Application-Network Cross Layer Multi-variable Cost Function for Application Layer Multicast of Multimedia Delivery over Convergent Networks

Tien Anh Le, Hang Nguyen, Manh Cuong Nguyen

HAL Id: hal-01271802

https://hal.science/hal-01271802

Submitted on 9 Feb 2016

Wireless Networks

Application-Network Cross Layer Multi-variable Cost Function for Application Layer Multicast of Multimedia Delivery over Convergent Networks
--Manuscript Draft--

Application-Network Cross Layer Multi-variable Cost Function for Application Layer Multicast of Multimedia Delivery over Convergent Networks

Tien Anh Le · Hang Nguyen · Manh Cuong Nguyen

Abstract Application Layer Multicast (ALM) algorithms are either similar or conceptually based on Network Layer Multicast's cost functions. In this research work, a new application-network cross layer multi-variable cost function is proposed. It optimizes the variable requirements and available resources from both the application and the network layers. It can dynamically update the available resources required for reaching a particular node on the ALM's media distribution tree. Mathematical derivation and theoretical analysis have been provided for the newly proposed cost function so that it can be applied in more general cases of different contexts. An evaluation platform of an overlay network built over a convergent underlay network comprised of a simulated Internet topology and a real 4G mobile WiMAX IEEE802.16e wireless network is constructed. The performance of the newly proposed cross-layer multi-variable cost function is evaluated and compared with conventional cost functions in Scalable Video Coding-based services such as video multicast and video conferencing services using the evaluation platform. Intensive evaluation results have shown that the newly proposed cross-layer multi-variable cost function outperforms the conventional cost function and improves the performance of multimedia services on dynamic convergent network environments.

Keywords Application layer multicast, application layer routing, cost function, resource provisioning, cross-layer optimization, end-to-end QoS, 4G wireless networks, WiMAX, IEEE802.16e, convergent networks, multimedia content delivery, multimedia services

1 Introduction

Multicast is the method of delivering data over a group of pre-registered destinations. Conceptually, multicast has a better performance than both unicast and broadcast when serving only a certain group of users. 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 multimedia conference or multi-player games. These types of multicast services put a big load on the unicast infrastructure of the Internet. Therefore, there is a demand for the design and deployment of multicast algorithms over the Internet. Multicast can be approached from either network layer or application layer. Regarding multicast in network layer, IP-Multicast[2] is the first attempt to solve this problem. It is so far the most efficient multicast mechanism for delivering data over each link of the network only once. However, many deploying problems are still preventing IP-Multicast from being supported worldwide[3]. Attempts have been made to overcome these problems. Explicit Multi-Unicast (XCAST)[4] 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. An alternative solution for Multicast is to build Application Level Multicast(ALM) over the underlay network.

TA Le, H Nguyen and MC Nguyen
Department of Wireless Network and Multimedia Services
Institut Mines-Telecom, Telecom SudParis
91011 Evry, France
E-mail: tien_anh.le, hang.nguyen, manh_cuong.nguyen@it-sudparis.eu

The key concept of ALM is the implementation of multi-casting 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. 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, then the data is actively distributed from the source node to intermediate peers until reaching all peers in the multicast tree[?]. In order to build that 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. The construction algorithm of the media distribution tree is based on the costs among participating peers. Eventually, the efficiency of the media distribution algorithm will mainly depend on the cost function being used. The proposed cost function should not be confused with conventional network layer-based cost functions since they are not working on the same layer. In fact, application layer cost function relies on network layer cost function for underlay's routing and only be responsible for routing on the overlay level of Application Layer Multicast.

The conventional cost functions are mainly based on classical network layer cost functions. Moreover they are either single variable or heuristic. In this paper, we propose a cross-layer multi-variable cost function to reflect the fact that, the applications and the networks have many different QoS parameters (bandwidth, delay, Packet Error Rate, Bit Error Rate, jitter...) and a cross-layer multi-variable cost function can better model the real network's available resources than a single variable cost function. Our cross-layer multi-variable cost function makes an optimization simultaneously over several QoS parameters, both from the application and the network layers; whereas conventional single variable cost functions are all optimizations for only one single QoS parameter. If a single-variable cost function can only obtain a local optimal value of the cost, our cross-layer multi-variable cost function can generate a bigger set of routes based on their multi-dimensional cost values. Therefore, the ALM algorithm can search and find a better optimum route from the bigger route set. Nowadays, the multimedia application has dynamic resource requirements. In order to support that, the newly proposed cost function considers both the requirements from the application layer and the available resources from the network layer to make a cross-layer optimization. This multi-variable characteristic and this cross-layer optimization make our proposed cost function more suitable for today's enriched and complexed multimedia services. A theoretical model has been built in order to obtain the exact mathematical expression of the cross-layer multi-variable cost function. The mathematical derivation of our cost function makes it extensible and applicable in more general cases of different contexts. All the steps of this mathematical derivation are also described in details to be further applied to find other forms of cost functions with other QoS parameters. For example, our proposed mathematical derivation and theoretical model can be applied to get the cost functions for different types of services with any arbitrary number of QoS requirements (even for $N \geq 3$).

