



**HAL**  
open science

# Performance Evaluation of Application-Aware Cost Function for Scalable Video Multicast Streaming Services on Overlay Networks

Tien Anh Le, Hang Nguyen

► **To cite this version:**

Tien Anh Le, Hang Nguyen. Performance Evaluation of Application-Aware Cost Function for Scalable Video Multicast Streaming Services on Overlay Networks. PIMRC2012-Track 4:Services, Applications, and Business (PIMRC2012-S

A), citeulike-article-id = 10721917, Sep 2012, Sydney, Australia. pp.1-6. hal-00703009

**HAL Id: hal-00703009**

**<https://hal.science/hal-00703009>**

Submitted on 31 May 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Performance evaluation of application-aware cost function for scalable video multicast streaming services on overlay networks

Tien Anh Le, Hang Nguyen

**Abstract**—Real time video multicast streaming services on overlay networks need to build a multicast tree for distributing the video contents to all clients. Link cost is very important in building this multicast tree. A good cost function can provide information for media routing algorithms to find the best way to distribute the media on overlay networks. In this research work, a multi-variable cost function is proposed. This cost function can calculate links' costs based on both network resources and application's requirements. The new cost function is then applied to build an overlay network for scalable video multicast streaming services of up to more than one thousand peers. An evaluation platform is proposed to validate the advanced performance of the newly proposed cost function in overlay network. The results have shown a better adaptability of the new cost function in the scalable video multicast streaming service on overlay networks.

**Index Terms**—overlay, application layer multicast routing; cost function; scalable video streaming;

## I. INTRODUCTION

The Internet was originally built for unicast or one-to-one applications. Nowadays, it has to serve a large number of multimedia services such as IPTV, multimedia conference... These types of multicast services put a big load on the unicast infrastructure of the Internet. IP-Multicast[1] is the first attempt to solve this problem. However, many deploying problems are still preventing IP-Multicast from being supported worldwide[2]. An alternative solution is Application Level Multicast(ALM). The key concept of ALM is the implementation of multicasting functionality as an application service instead of a network service. It has excellent advantages over IP-Multicast: easier and possibly immediate deployment over the Internet without any modification of the current infrastructure and adaptable to a specific application. In a tree-push ALM, a data distribution tree is built first, then the data is actively distributed from the source node to intermediate peers until reaching all peers in the multicast tree[3]. In order to build an ALM distribution tree, we must have costs of all available end-to-end links. Those costs can only be calculated by using a cost function.

Real time video multicast streaming services on overlay networks need to build a multicast tree for distributing the video contents to all clients. Link cost is very important in building this multicast tree. A good cost function can

Authors are with the Department of Wireless Networks and Multimedia Services, Telecom Sud Paris, 91011, France. Phone: +33 (0)160 76 66 63, Fax: +33 (0)160 76 45 78, E-mail: {Tien.Anh.Le, Hang.Nguyen}@it-sudparis.eu. This work was supported in part by POSEIDON, a French national project on multimedia services over 4G networks.

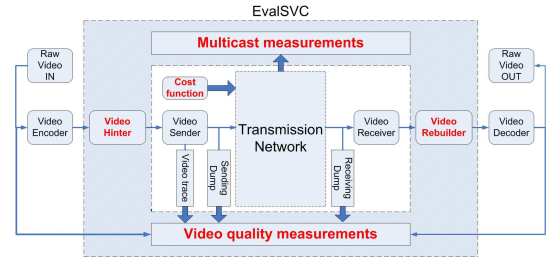


Fig. 1. EvalSVC and the evaluation of SVC-based services on overlay network constructed by the newly proposed multi-variable cost function.

