

Toward building an efficient Application Layer Multicast tree

Tien Anh Le, Hang Nguyen, Quang Hoang Nguyen

Abstract—Link cost is very important in media distribution trees. 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, a bandwidth-type cost function is proposed. The proposed cost function can calculate links' costs based on both network resources and application's requirements. It can help Application Layer Multicast (ALM) routing algorithms to avoid congestion before building media distributing trees. The derivation process has also been explained in details so that it can be further applied in other conditions to build other cost functions suitable for different requirements. The newly proposed cost function will then be applied in a popular Application Layer Multicast (NICE) to replace its old cost function. Possible modifications to NICE's algorithm will be discussed. Intensive simulation scenarios have also been carried out to validate the advanced performance of the newly proposed cost function in comparison with conventional functions. The simulation scenario has also been developed to show the adaptation of the cost function under real conditions when some peers use the WiMax access network to join the multicast group.

Index Terms—application layer multicast routing; cost function; resource allocation; traffic control; end-to-end QoS routing;

I. INTRODUCTION

Multicast is the essential routing mechanism required by one-to-many and many-to-many services. If unicast can compete or even replace multicast in low data-rate services (e.g. short message service, instance message...), multicast is so far the best choice for multimedia services such as video streaming, IPTV, multi-point video conferencing. IP Multicast[1] is the most efficient multicast mechanism to deliver the data over each link of the network only once, creating copies only when the links to the multiple destinations split. However, IP Multicast can only be deployed within a private network or on a network which can be fully managed by the service provider. The deployment of IP Multicast over the Internet has been facing many technical and business problems[2] which are preventing it from being used universally.

Attempts have been made to overcome these problems. Explicit Multi-Unicast (XCAST)[3] is an alternate multicast strategy to IP multicast that provides reception addresses of all destinations within each packet. As such, since the IP packet size is limited in general, XCAST cannot be used for multicast groups of large number of destinations. Tunnels and bridges can be made to connect IP Multicast islands with each other using unicast or Application Layer Multicast (ALM) as proposed in Island Multicast [4]. Between them, the method using ALM is more favorable since it can provide a more efficient mechanism connecting IP Multicast islands.

Application Layer Multicast can be used as the bridges between IP Multicast islands but it is also well-known for its

capability of being deployed as a stand-alone solution for multicasting service over the Internet. The key concept of ALM is the implementation of multicasting functionality as an application service instead of a network service. Although there are some drawbacks such as multiple copies of the same packet on the same link as well as typically constructing non-optimal trees, it has an excellent advantage that IP-Multicast cannot have: easier and possibly immediate deployment over the Internet, ability to adapt to a specific application.

Tree-push is a common approach for data delivery in Application Layer Multicast algorithms, especially when the multimedia quality is concerned[5]. In this approach, before the data distribution can take place, a media distribution tree must be built from all participating peers. The construction algorithm of the media distribution tree is based on the costs among participating peers. These costs are calculated by using a certain cost function. Eventually, the efficiency of the media distribution algorithm will mainly depend on the cost function being used.

The main contribution of this research is to propose a cost function considering requirements from both the multimedia service and available resource from the underlay network for building a more efficient ALM media distribution tree. The derivation process will also be explained in details so that it can be further applied in other conditions to build other cost functions suitable for different requirements. The implementation process of the new cost function in NICE, a very popular ALM algorithm, will also be studied to show its feasibility and its advance in performance. Intensive simulation scenarios in which a maximum of more than one thousand peers are served at the same time using conventional distance-type cost function and the newly proposed bandwidth-type cost function. The simulation scenario is also extended to cover the case when some peers are joining the multicast session from the WiMax access network. This scenario will show how well the newly proposed cost function can adapt with the real network conditions.

II. CONVENTIONAL COST FUNCTION IN NICE

NICE[6] is a popular ALM protocol, specially designed for large number of receiver sets. The protocol arranges peers into a layering hierarchy. The basic operation of the protocol is to create and maintain that hierarchy. The hierarchy is created by assigning peers into different layers. Peers in each layer are partitioned into a set of clusters whose sizes are from k to $3k-1$ (with k is a chosen constant number of peers who have the nearest "distance" to each other). Each peer will have a maximum distance (M_d) to all other peers within that cluster. The leader (or the

center) of a cluster will be elected as the one who has the minimum M_d . Firstly, all peers will join clusters of the first layer. Then leaders from all clusters of the first layer will form clusters on the second layer. There are at most $\log_k(N)$ layers with the highest layer has only single member. The data distribution in NICE is a source-specific tree. If a peer wants to multicast data, it will unicast that data to the leader of its cluster. That leader will then multicast the data to all members of all clusters that it is joining including leaders on upper layers. These leaders will then multicast the data to the rest.

