Efficient and Load-Balance Overlay Multicast Scheme with Path Diversity for Video Streaming

Chao-Lieh Chen[1], Jeng-Wei Lee[2], Jia-Ming Yang[2], and Yau-Hwang Kuo[2]

[1] Department of Electronic Engineering,
Kun-Shan University, Yung-Kang, Tainan County, Taiwan
frederic@ieee.org
[2] Department of Computer Science and Information Engineering
National Cheng Kung University, Tainan City, Taiwan
{lijw,abby,kuoyh}@cad.csie.ncku.edu.tw

Abstract. An overlay multicast is proposed to solve the scalability and deployment problems in IP Multicast. We propose a scheme, Topology-aware Load-balance Hierarchical Independent Tree (TLHIT), with topology-aware, load-balance and path diversity properties to improve the performance of overlay multicast. Compared to traditional methods, the proposed TLHIT constructs not only node-disjoint but also path-disjoint multicast trees where each node serves as an interior node in only one tree and different trees do not contain the same path. Moreover, TLHIT ensures load-balance property by building the multicast trees based on *n*-ary full tree. It ensures that each node serves almost the same amount of child nodes. Simulation results show that the reliability, efficiency, and load-balance properties of the proposed TLHIT are assured.

Keywords: overlay multicast, topology-aware, load-balance, hierarchical independent tree, node-disjoint, path-disjoint.

1 Introduction

With the rapid growth of internet technology, more and more one-to-many transmission services are developed including video streaming, distributed simulations, video-conferencing, multi-party games, content distribution, and so on. Thus, IP multicast at the network layer has been proposed for realization of these services. However, it has not been widely deployed yet because of high cost to upgrade the network infrastructure. Recently, overlay multicast is proposed as an alternative to provide the multicast services. In this way, the participating nodes organize themselves into an overlay structure and the efficiency of the overlay can be optimized by adapting to network dynamics and considering application level performance.

In overlay network, each participating node has potentially multiple paths to communicate with another node. Therefore, multi-tree multicast [1][2][3] is proposed to improve the fault-tolerance if compared to single-tree multicast [4][5]. However, how to use these trees more efficiently is still an open problem. Hence, multi-tree multicast with path diversity in overlay network attracts lots of interest in recent

T.-W. Kuo et al. (Eds.): EUC 2007, LNCS 4808, pp. 389–399, 2007.

years. The topology-aware hierarchical arrangement graph (THAG) [1] constructs multi-tree multicast applications with diverse paths. In THAG, all participating nodes are divided into a number of arrangement graphs and several node-disjoint multicast trees are embedded in each arrangement graph. Node-disjoint trees mean that any node serves as interior node in only one tree. Though THAG constructs node-disjoint trees, it leads to unbalance load problem because each node is responsible for handling traffics to different number of child nodes, especially the source node. The situation gets worse as the growth of multicast group members.

In this paper, we propose a load-balance scheme called Topology-aware Load-balance Hierarchical Independent Tree (TLHIT) scheme which construct a virtual graph (VG) at first, and the multicast trees are embedded in VG based on n-ary full tree. Therefore, each node in TLHIT serves almost the same number of child nodes. Moreover, the multicast trees in TLHIT are independent, where independent means that each tree is both node-disjoint and path-disjoint. Hence, the load-balance and fault-tolerant ability are further improved in TLHIT. Moreover, when the number of multicast group members is larger than the capacity of the constructed VG, TLHIT extends the original VG into a larger one by duplicating several child VGs and assembling these VGs into hierarchical structural. As a node joining the multicast service, TLHIT selects a suitable position in the VG not only in accordance with the network conditions but also keeps all the nodes in TLHIT with balance load.

The rest of the paper is organized as follows. In Section 2, background and relative work is introduced. In Section 3, we explain how to design the TLHIT. The simulation results are shown in Section 4 and the reliability, efficiency, and load-balance of THAG and TLHIT are compared in several simulations. The conclusions are drawn in Section 5.

2 Background and Related Work

In this section, we present some background knowledge about multi-tree and application-layer multicast streaming. They are multi-path streaming, multiple descryiption coding (MDC) and topology-aware multicast streaming. Each mechanism is implemented in application layer.

2.1 Multi-path Streaming

Unlike traditional methods which embed redundant bits into each packet to provide error-correction ability, the concept of multi-path is to transmit data through different paths and provide path diversity to avoid data sharing the same congested or troublesome interior nodes or links. It prevents multimedia data from burst error and degraded quality significantly. There are two major multi-path mechanisms -- relay server [6][7] and overlay network [8][9].

