Asian Journal of Information Technology 15 (6): 1139-1145, 2016
ISSN: 1682-3915


# Bandwidth Distribution Algorithm Based DDR Scheduler with Route Selection for Real Time Video Streaming in Peer-Peer Networks

[1]M. Chandrasekaran and [2]T. Premakumari
[1]Electronics and Communication Engineering, Govt. College of Engineering, Bargur
[2]Sona College of Technology, Salem

**Abstract:** The problem of video streaming has been discussed in different articles but suffers with various problems like streaming efficiency, throughput and bandwidth utilization. To solve these issues, a bandwidth distribution algorithm is discussed, which performs route selection based on various strategies. The method identifies set of resource peers and computes routes available to reach the peers. Each route has been identified for the intermediate nodes and their up/downlink capacity using the divisive algorithm. The divisive algorithm splits the route into Internal Bandwidth Section and External Bandwidth section. For each section being identified the method computes the bandwidth distribution for each section of nodes. Based on computed bandwidth distribution factors the method selects a single route to perform video streaming. Also, the method uses the available routes using Dynamic Data Rate (DDR) Scheduling. The scheduler initiates a number of streaming process with varying data rate to maximize the channel utilization and improves the video streaming performance.

**Key words:** Peer-peer networks, DDR scheduler, real time video streaming, bandwidth distribution, route selection

## INTRODUCTION

The peer to peer network has a number of distributed nodes placed in a different geographic region which communicates using the connection based protocol. In real world network, the resources are stored in different peers of the network which can be accessed by the rest of the nodes. In general data access, the nodes communicate between them to identify the routes available in order to access the data from any of the peer node. The request generated from any source peer $S_p$, pass through a number of intermediate nodes $N_k$, to reach the destination peer Dp. Each intermediate node has different bandwidth connections and capable of transferring different size of data bits. The throughput of transmitting the data between the source and destination peer is highly based on the bandwidth constraints. If the links between the peers have higher bandwidth then the throughput of data transmission would be higher. Also the other factors like latency, bandwidth utilization is also based on the bandwidth factors only.

However, the data packets sent from any source node has to be forwarded to reach the destination. To forward the data packets the nodes discover the routes to reach the destination and select a single route based on various measures. Some of the methods use, the hop count to select the shortest route and in some other approaches, the methods use the traffic constraints to select the route in such a way to reduce the latency. But in order to perform video streaming considering only the hop count and traffic factors will not support efficiently. So that the protocol has to consider more number of factors in selecting the route for video streaming.

The nodes of peer to peer network have various bandwidth conditions and the data rate also varies accordingly. The nodes stream the video content in a constant manner like streaming some fixed amount of data per second through a particular route. By doing streaming in such manner does not utilize the available bandwidth. For example, the video streaming in torrent networks streams the data in fixed rate which does not utilize the bandwidth an efficient manner. The DDR-Dynamic Data Rate scheduler can vary the data rate of video streaming in each route used according to various constraints. The DDR computes the bandwidth support for uplink and downlink using the internal and external bandwidth constraints. By setting, the data rate according to these varying conditions of internal and external bandwidth sections the bandwidth utilization can be improved. The selection of internal and external sections of bandwidth is

**Corresponding Author:** M. Chandrasekaran, Electronics and Communication Engineering, Govt. College of Engineering, Bargur

performed based on the hop count. The value of bandwidth boundary is selected based on the number of hop counts in each route.

Real time video streaming is to stream the video content simultaneously from a large number of peers at the same time. The earlier methods assign the streaming rate in a constant manner which cannot be modified on the fly but the real time video streaming approach can change the data rate in which the nodes can download the video file. On the other side the route selection is the major factor which affects the quality of video streaming. The route selection is performed based on various constraints and we focus on the bandwidth, hop count, delay, traffic and more in selecting the route. Using all the above-mentioned factors of any route the method computes data support value based on which a single route is selected. In our case, we select multiple routes for different resource peers and assigns data rates using the DDR scheduler.