Because of its multi-layer characteristics, Scalable Video Coding (SVC) is a very suitable video codec for multimedia applications designed to work on a convergent network. Besides, SVC contents are very suitable for multimedia applications such as multimedia conference or multi-player games because of their advantages over conventional video coding methods[7]. Scalable Video Coding is the multi-layer extension of Advanced Video Coding with the advantage of providing visual services for customers with convergent network conditions and terminals' capabilities. SVC-based multimedia applications dynamically change their QoS requirements during the multicast session and this can be supported very well with our newly proposed Application-Network cross layer multi-variable cost function. More specifically, video conferencing and video multicast services are the typical multimedia applications whose resource requirements are different. For example, a video conferencing service requires a small delay and a quite high bandwidth while video multicast may have a lower strain on the delay but the required bandwidth may be higher. In this research, we evaluate interesting and useful multimedia services built on scalable video coding and our newly proposed cost function. These services are the models for other one-to-many and many-to-many multimedia services. They can support convergent context of terminals with different available bandwidth, different screen-sizes and a variety of computational capacities. Our contributions are also to evaluate popular SVC-based services built on our newly proposed cross-layer multi-variable cost function. They are a scalable video coding-based multicast service built on an overlay network as a model for many video streaming or one-to-many services, and a scalable video coding-based conferencing service built on a convergent underlay network as a model for many many-to-many multimedia services

2

(such as video conference or multi-party multimedia games).

Wireless network is a very important part of telecommunications. With the coming of 4G wireless networks such as LTE and WiMAX, broadband wireless communication becomes a reality. WiMAX is an IEEE (Institute of Electrical and Electronics Engineers) specification also known as IEEE 802.16. Its most famous releases are IEEE 802.16d (2004), IEEE 802.16e (2005) and IEEE 802.16m (2009). LTE is a 3GPP (Third-Generation Partnership Project) standard. It has two main release groups: LTE and LTE-Advanced. Multimedia services are the killer applications of 4G wireless networks. Few or almost no research work has been found on the evaluation of cost functions and ALM algorithms on a wireless network in general and a real wireless network in particular. Convergent network comprised of Internet and wireless network is the current trend of telecommunications. Users can use different types of terminals and network infrastructures to connect to the multimedia services such as video conferencing, multi-party games, and video streaming. For example, in a same video conference, some users can participate to the conference from their smart phones using a mobile WiMAX wireless network whereas other users can participate from their fixed terminals through the Internet. WiMAX (Worldwide Interoperability for Microwave Access) IEEE 802.16e 2005 release is one of the 4G wireless communication technologies for delivering high-speed Internet service to large geographical areas. Here we use this IEEE 802.16e release which is also called "Mobile WiMax" release. We have conducted evaluations of multimedia services on a real WiMAX mobile network provided by POSEIDON, a French National Pole de Competitivite System@tic's project. We have evaluated the SVC-based multimedia services on an overlay network constructed from our newly proposed cost function. The overlay network is built over a convergent network of the simulated Internet topology and the real mobile WiMAX network. The available mobile WiMAX network comprises of a real access network and a real core network. The mobile WiMAX access network comprises of an Acatel-Lucent extended Base Station. The core network comprises of the Operations and Maintenance Centre, the Wireless Area Controller, the Authentication, Authorization, and Accounting, the Home Agent, the Dynamic Host Configuration Protocol, FTP and video servers. We also have two types of real WiMAX mobile terminals. The first type of WiMAX mobile terminal is an Alcatel-Lucent 9799 PCMCIA card. The second type of WiMAX mobile terminal is a Sequans USB card. Together, the dynamic multimedia applications and the convergent network comprised of a simulated Internet topology and the real mobile WiMAX network make an ideal platform for us to evaluate the performance of our cross-layer multi-variable cost function.

We are now able to evaluate performance of our newly proposed cost function for the SVC-based services (scalable video multicast and scalable video conferencing) on a convergent network of the simulated Internet topology and the real 4G mobile WiMAX network.

Our proposal is original because of the cross-layer multi-variable characteristics of the cost function. But it's also the scalable video coding-based multicast and video conferencing services and their performance evaluation results on the overlay network built over a convergent underlay network comprised of a simulated Internet topology and a real mobile WiMAX network that make our proposal unique. In this research, all of these requirements are for the very first time integrated and evaluated on a common platform to show the performance of the newly proposed cross-layer multi-variable cost function on the popular SVC-based services built over an overlay network whose underlay network is the convergence of a real mobile WiMAX network and a simulated Internet topology.

