IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-4, Issue-9 E-ISSN: 2347-2693

# Video on demand for Peer to Peer network

Virangana Waghchaure[1*] and Viresh Chapte[2]

[1,2]Computer Engineering, DYPSOET, Pune, Maharashtra

**Available online at: www.ijcseonline.org**

Received: 22/Aug/2016 Revised: 02/Sept/2016 Accepted: 20/Sept/2016 Published: 30/Sep/2016

**Abstract—** In the proposed system a social-network-aided efficient P2P live streaming is performed, To prove that system outperforms other representative systems in terms of overhead, video streaming efficiency and server load reduction, and the effectiveness of the system two strategies. Reducing the cost of system in structure maintenance and node communication. The files can be moved from a centralized files to a decentralize server to distribute load and solve shutdown problems. Also cloud can be used for faster availability of files. Identity of peers in a private network remains hidden behind their global endpoint, P2P applications cannot run between two peers in separate private networks.

*Keywords—* ***Peer-to-peer (P2P) P2P networks, social networks, Distributed video-on-demand (VoD)***

## I. Introduction

In that we ensure a video download with peer to peer network approach including social network techniques. Propose a Video on demand approach for P2P. Once connected, a peer downloads bits of the files in small pieces, downloading all the data it can get. Once the peer has some data, it can then begin to upload that data to other peer in the connected network. In this way, everyone downloading a file is also uploading the same file. These speeds up everyone's download speed. If 10,000 people are downloading the same file, it doesn't put a lot of stress on a central server. Instead, each downloader contributes upload bandwidth to other downloaders, ensuring the file download stays fast.

## II. Motivation

Now a days download the data from internet is most common part .everyone can download data from internet. many user download the same video from internet or many user already have that video.at the same time network traffic creates well as server slow down speed. In previous system peer to peer network server does not store data or if any peers delete data that time loss the data. Direct download from server may affect download speed. Also more downloads may affect server speed. In order to avoid the glitches created from direct download, our approach will be effective.so i think to do better for data downloading.

## II. Literature Survey

They propose an AQCS, adaptive queue-based chunk scheduling that can support the maximum streaming rate allowed by a P2P streaming system with small signaling overhead and short startup delay. AQCS is a distributed algorithm with minimum requirement on peers. Queue-based design enables peers to be self-adaptive to the bandwidth variations and peer churn, and automatically converges to the optimal operating point. The prototype of AQCS is implemented and various implementation issues are examined. The experiments over the Planet Lab further demonstrate AQCS's optimality and its robustness against changing system/network environment [1].

The problem of broadcasting in a live streaming is an unstructured network. The broadcasting problem has been studied extensively for edge-capacitated networks. They study through simulation the delay that users must wait in order to playback a video stream with a small number of skipped packets, and discuss the suitability of algorithms for live video streaming [2]. In this paper they answer that question by showing that a carefully-designed mesh-based system can achieve close-to-optimal stream rates. Specifically, implement and evaluate a design based on a mesh-based algorithm called DP/LU. Contrary to tree-based designs, DP/LU uses an unstructured overlay, which is easier to construct and is highly resistant to churn. In addition, they introduce mechanisms for overlay rewiring and source scheduling that lead to significant performance improvements[3,4].The success of file swarming mechanisms such as Bit Torrent has motivated a new approach for scalable streaming of live content that we call mesh-based Peer-to-Peer (P2P) streaming. In this approach, participating end-systems (or peers) form a randomly connected mesh and incorporate swarming content delivery to stream live content. The main design goal of PRIME is to minimize two performance bottlenecks, namely bandwidth bottleneck and content bottleneck. This leads to effective utilization of available resources to accommodate scalability and also minimizes content bottleneck[5]. The success of swarming content delivery has motivated a new approach to live peer-to-Peer (P2P) streaming that call mesh-based streaming. In this approach, participating peers form a random mesh and incorporate swarming content delivery to stream live content. Despite the growing popularity of this approach, neither the design tradeoffs nor the basic performance bottlenecks in mesh-basedP2P streaming are well understood[6,7]. This paper presented in mesh network[8]. That paper represents one can build multi-tree systems with simple and scalable algorithms, and can still yield fast convergence and robustness. This paper presents Chunky spread, a multi-tree, heterogeneous P2P multicast algorithm based on an unstructured overlay. Through simulation, show that Chunky spread can control load to within a few percent of a heterogeneous target load, and how this can be traded off for improvements in latency and tit-for-tat incentives[9,10]. In this paper, they suggest a novel hybrid tree/mesh design that leverages both overlays. The key idea is to identify a set of stable nodes to construct a tree-based backbone, called tree bone, with most of the data being pushed over this backbone. The results demonstrate the superior efficiency and robustness of this hybrid solution[11].

