 2011.

[49]. Meng Zhang at el “Survey of Caching Mechanisms in Information-Centric Networking,” IEEE Communication surveys & tutorials, Vol. 17,No. 3, Third Quarter 2015.

[50]. Athanasios at el ” Information centric network: Research challenges and opportunities”, pp 1-5,2015.

[51]. J. Wang, “A survey of Web caching schemes for the internet, ” SIGCOMM Comput, Commun. Rev., vol.29, no.5, pp.36-44, Oct. 1999

[52]. S. Podlipnig and L. B€osz€ormenyi, “A survey of Web cache replacement strategies,” Comput. Surveys, vol. 35, no. 4, pp. 374– 398, 2003.

[53]. Qingxia Chen, F. Richard Yu, Fellow, IEEE, Tao Huang, Renchao Xie, Jiang Liu and Yunjie Liu, “Joint Resource Allocation for Software-Defined Networking, Caching, and Computing”, 2018.

[54]. J. Karkazis et al “The Multi-Commodity Facilities Location Problem,”, vol. 32, No.9,pp.347-354, 1981

[55]. T. B. Boffey“Location problem arising in computer networks,”, vol. 40, No.4,pp.347-354, 1989