In brief, we have the following contributions:

- Propose a new cross-layer multi-variable cost function taking into account both the requirements from applications and the available resources from the networks,
- Propose in details the mathematical derivation process of the new cross-layer multi-variable cost function and analyze its performance. This analysis and derivation process can be further applied to find other forms of cost functions with other QoS parameters,
- Perform theoretical analysis to compare the newly found cost function with conventional ones,
- A scalable video coding-based multicast service built on an overlay network. This can be a model for many video streaming or one-to-many services,
- Evaluation of the new cost function in comparison with the conventional cost function over an overlay network in a scalable video coding-based multicast service setting,
- A scalable video coding-based conferencing service built on a convergent underlay network. This can be a model for many many-to-many multimedia services (such as video conference or multi-party multimedia games),

– Evaluation of the new cost function in comparison with the conventional cost function over an overlay network built on a convergent underlay network of a real mobile WiMAX network and a simulated Internet topology in a scalable video coding-based conferencing service setting,

The rest of the paper is organized as follows. Section 2 is a state of the art on conventional cost functions. The cross-layer multi-variable cost function is proposed together with the mathematical proofs in section 3. Theoretical analysis of the newly proposed cross-layer multi-variable cost function is explained in section 4. In section 5, the new cost function is evaluated for both video multicast and video conferencing services, on both overlay networks and real convergent networks of Internet and mobile WiMAX network. We make our conclusions in section 6.

2 Conventional approaches

The concept of cost function originally came from the theoretical Steiner tree problem in geometry. Nevertheless, it became a research topic in communication for solving problems on routing of multipoint connections[8]. Since routing is the main responsibility of the network layer in the OSI model, it is intuitable that conventional cost functions are mainly built based on the network layer's routing approach. We only consider application layer cost function in this research work, so network layer cost functions and routings such as the ones proposed in[9] are out of the scope of this research. Since the idea of Application Layer routings is new, the following state-of-the-art cost functions are so far the only related works that we can find.

Conventional cost functions are either empirical or heuristic[10][11]. Among all available cost functions for ALM routing that we have found, none of them has a mathematical derivation nor a clear citation. This does not mean those cost functions are not good since many of them are heuristically practical, but it is very difficult for designers to understand the reason why they should select a particular cost function and what are its exact reactions over network's conditions and application's requirements. Therefore a cost function with theoretical analysis and mathematical background is highly demanded for the ease of its applications. 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[12]. Also in[12], several kinds of cost functions have been investigated.

$$LinkCost = \frac{RsvBw + ReqBw}{LinkCap} \tag{1}$$

In which:

– RsvBw: The amount of bandwidth currently in use by existing connections,
– ReqBw: The amount of bandwidth requested by the newly arriving group of participants,
– LinkCap: Total bandwidth of the link.

The main idea of using the cost function Equ.1 is to choose a tree that is least-loaded and at the same time, to minimize the total amount of bandwidth to be consumed by the new connection. A variant which accounts for the length of the path (in number of hops) is also considered in [12] as follows:

$$LinkCost = 1 + \frac{RsvBw}{LinkCap} \tag{2}$$

The main idea of using the cost function of Equ.2 (a variation of Equ.1) is that hop count, a static link metric, plays a dominant role in the link cost.

$$LinkCost = \frac{1}{LinkCap - (RsvBw + ReqBw)} \tag{3}$$

In Equ.3, the cost increases exponentially with the utilization of the link. According to[12], Equ.3 appears to be more attractive than Equ.1 and Equ.2 because it can better distribute the load over the network by avoiding the use of highly loaded links, and thus links don't likely become saturated for future connections. Furthermore, it gives preference to shorter paths (with less number of link) as long as links are not heavily loaded.

4

In[**?**], the cost function Equ.4 is used because it has many desirable practical characteristics. It decreases with the delay, it is convex, it assigns infinitely high cost when the required delay guarantee approaches zero, and a fixed minimal link usage cost C^l, even if no guarantee is required. The constant θ determines how fast the cost grows for low delays and the constant A^l is used as a scaling constant.

$$C^l(d) = \frac{A^l}{d^\theta} \tag{4}$$

In[14], the cost function on each link is calculated by Equ.5

$$c_i(x) = \begin{cases} \dfrac{\kappa_i}{x - \kappa_i} & \text{, if } x > \kappa_i \\ \infty & \text{, if } x \le \kappa_i \end{cases} \tag{5}$$

In which:

− κ_i: Minimal delay that can be guaranteed on link i when utilizing all of its available resources,
− x_i: Requested delay of the new connection.

Although delay is used as a sample, any other QoS parameter such as bandwidth, jitter, packet loss can be used. However, the heuristics function Equ.5 is only single-variable, therefore it cannot consider other parameters simultaneously.

Another cost function Equ.6 considering several QoS parameters has been used in[15], again, without any mathematical proof.

$$C(\beta, \delta, \psi) = \delta + \frac{K_1}{\beta} + K_2.e^{\frac{\kappa_3}{\psi}} \tag{6}$$

In which:

− β: Residual bandwidth,
− δ: Residual buffer space,
− ψ: Estimated delay bound.

The scaling factor K_i allows us to modulate the relationship between β, δ, ψ even further, although it is still unclear how bandwidth, buffer, and delay units could be added exactly together[15],[16].