**Literature review:** There are a number of methods has been discussed for the problem of video streaming in peer to peer networks. We discuss some of the methods here in this study.

A bandwidth allocation scheme for peer to peer streaming in wireless local area network is discussed in (Kim, 2014). The work is focused on improving the multimedia streaming capability which supports video streaming. The scalability of video streaming in mobile peer to peer architecture is considered and investigates the effect of topology in throughput. The author proposes a bandwidth allocation scheme which is adaptive for various conditions.

A detailed survey of peer to peer video on demand streaming is conducted by Ghosh *et al.* (2014) which explores the various issues of video streaming like overhead in servers. The author also focused on building streaming applications which are interactive. The researcher discuss the flow of peer startup and how it uses the buffers available initially (Li *et al*., 2013). The author proposes a chunk based streaming approach and use a fixed padding scheme. The fixed padding scheme use an offset which is set to the neighbor referred value.

A smooth cache scheme is discussed by Roverso *et al.* (2012). To improve the performance of video streaming and the scheme works on the http protocol. The scheme supports varying bitrate streaming. A dynamic congestion avoidance mechanism is discussed in this to improve the efficiency of video streaming. An allocation scheme based on incentive is discussed by Chen and Zou (2012) to support video streaming in peer to peer networks. It works as a decentralized nature and support for online games.

A policy-driven approach is discussed for on-demand video streaming (Fabio *et al*., 2012). The method works based on the playback policies given and supports storage of blocks dynamically. The stored blocks can be redistributed later based on the demand. This enables the distribution of video blocks and streaming capability efficiently. An multicast system for peer to peer network has been presented which eliminates trees (Pai *et al*., 2005). In this approach the neighbor has to request the node to receive the packet. This eliminates packet duplication and improve the streaming performance.

To improve the streaming performance an adaptive approach is discussed (Agarwal and Rejaie, 2005). The method enables the receiver to intimate the quality of streaming and capability of receiving the stream. Based on the quality of receiver the source sends the stream towards the receiver. An data driven video streaming approach is discussed (Zhang *et al*., 2005). In this, the node performs streaming according to the availability of data. The receiver is notified with the amount of data available and the receiver performs streaming accordingly.

A live video streaming with push pull approach is discussed by Zhang *et al.* (2005). The sender push the data towards the receiver and the receiver pulls the available data in the channel. This improves the streaming performance where the data is large in size. A mesh based video streaming for peer to peer network is discussed by (Magharei and Rejaie, 2007). In this approach the receiver can receive the data directly from the sender and all the nodes has direct connection with others. In (Venkataraman and Francis, 2006), an unstructured end to end streaming is discussed. The method use a multi-tree, heterogeneous P2P multicast algorithm for efficient streaming which uses an unstructured overlay. An heterogeneous approach is discussed by Venkataraman and Francis (2006). The method use the random selection technique for the streaming of data and performs the route selection accordingly. An membership management algorithm has been discussed by Jiang and K. Nahrstedt (2006) and the method use the random selection of nodes to improve the streaming performance.

To improve the performance of video streaming performance, the bandwidth management algorithm is discussed by Kostic *et al.* (2005). The method maintains the bandwidth availability even at the congested situations and improves the streaming performance. Such method to improve the bit torrent network is discussed by (Bharambe *et al*., 2012). The researcher performs a detailed

comparative study video streaming approaches and an data driven approach is discussed by Bharambe *et al.* (2006), Li *et al.* (2006). The method is towards improving the video streaming in delay tolerant networks.

All the above-discussed approaches have the problem of selecting the peer from a number of available peer to perform video streaming. To solve this issue, in this study an efficient video streaming and peer selection approach is discussed.