In this paper, it support for video streaming a step further by presenting extensions to Bit Torrent for supporting live video streaming, which we have implemented in our Bit Torrent client called Tribler. tested that extensions both by running simulations and by deploying our implementation in a public trial in the Internet, using the optimal values of several parameters as found in the simulations[12,13]. The Bit Torrent protocol is by far the most popular protocol for offline peer-to-peer video distribution on the Internet. Bit Torrent has previously been extended to support the streaming of recorded video, that is, Video-on-Demand (VoD)[14].

In this paper, theoretically study the impact of this inherent delay constraint and derive the minimum delay bounds for P2P live streaming systems. They show that the bandwidth heterogeneity among peers can be exploited to significantly improve the delay performance of all peers. They further propose a conceptual snowball streaming algorithm to approach the minimum delay bound in a dynamic P2P networking environment. Propose system analysis and simulation suggest that the proposed algorithm has better delay performance and more robust than static balanced multi-tree-based streaming solutions. Insights brought forth by study can be used to guide the design of new P2P systems with shorter streaming delays.[15] In this paper, they performed a detailed analysis on 400 GB and 7 months of run-time traces from UUSee, a commercial P2P streaming system, and observed that available capacities on streaming servers are not able to keep up with the increasing demand imposed by hundreds of channels. they propose a novel online server capacity provisioning algorithm that proactively adjusts the server capacities available to each of the concurrent channels, such that the supply of server bandwidth in each channel dynamically adapts to the forecasted demand, taking into account the number of peers, the streaming quality, and the priorities of channels. The algorithm is able to learn over time, and has full ISP awareness to maximally constrain P2P traffic within ISP boundaries. To evaluate the effectiveness of our solution, our experimental studies are based on an implementation of the algorithm with actual channels of P2P streaming traffic, with real-world traces replayed within a server cluster.[16,17,18] In that paper, they propose a new protocol for multi-view P2P streaming, called divide-and-conquer (DAC), which efficiently solves the inter-channel bandwidth competition problem using a divide-and conquer strategy at the channel level, and thus is flexible to work with various streaming protocols. This makes DAC more suitable for upgrading current single-view P2P live streaming systems to multi-view P2P live streaming systems. Our extensive packet level simulations show that DAC is efficient in allocating the overall system bandwidth among competing channels, is flexible in working with various streaming protocols, and is scalable in supporting a large number of users and channels.[19,20,21]

paper describes a framework for future IPTV systems where millions of viewers can select multiple channels, view them in a customized way, and cooperate with each other in a peer-to-peer fashion for TV program deliver.[22,23,24] they propose a novel peer selection algorithm, called Channel-Aware Peer Selection (CAPS), where a peer selects its neighboring peers based on the channel subscription of the system, in order to efficiently utilize the bandwidth of all peers in the system, especially those peers watching multiple channels. The results of a large-scale simulation with 10,000 peers and 4 channels shows that CAPS can significantly improve the system performance over the straightforward Random Peer Selection (RPS), which is widely, used in single-view P2PMMS networks [25]They develop infinite-server queuing network models to analytically study the performance of multichannel P2P live video systems. models capture essential aspects of multichannel video systems, including peer channel switching, peer churn, peer bandwidth heterogeneity, and Zip file-like channel popularity. they apply the queuing network models to two P2P streaming designs: the isolated channel design (ISO) and the View-Upload Decoupling (VUD) design. For both of these designs, we develop efficient algorithms to calculate critical performance measures, develop an asymptotic theory to provide closed-form results when the number of peers approaches infinity, and derive near-optimal provisioning rules for assigning peers to groups in VUD. It uses the analytical results to compare VUD with ISO. they show that VUD design generally performs significantly better, particularly for systems with heterogeneous channel popularities and streaming rates.[26]

## III.Problem Statement

Client server create bottle neck for bandwidth of server multiple download. Resulting in low download speed to every client. Also this may lead to server crash. The transfer speed is affected by a number of variables, including the type of protocol, the amount of traffic on the server and the number of other computers that are downloading the file. If the file is both large and popular, the demands on the server are great, and the download will be slow.