- Relay server: For path-diversity, a relay server is responsible for relaying data to a destination. As shown in Figure 1, when node A wants to transmit data to node B, node A transmits partial data to the destination via a relay server indirectly and the other data to destination directly. However, this mechanism has some disadvantages. First, the performance of a relay server is limited and there is a tradeoff between cost and performance. Second, good deployment of relay servers is necessary and it affects overall system performance.

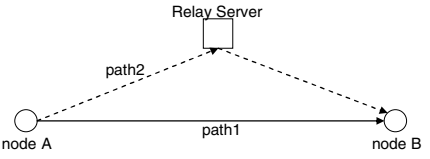

Fig. 1. Achieving path diversity using relay server

- Overlay network: Overlay network is built on top of another network. Each node in overlay network communicates with another node by virtual or logical links which correspond to a direct link or many physical links in the underlying network. It means that the source node has potentially multiple paths to communicate with all the other participating nodes through directly link or other relay nodes. Path diversity can be obtained by a good choice of relay nodes. However, communication between pair-wise nodes bypassing others potentially increases latency. The proposed TLHIT is based on overlay network and focus on how to achieve path diversity with acceptable latency.

2.2 Multiple Description Coding (MDC)

Multiple Description Coding (MDC) [10] is utilized when a media stream needs to be separated into several parts and transmitted in each multicast tree. In MDC, a media stream is encoded into several parts referred to as descriptions. Any combination of received descriptions can be used to decode the original media stream with acceptable quality. Media quality is improved as the number of received descriptions increases and the best media quality is obtained when all the descriptions are received.

2.3 Topology-Aware Hierarchical Arrangement Graph (THAG)

There are some related works on application-layer multicast for media streaming such as THAG [1], which embeds multicast trees in arrangement graphs and provides node-disjoint characteristics in each tree. In THAG, each node serves as interior node which is responsible for forwarding data to other nodes in exactly one multicast tree. Thus, the influence of any node failure is minimized and the fault-tolerance is improved. Figure 2 shows an example of arrangement graphs.

However, THAG does not consider the unbalance load problem of each node. As shown in Figure 2, the load of the source node is much heavier than all the other nodes. And the situation gets worse as the growth of the arrangement graph. Moreover, THAG only guarantees the interior nodes in each multicast tree are different, but it may share the same congested or troublesome path and deteriorate the system performance. Hence, in this paper, we propose a mechanism to build multicast trees with node-disjoint, path-disjoint and load-balance properties.

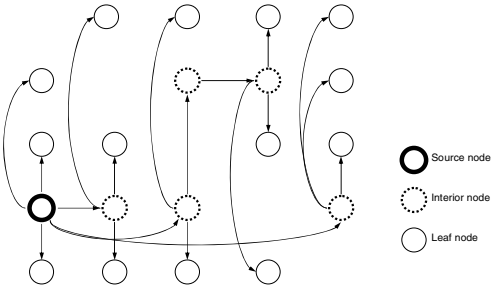

Fig. 2. One multicast tree in THAG

3 Topology-Aware Load-Balance Hierarchical Independent Tree (TLHIT)

In TLHIT, a virtual graph (VG) with n independent multicast trees is built. Independent trees mean that each participating node only acts as the interior node (i.e. non-leaf node) in one of n multicast trees and paths in each tree are distinct. Based on the structure of the VG, as one member joining the multicast group, it selects an appropriate vacant position in VG according to its network conditions. Further, TLHIT extends original VG into hierarchical structure by duplicating several VGs to support more multicast members.

3.1 Independent Multicast Trees in Virtual Graph

To enhance the performance of multi-tree multicast applications, TLHIT considers the following three requirements as constructing a VG:

- *Node-disjoint*: To mitigate the influence of node failure or leaving, we must make sure each participating node only acts as the interior node one time. It means that if one node is selected to be an interior node in one tree, it must be a leaf node in all the other trees.
- *Path-disjoint*: Transmission in the same path may result in path congestion or high relation of loss behavior [11], which increases end-to-end delay and reduces the performance of the MDC streaming. Hence, the path-disjoint is considered in our VG.
- *Load-balance*: In practice, the resources of a node are limited (e.g. computation capability and network bandwidth) and unbalanced load may affect scalability of the multi-tree multicast applications. Hence, the algorithm should achieve load-balance.