In [17], a single-variable cost function has been derived and its performance has been simulated under real conditions. The results have shown that, the single-variable cost function can only build a good multicast tree in certain conditions. A multi-variable cost function is highly demanded to build a more reliable multicast tree. From the best of our knowledge, conventional cost functions are mainly applied in network-layer routings. Our work has been one of the pioneers who have made full investigations on application layer multicast routing considering both network conditions and application's requirements.

3 Proposed cross-layer multi-variable cost function

Assuming that we have an overlay network with application peers and end-to-end links, in order to form a tree for data delivery, we need to know the 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 cross-layer multi-variable function. On each end-to-end link, we have to consider variable requirements from applications running on the ALM-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 varied frequently by the application. We have to also consider the maximum available resources of the underlay network. 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 (bottleneck) of all 3 physical links, the minimum guaranteed delay of the end-to-end link equals to the sum of all guaranteed delays on the 3 physical links.

QoS parameters can be either 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).

3.1 Problem formulation

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.*

3.2 Lemma 1: Bandwidth-type cost function

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 cost value at any time depends on its previous cost before there is any variation in the application's requirement,
- The increment of cost which is proportional to:
  - The previous cost, $f(x_w)$: If the previous cost is high, the trend is that the increment of the cost should be proportionally high to prevent the multicast from saturation. On the other hand, if the previous cost value is not very much high, the increment of the cost can be reasonably low to provide enough space for the cost to raise,
  - 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 $\frac{(\kappa_w - x_w - \Delta x_w)}{\kappa_w}$: It is obvious that if this ratio is high, so is the decrement of cost.

Finally, we have:

$$f(x_w + \Delta x_w) = f(x_w). \left[1 + \frac{\dfrac{\Delta x_w}{x_w + \Delta x_w}}{\dfrac{(\kappa_w - x_w - \Delta x_w)}{\kappa_w}} \right] \tag{7}$$

From Equ.7 we have:

$$\lim_{\Delta x_w \to 0} \frac{f(x_w + \Delta x_w) - f(x_w)}{\Delta x_w} = \lim_{\Delta x_w \to 0} f(x_w) \frac{\kappa_w}{(x_w + \Delta x_w)} \cdot \frac{1}{(\kappa_w - x_w - \Delta x_w)} \tag{8}$$

$$\Leftrightarrow f'(x_w) = f(x_w). \frac{\kappa_w}{x_w(\kappa_w - x_w)} \tag{9}$$

Replacing $f(x_w)$ by y and $f'(x_w)$ by $\frac{dy}{dx_w}$; from Equ.9 we have an ordinary differential equation:

$$\frac{dy}{dx_w} = y \frac{\kappa_w}{x_w(\kappa_w - x_w)} \tag{10}$$

Solve the ordinary differential equation Equ.10, we find the bandwidth-type cost function:

$$\frac{dy}{y} = \frac{\kappa_w}{x_w(\kappa_w - x_w)} dx_w \Leftrightarrow \int \frac{dy}{y} = \int \frac{\kappa_w}{x_w(\kappa_w - x_w)} dx_w$$

$$\Leftrightarrow \int \frac{x_w + (\kappa_w - x_w)}{x_w(\kappa_w - x_w)} dx_w = -\int \frac{d(\kappa_w - x_w)}{\kappa_w - x_w} + \int \frac{dx_w}{x_w}$$

$$\Leftrightarrow \ln(y) = \ln(x_w) - \ln(\kappa_w - x_w) + c \Leftrightarrow y = \frac{\Phi.x_w}{(\kappa_w - x_w)} \tag{11}$$

3.3 Lemma 2: Delay-type cost function

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 Equ.10 we have:

$$\frac{dy}{d\dot{x}_d} = y\frac{\dot{\kappa}_d}{\dot{x}_d(\dot{\kappa}_d - \dot{x}_d)}$$

$$\Leftrightarrow -\frac{dy}{\frac{dx_d}{x_d^2}} = y\frac{\frac{1}{\kappa_d}}{\frac{1}{x_d}(\frac{1}{\kappa_d} - \frac{1}{x_d})}$$

$$\Leftrightarrow \frac{dy}{dx_d} = y\frac{1}{\kappa_d - x_d} \tag{12}$$

Equation 12 is the ordinary differential equation to derive the delay-type cost function. From Equ.12, we have:

$$\frac{dy}{y} = \frac{dx_d}{\kappa_d - x_d} \Leftrightarrow \int \frac{dy}{y} = -\int \frac{d(x_d - \kappa_d)}{x_d - \kappa_d}$$

$$\Leftrightarrow \ln(y) = -\ln(x_d - \kappa_d) + \ln(c)$$

$$\Leftrightarrow y = \frac{c}{x_d - \kappa_d} \Leftrightarrow y = \frac{\Psi.\kappa_d}{x_d - \kappa_d} \tag{13}$$

3.4 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 Equ.10 and Equ.12, we have:

$$\frac{x_w(\kappa_w - x_w)}{\kappa_w}u_{x_w} + (\kappa_d - x_d)u_{x_d} = u \tag{14}$$