## MATERIALS AND METHODS

**Ddr scheduling based real time video streaming:** The dynamic data rate scheduler uses the divisive algorithm to compute the internal bandwidth support and external bandwidth support. Using the computed bandwidth support values the method computes the weight for each route to select a single route to stream data from the available location. Also, the scheduler monitors the changing bandwidth conditions and modifies the data rate in which the node can stream data from the peer. The bandwidth distribution algorithm distributes the available bandwidth between a number of peers of network in the internal/external bandwidth sections. The entire process can be split into a number of stages, namely route discovery, Bandwidth Distribution algorithm, DDR scheduling, route selection. We discuss each of the stages in detail in this study.

**Route discovery:** The route discovery process generates RDR-Route Discovery Request and multicast to the neighbor nodes and propagated to all the nodes of the network. The receiving node generates reply about the presence of route to reach the destination. From the reply being received from the neighbor node, the method extracts the routes and store them into the route table. The discovered routes are used to perform bandwidth distribution and route selection. From the reply being received the method extracts various features like list of hop names, uplink/downlink capacity, traffic and delay to store them in the route table.

**Pseudo code of route discovery:**

```
Input: Neighbor Table Nt
Output: Route Table Rt
Begin
    Generate Route Discovery Request RDR.
    For each neighbor Ni from Nt
                Send the RDR
    End
    For each neighbor Ni from Nt
                Receive RDREP.
                Extract Route from RDREP.
                Ri = Route
                Extract host names.
                Hn = ΣHostϵRi//Identifies the list of host present in
the route Ri                Compute hop count Hc
                Hc = size(Hn).
                For each host Hi from Hn
        Extract Traffic HT = ∫_{i=1}^{size(Hn)} Hn(Hi)Traffic
        Extract Delay HD = ∫_{i=1}^{size(Hn)} Hn(Hi)Delay
        Extract Uplink Capacity HUC =
        ∫_{i=1}^{size(Hn)} Hn(Hi)unlinkcap
        Extract Downlink Capacity HDC =
        ∫_{i=1}^{size(Hn)} Hn(Hi)dlinkcap
        Add to route table
          {HT, HD, HUC, HDC}
        Ri = Ri
        Rt = ∑(Rjϵ RT)∪Ri
                End
    End
End
```

The algorithm depicts that the source node sends the RDR-Route request to all its neighbors. The source node is replied with the list of routes available by the neighbors. From the routes obtained, the node identifies the list of host names and for each of them, the node computes the traffic, delay, uplink capacity and downlink capacity. The node stores the routes in the route table and each route has the features above mentioned.

**Route selection:** The route selection is the process of selecting a single route from available routes based on certain factors. In this method, we first use the divisive algorithm which returns the streaming support factor for each of the route. Based on the streaming support factor being returned the method selects the single route from available routes. The selected route will be used to perform video streaming.

**Pseudo code of route selection:**

```
Input: Route Table Rt
Output: Route Ri
Begin
    For each route Ri
                Streaming support factor SSF = Divisive-
Algorithm(Ri).
    End
    Chose the route with more SSF value.
                Ri = ∫_{i=1}^{Size(Rt)} Max(Ri,SSF)  = //The function
identifies the route with more
            streaming support factor
End
```

The above discussed algorithm computes the streaming support factor for each of the route using the divisive algorithm. Based on the SSF value a single route will be selected to perform video streaming.

**Divisive approach:** The divisive approach splits the route given in to inter bandwidth section and external

bandwidth section. To split the sections the method computes the threshold value using the hop count of the route. The nodes or hops within the threshold are assigned to the internal bandwidth section and rest is to the external bandwidth section. For each section of assigned nodes, the method computes streaming support factor using the factors stored in the route table. The streaming support factor represents the support of the route to stream video content. Using both the support factors, the method computes the cumulative support value and returned.