For satisfying the three conditions, TLHIT constructs a VG which contains n multicast trees. The parameter n is user-defined and can be decided by the number of descriptions in the MDC method. At first, the nodes in VG are separated into n root sets. All nodes in a root set are organized into an n-ary full tree and leaf nodes are responsible for connecting the other root sets. Each root set forms a multicast tree.

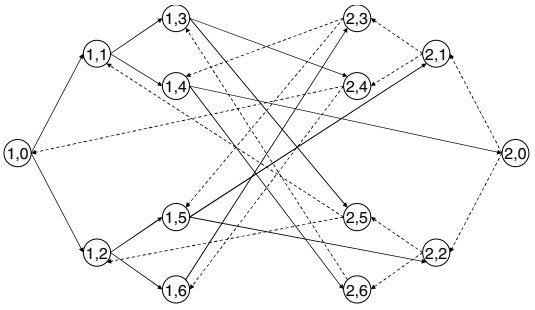

Fig. 3. VG with two multicast trees

Figure 3 illustrates a construction of the VG with two root sets. The solid and dotted lines represent different multicast trees. If the media stream is fragmented into n descriptions, i-th description is transmitted by i-th root set. Hence, via this simple concept, TLHIT guarantees that each node transmits only one description and receives the other descriptions from the other root sets and therefore the first requirement, node-disjoint, is achieved. The topology generate algorithm is described below. To meet the path-disjoint requirement, TLHIT must guarantee that each description is transmitted in different link. Denote $N_{s,i}$ as the node with a pseudo-address i in the s-th root set. Line 6 to line 17 show that when the number of child of leaf node $N_{s,i}$ in root set s is less than n, $N_{s,i}$ chooses an unselected node $N_{m,j}$ as its child node. Line 8 ensures $N_{m,j}$ not to choose $N_{s,i}$ as his child node when m-th set becomes root set. Therefore, path-disjoin is achieved. Moreover, the multicast trees are based on n-ary full tree such that the interior nodes in each tree are responsible for the same number of child nodes. Line 7 and line 14 ensure that each node have at most n child nodes. Therefore, TLHIT satisfies the requirement of load-balance. Line 4 ensures each participating nodes to be selected in each multicast tree. The double-slashes are remarks.

```
0 Algorithm Topology-generate
1 Input: n root sets 1, ..., n; //Each of which contains k nodes
  (k=1+n+n^2). The nodes in each root set are organized into a n-
  ary full tree.
2 Input: Node addresses N_{s,j}; // Each node is given a pseudo-
  address where s is the root set the node belongs to and j is
  the node ID. The pseudo-address of root node is N_{s,0}. The
  pseudo-addresses of child nodes of the parent node N_{s,j} are
  N_{s,n*j+1}, N_{s,n*j+2}, …, and N_{s,n*j+n}.
3 For each root set s= 1, 2, …, n do {
4     Mark all the nodes as unselected except the nodes in root
    set s;
5     While there is any node marked as unselect do {
6         For each leaf node N_{s,i}, i=n+1, n+2, …,n^2+n, in root set
        s, do {
7             While number of child of N_{s,i} ≤ n, do {
8                 j = (i+1)%k; //modulus %
9                 For m=1, 2, …, n, m≠s, do {
10                    If N_{m,j} is marked as unselect, then {
```

```
11                          s = s ∩ N_{m,j}; // N_{m,j} chooses Node N_{s,i}
                                              as parent
12                          Marked N_{m,j} as selected;
13                      }
14                      If number of child of N_{s,i} ≥ n, then
15                          Break;
16                  }
17              } // number of child of N_{s,i} ≥ n
18          } // end for each leaf node
19      } // each node is selected
20 } // end for each s
```

3.2 Extending Virtual Graph to Hierarchical Structure

This section describes how to extend the VG into a hierarchical structure for accommodating more multicast group members. As presented above, the capacity of VG is very limited. Typically, there are at most 14 nodes in the VG with two multicast trees. Hence, TLHIT extends the original VG into a large one when the number of multicast group members is larger than the capacity of the VG. In addition to the tree requirements discussed above, the extended VG must also remain the structure of serving nodes unchanged. This ensures multimedia services are not affected by the extending process.

