

# Analysis of Fault Tolerance in Peer to Peer Video on Demand System Using V Chaining

**Hareesh K.**

Research scholar  
Jawaharlal Nehru Technological University,  
Anantapur, A.P India  
mail\_hareeshk@yahoo.com

**Manjaiah D.H.**

Professor & Chairman  
Department of Computer Sciences,  
Mangalore University, Mangalore  
yilm321@yahoo.co.in

**Abstract** - The video on demand system is one of the streaming applications widely access remote video programs over the Internet. One of the variant of VoD system is peer to peer (p2p) system. In P2P VoD system peers are frequently fails while chaining. To overcome the failure of peers in chaining fault tolerance mechanism is used. We have proposed VChaining mechanism on Continuous Time Markov Chain model. A birth death process is used to model our proposed mechanism. The parameters used in our model as arrival of requests of peers versus failure of peers, normal versus recovery, the average load on the system, the peer band width and buffer in the system, the server bandwidth and server load. We have simulated above parameters using Video Chaining (V Chaining) mechanism. We have compared simulation results with existing mechanism such as optimal, accelerated chaining mechanism. Our simulation results fairer among all the chaining mechanisms.

**Keywords** - Vod, Peer, Chaining, Streams, Segments, Peer-to-Peer.

## I. INTRODUCTION

The Video on Demand (VoD) systems are model on the client server architecture. Traditional client and or server architectures for VoD services cannot provide video streams to a large number of concurrent users. Peer-to-Peer (P2P) technology has been incorporated into VoD applications by utilizing the uplink bandwidths of the peers [1-5] which is overcome the server upload burden. The VoD system is one of the popular applications in the Internet .Users can download the movies from the multimedia servers, user can watch live streaming videos [6-9] also share their personal videos with each other. The multimedia server stores the video titles, popularity, index, size, buffer and bandwidth for the videos. When the user requests for the video title to the multimedia server, if the requested video title found in the server then video stream is downloaded to the client for playback. As the number of user increases system load significantly increases due to this there is rapid consumption of bandwidth. The design of the VoD system requires high data rates to serve the large number of concurrent clients. So, the design of such VoD systems faces significant challenges. To further balance the load on the server parallel and distributed server architectures are used. Each these servers serve only few numbers of users rather than serving all the users. The drawbacks of these servers are it requires high end hardware with large amount of bandwidth and buffer, so that the costs of upgrading these servers are high.

To further decrease the load on these servers, Hua *et al.* proposed in [10] a client side caching stream is called as

earthworm. The interval caching scheme is similar to buffering scheme, here each client not only playback the received video streams but also forwards the streams to other clients with adequate buffering delay. This scheme is further extended as forward, backward, adaptive and optimal chaining, the drawback of these schemes which exploits client resources such as buffer and uplink bandwidth. However, demand for the quality videos and longer duration videos are expected in the near future [11]. For such applications the existing chaining schemes fails in meeting the reliability requirements due to this tremendous transfer of same data occurs in the clients which will degrade overall performance of the system. In chaining based systems the failure of any peers may corrupts a stream of video chain due to several reasons such as stop sending streams from the neighboring peers (selfish peers), due to hardware or software failure etc., To address this problem we have present the some of the fault detection scenarios with recovery system process. In video chaining process is a client-server model, a peer can act as miniature as well as client or vice versa. Once a peer fails, its downstream peer will identify this after receives the video streams from this peer after certain timeout. The fault will identify by the peers play out of video streams of the requested movie through VCR operations. In the e buffered video streams/segments, some of the segments are missing because these segments are not receive from its immediate upstream peer. Even we can find the missing segments from the segment offset of the segments of the entire video.

To recover the fault, the downstream peer send fault peer to nearby proxy server as well as all the peers which are chained in the system. On the other hand upstream peer will send the FAULT control message will broadcast the entire chained peer as well as nearby proxy server. Then the proxy server sends the SELECT message may contain new possible optimal paths [12] for chaining, longest alive time of the peers, fault peer etc., Then the peer will receive SELECT message after certain timeout and respond back to the proxy server. Now the proxy server sends the new optimal paths, then the peer may select best optimal path and starts streaming from the next neighboring peer otherwise it streams from the proxy server.

In our approach, we build a effective VoD system. Unlike a client server VoD system, videos streams are distributed into proxy servers, while this video streams are placed in the different proxy servers, eliminates the different storage overheads like replications in current Peer-to-Peer system. A Peer-to-Peer system cannot

function if any one of its peer fails. To solve this problem we have developed a fault tolerance in VoD system using a Markov Chain. However, building a reliable VoD system is a challenging task; we investigated the streaming, i.e., scheduling the video streams among the source peer from the requested peer is based on the optimal paths. The selection of the nearer peers is based on optimal paths [9]. While chaining among the peers we observed frequent break in the chain. To overcome these stream failures among the peers, In this paper we have analyzed the fault tolerance in peer to peer VoD system using V Chaining mechanism is model on Continuous Time Markov Chain (CTMC) model.. The rest of the paper is organized as follows: Section II reviews some previous related works; Section III presents an overview of the VoD architecture; Section IV presents Chaining mechanism; Section V presents V-Chaining algorithm; Section VI presents the fault tolerance Section VII presents analysis of fault tolerance in p2p VoD system Section VIII presents Simulation ad Results; section IX concludes the paper.

## II. RELATED WORK

In this section present the brief discussions of the previous research works in the various chaining approaches of video on demand system.

### A. Standard Chaining

In standard chaining [13] chaining between each peer happens that, the first peer in the chain receive the video streams from the server and subsequent peers in the chain receive their video stream from their neighboring peers, which are in the chaining. As a result of this, we have observed some of the disadvantages that, the video streams are pipelined through peers belonging to the same chain. Here a new chain has to be restarted every time the inter arrival between the two successive peers requests exceed the buffer capacity of the previous peer buffer. Since, the chaining among the peers requires small buffers in the peers. In our scheme video streams are not pipelined through the peers belonging to same chain. A new chain has not restarted every time the inter arrival between the two successive peers exceed the buffer capacity of the previous peer buffer. The user can get the video stream from the neighboring peers without any duplication.