In which $u_{x_w} = \frac{\partial u}{\partial x_w}$, and $u_{x_d} = \frac{\partial u}{\partial x_d}$.
Equation 14 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:

$$\frac{\partial x_w}{\partial s}.u_{x_w} + \frac{\partial x_d}{\partial s}.u_{x_d} = \frac{\partial u}{\partial s} \tag{15}$$

Compare Equ.14 and Equ.15, we have:

$$\begin{cases} \dfrac{\partial x_w}{\partial s} = \dfrac{x_w(\kappa_w - x_w)}{\kappa_w} \\ \dfrac{\partial x_d}{\partial s} = \kappa_d - x_d \\ \dfrac{\partial u}{\partial s} = u \end{cases} \tag{16}$$

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, we can write:

$$\frac{d\phi}{ds} = 0 \Leftrightarrow \frac{\partial \phi}{\partial x_w}.\frac{\partial x_w}{\partial s} + \frac{\partial \phi}{\partial x_d}.\frac{\partial x_d}{\partial s} + \frac{\partial \phi}{\partial u}.\frac{\partial u}{ds} = 0$$

$$\Leftrightarrow \frac{\partial \phi}{\partial x_w}.\frac{x_w(\kappa_w - x_w)}{\kappa_w} + \frac{\partial \phi}{\partial x_d}.(\kappa_d - x_d) + \frac{\partial \phi}{\partial u}.u = 0 \tag{17}$$

Equation 17 is the *orthogonality property* of the vector (x_w, x_d), we can use it to check whether ϕ has been obtained correctly.
In order to solve Equ.14 we have to find two constants of integration from Equ.16.

3.4.1 Finding the first constant of integration:

From Equ.16, we have:

$$\frac{\dfrac{dx_w}{ds}}{\dfrac{du}{ds}} = \frac{\dfrac{x_w(\kappa_w - x_w)}{\kappa_w}}{u} \Leftrightarrow \frac{du}{u} = \frac{\kappa_w.dx_w}{x_w(\kappa_w - x_w)} \tag{18}$$

Since Equ.18 and Equ.10 have an identical form, we can use Lemma 1 to achieve the first constant of integration Equ.19 :

$$u = \frac{\Phi.x_w}{(\kappa_w - x_w)} \Leftrightarrow \Phi = \frac{(\kappa_w - x_w)u}{x_w} \tag{19}$$

We now check the orthogonality property of the first constant of integration Equ.19 by confirming Equ.17:

$$\frac{\partial \Phi}{\partial x_w}.\frac{x_w(\kappa_w - x_w)}{\kappa_w} + \frac{\partial \Phi}{\partial x_d}.(\kappa_d - x_d) + \frac{\partial \Phi}{\partial u}u$$

$$= \frac{\partial(\dfrac{(\kappa_w - x_w)u}{x_w})}{\partial x_w}.\frac{x_w(\kappa_w - x_w)}{\kappa_w} + ..$$

$$.. + \frac{\partial(\dfrac{(\kappa_w - x_w)u}{x_w})}{\partial x_d}(\kappa_d - x_d) + \frac{\partial(\dfrac{(\kappa_w - x_w)u}{x_w})}{\partial u}u$$

$$= \frac{-\kappa_w.u}{x_w^2}.\frac{x_w(\kappa_w - x_w)}{\kappa_w} + 0.(\kappa_d - x_d) + \frac{(\kappa_w - x_w)}{x_w}.u = 0 \tag{20}$$

By Equ.20 and Equ.17, we can confirm that the first constant of integration has been correctly found.

3.4.2 Finding the second constant of integration:

Similarly, using Equ.16 and Equ.13, we can find the second constant of integration having the form of:

$$\Psi = \frac{(x_d - \kappa_d)u}{\kappa_d} \tag{21}$$

We now check the orthogonality property of the first constant of integration Equ.21 by confirming Equ.17:

$$\frac{\partial \Psi}{\partial x_w}.\frac{x_w(\kappa_w - x_w)}{\kappa_w} + \frac{\partial \Psi}{\partial x_d}.(\kappa_d - x_d) + \frac{\partial \Psi}{\partial u}u$$

$$= \frac{\partial(\dfrac{(x_d - \kappa_d)u}{\kappa_d})}{\partial x_w}.\frac{x_w(\kappa_w - x_w)}{\kappa_w} + ..$$

$$.. + \frac{\partial(\dfrac{(x_d - \kappa_d)u}{\kappa_d})}{\partial x_d}(\kappa_d - x_d) + \frac{\partial(\dfrac{(x_d - \kappa_d)u}{\kappa_d})}{\partial u}u$$

$$= 0.\frac{x_w(\kappa_w - x_w)}{\kappa_w} + \frac{u}{\kappa_d}.(\kappa_d - x_d) + \frac{(x_d - \kappa_d)}{\kappa_d}.u = 0 \tag{22}$$

By Equ.22 and Equ.17, we can confirm that the second constant of integration has been correctly found.