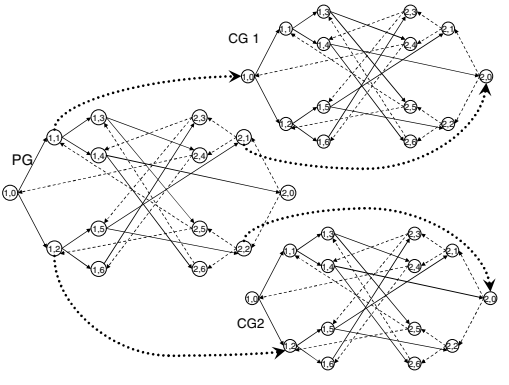

Fig. 4. Hierarchical structure with two root sets

When an "over-capacity" event is triggered, TLHIT duplicates several child VGs (short for CGs) and connects the CGs to the original parent VG (short for PG). The path-disjoint requirement is still retained as the CGs inherits the characteristics of the PG and each leaf node does not have direct connection to each other. Hence, the remaining problem is how to retain the node-disjoint and load-balance properties. The problem can be reduced to determinations of the number of CGs to duplicate and the optimal source nodes positions in CGs. In the PG, let $n_{nonroot}^{(\min)}$ be the total number of nodes satisfying the two conditions that first having connections to any CGs and second having minimum hop count to their root. Then, the number of CGs to

duplicate is $n_{CGs} = \frac{n_{nonroot}^{(\min)}}{n}$. Figure 4 shows an example of hierarchical structure with two root sets. The nodes $N_{1,1}$, $N_{1,2}$, $N_{2,1}$ and $N_{2,2}$ in PG satisfy the two conditions to become the source node of CGs. Hence, when the original VG is full-filled, two CGs are duplicated. If another "over-capacity" event is triggered again, 8 CGs will be created since there are 16 nodes satisfying the two conditions (i.e. $N_{1,3}$, $N_{1,4}$, $N_{1,5}$, $N_{1,6}$, $N_{2,3}$, $N_{2,4}$, $N_{2,5}$ and $N_{2,6}$ in PG, $N_{1,1}$, $N_{1,2}$, $N_{2,1}$ and $N_{2,2}$ in CG1, $N_{1,1}$, $N_{1,2}$, $N_{2,1}$ and $N_{2,2}$ in CG 2). Node-disjoint is assured since each source node of a CG belongs to a different root set in PG.

3.3 Member Joining to Virtual Graph

When a new member joins a multicast group, TLHIT assigns it to an appropriate vacant position in the VG according to end-to-end delays between the joining node and source nodes and ensures neighbor nodes are either in the same VG or in the CGs produced by the VG. But the ancestor of the vacant position is responsible for all its data transmission jobs. Hence, inappropriate member join will cause unbalanced load. Considering the tradeoff between the end-to-end delay and load balance, we divide the member joining into two states called *the locating* and *the replacing states*. During the locating state, a member is assigned to a vacant position according to loading situation while during the replacing state each node periodically detects the network condition and adjusts its position in the VG to enhance overall system performance.

The member join algorithm, MemberJoin(v_i, G), is described below. Suppose the node v_i is joining the multicast group. Let G denotes the original VG or one of the CGs closest to v_i, and s is the closest source node in G.

```
0 Algorithm MemberJoin(v_i, G)
1 Input: v_i; //the new join node.
2 Input: G; //one of the VGs which is closest to v_i
3 If there is any vacant position in G, then {//Locating state
4     v_i joins G by replacing a specific vacant position;
5     return;
6 } //end Locating state
7 Else {//Replacing state
8     Calculates end-to-end delay D(v_i, s);
9     For each v_j in G do {
10         If D(v_j, s) < D(v_i, s) then {
11        v_i = v_j; // v_j replaces v_i and joins G;
12        } //end delay comparison
13    } //end for each V_j
14    find G' which is a CG of G closest to v_i;
15    MemberJoin(v_j, G'); //recursive
16 } //end Replacing state
```

As shown in the member join algorithm above, line 4 to line 6 refers to the joining procedure when v_i is in the locating state that it does not belong to any VG. This situation may happen either when the first time this node joins the multicast group or when another member node has shorter end-to-end latency to the source node than it. During locating state, the v_i searches for a root set in G with maximum number of vacant positions. Then, a member is assigned to the vacant position closest to the

source of the root set. As shown starting from line 7, when VG contains no vacant position, v_i enters into replacing state. During replacing state, v_i calculates the end-to-end delay D(v_i, s) and compares to D(v_j, s) of each node j in the root set of s. As shown in line 9 to line 13, if a node v_j having D(v_j, s) larger than D(v_i, s), v_j is replaced with v_i. Finally, v_j enters locating state and searches for another vacant position. Otherwise, as shown in lines 14 to 15, the node v_i joins G' which is a CG of G closest to it. Because the multicast group members may change as time goes by. Each node in TLHIT runs member join algorithm periodically to see whether there exist a better position or not. Thus, we can ensure each node is always in proper position in TLHIT.