provide information for media routing algorithms to find the best way to distribute the media on overlay networks. In this research work, a multi-variable cost function is proposed. This cost function can calculate links' costs based on both network resources and application's requirements. The new cost function is then applied to build an overlay network for scalable video multicast streaming services of up to more than one thousand peers. An evaluation platform is proposed to validate the advanced performance of the newly proposed cost function in overlay network. The main contribution of this research is to design a new multi-variable cost function of the end-to-end delay and bandwidth taking into account advantages of application layer links. The research work is an extension to the original work proposed in[4] and[5]. After having proposed the new multi-variable cost function, it is necessary to evaluate its performance. Since it has been proved that SVC-content can resist better in the overlay network[6][7], therefore an evaluation platform of the newly proposed multi-variable cost function with SVC content is highly required. The general diagram of the evaluation process is demonstrated in Fig.1. We have built an extended evaluation platform from our original evaluation platform for scalable video transmission (EvalSVC[8]). This platform provides measurement data of a real-time scalable video streaming service on a simulated overlay network with up to more than one thousand peers. The collected measurements have shown the adaptability of the new cost function in such a fast changing conditions as of overlay networks. The rest of the paper is organized as follows. Section II provides a survey on the state-of-the-art cost functions and analyze their limitations. The proposed multi-variable cost function will be described and derived in section III. By analyzing intensive simulation result in section IV, we will compare the new multi-variable cost function with conven-

tional ones on overlay networks to show its advantages. Finally, we conclude and give out some possible future works in Section V.

## II. CONVENTIONAL COST FUNCTIONS

Conventional cost functions are either empirical or heuristic. Among all available cost functions for ALM routing that we have found, neither of them has a mathematical derivation nor a clear citation. In most of the ALM routing algorithms, the state of the network, on which the routing algorithm is presented, readily associates some costs with each link. Thus they do not address how the link cost function should be defined so as to efficiently distribute allocated resources over the network[9]. Also in[9], several kinds of cost functions have been investigated.

In[10], the cost function decreases with the delay, convex, assigns infinitely high cost when the required delay guarantee approaches zero, and a fixed minimal link usage cost, even if no guarantee is required. A constant determines how fast the cost grows for low delays and another constant is used as the scaling constant. Although delay is used as a sample, any other QoS parameter such as bandwidth, jitter, packet loss can be used, but not simultaneously. The heuristics cost functions is only single-variable, therefore it cannot consider other parameters simultaneously.

Another cost function considering several QoS parameters has been used in[11], again, without any mathematical proof.

A scaling factor allows us to modulate the relationship between parameters even further, although it is still unclear exactly how bandwidth, buffer, and delay units could be added together[11]. This again raises need for a new cost function.

## III. PROPOSED MULTI-VARIABLE COST FUNCTION

Assuming that we have an overlay with application peers and end-to-end-links, in order to form a tree for data delivery, we need costs of all those end-to-end links. These costs must be calculated by a cost function. To take into account several QoS parameters simultaneously, the cost function must be a multi-variable function. QoS parameters can be a bandwidth-type (meaning that the requested bandwidth is always smaller than or equal to the maximum available bandwidth) or delay-type (meaning that the requested delay is always greater than or equal to the minimum available delay). On each end-to-end link, we have to consider variable requirements from applications running on the P2P-based overlay. For example, an application can be a scalable video service with different video coding layers or it can be a multimedia flux comprising of video, audio, text, data sub-streams, each has different bandwidth and delay requirements. Those requirements are changed frequently by the application. We have to also consider the maximum available resources of the underlay. For example, if an end-to-end link is built upon 3 physical links, each has its own available bandwidth and delay. Then the maximum available bandwidth of the end-to-end link equals to the minimum available bandwidth (bottle-

neck) of all 3 physical links, the minimum guaranteed delay of the end-to-end link equals to the sum of all delays on the 3 physical links.

### A. Problem formation

**Problem:** Find a multi-variable cost function which can simultaneously consider varied bandwidth and delay requests from the application and maximum guaranteed resources from the underlay network. The cost function must be able to assign increasingly higher costs for nearly-saturated end-to-end links to prevent congestion.

### B. Single variable cost function

Assume we have on the end-to-end link  $i$ : A total available bandwidth of  $\kappa_w$ , and a requested bandwidth of  $x_w$ , we must find the bandwidth-type cost function:  $f(x_w)$ . Since  $\kappa_w$  is the maximum available bandwidth when using all available resources on link  $i$ , so  $0 \leq x_w \leq \kappa_w$ . With time, according to the application's requirements,  $x_w$  may be varied by an amount of  $\Delta x_w$  causing the cost to have the current value of  $f(x_w + \Delta x_w)$ , so this current value of the cost function depends on:

- The previous cost:  $f(x_w)$ ,
- The increment of cost which is proportional to:
  - The previous cost:  $f(x_w)$ ,
  - The ratio between the increment of requested bandwidth and the total requested bandwidth:  $\frac{\Delta x_w}{x_w + \Delta x_w}$ ,
- The decrement of cost which is proportional to:
  - The ratio between the decrement of the remaining available bandwidth and the maximum available bandwidth  $\frac{(\kappa_w - x_w - \Delta x_w)}{\kappa_w}$ .