3.4.3 The general solution:

The equation $\Phi(x_w, x_d, u) = $ constant, describes a relationship among x_w, x_d, u such as shown in Equ.16. Notice that if $\Phi(x_w, x_d, u)$ is a constant, then $\mathcal{G}(\Phi)$ is also a constant ($\mathcal{G}(.)$ is any arbitrary function). Similarly, if $\Psi(x_w, x_d, u)$ is a constant, then $\mathcal{H}(\Psi)$ is also a constant ($\mathcal{H}(.)$ is any arbitrary function). We can set these two constants equal so that:

$$\mathcal{G}(\Phi) = \mathcal{H}(\Psi) \tag{23}$$

This provides a more general expression in x_w, x_d, u that solves the partial differential equation Equ.14. The two arbitrary functions in Equ.23 may be merged into one by letting $\mathcal{F}(.) = \mathcal{G}^1(\mathcal{H}(.))$, then:

$$\Phi = \mathcal{F}(\Psi) \tag{24}$$

Equ.24 is the general solution of the partial differential equation Equ.14.

3.4.4 Checking the general solution:

From Equ.19, Equ.21, Equ.24, we have:

$$\frac{(\kappa_w - x_w)u}{x_w} = \mathcal{F}\left(\frac{(x_d - \kappa_d)u}{\kappa_d}\right) \tag{25}$$

We will now check whether Equ.25 is indeed a solution, and that the function $\mathcal{F}$ can be completely general. Let u be expressed as a function of x_w and x_d, where x_w and x_d are still independent.
We take the total derivative of Equ.25 to x_w and solve for u_{x_w}. Denoting $\mathcal{F}\left(\frac{(x_d - \kappa_d)u}{\kappa_d}\right) = \mathcal{F}(x_d, u)$, we have:

$$\frac{\kappa_w - x_w}{x_w}.u_{x_w} - \frac{\kappa_w}{x_w{}^2}.u = \frac{x_d - \kappa_d}{\kappa_d}.u_{x_w}.\mathcal{F}'(x_d, u)$$

$$\Leftrightarrow u_{x_w} = \frac{\kappa_w.\kappa_d.u}{x_w\left[(\kappa_w - x_w)\kappa_d - x_w(x_d - \kappa_d)\mathcal{F}'(x_d, u)\right]} \tag{26}$$

Similarly, taking the total derivative of Equ.25 to x_d and solve for u_{x_d}, we have:

$$\frac{(\kappa_w - x_w)u_{x_d}}{x_w} = \mathcal{F}'(x_d, u).\left[\frac{u + (x_d - \kappa_d)u_{x_d}}{\kappa_d}\right]$$

$$\Leftrightarrow u_{x_d} = \frac{\mathcal{F}'(x_d, u).u.x_w}{(\kappa_w - x_w)\kappa_d - \mathcal{F}'(x_d, u).(x_d - \kappa_d).x_w} \tag{27}$$

Replacing Equ.26 and Equ.27 into the left-hand side of Equ.14 we have:

$$\frac{x_w(\kappa_w - x_w)}{\kappa_w}.\frac{\kappa_w.\kappa_d.u}{x_w.\left[(\kappa_w - x_w)\kappa_d - x_w(x_d - \kappa_d).\mathcal{F}'(x_d, u)\right]} +$$

$$... + (\kappa_d - x_d).\frac{\mathcal{F}'(x_d, u).u.x_w}{(\kappa_w - x_w)\kappa_d - \mathcal{F}'(x_d, u)(x_d - \kappa_d).x_w}$$

$$= \frac{(\kappa_w - x_w).\kappa_d - (x_d - \kappa_d).\mathcal{F}'(x_d, u).x_w}{(\kappa_w - x_w)\kappa_d - x_w(x_d - \kappa_d).\mathcal{F}'(x_d, u)}.u = u$$

We can obtain the right-hand side of Equ.14, so Equ.25 is exactly the general solution of the partial differential equation Equ.14.

3.4.5 Fitting boundary conditions to the general solution:

Equation Equ.25 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 two independent QoS parameters: *requested bandwidth* (x_w) and *requested delay* (x_d), and their partial cost functions Equ.11 and Equ.13, we have these boundary conditions:

$$\begin{cases} \dfrac{x_w}{\kappa_w - x_w} = t^3 \\ \dfrac{\kappa_d}{x_d - \kappa_d} = t \\ u = t^2 \end{cases} \tag{28}$$

Replacing Equ.28 into Equ.25 we have:

$$\mathcal{F}(\frac{t^2}{t}) = \frac{t^2}{t^3} \Leftrightarrow \mathcal{F}(t) = \frac{1}{t} \tag{29}$$

9

From Equ.29 and Equ.25, we have:

$$\mathcal{F}\left(\frac{(x_d - \kappa_d).u}{\kappa_d}\right) = \frac{\kappa_d}{(x_d - \kappa_d).u} = \frac{(\kappa_w - x_w)u}{x_w} \tag{30}$$

The specific solution of Equ.14 is therefore:

$$u(x_w, x_d) = \sqrt{\frac{x_w}{\kappa_w - x_w} . \frac{\kappa_d}{x_d - \kappa_d}} \tag{31}$$

3.5 N-variable cost function with $N \geq 3$

Naturally, one may concern about the existence of a cross-layer multi-variable cost function when the number of variable is more than two. More specifically, what happens if we want to consider not only bandwidth and delay requirements from the application but also the other popular requirements such as jitter, packet-loss rate etc.. We may have full list of them. Thus, how to obtain the cost function in such case is a right question to ask.