4 Simulation Results

In this section, the metrics of evaluating the performance of THAG and TLHIT are described as follows:

- *Average received descriptions (ARD)*: ARD represents the average number of descriptions received by the nodes. Video quality improves as the number of received descriptions increases. Hence, ARD represents not only fault-tolerant but also QoS of a system.
- *Stretch*: stretch presents the average number of interior nodes from the source to each participating node in overlay multicast trees. The stretch shows propagation delay in TLHIT comparing to the unicast case.
- *Stress*: stress represents the number of descriptions a node needs to forward. This metric shows whether this system is load-balance or not.
- *Delay Distribution*: this metric shows the difference of propagation delay of each node.

In the simulations, both THAG and TLHIT construct two virtual graphs with 1750 nodes, respectively. A virtual graph is constructed according to G_6 and s trees are embedded where s varies from 2 to 6. The value of propagation delay between any pair of nodes is randomly generated ranging from 20ms to 120ms. We use the same topology in THAG and TLHIT.

4.1 Average Received Descriptions and Stretch

Figure 5 presents ARD v.s. the number of trees with different node failure probabilities 2%, 5%, and 10%. The simulation results represents that the nodes using TLHIT receives more descriptions than those using THAG and it means the fault-tolerant of TLHIT is better than THAG. Hence, the multicast group member in TLHIT gets a better video quality. Furthermore, only the first source node needs to execute the topology-generation algorithm in TLHIT. The member join algorithm of TLHIT is based on the same algorithm of THAG. Hence, the complexity of TLHIT and THAG system is the same, but the performance of TLHIT is better than THAG. Straight lines also indicate that the TLHIT are not affected by the number of trees. Because of the node-disjoint property, the TLHIT will not generate enormous number of losses when node failure occurs. And then we compare the stretch in TLHIT and THAG in cases of different number of nodes in the system. As shown in Figure 6, the stretch of TLHIT is smaller than that of THAG. The result means that the data delivery in TLHIT has shorter latency than that in THAG.

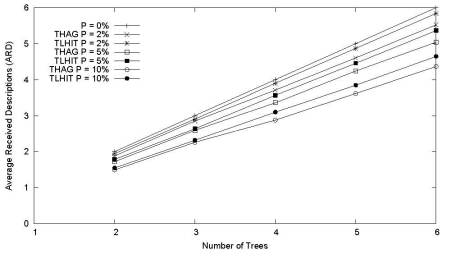

Fig. 5. Comparison of average received descriptions in THAG and TLHIT when group size = 1750

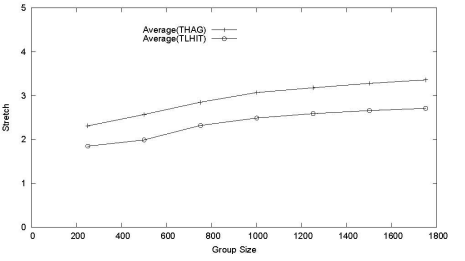

Fig. 6. Stretch versus group size, s = 6

4.2 Stress and Delay Distribution

As shown in Figure 7, 23% of the group members in THAG forward less than five descriptions which are nearly idle and 12% of the group members need to forward more than ten descriptions. It means that the duty of data forwarding for each group member is unfair in THAG. In TLHIT, 94% of the group members forward the same number of descriptions. The result indicates that the load-balance of TLHIT is assured.

In THAG and TLHIT systems, one description is transmitted to each group member through different number of interior nodes. So, each group member will experience different latency transmitting this description. The delay distribution represents the difference of propagation delays among all group members. As shown in Figure 8, the variance of propagation delay in THAG is much larger than in TLHIT. In TLHIT, the propagation delay is centralized from 200ms to 300ms. Only a few members' delay value is greater than 450ms. On the contrary, the maximum propagation delay is up to 550ms and the variance of the delay distribution is large in THAG. Figure 8 also shows the advantage of load-balance.