NICE is a very popular sample for a tree-pushed multicast algorithm. It depends heavily on "distances" which are actually calculated by a delay-type cost function. The delay is measured by a very simple end-to-end latency which is obtained by using a simple *ping* between each pair of peers. Other conventional cost functions can also be applied to calculate the distances among peers in NICE. In [7][8][9], some cost functions have been proposed. However, all of them only consider the network resource when calculating the cost, none of them consider requirements from the multimedia applications which are also varied greatly during the communication session. In [10] and [11], conventional cost functions have started to consider application's requirements. However, there was neither mathematical derivation nor theoretical analysis for these cost functions so one may easily get loss when trying to apply this cost function in other conditions.

III. PROPOSED BANDWIDTH-TYPE 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. On each end-to-end link, we have to consider variable requirements from applications running on the NICE ALM. 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. Then the maximum available bandwidth of the end-to-end link equals to the minimum available bandwidth (bottleneck) of all 3 physical links.

Assume we have on the end-to-end *link i*: A total available bandwidth of κ_w , and a requested bandwidth of x_w , we must find the bandwidth-type cost function: $f(x_w)$. Since κ_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 Δx_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:

$$\begin{aligned} & \lim_{\Delta x_w \rightarrow 0} \frac{f(x_w + \Delta x_w) - f(x_w)}{\Delta x_w} \\ &= \lim_{\Delta x_w \rightarrow 0} f(x_w) \frac{\kappa_w}{(x_w + \Delta x_w)} \cdot \frac{1}{(\kappa_w - x_w - \Delta x_w)} \\ &\Leftrightarrow f'(x_w) = f(x_w) \cdot \frac{\kappa_w}{x_w(\kappa_w - x_w)} \end{aligned} \quad (2)$$

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:

$$\begin{aligned} \frac{dy}{y} &= \frac{\kappa_w}{x_w(\kappa_w - x_w)} dx_w \\ \Leftrightarrow \ln(y) &= \ln(x_w) - \ln(\kappa_w - x_w) + c \\ \Leftrightarrow y &= \frac{\Phi \cdot x_w}{(\kappa_w - x_w)} \end{aligned} \quad (4)$$

IV. SIMULATION METHODOLOGY AND RESULTS

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

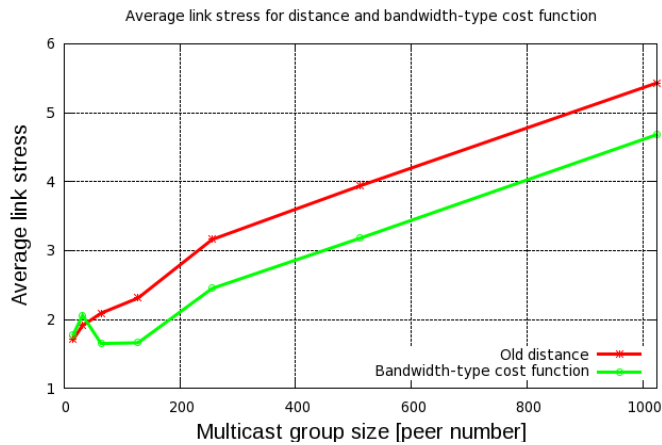


Fig. 1. Average link stress comparison for the NICE data-plan using the old and newly proposed bandwidth-type cost functions. Transmitting data is obtained from a real SVC transmission session.

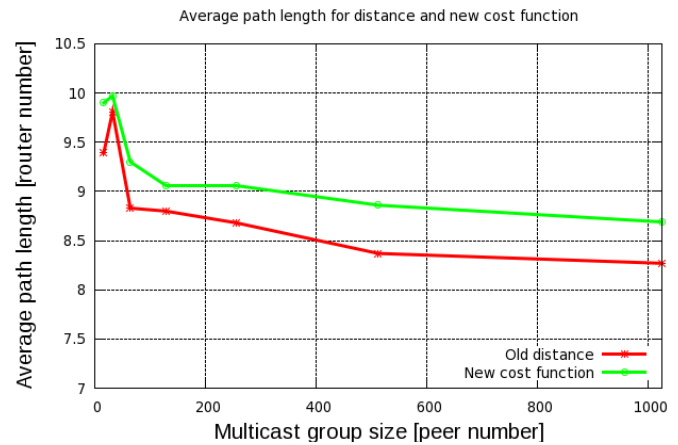


Fig. 2. Average path length (physical router counting) comparison for the NICE data-plan using the old and newly proposed bandwidth-type cost functions. Transmitting data is obtained from a real SVC transmission session..

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 rebuilding the entire NICE tree. In its original paper[6], 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-type cost function. 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**[16]. 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 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.