Finally, we have:

$$f(x_w + \Delta x_w) = f(x_w) \cdot \left[ 1 + \frac{\frac{\Delta x_w}{x_w + \Delta x_w}}{\frac{(\kappa_w - x_w - \Delta x_w)}{\kappa_w}} \right] \quad (1)$$

From (1) we have:

$$\Leftrightarrow f'(x_w) = f(x_w) \cdot \frac{\kappa_w}{x_w(\kappa_w - x_w)} \quad (2)$$

Replacing  $f(x_w)$  by  $y$  and  $f'(x_w)$  by  $\frac{dy}{dx_w}$ ; from (2) we have an ordinary differential equation:

$$\frac{dy}{dx_w} = y \frac{\kappa_w}{x_w(\kappa_w - x_w)} \quad (3)$$

Solve the ordinary differential equation (3), we find the bandwidth-type cost function:

$$y = \frac{\Phi \cdot x_w}{(\kappa_w - x_w)} \quad (4)$$

We can see that, the required delay parameter ( $x_d$ ) has a reversed characteristic against the required bandwidth

parameter  $(x_w)$ . So by replacing  $\dot{x}_d = \frac{1}{x_d}$ ,  $\dot{\kappa}_d = \frac{1}{\kappa_d}$ , and  $d\dot{x}_d = d(\frac{1}{x_d}) = -\frac{dx_d}{x_d^2}$  into (3) we have:

$$\frac{dy}{dx_d} = y \frac{1}{\kappa_d - x_d} \quad (5)$$

Equation (5) is the ordinary differential equation to derive the delay-type cost function. From (5), we have:

$$y = \frac{\Psi \cdot \kappa_d}{x_d - \kappa_d}$$

### C. Derivation of the multi-variable cost function

We now try to derive the **bandwidth-delay cost function**  $u(x_w, x_d)$  considering two independent QoS parameters: bandwidth  $(x_w)$  and delay  $(x_d)$  at the same time. From (3) and (5), we have:

$$\frac{x_w(\kappa_w - x_w)}{\kappa_w} u_{x_w} + (\kappa_d - x_d) u_{x_d} = u \quad (6)$$

In which  $u_{x_w} = \frac{\partial u}{\partial x_w}$ , and  $u_{x_d} = \frac{\partial u}{\partial x_d}$ .

Equation (6) is a *quasi linear first order partial differential equation*, we will solve it to obtain our bandwidth-delay cost function.

Let  $x_w = x_w(s)$ ,  $x_d = x_d(s)$ ,  $u = u(x_w(s), x_d(s))$ , then:

$$\frac{\partial x_w}{\partial s} \cdot u_{x_w} + \frac{\partial x_d}{\partial s} \cdot u_{x_d} = \frac{\partial u}{\partial s} \quad (7)$$

Compare (6) and (7), we have:

$$\left\{ \begin{array}{l} \frac{\partial x_w}{\partial s} = \frac{x_w(\kappa_w - x_w)}{\kappa_w} \\ \frac{\partial x_d}{\partial s} = \kappa_d - x_d \\ \frac{\partial u}{\partial s} = u \end{array} \right. \quad (8)$$

A *constant of integration* is obtained by eliminating  $s$  from two or more equations and integrating out. Such integration generates an arbitrary integration constant, which may be viewed as a function of all the variables, but it is constant with respect to  $s$ . Let  $\phi(x_w, x_d, u)$  be a constant of integration, since it is constant with respect to  $s$ , or  $\frac{d\phi}{ds} = 0$ , we can write:

$$\frac{\partial \phi}{\partial x_w} \cdot \frac{x_w(\kappa_w - x_w)}{\kappa_w} + \frac{\partial \phi}{\partial x_d} \cdot (\kappa_d - x_d) + \frac{\partial \phi}{\partial u} \cdot u = 0 \quad (9)$$

Equation (9) is the *orthogonality property* of the vector  $(x_w, x_d)$ , we can use it to check whether  $\phi$  has been obtained correctly.