**Pseudo code:**

Input: Route Table Rt, Route Ri
Output: Streaming support factor CSSF.
Begin
Compute hop count hc = $\int \sum$Hops∈Ri
Compute hop threshold Ht = 2/3×hC
Extract Internal Bandwidth Section Nodes.

$$IBSN = \int_{i=ht}^{ht} \sum Hop(i) \in Ri$$

Extract External Bandwidth Section Nodes

$$EBSN = \int_{i=ht}^{ht} \sum Hop(i) \in Ri$$

For each hop of IBSN
Compute Streaming support value SSV

$$ISSV = \int_{i=t}^{Size(IBSN)} IBSN(i)Traffic/IBSN(i)Bw$$

$$IBSN(i)Delay / hC - i$$

$$IBSN(i)Uplinkcap / IBSN(i)Dlinkcap$$

$$ISSV = \sum Issv/size\ (IBSN)$$

$$ESSV = \int_{i=t}^{Size(EBSN)} EBSN(i)Traffic/IBSN(i)Bw$$

$$EBSN(i)Delay / hC - i$$

$$EBSN(i)Uplinkcap / IBSN(i)Dlinkcap$$

$$ESSV = \Sigma Essv/Size\ (EBSN)$$

End
Compute Streaming support value SSV.
End
Compute cumulative streaming support factor.,
CSSF = ISSV×ESSV
End

The above presented algorithm computes the cumulative streaming support factor for each route being identified and used to perform route selection.

**DDR Scheduling:** The dynamic data rate scheduling approach uses various functional components of the proposed method. The method starts with discovering the available routes with the information that the resource peers. For each resource peer available the method discovers the routes and for each route, the method computes the streaming support factor using the divisive algorithm. Based on the computed value, the method selects a single route using the route selection approach. The selected route will be used to perform video streaming where the data rate for the route is set based on the support factor. Later the value of data rate is modified based on the data rate received and streaming support factor.

**Pseudo code:**

Input: Neighbor Table Nt, Resource Requested Rr
Output: Null
begin
Identify set of all peers having the requested resource Rr.
Peer set Ps = Σpeers (Network)∈Rr
For each peer Pi from Ps
Compute available routes Rs.
Rs = Route Discovery (Pi)

$$Rs = \int_{i=1}^{Size(ps)} \sum Routes \in Network$$

Selected Route Sr.
Sr = Route-Selection(Rs)
Compute streaming support factor SSF
SSF = Divisive (Sr)
Compute data rate Dr
Dr = SSF/Ri.Bandwidth
Start Streaming.
End
While True
For each route Ri
Monitor data rate Dr
Compute SSF
Compute Dr value
Modify Data rate
End
End
End

The above discussed algorithm performs dynamic data rate scheduling and performs video streaming in an efficient manner.

**Experimental setup:** The divisive algorithm based dynamic data rate scheduler for video streaming has been implemented using java and has been tested for its efficiency using the following setup mentioned.

Table 1 shows the simulation parameters used to evaluate the proposed method. The method has been evaluated with a different number of peers and window size and number of requests.

**Case study:** The proposed approach has been evaluated for its efficiency in video streaming as follows: The video resource Vi has been placed in K locations of the network, where the network has 200 number of peers. The user request Ur has been received and the method identifies the requested resource Ri which denotes the video Vi.

Table 1: Simulation parameters of proposed method

| Parameters | Value |
|---|---|
| No. of peers | 200 |
| No. of Resources | 10 |
| Average window size | 4 |
| No. of requests | 100 |