## IV. Propose System

This paper are summarized as follows dissertation work falls into the category known as Peer to peer networking. Peer-to-peer (P2P) computing or networking is a distributed application architecture that partitions tasks or workloads Between peers

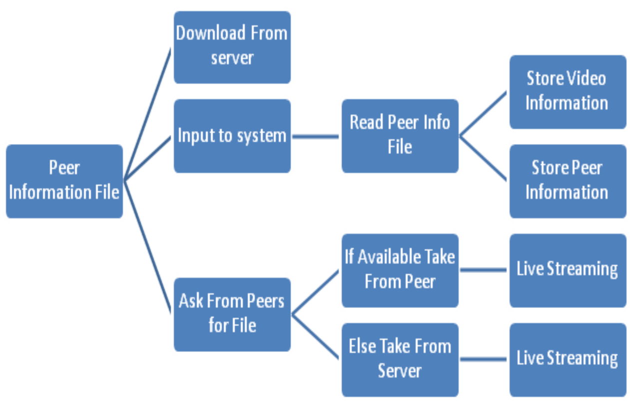



Fig Architecture diagram for video on demand for peer to peer network

The user will select the video that he/she wants to download. For e.g. that video size is 3.37 MB. The user will get a Meta file which will contains the information about the video. Meta file contains all the information of the file including its size and the other users that are downloading the video. User will automatically get joined to the network. If there is no peer for downloading the video then user will get joined directly to the server[27,28].

After downloading the metafile server will divide the video into chunks. Large files are broken into pieces of size between 64 KB and 1 MB. We consider 3.37 MB video is divided into 13 parts. If 6 peers are available in network that time peer5 want to download one 3.37 MB video that time peer2 send some available parts(i.e.3 parts) other side peer 1 also send some available parts(i.e.2 parts) to peer 5& peer 3will send other parts(i.e. 4 parts) and remaining parts(i.e.4 parts) will be send server that parts does not available on peer1…peer6 . If any user uploading content gets disconnected directly from network then the user downloading that part gets directly connected to server in order to avoid time delay. After download is complete, it will merge all the video parts.

1. Connect To server: Peer connects to server when the application executes. Connection takes places by client sending its Ip address and the information of video parts it has.
2. Peers Information. Server stores basic information of all peers including its Ip address and the last active time. When peer interested in live streaming connects, the server provides all the information to the peer[29,30].
3. Connection with other peers: After getting all the information about other peers, server tries to connect with other peers. But sometime it is impossible to connect to a particular peer due to Network address translation systems.
4. 3 way handshake: A peer node sends a SYN data packet over an IP network to a other peer on the same or an external network. The objective of this packet is to ask/infer if the other peer is open for new connection.
   - The target peer must have open ports that can accept and initiate new connections. When the server receives the SYN packet from the client node, it responds and returns a confirmation receipt - the ACK packet or SYN/ACK packet.
   - The client node receives the SYN/ACK from the target peer and responds with an ACK packet.

5. data upload & download: After connection establishment, peer starts asking data. This leads to exchanging of video chunks among peers.

6. Live steaming: User can anytime starts live streaming of videos. Since live streaming runs on different thread & takes data from above module.

### A Initialization.

Consider a set of $S$. Is An multimedia
Let S be server.{S1}
Let P is a number of peers {p1,p2,p3……$p_n$ }
$U_s$ is a video uploaded speed
$D_s$ is a video Downloaded speed
$T_p$ is a Total number of Peers connected
$E_t$ is a Estimated time for Downloading video
V is a Video.

### B Distributed hash table algorithm

A distributed hash table stores key-value pairs by assigning keys to different computers (known as "nodes"); a node will store the values for all the keys for which it is responsible. find. Successor (k) the basic approach is to pass the query to a node's successor, if it cannot find the key locally. This will lead to a O (N) query time. To avoid the linear search, Chord implements a faster search method by requiring each node to keep a finger table containing up to m.the Successor ((n+$2^{i-}$

[1])mod2[m]). Whenever a new node joins, three invariants should be maintained (the first two ensure correctness and the last one keeps querying fast). To ensure correct lookups, all successor pointers must be up to date. Stabilization protocol running periodically in the background [31,32].