In order to solve (6) we have to find two constants of integration from (8).

From (8), we have:

$$\frac{dx_w}{ds} = \frac{x_w(\kappa_w - x_w)}{\kappa_w} \Leftrightarrow \frac{du}{u} = \frac{\kappa_w \cdot dx_w}{x_w(\kappa_w - x_w)} \quad (10)$$

Since (10) and (3) have an identical form, we can use Lemma 1 to achieve the first constant of integration (11):

$$u = \frac{\Phi \cdot x_w}{(\kappa_w - x_w)} \Leftrightarrow \Phi = \frac{(\kappa_w - x_w)u}{x_w} \quad (11)$$

Similarly, using (8) and (6), we can find the second constant of integration having the form of:

$$\Psi = \frac{(x_d - \kappa_d)u}{\kappa_d} \quad (12)$$

### D. The general solution:

The equation  $\Phi(x_w, x_d, u) = \text{constant}$ , describes a relationship among  $x_w$ ,  $x_d$ ,  $u$  such as shown in (8). Notice that if  $\Phi(x_w, x_d, u)$  is a constant, then  $\mathcal{G}(\Phi)$  is also a constant (in which  $\mathcal{G}(\cdot)$  is any arbitrary function). Similarly, if  $\Psi(x_w, x_d, u)$  is a constant, then  $\mathcal{H}(\Psi)$  is also a constant (in which  $\mathcal{H}(\cdot)$  is any arbitrary function). We can set these two constants equal so that:

$$\mathcal{G}(\Phi) = \mathcal{H}(\Psi) \quad (13)$$

This provides a more general expression in  $x_w$ ,  $x_d$ ,  $u$  that solves the partial differential equation (6). The two arbitrary functions in (13) may be merged into one by letting  $\mathcal{F}(\cdot) = \mathcal{G}^1(\mathcal{H}(\cdot))$ , then:

$$\Phi = \mathcal{F}(\Psi) \quad (14)$$

(14) is the general solution of the partial differential equation (6). From (11), (12), (14), we have:

$$\frac{(\kappa_w - x_w)u}{x_w} = \mathcal{F}\left(\frac{(x_d - \kappa_d)u}{\kappa_d}\right) \quad (15)$$

#### D.1 Fitting boundary conditions to the general solution:

Equation (15) provides us a general solution comprising of a family of arbitrary functions. We need to fix to a certain bandwidth-delay cost function by assigning boundary conditions to this general solution. From the natural characteristics of 2 independent QoS parameters: *requested bandwidth*  $(x_w)$  and *requested delay*  $(x_d)$ , and their partial cost functions (4) and (6), we have these boundary conditions:

$$\left\{ \begin{array}{l} \frac{x_w}{\kappa_w - x_w} = t^3 \\ \frac{x_d - \kappa_d}{u} = t \end{array} \right. \quad (16)$$

Replacing (16) into (15) we have:

$$\mathcal{F}\left(\frac{t^2}{t}\right) = \frac{t^2}{t^3} \Leftrightarrow \mathcal{F}(t) = \frac{1}{t} \quad (17)$$

From (17) and (15), we have:

$$\mathcal{F}\left(\frac{(x_d - \kappa_d) \cdot u}{\kappa_d}\right) = \frac{\kappa_d}{(x_d - \kappa_d) \cdot u} = \frac{(\kappa_w - x_w)u}{x_w} \quad (18)$$

The specific solution of (6) is therefore:

$$u(x_w, x_d) = \sqrt{\frac{x_w}{\kappa_w - x_w} \cdot \frac{\kappa_d}{x_d - \kappa_d}} \quad (19)$$

Recursively, we can see that, the specific multi-variable cost function equals to the average multiplication of all partial cost functions:

$$u(x_1, x_2, \dots, x_n) = \sqrt[n]{\prod_{i=1}^n f_i(x_i)} \quad (20)$$

In general, we can build a cost function for as many variables as possible given separated partial cost functions. However, while a multi-variable cost function can consider many QoS parameters at the same time, it should be noticed that the multi-variable cost function does not always give a better result than the single-variable cost function. For example, the cost function with bandwidth, delay, and packet-loss can build a better multicast tree if many peers are using wireless access network with a high packet loss rate to join the multicast tree but when most of the peers are using a wired access network with a low packet loss rate, then that three-variable cost function may build a worse multicast tree than the two-variable cost function of only bandwidth and delay. Therefore, a N-variable cost function with  $N \geq 3$  should be designed and applied with care.