### B. Advanced Chaining

The main disadvantages of advanced chaining [14] in its poor performance observed in the inter arrival rates of the requests of peer. More specifically, whenever the time interval between two consecutive requests among the peers exceed time of the video time in minutes. In this chaining we have studied that, every inter arrival of requests exceed the duration of video, to overcome the above problem by inserting every time in minutes the idle peers, and these idle peers will delay the video stream among the peers. In our proposed mechanism, we have not inserting idle peers, the differences of inter arrival of requests will not exceed the duration of video, and in turn our mechanism will not delay the video streams once each peer upload the video stream to the requested peer our

proposed chaining algorithm checks certain conditions discussed in section 4. If the condition is satisfied the peer will starts streaming the video to the requested peer.

### C. Optimal Chaining

This chaining [15] [16] also managing all the peers buffer's as a single shared resource. As we have noticed as a result of this, peers can borrow the buffers from the neighboring peers in order to bridge the gaps between the incoming requests of several peers. In this mechanism also introduced the streaming proxy servers in order to increase the chaining responses and resilience. In optimal chaining approach, if a failure occurs on a chain single shared resource causes entire system will fail instead in our scheme we use multi shared resources.

### D. Expanded chaining

Expanded chaining is also known as cooperative video distribution protocol [17] it also improve the chaining by taking the advantage of large buffer size of all the peers in the chaining. In this mechanism the main drawback that, the peer will disconnect also stop forwarding the video stream once they have finished the playback of the video. But in standard chaining, each peer forwards the video stream to the next immediate peers starts sending the beginning of the video streams, when a peer has finished playing the video the chain will disconnect and also stop transmitting the video stream. But the streaming server will transmit the remaining parts of the video. In our scheme we are not using the larger buffer size while involved chaining among all the peers, instead we have used the peer buffer based on the streaming time line of the video.

### E. Accelerated Chaining

In case of standard and advanced chaining completely eliminate the server workload that the peer will always keep forwarding the video stream the next neighboring peers in the chain once they have finished playing the video. In above mechanism assumption is not reasonable because the most of the peers are disconnect once they played the video and most of the significant number of peers will also disconnect without playing the full length of video. In accelerated chaining [18] peers will forwards the video stream to the neighboring peers in the chain at slightly higher rate compared to other chaining mechanisms and on the video consumption rate. In accelerated chaining higher rate forwarding the video stream overflows the buffer due to this, we have forwarding the video stream based on the availability of bandwidth and playback rate.

In the above chaining approaches in the VoD system the chaining among the peers may fail due to stringent bandwidth and buffer, hardware, software and peer failure. To overcome these problems we have analyzed the fault tolerance in peer to peer video on demand system is model on continuous time markov chain model.

## III. VOD ARCHITECTURE

The peer to peer video on demand (P2P VoD) architecture shown in Figure 1 consists of a Media server, proxy servers and peers in a cluster. The multimedia

server contains movie database in which there will be a collection of video data files. The video data file's information such as index popularity, minimum bandwidth requirement and minimum buffer capabilities are also stored in the Media server. Proxy server in this architecture is used to cache video data files for the nearer peers in a cluster. The purpose of proxy server is to reduce the load on the multimedia server by caching the video data files. A cluster is a logical connectivity of peers which is headed by a proxy server. A peer can be a seed peer or a non-seed peer. A seed peer is a client, which has sufficient bandwidth and buffer capacity as well as it can store and forward video data files to other peers. A non seed peer is a client, which has only required configuration and it can only playback the received video data file and cannot store or forward the video data files to other peers

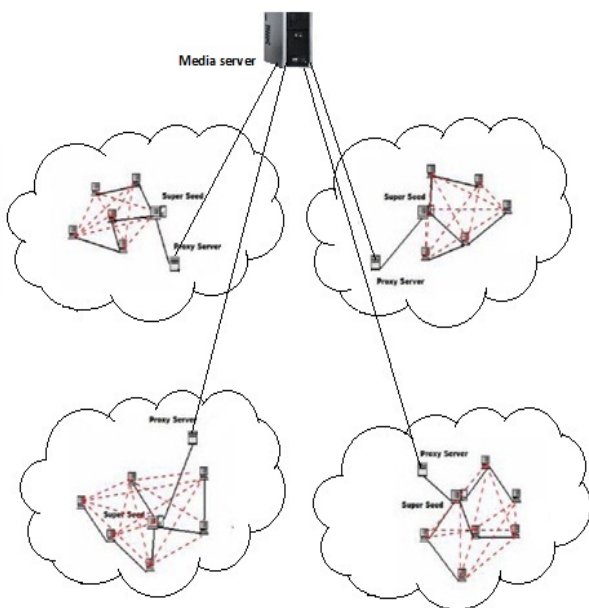


Fig.1. Peer to Peer VoD Architecture

#### A. Working Principle

Initially, suppose if a peer makes a request to the media server, and then the server downloads the entire video data files to the nearest proxy server of the requesting peer. In the *first* case, it is assumed that none of the peers are requested for the same movie. Thereby, after downloading the movie to the proxy server, the proxy server will transmit the movie to the requesting peer. Subsequently, if another peer from the same cluster makes a request for the same movie to the media server. Then the media server looks in to its current streaming movies database for the nearest proxy server and its availability of the movies. If such entry is found then media server redirects the requesting peer to the nearest proxy server. Again the proxy server applies the same procedure to find out that any of the peers in the cluster has the movie in its buffer. If such peer is found, then the requesting peer will be redirected to that peer which has the same movie. Then on the transmission occurs from that peer to the requesting peer. The transmission of video data file from that of peer to another peer is called *chaining*. However, if entry is not

found in the proxy server, then the proxy server starts transmitting the video data files to the requesting peer. Elsewhere, if the entry is not found in the media server then the procedure is followed as if it is a first request from the cluster.

Suppose, if another request from different cluster occurs for the same movie to the media server. Then the media server redirects to the nearest proxy server of the requesting peer to transmit the video data file to the proxy server of the requesting peer. Now instead of downloading the video data file from media server to the nearest proxy server. The download happens from another proxy server which has the movie to the nearest proxy server of the requesting peer. Therefore, the same procedure is carried out for the transmission of movies among the clusters. If none of the proxy servers has the same movie then the media server downloads to the nearest proxy server and then the proxy server transmits the movie to the requesting peer.