Recursively from Equ.31, we can see that, the specific multi-variable cost function equals to the average multiplication of all partial cost functions $f_i(x_i)$:

$$u(x_1, x_2, ..., x_n) = \sqrt[n]{\prod_{i=1}^{n} f_i(x_i)} \tag{32}$$

In which:

- $x_1, x_2, ..., x_n$: List of application requirements (e.g. bandwidth, delay, jitter, packet-loss rate etc.),
- $f_i(x_i)$: Partial cost functions.

In order to obtain the N-variable cost function with $N \geq 3$, we first need to define the list of application requirements $(x_1, x_2, ..., x_n)$. Afterwards, we must find a partial cost function for each of these application requirements. The processes described in subsections 3.2 and 3.3 can be followed to obtain the partial cost functions.

In general, we can build a cost function for as many variables as possible given separated partial cost functions (e.g. jitter-type and packet-loss rate type cost functions). However, while a cross-layer multi-variable cost function can consider many QoS parameters at the same time, it should be noticed that the cross-layer 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.

4 Theoretical analysis of the newly proposed cross-layer multi-variable cost function

Conventional cost functions can be divided into two groups. The first group comprises of single-variable cost functions given by Equ.1, Equ.2, Equ.3, Equ.4, and Equ.5. The other group comprises of multi-variable cost functions such as Equ.6. They are described in section 3. Figure 1(a) shows the value of a bandwidth-type single-variable cost function versus the requested bandwidth. Since $x_w < \kappa_w$, we only consider the left domain of the graph. We can see that, when the requested bandwidth approaches κ_w, the link cost increases rapidly to infinity. Figure 1(b) shows the value of a delay-type single-variable cost function versus the requested delay. Since $x_d > \kappa_d$, we only consider the right domain of the graph. We can see that, when the requested delay approaches κ_d, the link cost increases rapidly to infinity. Nevertheless, two single-variable functions failed to consider other important QoS parameters simultaneously. Figure 2 shows the value of the newly proposed cross-layer multi-variable cost function. In this graph, only two QoS parameters (requested bandwidth and delay) are shown for

demonstrative purposes. We only use the right part of the graph where $x_w < \kappa_w$ and $x_d > \kappa_d$. The main difference is that, the link cost will only increase rapidly when both requested bandwidth x_w and delay x_d excess the maximum resources κ_w and κ_d. When only a single parameter excesses its limited resource, a high cost will be assigned, however, it should not be high enough to block the entire link since we still have resources to assign for the other parameter. For example, when x_d excesses κ_d, the cost increases to a high value, however, if we still have much bandwidth to assign, then the increased link cost will not be sufficient to block the entire link. The link cost will only increase rapidly if both the requested bandwidth and delay simultaneously excess their available resources (which are the total resources that the link can provide).

Compared to the heuristic multi-variable cost function Equ.6, our new cross-layer multi-variable cost function has many advantages. Firstly, it has considered the requested QoS parameters from the application instead of only the residual resource from the underlay network. This is a major advantage since application's requirements are usually varied. Secondly, although the scaling factors K_i allows us to assign different weights for different parameters, it is not clear how to assign values for these factors in practice, therefore, there will be implementation problems. Thirdly, it is really unclear how we can add all bandwidth, buffer and delay together without any unit impairment. From the above analysis, we can conclude that our new cross-layer multi-variable cost function owns better characteristics than conventional cost functions.

5 Proposed cost function performance on multimedia delivery platforms

After having the new cross-layer multi-variable cost function, it is necessary to evaluate its performance. Since it has been proved that SVC-content can resist better in the convergent environment of the overlay network, an evaluation platform of the newly proposed cross-layer multi-variable cost function with SVC content is highly required. EvalSVC is such a new evaluation platform for SVC content transmission [**?**]. The evaluation platform is shown in Fig.3. In this platform, the performance of the new cost function can be evaluated by comparing the video-related and multicast-related measurements of video transmissions on different transmission network's conditions.

We will use the four popular metrics for evaluation: (i) average link stress, (ii) link stress, (iii) average end-to-end delay, and (iv) end-to-end delay to compare the performance between the new cost function and the conventional NICE's distance function. The reason for choosing these metrics is because they are usually used to compare the performance of Application Layer Multicast systems. If only one link is considered, the average values of average link stress and the average end-to-end delay becomes the exact values of link stress and end-to-end delay.