#### IV. EVALUATION OF THE PROPOSED COST FUNCTION ON OVERLAY NETWORKS

##### A. Platform's architecture and settings

To compare the performance of the new cost function with a popular cost function, the evaluation process is set up as demonstrated in Fig.2. We set up an OverSim[12] simulation scenario based on NICE. The main goal of the simulation is to show the advanced performance of a representative ALM algorithm (e.g. NICE) when applying our new cost function. The simulation may only show the advanced performance for NICE but the simulation methodology (which is protocol-independence) can be generalized to any ALM algorithm using a different cost function other than our new cost function for building the data delivery tree. The simulation plan will build an overlay of a varied number of peers (e.g., varied group sizes of 16, 32, 64, 128, 256, 512, and 1024) running on an underlay network topology generated by GT-ITM[13]. Each topology was a two-level hierarchical transit-stub topology, containing 1250 nodes and about 6000 physical links[14]. Each physical link will have random values of delay, bandwidth, and PER (Packet Error Ratio). We will use the simulation plan described in[15] for comparison and confirmation purposes. We use similar performance metrics commonly applied by all ALM algorithms to validate the advanced performance of the newly-proposed cost function. The  $x_w$  parameter can be obtained by investigating the sending and receiving dump files of a Scalable Video Coding unicast.

NICE only uses a delay-type cost function to build and to maintain its ALM tree (with a clustering, layering structure). By sending and receiving periodic heartbeat messages containing delays between nodes within a cluster, peers will decide whether it should elect a new cluster-leader. Changing cluster-leaders provokes changing and

TABLE I  
SIMULATION PARAMETERS OF THE SVC TRANSMISSION ON OVERLAY NETWORK CONSTRUCTED FROM MULTI-VARIABLE AND CONVENTIONAL COST FUNCTION.

Parameters	Values
Purpose	Evaluation of the new multi-variable cost function for Video Multicast service
Encoding	SNR SVC
Video size	CIF
Transmission network	Overlay network
Service	Application Layer Multicast of SVC video
Network simulation tool	Oversim
Number of peer	1-1024 peers
Underlay network	Internet topology generated by GT-ITM
Cost functions	<ul style="list-style-type: none"> <li>• New multi-variable cost function,</li> <li>• NICE's popular cost function</li> </ul>
Overlay measurements	Average link stress, average end-to-end delay

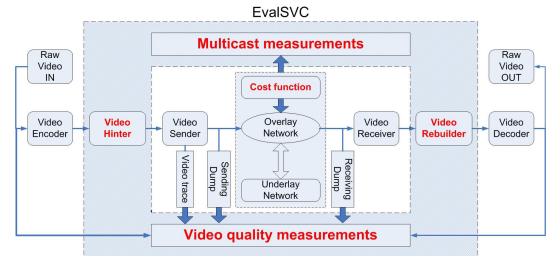


Fig. 2. EvalSVC and the performance evaluation of SVC transmission on overlay network constructed from multi-variable cost function.

rebuilding the entire NICE tree. In its original paper[16], authors of NICE implemented the delay-type cost function simply by using an end-to-end delay parameter. We now want to apply our new bandwidth-delay cost function obtained in (19).