#### IV. CHAINING MECHANISM

Now we proposed a mechanism for video transmission between the peers, which is known as V-Chaining mechanism. Here the time line of streaming session is taken as a reference to chain the peers. The time line is scale that units the inter arrival time between the peer. Let us consider the duration of movie D minutes. Then the factor in the timeline starts from 0 to D minutes. Let us assume that the chain has to be established between two peers that is P<sub>1</sub> and P<sub>2</sub>. Let the inter arrival time be AT<sub>12</sub> between the peers P<sub>1</sub> and P<sub>2</sub>, where AT<sub>12</sub> is the difference between the requests of peers P<sub>1</sub> and P<sub>2</sub>. Let B<sub>12</sub> be the bandwidth capacity between P<sub>1</sub> and P<sub>2</sub>. Let b<sub>1</sub> and b<sub>2</sub> be the buffer capacity of P<sub>1</sub> and P<sub>2</sub>. Now the downloading time of P<sub>1</sub> from source is calculated based on the size of the Movie S and its reception rate R<sub>1</sub>. The first condition is checked for chaining and is given as

$$AT_{12} < \frac{S}{R_1} \text{ ----- (1)}$$

If the condition in equation 1 satisfies, then a chain is established between P<sub>1</sub> and P<sub>2</sub>. If the condition fails, then total playback rate Pr<sub>1</sub> of P<sub>1</sub> is calculated based on the video consumption rate. Now, the inter arrival time AT<sub>12</sub> and the total playback rate of P<sub>1</sub> is checked as second condition and is given as

$$AT_{12} < \frac{S}{Pr_1} \text{ ----- (2)}$$

If the condition in equation 2 satisfies, then a chain is established between P<sub>1</sub> and P<sub>2</sub>. If this condition fails, contains of buffer b<sub>1</sub> of P<sub>1</sub> is observed within streaming timeline. Now, the observation is estimated based on the transmission rate r<sub>1</sub> of P<sub>1</sub> from source and the buffer b<sub>1</sub> contains within the streaming timeline of the movie's duration D. The inter-arrival time AT<sub>12</sub> and the buffer b<sub>1</sub> contain is checked as third condition and is given as

$$AT_{12} < \frac{b_1}{r_1} \text{ ----- (3)}$$

If the condition in equation 3 satisfies, then a chain is established between P<sub>1</sub> and P<sub>2</sub>. If none of the above

condition is satisfied, then the video is obtained directly either from proxy server or multimedia server to peer P<sub>2</sub>. The above scenario of chaining between P<sub>1</sub> and P<sub>2</sub> can be generalized for any chaining between P<sub>1</sub> and P<sub>2</sub>. Therefore; the generalized equations are derived as given in the below equations.

$$AT_{ij} < \frac{S}{R_i} \text{-----(4)}$$

$$AT_{ij} < \frac{S}{Pr_i} \text{-----(5)}$$

$$AT_{ij} < \frac{b_i}{r_i} \text{-----(6)}$$

The generalized equations will be applied in the V-Chaining algorithm for j<sup>th</sup> peer to establish a chain from i<sup>th</sup> peer.

### V. V-CHAINING ALGORITHM

#### Nomenclature

AT<sub>ij</sub>:: Inter arrival time between i<sup>th</sup> and j<sup>th</sup> peer , S:: size of the movie, R :: reception rate, P<sub>r</sub>:: playback rate, b::

buffer of the peer, r:: transmission rate of the movie

Step 1: Request arrival of j<sup>th</sup> peer

Step 2: Redirection from Server to i<sup>th</sup> peer

Step 3: Calculate AT<sub>ij</sub>=Arrival of j<sup>th</sup> peer - arrival of i<sup>th</sup> peer

Step 4: If  $AT_{ij} < \frac{S}{R_i}$  then establish chain between i<sup>th</sup> and j<sup>th</sup> peer

Step 5: else if  $AT_{ij} < \frac{S}{Pr_i}$  then establish chain between i<sup>th</sup> and j<sup>th</sup> peer

Step 6: else if  $AT_{ij} < \frac{b_i}{r_i}$  then establish chain between i<sup>th</sup> and j<sup>th</sup> peer

Step 7: peer P<sub>i</sub> starts chaining with peer P<sub>j</sub>.

Step 8: If peer P<sub>k</sub> is fails, then the peer P<sub>j</sub> FAULT message send to proxy server as well as all the peers involved in chaining.

Step 9: proxy server reply with SELECT message contains new optimal paths alive time of peer, fault peers list etc.,

Step 10: Peer P<sub>i</sub> selects optimal path [12] from SELECT message.

Step 11: Peer P<sub>i</sub> starts streaming from next immediate peer P<sub>j</sub>

Step 12: If peer P<sub>i</sub> is fails, then repeats the step 8 to 11

Step 13: else receive from server or from the proxy

#### A. Performance Analysis

In order to evaluate the performance of V-Chaining algorithm, we have considered bandwidth and buffer are the two very important parameters for analysis of the algorithm. The bandwidth utilization is measured at the server and as well as at the peer ends. The load on the server is measured in terms of bandwidth utilization at the server. The load at the server shall gradually decrease as the number of peers establishes the chains. Meanwhile, the load on the peer are collectively measured based on the bandwidth utilization at the peers. Since, the arrival of the peers follows a Poisson's process. Therefore, in our model, the probability density of peer's arrival is given as

$$P(t) = \lambda e^{-\lambda t} \text{-----(7)}$$

Where at time t on the streaming time line an average of λ peers arrive. The average bandwidth utilization at the server end is measured as

$$B_{Avg}^S = \int_0^T D \lambda e^{-\lambda t} = D \lambda e^{-\lambda t} \text{-----(8)}$$

Therefore, the server load is calculated for an average of λ peers and is given as

$$S_L = \lambda B_{Avg}^S \text{-----(9)}$$

The average bandwidth utilization at the peers are measured as

$$B_{Avg}^P = \int_0^T \frac{S \cdot D}{r \cdot \lambda} e^{-\lambda t} = \frac{S \cdot D}{r \cdot \lambda} e^{-\lambda t} \text{----- (10)}$$

Therefore, the collective load on peers is calculated for an average of λ peers is given as

$$P_L = \lambda B_{Avg}^P \text{-----(11)}$$

The other important parameter of buffer is measured in terms of effective utilization of buffers at the peers. For each peer buffer utilization measured in terms of video contains in its storage for the duration of the movie D and the video consumption rate that is playback rate. Therefore, the average buffer utilization of buffer in a peer is given by

$$b_{Avg}^P = \int_0^D \left( \frac{S}{r} - P_r \right) dt \text{-----(12)}$$

The average buffer utilization of buffers in all the peers is given by

$$b_{Avg}^{Ps} = \lambda b_{Avg}^P \text{-----(13)}$$

### VI. FAULT TOLERANCE

The fault tolerance in P2P Video on demand system using VChaining mechanism is analyzed based on the Continuous Time Markov Chain model is used for chaining among the peers. We use a simple two state Markov model for chaining among the peers. Each peer is allowable to be ACTIVE or INACTIVE State. ACTIVE state means that, the peer is starts streaming and INACTIVE state means that, when the peer is fails to stream due to the stop fail, hardware and software fail or Interactive controls. The two state Markov model is shown in Figure 2.