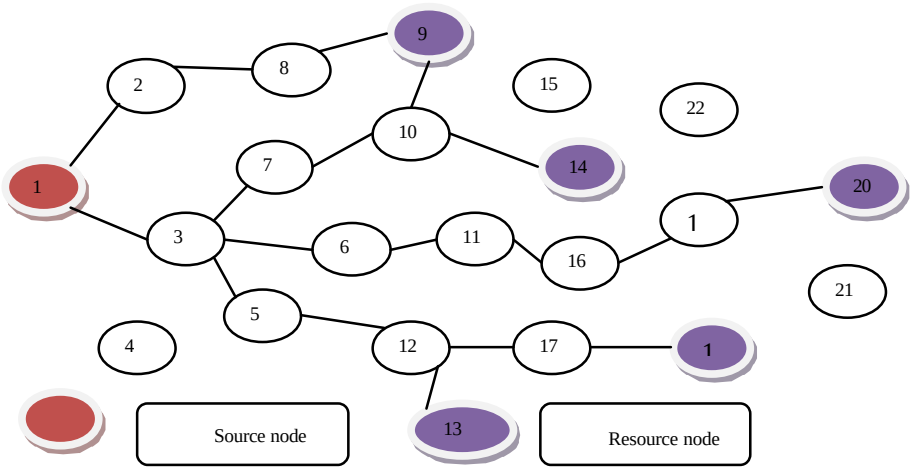


Fig. 1: Example scenario of video streaming

Table 2: Details of route selection

| Resource peers | Routes | Video streaming factor | Amount of stream allocated () | Selected route |
|---|---|---|---|---|
| 9 | 1-2-8-9 | 9.3 | 46 | Yes |
| | 1-3-7-10-9 | 8.2 | - | No |
| 13 | 1-3-5-12-13 | 6.8 | 27 | Yes |
| 14 | 1-3-7-10-14 | 6.8 | 27 | Yes |
| 18 | 1-3-5-12-17-18 | 4.1 | - | No |
| 20 | 1-3-6-11-16-19-20 | 1.6 | - | No |

Upon identifying the requested video content the method identifies the set of locations Ls, where the resource is available. Once the locations are identified, the method computes the routes available to reach the destination peers P, which contains the requested resource. For each route of Pi, available from R, the method computes the streaming factor Sf. From computed streaming factor Sf, the method performs route selection and assigns the amount of data to be streamed from the peer Pi through the route Ri.

Figure 1 shows the small region of the peer to peer network where the video streaming is performed and it has five different resource peers from where the resource can be streamed.

Table 2, shows the computed values of video streaming factor and routes available to reach the resource peers.

The method computes the video streaming factor according to the hop count, traffic and many parameters. Based on the video streaming factor the DDR scheduler schedules the video streaming and performs route selection. In Fig. 1, there are five resource peers and the topology has different routes to reach the resource peers. According to the video streaming factor, the method selects only three peers for the streaming and assigns different amount of data to be streamed according to the factor computed.

## RESULTS AND DISCUSSION

The proposed divisive algorithm with DDR scheduling has been implemented and simulated and the performance of the protocol has been evaluated using the various size of video file streaming. The method has been tested with a different number of resources and different number of peers with a number of users in the test environment. The method has produced efficient results in all the factors of quality of service in video streaming or data streaming in peer to peer networks.

Figure 2, shows the comparison result of video streaming efficiency and the graph shows that the proposed method has produced efficient results than others.

Figure 3 shows the comparison result of video streaming latency produced by different methods and it shows clearly that the proposed method has produced less latency than others.

Figure 4 shows the comparative result on bandwidth utilization efficiency. The result shows that the proposed method has produced more bandwidth utilization efficiency than others.

Figure 5 shows the comparative result in throughput performance produced by different methods on video streaming. The values show that the proposed method has produced more throughput performance than other methods.

Figure 6, shows the comparison of data reduction in the form of routes produced by different methods. The result shows that the proposed DDR technique has produced more data reduction to improve the performance of video streaming.

Figure 7 shows the comparison of time complexity produced by different methods on selection the routes