Fig. 7. Stress distributions of THAT and TLHIT when s = 6 and group size = 1750

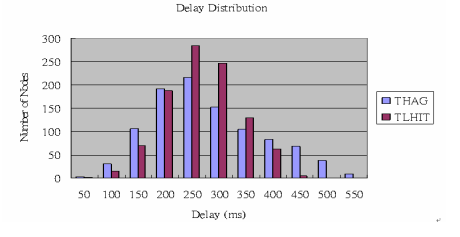

Fig. 8. Delay distribution in THAG and TLHIT, s = 6, group size = 1750

The simulation result of ARD, stretch, stress, and delay distribution show that the proposed TLHIT provides more reliable and efficient multicast in overlay networks.

5 Conclusions and Future Work

In this paper, path diversity using independent trees in overlay multicast is proposed to improve the performance of media streaming service. Two schemes to construct diverse paths for participating node are compared. One is THAG and the other is TLHIT. THAG makes all the multicast trees node-disjoint. In addition to the node-disjoint property, TLHIT builds multicast trees which are path-disjoint and load-balance to minimize the influence of failures. The reliability and efficiency of THAG and TLHIT are compared through several simulations. The average received descriptions (ARD) shows that each node has higher probability to receive more descriptions in TLHIT. The stretch and delay distribution show that each node experiences a shorter latency in TLHIT and the delay variance of each node is small. Moreover, the stress shows that the duty of each node is much more balanced in TLHIT. Hence, the simulation results indicate that TLHIT is a more reliable, efficient and load-balance scheme for multimedia streaming service.

In future, we intend to further improve the TLHIT scheme with the ability of detecting limits of participating nodes. A powerful node with higher bandwidth in the multicast group should undertake more data transmissions. Thus, we can avoid the bottleneck caused by weak nodes and this context-aware TLHIT provide optimal performance for video streaming.

Acknowledgement

The authors would like to thank the National Science Council in Taiwan R.O.C for supporting this research, which is part of the three projects numbered NSC 95-2221-E-168-029, NSC 94-2213-E-006-081 and NSC 95-2219-E-006-007.

References

[1] Tian, R., Zhang, Q., Xiang, Z., Xiong, Y., Li, X., Zhu, W.: Robust and Efficient Path Diversity in Application-Layer Multicast for Video Streaming. IEEE Transactions on Circuits and Systems for Video Technology 15(8), 961–972 (2005)

[2] Padmanabhan, V.N., Wang, H.J., Chou, P.A., Sripanidkulchai, K.: Distributing streaming media content using cooperative networking. In: Proc. ACM NOSSDAV, Miami Beach, FL, pp. 177–186 (May 2002)

[3] Castro, M., Druschel, P., Kermarrec, A.-M., Nandi, A., Rowstron, A., Singh, A.: SplitStream: High-bandwidth content distribution in a cooperative environment. In: Kaashoek, M.F., Stoica, I. (eds.) IPTPS 2003. LNCS, vol. 2735, Springer, Heidelberg (2003)

[4] Banerjee, S., Bhattacharjee, B., Kommareddy, C.: Scalable application-layer multicast. In: Proc. ACM SIGCOMM, pp. 205–217 (August 2002)

[5] Chu, Y., Rao, S., Seshan, S., Zhang, H.: Enabling conferencing applications on the internet using an overlay multicast architecture. In: Proc. ACM SIGCOMM, pp. 55–67 (August 2001)

[6] Liang, Y.J., Steinbach, E.G., Girod, B.: Real-time voice communication over the Internet using packet path diversity. In: Proc. ACM Multimedia 2001, (September/October 2001) pp. 431–440 (2001)
[7] Aopstolopoulos, J.: Reliable Video Communication over Lossy Packet Networks using Multiple State Encoding and Path Diversity. Visual Communications and Image Processing, 392–409 (January 2001)
[8] Andersen, D., Balakrishnan, H., Kaashoek, F., Morris, R.: Resilient Overlay Networks. In: Proc. 18th ACM Symposium on Operating Systems Principles, Banff Canada, pp. 131–145 (October 2001)
[9] Chu, Y., Rao, S., Seshan, S., Zhang, H.: A case for end system multicast. In: Proceedings of ACM SIGMETRICS, pp. 1–12 (June 2000)
[10] Goyal, V.K.: Multiple description coding: Compression meets the network. Signal Processing Magazine 18(5), 74–93 (2001)
[11] Bolot, J.: End-to-End Packet Delay and Loss Behavior in the Internet. In: Proceedings of ACM SIGCOMM, pp. 289–298 (September 1993)