Fig.1 shows that the newly proposed cost function when

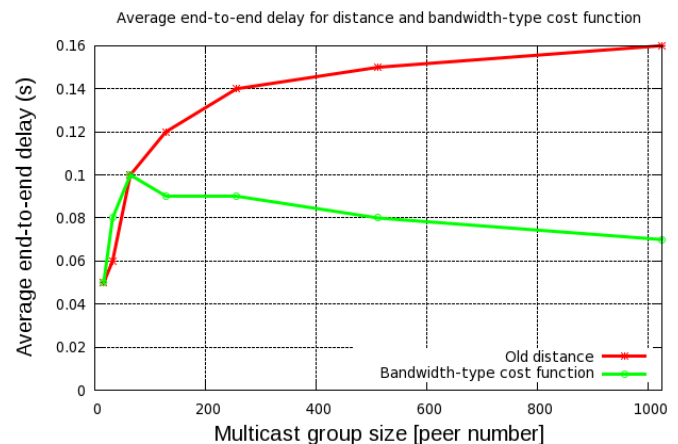


Fig. 3. Average end-to-end delay performance.

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. The result in Fig.2 means that, in average, a packet has to go over a longer physical route when applying the newly proposed cost function. However, Fig.3 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[17]. 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.3 that, when the group size is smaller than 64, the average end-to-end delays when applying new and conven-

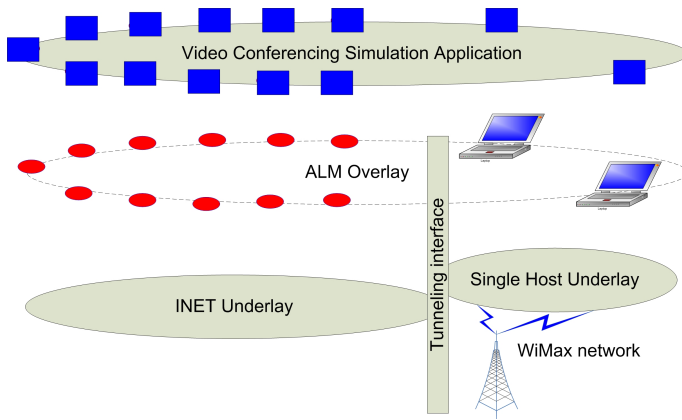


Fig. 4. Extended simulation scenario with 2 real Wimax terminals.

tional 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. EXTENDED SIMULATION SCENARIO

To see the adaptation of the newly proposed cost function in the real network conditions, we implement a testbed based on both the Oversim-based simulation platform and two real WiMax terminals. The Oversim-based simulation platform is reused from the previous simulation scenarios. The WiMax access network comprises of an Acatel-Lucent base station (9710 C-WBS). The first WiMax terminal is an Alcatel-Lucent 9799 PCMCIA card. The second terminal is a Sequans USB card. IEEE 802.16e-2005 state of the art Scalable OFDMA based Technology is applied. Fig.4 illustrates the integration between the Oversim-based simulation platform and the WiMax access network.

The results show that, when some peers are using a WiMax access network to participate in the multicast tree, the performance of a single variable cost function is not good enough to give a really better performance since the access network is usually very varied in QoS. More investigations need to be done for this type of access network.

VI. CONCLUSION AND FUTURE WORKS

In this research, a new bandwidth-type cost function has been proposed. The mathematical derivation process has also been described in details so that ones can apply it to obtain other cost functions according to their specific requirements. The newly found cost function has considered dynamic requirements of the application and the underlay network. Intensive simulation 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 multicast session. A real testbed has been developed from the simulation scenarios to illustrate the adaptation of the newly proposed cost function in the case when some peers are using the WiMAX access network to connect to the multicast group. Base on

this result, multi-variable cost function should be considered. 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.

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] R. Boivie, N. Feldman, Y. Imai, W. Livens, D. Ooms, and O. Paridaens, "Explicit multicast (Xcast) concepts and options," *Request for Comments (RFC)*, vol. 5058, 2007.
- [4] Xing Jin, Kan-Leung Cheng, and S. H. G. Chan, "Island multicast: Combining ip multicast with overlay data distribution," *Multimedia, IEEE Transactions on*, vol. 11, no. 5, aug 2009.
- [5] J. N. Hwang, "Multimedia Networking: From Theory to Practice," 2009.
- [6] 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.
- [7] 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.
- [8] R. Widyono, "The design and evaluation of routing algorithms for real-time channels," *International Computer Science Institute, TR-94-024*, 1994.
- [9] A. Bueno, P. Vila, and R. Fabregat, "Multicast extension of unicast charging for qos services," in *Proceedings of 4 th IEEE European Conference on Universal Multiservice Networks (ECUMN)*. Citeseer.
- [10] D. H. Lorenz and A. Orda, "Optimal partition of QoS requirements on unicast paths and multicast trees," *IEEE/ACM Transactions on Networking (TON)*, vol. 10, no. 1, pp. 102–114, 2002.
- [11] D. Raz and Y. Shavitt, "Optimal partition of QoS requirements with discrete cost functions," *IEEE Journal on selected areas in communications*, vol. 18, no. 12, pp. 2593–2602, 2000.
- [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] 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.
- [17] R. Itu-T and I. Recommend, "G. 114," *One-way transmission time*, vol. 18, 2000.