### C Routing algorithm

Routing algorithm is most important becase Choosing the next neighbor to request content from is an important part of freenet's behavior, as it is this that it allows it to adapt to usage patterns and network changes .

1. Node n, Table t
2. For each n If( request = = t.value)

   {

   Return n.key

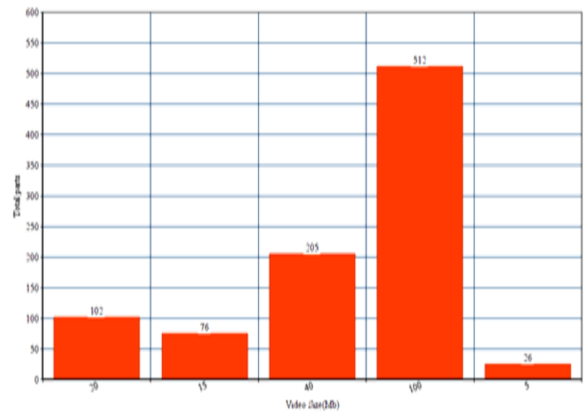
   } //It is important to note that it is the keys which are being compared for closeness, not the files, so there is no semantic similarity implied.
3. File f=get File(n.key)// When the network is set up there will be no performance records for any nodes, so requests are sent to node which is essentially chosen at random.

   // If it successfully returns the requested file, then an entry will be created for that node with the relevant key. Requests for similar keys will then be routed towards the successful node, and as a result it will eventually start to specialize in these keys.

### V Experimental result

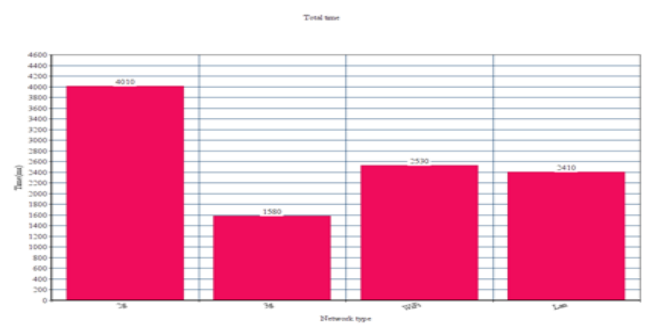
1.Total parts of video Available on Different Peers or one Peer

| Sr.No. | Video size | Total size |
|---|---|---|
| 1 | 20MB | 102 |
| 2 | 15 MB | 76 |
| 3 | 40 MB | 205 |
| 4 | 100 MB | 512 |
| 5 | 5 MB | 26 |




2.Total time required to initialize connections for download 3.37 MB video

| Sr.No | Network type | bandwidth | Time(ms) |
|---|---|---|---|
| 1 | 2G | 25kbps | 4010 |
| 2 | 3G | 1024 kBps | 1580 |
| 3 | WI_FI | 256kBps | 2530 |
| 4 | Lan | 128 Mbps | 2410 |




3.Time required to download and upload parts for live streaming(Consider parts of 200 Kb each)

| Sr. No. | Network type | bandwidth | Download time(ms) | Upload time(ms) |
|---|---|---|---|---|
| 1 | 2G | 25kbps | 28.57 | 66.66 |
| 2 | 3G | 1024kBps | 3.63 | 8.05 |
| 3 | WI-FI | 256kBps | 1.85 | 2.06 |
| 4 | Lan | 128kBps | 1.44 | 2.04 |