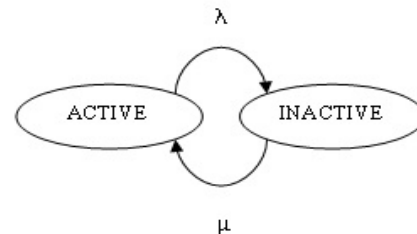


Fig.2. Two state Markov Model

The amount of time spends by the peers in the chaining is exponentially distributed with the parameter λ and μ for ACTIVE and INACTIVE state respectively. Consider a number of movies M1, M2, M3.....M<sub>n</sub> with varying segment sizes, streaming in N different peers. Each movie is fragmented into non overlap S segments, such that the size of the movie M= S si=i1. Each Peer involved in chaining is chances of break the chain due to various reasons such as stop fail, hardware and software fail or

Interactive controls. This leads to the unreliable streaming of segments to the peer. Initial segments of the movies are always streamed from the proxy server. Later parts of the movies are streamed from the nearer peer using chaining mechanism. Each peer has sufficient initial buffer to store the movies for chaining and as well as for playback rate. The mean joining rate of the peer in the chain is  $\lambda_j$  for the  $j$ th movie is Poisson distributed and the mean break of the chain in the system is  $\mu_j$  for the  $j$ th movie. The inter joining time of the chain is exponentially distributed with mean is  $1/\lambda_j$ . The total number of segments arrived to a peer is  $S_j g$  where  $g$  is the number of segments for the  $j$ th movie. The behavior of this system is modeled as Continuous Time Markov Chain (CTMC) known as Birth-Death process.

A birth-death process is used to model our P2P VoD system in which peers arrive one at a time and not as a group. The state of the system can be represented by the number of peer's  $k$  in the system. Arrival of new peers changes the state to  $k+1$ , this process is called as Birth. Similarly failure of a peer changes the system state  $k-1$ , this process is called as death. The number of peers in a system can therefore model as birth-death process.

Let us derive an equation for state probability  $P_n$  for a general birth-death process of our P2P VoD system. The analysis is based on Continuous Time Markov Chain of M/M/1 model.

The state transition diagram of our birth-death process to our P2P VoD system is as shown in Figure3.

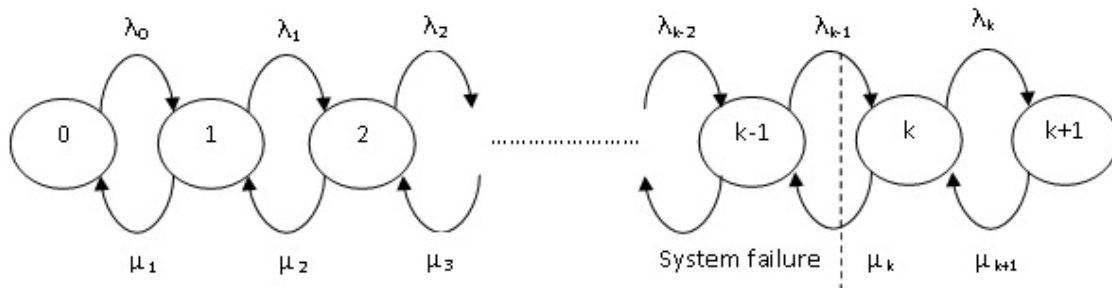


Fig.3. Birth-death process to our P2P VoD system

When a system is in state  $n$  it has  $n$  peers in it. When new arrival of a peer takes place at a rate of  $\lambda_n$ . The failure of a peer takes place at a rate of  $\mu_n$ . Both the new arrival of peers and failure rates of peers are exponentially distributed.

The steady-state probability of our birth-death process being in state  $k$  is given as follows.

Suppose the system is in the state  $k$  at certain time  $t$ . There are  $k$  peers in the system. In the next time interval of duration  $\Delta t$ , the system can move to state  $k-1$  or  $k+1$  with the following probabilities. Probability of one arrival of peer in

$$P[n(t+\Delta t)=k+1/n(t)=k] = \lambda_k \Delta t$$

Probability of one failure of peer in

$$P[n(t+\Delta t)=k-1/n(t)=k] = \mu_k \Delta t$$

If there are no new arrivals or failures of peers, the system will stay in state  $k$  and thus

$$P[n(t+\Delta t)=k/n(t)=k] = 1 - \lambda_k \Delta t - \mu_k \Delta t$$

We are assuming  $\Delta t$  to be average of 0.5 hours that there is probability of two arrivals or two failure occurring during this interval.

The solution for the steady state probability is derived with following equation.

$$P_k = \frac{\lambda_0 \lambda_1 \dots \lambda_{k-1}}{\mu_1 \mu_2 \dots \mu_k} \quad \text{----(14)}$$

$$P_k = P_0 \prod_{j=1}^{k-1} \frac{\lambda_j}{\mu_{j+1}} \quad \text{-----(15)}$$

where  $k=1,2,\dots,\infty$

The equilibrium probabilities  $P_k$  in terms of  $P_0$  computed using additional condition that the sum of all probabilities must be equal to 1, which is derived with equation is as follows

$$P_0 = \frac{1}{1 + \sum_{k=1}^{\infty} \prod_{j=0}^{k-1} \frac{\lambda_j}{\mu_{j+1}}} \quad \text{----- (16)}$$

Thus, as the number of chain increases the system reaches the steady state and there will be smooth streaming of video streams among the peers.