Costs of all end-to-end links will be calculated and NICE will use them instead of the conventional delay cost to run their algorithm on. We will compare performances of two cases mainly by using two metrics: **average link stress**, and **average link stretch**[17]. The average link stress metric is defined by the mean value of identical packets sent by a protocol over each underlay link. To calculate the average link stress of the network, instead of standing on each link and counting identical packets, we let the nodes (peers/routers) count the link stress of all their links, and then take a half of the sum. The reason for doing so is

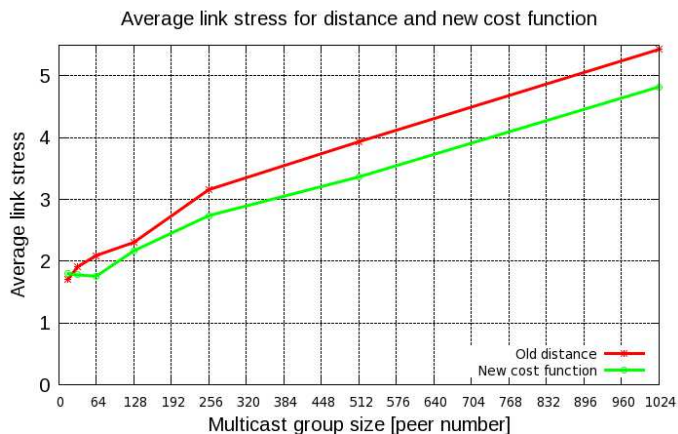


Fig. 3. Average link stress comparison for the NICE data-plan using the old and new cost functions. Transmitting data is obtained from a real SVC transmission session.

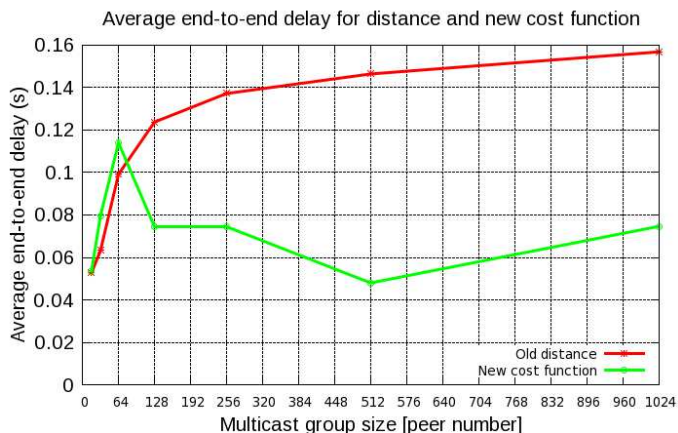


Fig. 4. Average end-to-end delay performance.

because in OverSim, it is easier to control nodes than links, meanwhile any physical link is always formed just by 2 nodes. The average link stretch is the ratio of average path length of the members of a protocol to the average path length of the members in the multi-unicast protocol. In our implementation, we just concentrate on the numerator: the **average path length** (mean value of actual hops) that a data packet must go through from source to destination. For each packet received at an overlay peer, we will take its Time-To-Live information which is actually the hop-count value that it had to go through. Note that we just need to count the path length of packets routed by the ALM protocol, so we take the calculation at the overlay layer, not at the underlay layer.

### B. Performance evaluation results

Figure 3 shows that the newly proposed cost function when applied by NICE can reduce the average link stress that a link has to take to a smaller value than the original NICE's distance function. Fig.4 shows that the average end-to-end delay when applying the new cost function is much smaller than the old distance function especially when the group size increases. Even when the number

of participants is 1024, the average end-to-end delay of the new cost function is just about 79 ms which is much smaller than the limitation value of 150 ms recommended by ITU-T for real-time communication services[18]. From the results we can see that, the new cost function can avoid multiple replication of packets on access links and so reduce the average link stress. Even though a packet may have to go through more physical hops in order to reach its destination, the new cost function can still guarantee a half-smaller average end-to-end delay than the conventional distance function. It should be noticed from Fig.4 that, when the group size is smaller than 64, the average end-to-end delays when applying new and conventional cost functions are similar since there are not many better options for NICE to choose from. However, when the group size is large, the new cost function can give out more routes for NICE to build its media distribution tree resulting in a much better average end-to-end delay than the conventional cost function.