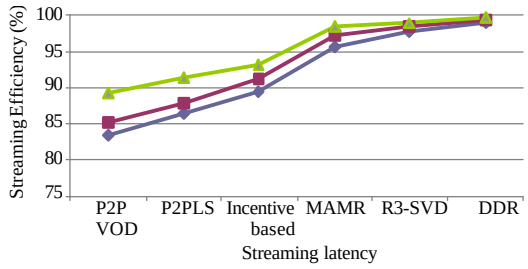


Fig. 2: Comparison of video streaming efficiency

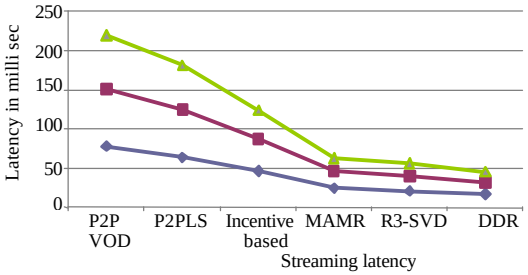


Fig. 3: Comparison of latency in video streaming

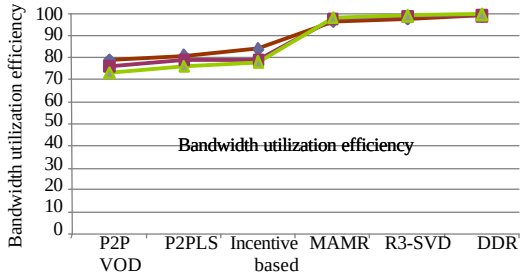


Fig. 4: Comparison of bandwidth utilization efficiency

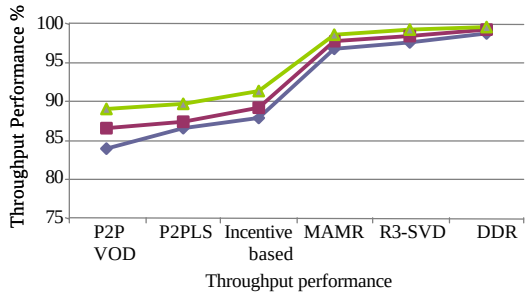


Fig. 5: Comparison of throughput performance

and performing video streaming. The result shows that the proposed method has produced less time complexity by reducing the data and choosing the best efficient route for video streaming.

Figure 8, shows the comparison of route selection efficiency produced by different methods and shows that the proposed DDR approach has produced more route selection efficiency than other methods.

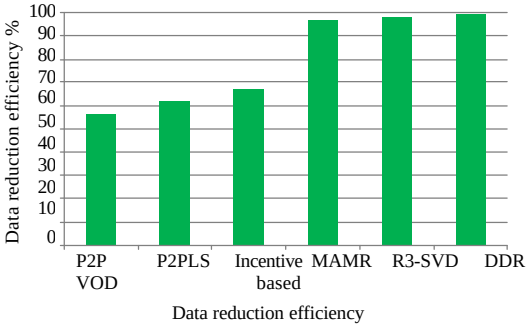


Fig. 6: Comparison of data reduction produced

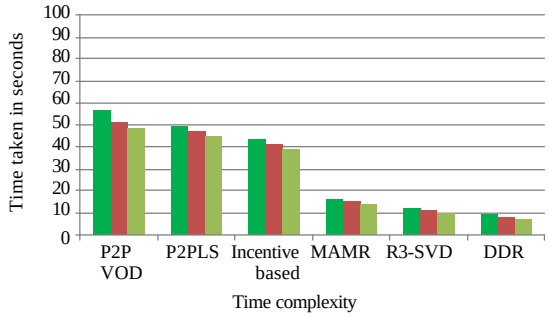


Fig. 7: Comparison of time complexity

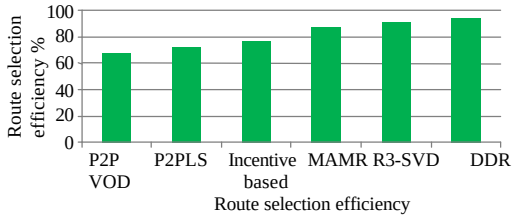


Fig. 8: Comparison of route selection efficiency

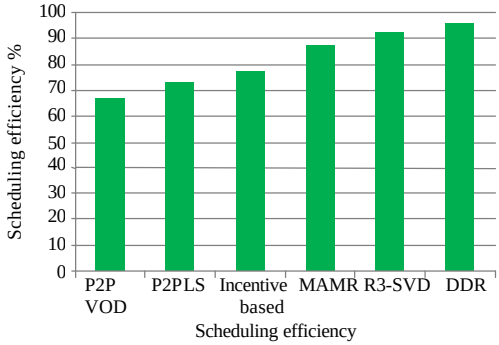


Fig. 9: Comparison of scheduling efficiency

Figure 9, shows the comparison of scheduling efficiency produced by different methods and shows that the proposed method has produced higher scheduling efficiency than other methods.