## VII. ANALYSIS OF FAULT TOLERANCE

Let  $k$  is the number of states; each state peers are available in M/M/1 model, the chaining among the peers is based on the condition of duration of the movie  $D$  and play back rate of that peer. Let  $\lambda$  is exponential distribution with timeline factor peer availability. Let  $\mu$  is also exponentially distributed with failure rate based on stop fail, hardware and software fail or Interactive controls.

In chaining based systems the failure of any peers may corrupts a video chain due to several reasons such as stop sending streams from the neighboring peers (selfish peers), due to hardware or software failure etc., To address this problem we have present the some of the fault detection scenarios with recovery system process. In video chaining process is a client-server model, a peer can act as miniature as well as client or vice versa. Once a peer fails, its downstream peer will identify this after receives the video streams from this peer after certain timeout.

The fault will identify by the peers play out of video streams of the requested movie through VCR operations. In the buffered video streams/segments, some of the

segments are missing because these segments are not received from its immediate upstream peer. Even we can find the missing segments from the segment offset of the segments of the entire video. To recover the fault, the downstream peer send fault peer to nearby proxy server as well as all the peers which are chained in the system. On the other hand upstream peer will send the FAULT control message will broadcast the entire chained peer as well as nearby proxy server. Then the proxy server sends the SELECT message may contain new possible optimal paths [12] for chaining, longest alive time of the peers, fault peer etc., Then the peer will receive SELECT message after certain timeout and respond back to the proxy server. Now the proxy server sends the new optimal paths, then the peer may select best optimal path and starts streaming from the next neighboring peer otherwise it streams from the proxy server.

In case of first scenario, let us consider three peers P1, P2 and P3 before the peer fails number of peers in the chaining is 3 after the peer fail it is 2. Assume peer P2 of downstream peer fails, then the peer P3 will identify the missing segments receives after certain timeout. Then peer P2 will immediately send fault peer information FAULT message to the immediate peers in the chaining as well as proxy server. Then the neighboring peer will receive SELECT message from the proxy server after certain timeout and respond back to the proxy server. Now the proxy server sends the new optimal paths, then the peer may select best optimal path and starts streaming from the next neighboring peer otherwise it streams from the proxy server.

In case of second scenario, let us consider the five peers P1, P2, P3, P4 and P5, before the peer fails number of peers in the chaining is 4 after the peer fail it is 3 Peer P1 starts streaming to Peer P2, Assume peer P2 of downstream peer fails, then the peer P3 will identify the missing segments receives after certain timeout. Then peer P3 will immediately send fault peer information FAULT message to the peers in the chaining as well as proxy server.

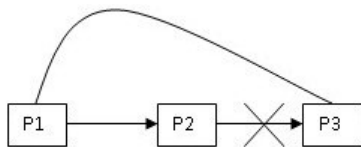


Fig.4. Chaining session failure of peer P2

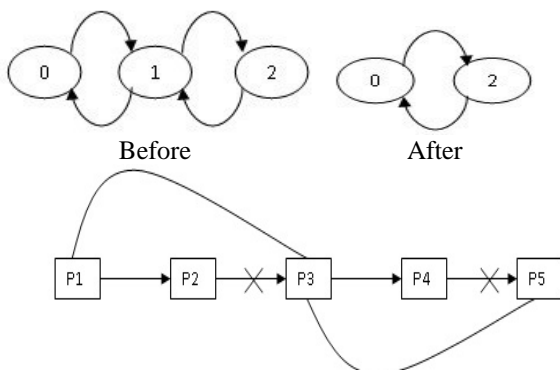
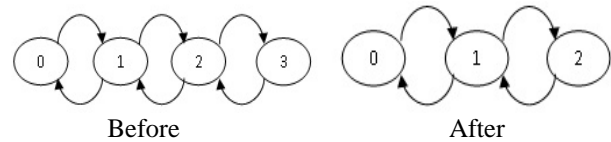


Fig.5. chaining session failure of peers P2 and P4



Then the neighboring peer will receive SELECT message from the proxy server after certain timeout and respond back to the proxy server. Now the proxy server sends the new optimal paths, then the peer may select best optimal path and starts streaming from the next neighboring peer P1 otherwise it streams from the proxy server. Assume peer P4 of downstream peer fails, then the peer P5 will identify the missing segments receives after certain timeout. Then peer P5 will immediately send fault peer information FAULT message to the peers in the chaining as well as proxy server. Then the neighboring peer will receive SELECT message from the proxy server after certain timeout and respond back to the proxy server. Now the proxy server sends the new optimal paths, then the peer may select best optimal path and starts streaming from the next neighboring peer P4 otherwise it streams from the proxy server.

In case of third scenario, let us consider the three peers P1, P2, and P3 with proxy server Px. Assume peer P2 of downstream peer fails, then the peer P3 will identify the missing segments receives after certain timeout. Then peer P3 will immediately send fault peer information FAULT message to the peers in the chaining as well as proxy server.

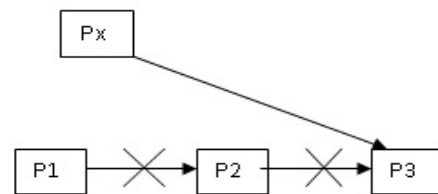


Fig.6. chaining session failure of peers P2 and P3

Then the neighboring peer will receive SELECT message from the proxy server after certain timeout and respond back to the proxy server. Now the proxy server sends the new optimal paths, then the peer may select best optimal path and starts streaming from the next neighboring peer P1, if the peer P1 is also fails to stream, it streams from the proxy server. Now the consumption of bandwidth usage will more, because requested video streams are directly stream from the proxy server.

## VIII. SIMULATION

The performance of the proposed architecture is evaluated through extensive simulation using MATLAB software. We analyze the performance of the system under various parameters that is different Poisson arrival requests into the system will offered different levels of cooperation presented by the seed peers. Our simulation program assumed that request arrivals for particular videos were distributed according to a Poisson process simulated the requests for a single two-hour MPEG-2 video. We did

not consider higher arrival rates as they seemed unrealistic. Each simulation run involved at least 5,000 arrivals over a simulated time period of at least 5000 minutes. The simulation model is evaluated for several times, we have measure the following parameters

1. Arrival of requests of peers versus failure of peers
2. Normal versus recovery
3. The average load on the system
4. The peer band width and buffer in the system
5. The server bandwidth and server load of the various chaining approaches is compared with our proposed approach.