## V. CONCLUSION AND FUTURE WORKS

In this research, a new multi-variable cost function has been proposed. The mathematical derivation process has also been described in details so that ones can apply it to obtain other multi-variable cost functions according to their specific requirements. The newly found cost function has considered dynamic requirements of the application and the underlay network. An evaluation platform and a scalable video streaming service have been built to validate the newly proposed cost function. Evaluation results show that the new cost function can greatly reduce the average link stress and average end-to-end delay (two very important metrics in multimedia services) for the scalable video multicast streaming service. For future works, a new ALM can be designed based on the newly proposed cost function. The result can be further applied to improve the performance of any ALM algorithms who are using conventional cost functions to build their data delivery tree.

## VI. ACKNOWLEDGMENT

The research work is supported in part by Poseidon, a French national research project on the evaluation of multimedia services on 4G networks. The authors are grateful to Quang Hoang Nguyen for his contributions in building the simulation environment for EvalSVC.

## REFERENCES

- [1] S. E. Deering and D. R. Cheriton, "Multicast routing in datagram internetworks and extended LANs," *ACM Transactions on Computer Systems (TOCS)*, vol. 8, no. 2, pp. 85–110, 1990.
- [2] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for the IP multicast service and architecture," *IEEE Network*, vol. 14, no. 1, pp. 78–88, 2000.
- [3] M. Hosseini, D. T. Ahmed, S. Shirmohammadi, and N. D. Georganas, "A survey of application-layer multicast protocols," *IEEE Communications Surveys & Tutorials*, vol. 9, no. 3, pp. 58–74, 2007.
- [4] Tien A. Le, Hang Nguyen, and Hongguang Zhang, "Multi-variable cost function for Application Layer Multicast routing," in *IEEE Globecom 2010 - Communications Software, Services and Multimedia Applications Symposium (GC10 - CSSMA)*, Miami, Florida, USA, Dec. 2010.

- [5] Tien A. Le, Hang Nguyen, and Quang H. Nguyen, "Toward building an efficient Application Layer Multicast tree," in *IEEE-RIVF 2010 International Conference on Computing and Telecommunication Technologies*, 2010.
- [6] C. Luo, W. Wang, J. Tang, J. Sun, and J. Li, "A Multiparty Videoconferencing System Over an Application-Level Multicast Protocol," *IEEE Transactions on Multimedia*, vol. 9, no. 8, pp. 1621–1632, 2007.
- [7] Tien A. Le, Hang Nguyen, and Hongguang Zhang, "Scalable Video transmission on overlay networks," in *Second International Conferences on Advances in Multimedia*, Athens, Greece, June 2010, pp. 180–184.
- [8] Tien A. Le, Hang Nguyen, and Hongguang Zhang, "EvalSVC - An evaluation platform for scalable video coding transmission," in *Consumer Electronics (ISCE), 2010 IEEE 14th International Symposium on*, 2010, pp. 1–6.
- [9] I. Matta and L. Guo, "On routing real-time multicast connections," in *IEEE International Symposium on Computers and Communications, 1999. Proceedings*, 1999, pp. 65–71.
- [10] D. H. Lorenz, A. Orda, and D. Raz, "Optimal partition of QoS requirements for many-to-many connections," in *IEEE INFOCOM*. Citeseer, 2003, vol. 3, pp. 1670–1679.
- [11] R. Widjono, "The design and evaluation of routing algorithms for real-time channels," *International Computer Science Institute, TR-94-024*, 1994.
- [12] I. Baumgart, B. Heep, and S. Krause, "OverSim: A flexible overlay network simulation framework," in *Proceedings of 10th IEEE Global Internet Symposium (GI'07) in conjunction with IEEE INFOCOM*. Citeseer, 2007, vol. 7, pp. 79–84.
- [13] K. Calvert and E. Zegura, "GT internetwork topology models (GT-ITM)," 1997.
- [14] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *Proceedings IEEE INFOCOM'96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation*, 1996, vol. 2.
- [15] D. Constantinescu, *Overlay multicast networks: elements, architectures and performance*, Department of Telecommunication Systems, School of Engineering, Blekinge Institute of Technology.
- [16] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," in *Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2002, p. 217.
- [17] D. Constantinescu and A. Popescu, "Implementation of Application Layer Multicast in OverSim," in *4th Euro-FGI Workshop on "New Trends in Modelling, Quantitative Methods and Measurements"*. Citeseer.
- [18] R. Itu-T and I. Recommend, "G. 114," *One-way transmission time*, vol. 18, 2000.