Table 3: Comparative analysis of various parameters

| Method name | Streaming with accuracy No. of nodes | | | Streaming latency in milli seconds with No of nodes | | | Bandwidth utilization seconds with No. of nodes | | | Throughput Performance with No. of nodes | | | Time complexity in seconds with No. of nodes | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 70 | 100 | 200 | 70 | 100 | 200 | 70 | 100 | 200 | 70 | 100 | 200 | 70 | 100 | 200 |
| P2PVOD | 83.4 | 85.2 | 89.3 | 79 | 72 | 68 | 79 | 76 | 73 | 84 | 86.7 | 89.2 | 56 | 51 | 48 |
| P2PLS | 86.5 | 87.8 | 91.3 | 64 | 61 | 56 | 81 | 79 | 76 | 86.5 | 87.4 | 89.7 | 49 | 47 | 45 |
| Incentive Based | 89.4 | 91.2 | 93.2 | 46 | 41 | 37 | 84 | 79 | 78 | 87.8 | 89.3 | 91.4 | 44 | 41 | 39 |
| MAMR | 95.7 | 97.2 | 98.4 | 25 | 21 | 17 | 96.7 | 97.4 | 98.1 | 96.7 | 97.8 | 98.6 | 16 | 15 | 14 |
| R3-SVD | 97.8 | 98.4 | 98.9 | 21 | 19 | 16 | 97.6 | 98.3 | 98.9 | 97.6 | 98.4 | 99.2 | 12 | 11 | 10 |
| DDR | 98.9 | 99.3 | 99.7 | 17 | 15 | 13 | 98.8 | 99.2 | 99.6 | 98.7 | 99.2 | 99.6 | 9 | 8 | 7 |

Table 3 shows the comparative results on various parameters produced by different methods. The values show that the proposed method has produced more efficient results than other methods.

## CONCLUSION

In this study, an efficient bandwidth distribution algorithm with dynamic data rate scheduling has been presented. The method discovers the routes available and for each route, the method computes the streaming support factor using the divisive algorithm. Based on computed streaming support factor the method selects a single route and streams the video content. The method computes the data rate to be streamed based on the streaming support factor and bandwidth. The method monitors and modifies the data rate in a dynamic manner and improves the performance of video streaming with higher bandwidth utilization. Also,, the method produces less time complexity and latency.

## REFERENCES

Agarwal, V. and R. Rejaie, 2005. Adaptive multi-source streaming in heterogeneous peer-to-peer networks. Proceedings of the Multimedia Computing and Networking Conference, January 17-20, 2005, San Jose, CA., USA -.

Bharambe, A.R., C. Herley and V.N. Padmanabhan, 2006. Analyzing and improving a bittorrent networks performance mechanisms. Proceedings of the 25th IEEE International Conference on Computer Communications, April 23-29, 2006, Barcelona, Spain, pp: 1-12.

Chen, L. and J. Zou, 2012. Incentive-based Bandwidth Auction for Scalable Streaming in Peer-to-Peer Networks. In: Advances on Digital Television and Wireless Multimedia Communications, Zhang, W., X. Yang, Z. Xu, P. An, Q. Liu and Y. Lu (Eds.). Springer, Berlin, Germany, ISBN: 978-3-642-34594-4, pp: 363-371.

Ghosh, D., P. Rajan and M. Pandey, 2014. P2P-VoD Streaming. In: Advanced Computing, Networking and Informatics-Volume 2, Kundu, M.K., D.P. Mohapatra, A. Konar and A. Chakraborty (Eds.). Springer International Publishing, USA., ISBN: 978-3-319-07349-1, pp: 169-180.

Hecht, F.V., T. Bocek, F.R. Santos and B. Stiller, 2012. Playback policies for live and on-demand P2P video streaming. Proceedings of the 11th International IFIP TC 6 Networking Conference, May 21-25, 2012, Prague, Czech Republic, pp: 15-28.