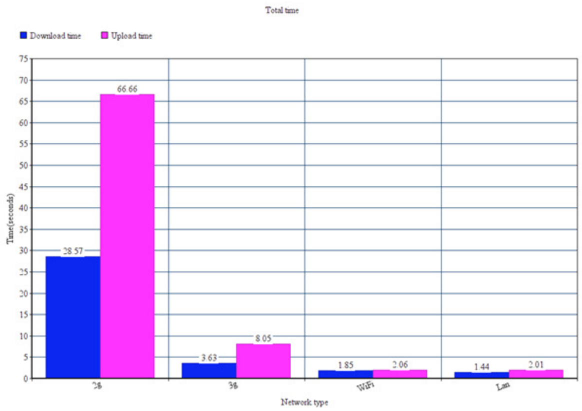



Figure: Uploading and downloading Time

## VI Conclusion

In this paper, propose peer to peer live streaming system. It is social network aided video on demand for peer to peer network and it used for downloads the media from more than one peers as well as server.it reduce the server load and increases the download speed. Also reduce the establishment cost. Distributed hash table and routing algorithm is used. In Our approach help to manage user preferences.in our work more accuracy in Delivering the media. Also costs can be reduced In future a cloud can be used for faster availability

## REFERENCES

# References


[1] F. Picconi and L. Massoulie, "Is there a future for mesh-based live video streaming?," in Proc. P2P, 2008, pp. 289-298.

[2] N. Magharei and R. Rejaie, "PRIME: Peer-to-peer receiver-driven mesh-based streaming," in Proc. IEEE INFOCOM, 2007, pp. 1415-1423.

[3] T. Moscibroda ; Oregon Univ., Eugene ; R. Rejaie, "PRIME: Peer-to-Peer Receiver-drIven MEsh-Based Streaming", IEEE INFOCOM 2007 - 26th IEEE .

[4] J. Venkataraman and P. Francis, "Chunkyspread: Multi-tree unstructured peer-to-peer multicast," in Proc. IPTPS, 2006, pp. 1-10.

[5] F. Wang, Y. Xiong, and J. Liu, "mTreebone: A hybrid tree/mesh overlay for application layer live video multicast," in Proc. ICDCS, 2007, p. 49.

[6] J. Mol, A. Bakker, J. Pouwelse, D. Epema, and H. Sips, "The design and deployment of a bittorrent live video streaming solution," in Proc. ISM, 2009, pp. 342-349.

[7] Y. Liu, "Delay bounds of chunk-based peer-to-peer video streaming," IEEE/ACM Trans. Netw., vol. 18, no. 4, pp. 1195-1206, Aug. 2010.

[8] Shivlal Mewada, Umesh Kumar Singh and Pradeep Kumar Sharma, "A Novel Security Based Model for Wireless Mesh Networks", International Journal of Scientific Research in Network Security and Communication, Volume-01, Issue-01, pp (11-15), Mar -Apr 2013.

[9] Ramesh Kagalkar and Dr. Nagaraja H.N, "New Methodology for Translation of Static Sign Symbol to Words in Kannada Language"," International Journal of Computer Applications(IJCA), Volume 121, No.20,PageNo. 25-30, July-2015.

[10] Ramesh M. Kagalkar, Dr. Nagaraja H.N and Dr. S.V Gumaste",A Novel Technical Approach for Implementing Static Hand Gesture Recognition", International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), Volume 4, Issue 7, July 2015.

[11] Ramesh M. Kagalkar and Dr. S.V Gumaste, "Automatic Graph Based Clustering for Image Searching and Retrieval from Database", CiiT International Journal of Software Engineering and Technology, Volume 8, No 2, 2016.

[12] Ramesh M. Kagalkar and Dr. S.V Gumaste, "Review Paper: Detail Study for Sign Language Recognition Techniques", CiiT International Journal of Digital Image Processing, Volume 8, No 3, 2016.

[13] Rashmi B. Hiremath and Ramesh Kagalkar, "A Methodology for Sign Language Video Analysis and Translation into Text in Hindi Language",CiiT International Journal of Fuzzy System, Volume 8, No 5, 2016.

[14] Rashmi B. Hiremath and Ramesh Kagalkar, "Methodology for Sign Language Video Interpretation in Hindi Text Language", International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), Volume. 4, Issue 5, May 2016.

[15] Amitkumar and Ramesh Kagalkar, "Sign Language Recognition for Deaf User", Internal Journal for Research in Applied Science and Engineering Technology (IJRASET), Volume 2, Issue 12, December 2014.

[16] Amit kumar and Ramesh Kagalkar, "Advanced Marathi Sign Language Recognition using Computer Vision", International Journal of Computer Applications (IJCA), Volume 118, No. 13, May 2015.

[17] Amit kumar and Ramesh Kagalkar, "Methodology for Translation of Sign Language into Textual Version in Marathi", CiiT, International Journal of Digital Image Processing, Volume 07, No.08, Aug 2015.