The topology used in our simulation consists of a single media server and 5 cluster based network. Each cluster constitutes a proxy server and 500 peers which includes the seed peers and the non seed peers. The media server consists of MPEG-2 movie files with duration ranging from 4800 min to 6000 min and their popularity is based on Zipf's law. The media server bandwidth capacity of 100 Mbps and the capacity of the buffer is 5000 MB. The proxy server bandwidth capacity is ranging from 30 Mbps to 50 Mbps and buffer capacity ranging from 500 MB to 1000 MB. Each seed peer has a bandwidth capacity of 5 MB to 10 MB and buffer capacity of 1000 MB. The non seed peer has a bandwidth capacity of less than 5MB and buffer capacity of less than 800MB. The proxy server maintains a database of currently streamed/streaming movies and a list of seed peers. The multimedia server contains the information about the movies such as index, size, duration, popularity etc., and this server also maintains two separate files for each of the movie. One file contains the segmented parts of the movie and other file contains the frames of the scene boundary locations. The average duration of each movie is 7200 seconds. The total number of movies requested in a cluster is less than 300 requests during the peak duration. It follows a Poisson request rate with mean  $\lambda=50$  during the normal durations. The average normal playback rate of each movie is 2 MB. The operations are executed on a timeline between 500 minutes and 5000 minutes in a cluster and follows the Poisson request rate with mean  $\lambda=50$ .

System parameters	Default values
Media server bandwidth	100 Mbps
Media server buffer	500MB to 1000MB
Proxy server bandwidth	30 Mbps to 50Mbps
Total number of peers in each of the Clusters	500 peers
Seed peer bandwidth capacity	5Mbps to 10Mbps
Seed peer buffer capacity	1000MB
Non-Seed peer bandwidth capacity	< 5Mbps
Non -Seed peer buffer capacity	< 800MB
The average duration of each movie	7200 seconds
Total number of movies requested in a cluster	< 300 requests during peak duration
Mean arrival of requests	$\lambda=50$ during normal duration
Average playback rate of each movie	2Mbps
Simulation time lines	500 to 5000 minutes

Table1: System parameters with their default values

## IX. RESULTS

The simulation model is executed for several times and the result shown is an average of all simulation trails carried out in all 5 clusters. During the simulation, we have observed from the Figure 1,the requests of peer's of videos arrive to the system according to Poisson's process with the arrival rate  $\lambda$ .we first measured the average arrival rate of requests of peers into the system. The mean arrival rate is varying between 100 to 460 requests of peers.

In Figure 7 we have observed that,the average nubmer of requests of peers arrive to the system is varying between 100 to 460 peers according to the poissons process.When the requests of movie from the peers arrive to the system,it checks the difference of interarrival time of the new peer and other peers of the existing chain, which is active is less than the duration of movie D minutes,The various conditions with which the chaining occurs will be clearly disucussed in section in section 4. For this requests of peers to be served immediately from the active chain and also add them to the currently available existing active chain.

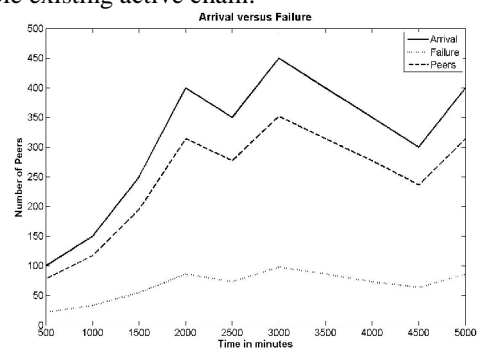


Fig.7. The average nubmer of requests

Hence our proposed VChaining approach reduce the number of peers requests rejections and increase the serving of requests of peers. In figure 7 also shows that number of peers in the system is claculated based on the difference of the number of peers arrival into the system and number of peers fails in the system.

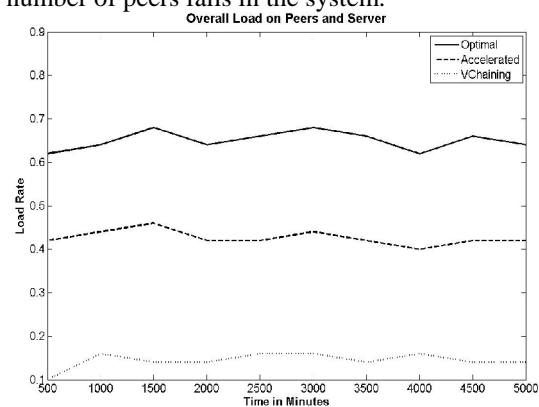


Fig.8. Shows the load on the peers and server

Figure 8 shows the load on the server versus number of requests. When the requests of videos from the peers arrive to the system,it checks the difference of interarrival time of the new peer and other peers of the existing chain,

which is active is less than the duration of movie D minutes, The various conditions with which the chaining occurs will be clearly discussed in section in section 4. For this requests of peers to be served immediately from the active chain and also add them to the currently available existing active chain. We measured the load on the server in terms of bandwidth utilization at the server. As the inter arrival of requests of peers for the movie varies between 0 to 5000 minutes the load on the server is between 10% to 18% in case of our proposed approach V-Chaining, which is less compared to all the existing chaining approaches, because the requested video streams is available in the nearby peers, accordingly the requested peer chain among the peers establishes the chaining. As in case of other approaches the peers are chaining, if the condition discussed in section 4 will not satisfied, then the video streams are directly streamed from the server. Hence the load on the server gradually increases. Therefore we observed in optimal and accelerated chaining the load on the server will be varies between 40% to 45% and 60% to 70% respectively.