Jiang, J. and K. Nahrstedt, 2006. Randpeer: Membership management for QoS sensitive peer-to-peer applications. Proceedings of the IEEE 25th Conference on Computer Communications, April 23-29, 2006, Barcelona, Spain, pp: 1-10.

Kim, G.H., 2014. A Bandwidth Allocation Mechanism in Mobile P2P Streaming in the Wireless LAN. In: Ubiquitous Information Technologies and Applications, Jeong, Y.S., Y.H. Park, C.H. Hsu and J.J. Park (Eds.). Springer, New York, USA., ISBN: 9783642416712, pp: 781-787.

Kostic, D., R. Braud, C. Killian, E. Vandekieft, J.W. Anderson, A.C. Snoeren and A. Vahdat, 2005. Maintaining high bandwidth under dynamic network conditions. Proceedings of the USENIX Annual Technical Conference, April 10-15, 2005, Anaheim, CA., USA -.

Li, C., Y. Chen, B. Zhang, C. Li and C. Chen, 2015. Peer startup process and initial offset placement in Peer-to-Peer (P2P) live streaming systems. Peer-to-Peer Networking Applic., 8: 137-155.

Li, D., Y. Cui, K. Xu and J. Wu, 2006. Segment-sending schedule in data-driven overlay network. Proceedings of the IEEE International Conference on Communications, Volume 1, June 11-15, 2006, Istanbul, Turkey, pp: 6-11.

Magharei, N., R. Rejaie and Y. Guo, 2007. Mesh or multiple-tree: A comparative study of live P2P streaming approaches. Proceedings of the 26th IEEE International Conference on Computer Communications, May 6-12, 2007, Barcelona, Spain, pp: 1424-1432.

Moscibroda, T. and R. Rejaie, 2007. PRIME: Peer-to-peer receiver-driven mesh-based streaming. Proceedings of the 26th IEEE International Conference on Computer Communications, May 6-12, 2007, Anchorage, AK., USA., pp: 1415-1423.

Pai, V., K. Kumar, K. Tamilmani, V. Sambamurthy and A.E. Mohr, 2005. Chainsaw: Eliminating trees from overlay multicast. Proceedings of the 4th International Workshop on Peer-to-Peer Systems IV, February 24-25, 2005, Ithaca, NY., USA., pp: 127-140.

Roverso, R., S. El-Ansary and S. Haridi, 2012. Smoothcache: HTTP-live streaming goes peer-to-peer. Proceedings of the 11th International IFIP TC 6 Networking Conference, May 21-25, 2012, Prague, Czech Republic, pp: 29-43.

Saroiu, S., P.K. Gummadi, S.D. Gribble, 2002. A measurement study of peer-to-peer file sharing systems. Proceedings of the Multimedia Computing and Networking, January 18-25, 2002, San Jose, USA., pp: 1-15.

Venkataraman, V. and P. Francis, 2006. Chunkyspread: Multi-tree unstructured peer-to-peer multicast. Proceedings of the 5th International Workshop on Peer-to-Peer Systems, February 27-28, 2006, Santa Barbara, CA., USA., pp: 1-6.

Venkataraman, V. and P. Francis, 2006. On heterogeneous overlay construction and random node selection in unstructured P2P networks. Proceedings of the IEEE 25th Conference on Computer Communications, April 23-29, 2006, Barcelona, Spain, pp: 1-12.

Zhang, M., J.G. Luo, L. Zhao and S.Q. Yang, 2005. A peer-to-peer network for live media streaming using a push-pull approach. Proceedings of the 13th Annual ACM International Conference on Multimedia, November 6-12, 2005, Singapore, pp: 287-290.

Zhang, X., J. Liu, B. Li and T.S.P. Yum, 2005. CoolStreaming/DONet: A data-driven overlay network for peer-to-peer live media streaming. Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Volume 3, March 13-17, 2005, Miami, FL., USA., pp: 2102-2111.