[18] Mrunmayee and Ramesh Kagalkar, "A Review On Conversion of Image To Text as Well as Speech using Edge Detection and Image Segmentation", International Journal of Science and Research (IJSR), Volume 3, Issue 11, November 2014.

[19] Mrunmayee Patil and Ramesh Kagalkar, "An Automatic Approach for Translating Simple Images into Text Descriptions and Speech for Visually Impaired People", International Journal of Computer Applications (IJCA), Volume 118, No. 3, May 2015.

[20] Kaveri Kamble and Ramesh Kagalkar, "A Review: Translation of Text to Speech Conversion for Hindi Language", International Journal of Science and Research (IJSR), Volume 3, Issue 11, November 2014.

[21] Kaveri Kamble and Ramesh Kagalkar, "Audio Visual Speech Synthesis and Speech Recognition for Hindi Language" ,International Journal of Computer Science and Information Technologies (IJCSIT), CiiT International Journal of Data Mining Knowledge Engineering, Volume 6, Issue 2, April 2015.

[22] Kaveri Kamble and Ramesh Kagalkar " A Novel Approach for Hindi Text Description to Speech and Expressive Speech Synthesis" International Journal of Applied Information Systems (IJAIS), Volume 8, No.7, May 2015.

[23] Shivaji Chaudhari and Ramesh Kagalkar "A Review of

Automatic Speaker Recognition and Identifying Speaker Emotion Using Voice Signal", International Journal of Science and Research (IJSR), Volume 3, Issue 11, November 2014.

[24] Shivaji Chaudhari and Ramesh Kagalkar, "Automatic Speaker Age Estimation and Gender Dependent Emotion Recognition", International Journal of Computer Applications (IJCA), Volume 117, No. 17, May 2015.

[25] Shivaji J. Chaudhari and Ramesh M Kagalkar, "A Methodology for Efficient Gender Dependent Speaker Age and Emotion Identification System", International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), Volume 4, Issue 7, July 2015.

[26] Ajay R. Kadam and Ramesh Kagalkar, "Audio Scenarios Detection Technique", International Journal of Computer Applications (IJCA), Volume 120, No. 16, June 2015.

[27] Ajay R. Kadam and Ramesh Kagalkar, "Predictive Sound Recognition System", International Journal of Advance Research in Computer Science and Management Studies (IJARCSMS), Volume 2, Issue 11.

[28] Swati Sargule and Ramesh M Kagalkar, "Hindi Language Document Summarization using Context Based Indexing Model", CiiT International Journal of Data Mining Knowledge Engineering, Volume 08, No. 01, Jan 2016.

[29] Swati Sargule and Ramesh M Kagalkar, "Methodology of Context Centered Term Indexing Style Intended For Hindi Language Document Summarization", CiiT International Journal of Software Engineering, Volume 8, No 5, 2016.

[30] Rachana Palaskar, Shweta Pandey, Ashwini Telang, Akshada Wagh and Ramesh Kagalkar," An Automatic Monitoring and Swing the Baby Cradle for Infant Care", International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), Volume 4, Issue 12, December 2015.

[31] Ramesh M Kagalkar, Kajal Chavan, Asmita Jadhav, Ravina Patil and Asmita Rawool, "Self-Educating Tool Kit for Kids" CiiT Software Engineering and Technology, Volume 8, No 1, 2016.

[32] Vandana D. Edke and Ramesh M. Kagalkar, "Video Object Description of Short Videos in Hindi Text Language", International Journal of Computational Intelligence Research, Volume 12, Number 2 (2016), pp. 103-116 © Research India Publications.

**Authors Profile**

**Virangana Waghchaure** is M.E 2nd year student of Computer Engg. Department, Dr. D. Y. Patil School of Engg. And Technology, Lohegaon, Pune. And her research interest includes computer engineering & networking.
E-mail-viranganawaghchaure@gmail.com

**Viresh Chapte:** presently working as an Assistant. Professor, Department of Computer Engineering, Dr. D. Y. Patil School Of Engineering and Technology, Charoli, B.K.Via –Lohegaon, Pune, Maharashtra, India. He has 8.5 years of teaching experience at various institutions.
E-mail-Viresh .Chapte@dypic.in