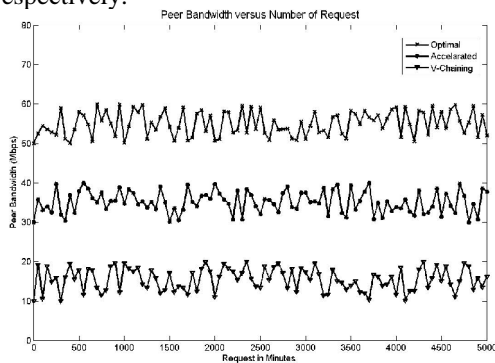


Fig.9. shows the peer bandwidth versus number of requests

Figure 9 shows the peer bandwidth versus number of requests. We have observed the peer bandwidth utilization based on the number of inter-arrival of requests of movie by the peer. When the requests of videos from the peers arrive to the proxy server, it redirects to the requested peer movie id is available in the near by peers, accordingly the peer checks the Condition, which is discussed in section 4, if the condition is Satisfied the peer upload the video streams through uplink to the requested peer chain among the peers establishes the chaining. Hence, our proposed V-Chaining mechanism 10 to 20 Mbps of the peer bandwidth utilization, which is less compared to all the existing approaches. As in existing approaches the peers are fails in satisfy the condition discussed in section 4. Therefore the bandwidth utilization in peers in case of optimal and accelerated chaining is 50 to 60 Mbps and 30 to 40 Mbps respectively.

In Figure 10 shows the normal chaining among the peers with recovered video streams from the failure of peers. In our proposed V Chaining, the chaining among any peers which satisfies certain conditions as we have discussed in section 5 is starts streaming video streams to the requested peer. As we have discussed in section 7, during the chaining process, the normal streaming of video streams among the peers which is actively involved in the

chaining of same movie of video streams, the consumption of bandwidth is less. During the simulation we have observed that average of 15 Mbps of bandwidth usage for normal chaining among the peers. The failure of peers will identified as we have discussed in section 7. Once the failure of peers identified, then neighboring peer will immediately send fault peer information FAULT message to the peers in the chaining as well as proxy server.

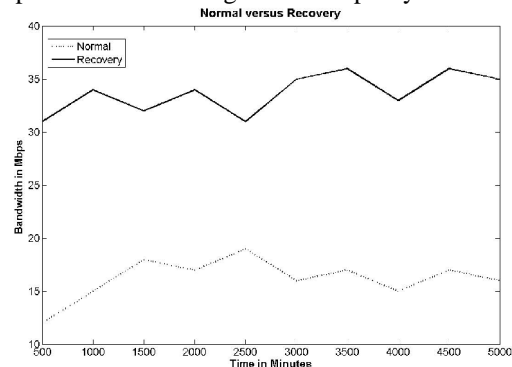


Fig.10. Shows the normal chaining versus video streams from the failure of peers.

Then the neighboring peer will receive SELECT message from the proxy server after certain timeout and respond back to the proxy server. Now the proxy server sends the new optimal paths, then the peer may select best optimal path and starts streaming from the next neighboring peer, If the next neighboring peer peer is also fails to stream, it streams from the proxy server. Now the consumption of bandwidth usage will more, because requested video streams are directly stream from the proxy server. During the simulation we have observed that average of 33 Mbps of bandwidth usage for recovered video streams from the failure of peers.

Figure 11 shows the server bandwidth with the number of requests. We have observed that, As in case of our proposed V-Chaining mechanism, consumption of bandwidth is 22 Mbps, which is very less compared all the above approaches because When the requests of movie from the peers arrive to the system, it checks the difference of interarrival time of the new peer and other peers of the existing chain, which is active is less than the duration of movie D minutes, The various conditions with which the chaining occurs will be clearly discussed in section in section 4. For this requests of peers to be served immediately from the active chain and also add them to the currently available existing active chain due to this consumption of server bandwidth is 22 Mbps in our proposed approach. Accordingly numer of peers chain among with other peers. As the chain fails the peer streams the video from the server. The server bandwidth measured based on the inter arrival of requests, initially less bandwidth consumption in all the approaches as the time increases the bandwidth also increases up to 80 Mbps consumption in case of standard chaining. Also we observed in optimal and accelerated chaining bandwidth consumption up to 55 Mbps and 40 Mbps respectively

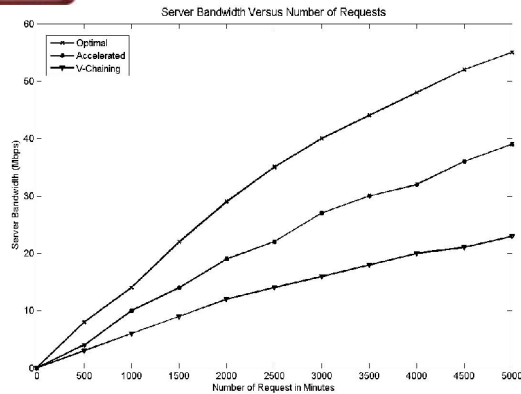


Fig.11. Shows the server bandwidth with the number of requests

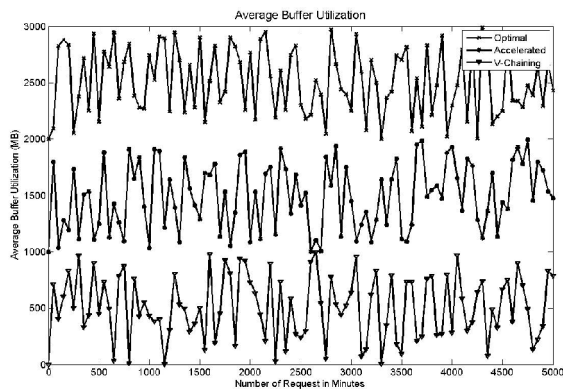


Fig.12. Shows the average buffer utilization in each peer versus number of requests

Figure 12 shows the average buffer utilization in each peer versus number of requests of the movie and play back rate. We have observed that, the buffer utilization for each is measured in terms of video contains in its storage for the duration of the movie  $D$  and the video consumption rate that is playback rate, which is clearly discussed in section 6. We have observed that in case of our proposed V-Chaining mechanism it checks the buffer as discussed the condition in section 4, an average of 300 to 600 MB of buffer utilization in each of peer for its chaining among the peers. Accordingly we have measured from the simulation, in case of optimal and accelerated chaining approaches approximately utilize the buffer as 2000MB and 1000MB respectively.

## X. CONCLUSION

Thus one of the variant of VoD system is peer to peer (p2p) system. In P2P VoD system peers chaining among the peers are frequently fails in streams while chaining. In the existing chaining approaches in the VoD system the chaining among the peers may fail due to stringent bandwidth and buffer, hardware, software and peer failure. To overcome some of these problems we have analyzed the fault tolerance in peer to peer video on demand system is model on continuous time markov model. In this paper we have proposed V Chaining mechanism on Continuous Time Markov Chain model. A birth death process is used to model our proposed mechanism. In this model we have

also shows that the number of chain increases the system reaches the steady state and there will be smooth streaming of video streams among the peers. The parameters used our model as arrival of requests of peers versus failure of peers, normal versus recovery, the average load on the system, the peer band width and buffer in the system, the server bandwidth and server load. We have simulated above parameters using V Chaining mechanism. We have compared our simulation results with existing mechanism such as optimal, accelerated chaining mechanism. Thus our simulation used in V chaining approach results fairer among all the chaining mechanisms. Further enhance of this paper in the VCR functionality of the peer may fails in chaining among the peers.

## REFERENCES

- [1] J. Li, "PeerStreaming: an on-demand peer-to-peer media streaming solution based on a receiver-driven streaming protocol," in Proceedings of the IEEE 7th Workshop on Multimedia Signal Processing (MMSP '05), pp. 1-4, November 2005
- [2] H. Chi, Q. Zhang, J. Jia, and X. Shen, "Efficient search and scheduling in P2P-based media-on-demand streaming service," IEEE Journal on Selected Areas in Communications, vol. 25, no. 1, pp. 119-130, 2007
- [3] S. Annappureddy, C. Gkantsidis, and P. Rodriguez, "Providing video-on-demand using peer-to-peer networks," in Proceedings of the International Conference on World Wide Web, 2006.
- [4] C. Huang, J. Li, and K. W. Ross, "Peer-assisted VoD: making Internet video distribution cheap," in Proceedings of the IPTPS, 2007.
- [5] Y. Huang, T. Z. J. Fu, D. M. Chiu, J. C. S. Lui, and C. Huang, "Challenges, design and analysis of a large-scale p2p-vod system," in Proceedings of the ACM SIGCOMM Conference on Data Communication (SIGCOMM '08), pp. 375-388, August 2008.
- [6] Y. Guo, K. Suh, J. Kurose, D. Towsley, "A Peer-to-Peer On-Demand Streaming Service and Its Performance Evaluation," in IEEE, ICME 2003
- [7] Shaoquan Zhang, Ziyu Shao, and Minghua Chen. "Optimal distributed p2p streaming under node degree bounds". In Proc. of IEEE ICNP, 2010.
- [8] H. Zhang, M. Chen, A. Parekh, and K. Ramchandran. "An Adaptive Multi-channel P2P Video on-Demand System using Plug-and-Play Helpers", In UC Berkeley Technical Report, 2010.
- [9] S. Pawar, S. Rouayheb, H. Zhang, K. Lee, and K. Ramchandran. "Codes for a distributed caching based video-on-demand system", In Asilomar Conference on Signals, Systems, and Computers, 2011.
- [10] K. A. Hua, S. Sheu, and J. Z. Wang "Earthworm: a network memory management technique for large-scale distributed multimedia applications," in Proc. IEEE INFOCOM '97, vol. 3, Kobe, Japan, Apr. 1997, pp. 990-997.
- [11] Z. Liu, Y. Shen, S. Panwar, K. W. Ross, and Y. Wang, "Efficient substream encoding for P2P video," to appear in 16th Packet Video Workshop, Lausanne, 2007.
- [12] Hareesh K and Manjaiah D H "Quality of Service in Peer to Peer Video on demand System V Chaining Mechanism" International Journal of Computer and Information Technology (ISSN: 2279 - 0764) Volume 02- Issue 01, January 2013 in press
- [13] S. Sheu, K. A. Hua, and W. Tavanapong 1997, 'Chaining: a generalized batching technique for video-on-demand systems', Proc. IEEE International Conference on Multimedia Computing and Systems (ICMS '97), Berlin, Germany, pp. 110-117
- [14] F. Lin, C. Zheng, X. Wang, X. Xue, "ACVoD: A peer-to-peer based video-on-demand scheme in broadband residential access networks", International Journal of Ad Hoc and Ubiquitous Computing, 2(4)4, 2007.

- [15] T.-C. Su, S.-Y. Huang, C.-L. Chan and J.-S. Wang, "Optimal chaining scheme for video-on-demand applications on collaborative networks", *IEEE Transactions on Multimedia*, 7(5): 972-980, 2005
- [16] T.-C. Su, S.-Y. Huang, C.-L. Chan and J.-S. Wang, "Optimal chaining and implementation for large scale multimedia streaming", Proc. 2002 IEEE International Conference on Multimedia and Expo (ICME 2002), Lausanne, Switzerland, Vol.1, August, pp. 385-388
- [17] J.-F. Pâris 2005, "A cooperative distribution protocol for video-on-demand", Proc. 6th Mexican International Conference on Computer Science, Puebla, Mexico, pp. 240-246.
- [18] J.-F. Pâris, A. Amer and Darrell D. E. Long, "Accelerated chaining: a better way to harness peer power in video-on-demand applications", Proc. 26th ACM Symposium on Applied Computing (SAC 2011), Taichung, Taiwan, in press, 2011.

## AUTHOR'S PROFILE

### Hareesh K.



is a Research Scholar, Jawaharlal Nehru Technological University, Anantapur, A.P, India. He received his BE in Computer Science and Engineering from Bangalore University and MTech in Computer Science and Engineering from Visvesvaraya Technological University, Belgaum, India. He has more than 14 years of extensive

academic, Industry and Research experience. His area of interest includes Computer communication network, Web Technology, mobile ad hoc networks, multimedia applications, Database management systems, Internet technologies and quality of service. His current research includes P2P Video-on-Demand systems.



### Dr. Manjaiah D.H.

is a Professor and Chairman in the Department of Computer Sciences, Mangalore University, and Mangalore, India. He received Ph.D degree from University of Mangalore, M.Tech from NITK Surathkal. and B.E. from Mysore University, India. He has more than 18 years of extensive academic,

Industry and Research experience. He has authored more than 50 research papers in International / National reputed journals and conferences. His current research focuses on protocol validation and verification, network performance and analysis, Advanced Computer Networking, Mobile / Wireless Communication, Wireless Sensor Networks, Operations Research, E-commerce, Internet Technology and Web Programming. He is the expert committee member of various technical bodies like AICTE, various technical Institutions and Universities in INDIA. He has worked at many technical bodies like CSI, ISTE, ACS, IAENG, WASET, IACSIT and ISOC. He had been actively involving in chairing technical sessions of various International & National Conferences and reviewer of the Journals.