Average link stress is defined in terms of the mean value of identical packets due to overlay forwarding, carried over a physical access link. This metric is equal to 1 for IP multicast. The lower the average link stress, the better the performance.

$$
\begin{cases}
\overline{stress(s_{id}, p_{id})} = \dfrac{1}{N} \sum_{i=0}^{N-1} stress(s_{id}, p_{id})_i \\[2ex]
\overline{stress} = \dfrac{1}{N} \sum_{s_{id}=0}^{N-1} \left(\dfrac{1}{P_{sid}} \sum_{p_{id}=0}^{P_{sid}-1} \overline{stress(s_{id}, p_{id})} \right)
\end{cases}
\tag{33}
$$

In which:

- p_{id}: Packet with an identity of id,
- s_{id}: Source with an identity of id,
- $stress(s_{id}, p_{id})_i$: Stress of link i caused by packet p_{id} being sent from the source s_{id},
- N: Total number of overlay nodes participating in the simulation platform,
- $\overline{stress(s_{id}, p_{id})}$: Average value of the link stress $stress(s_{id}, p_{id})_i$,
- P_{sid}: Total number of packets sent from s_{id},
- $\overline{stress}$: Average link stress on the entire overlay network.

Link stress is the exact value when only one link (real mobile WiMAX wireless link) is involved. This value can be found by replacing $N = 1$ into Equ.33.

11

Average end-to-end delay is defined as the average value of the end-to-end delay on the entire overlay network as demonstrated on Fig.4.

$$\begin{cases} \overline{peercount} = \dfrac{1}{P} \sum_{sid=0}^{N-1} \sum_{pid=0}^{P_{sid-1}} \overline{peercount(sid, pid))} \\ \overline{e2edelay} = \overline{peercount}.\overline{hop2hopdelay} \end{cases} \quad (34)$$

In which:

- $\overline{peercount}$: Average number of peers that a packet must travel from its source node to its destination,
- p_{id}: Packet with an identity of id,
- s_{id}: Source with an identity of id,
- $\overline{peercount(sid, pid))}$: Average number of a particular packet with a source identity of s_{id} and a packet identity of p_{id},
- $\overline{hop2hopdelay}$: Average delay between 2 hops,
- $\overline{peercount}$: Average number of peers that a particular packet must travel from a sending node to a receiving node,
- $\overline{e2edelay}$: Average end-to-end delay,

End-to-end delay is the exact value when only one link (real mobile WiMAX wireless link) is involved. This value can be found by replacing $N = 1$ into Equ.34.

5.1 Video multicast service on overlay network

To compare the performance of the new cost function with a popular cost function, the evaluation platform is set up as described in Fig.5. We set up an OverSim[19] simulation scenario based on NICE. The main goal of the simulation is to show that the newly proposed cost function can perform better than the popular NICE's distance function in terms of the average link stress and the average end-to-end delay. In this research, we only choose to compare our newly proposed cost function with NICE's distance function, but the method and the platform can be applied to compare it with any other cost function.

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 of the Internet generated by GT-ITM[20]. Each topology was a two-level hierarchical transit-stub topology, containing 1250 nodes and about 6000 physical links[21]. Each physical link will have random values of delay, bandwidth, and PER (Packet Error Ratio). We will use the simulation plan described in[22] for performance comparison and evaluation purposes.

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 periodical heartbeat messages containing delays between nodes within a cluster, peers will decide whether it should elect a new cluster's leader. Changing cluster's leaders provokes changing and rebuilding the entire NICE tree. In its original paper[?], 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 cost function obtained from Equ.31.

Figure 6 shows a sample ALM data delivery. We assign the value of κ_w on the end-to-end *link A-B* by selecting the minimum available bandwidth among all available bandwidths of physical links connecting *peer A* to *peer B* (e.g. links A-1, 1-2, B-2). Similarly, we get the value of κ_w on all other end-to-end links (links A-B, B-C, and A-D). The required bandwidth x_w on all end-to-end links has the same value equalling to the required bandwidth of the source. The minimum guaranteed end-to-end delay κ_d is equal to the sum of all physical links' delays. Depending on the application, we have a delay limit from the sending peer to receiving peers. For example, in Fig.6 (b), peer A delivers data to all other peers simultaneously with the same delay requirement, so the end-to-end required delay for delivering data to the farthest peer (peer D) should be smaller than the delay limit. On other relaying end-to-end links (e.g. links A-B, B-C), the required delays are partitioned proportionally to their minimum guaranteed end-to-end delay κ_d, respectively. We run an application on overlay peers. The application continuously changes values of its required bandwidth (x_w) and delay (x_d). 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 in two cases mainly by using two metrics: average link stress and average end-to-end delay

defined previously[24]. 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 total 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 (the 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 has 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. All simulation evaluation parameters are in Table 1.

Figure 7 shows that the newly proposed cost function when applied by NICE can reduce the average link stress to a smaller value than the original NICE's distance function.

Figure 8 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 still smaller than the limitation value of 150 ms recommended by ITU-T for real-time communication services[25]. From the results we can see that, the new cost function can avoid multiple replication of packets on access links and therefore 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 smaller average end-to-end delay than the conventional distance function. It should be noticed from Fig.8 that, 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.

5.2 Video conferencing service in convergent underlay network (WiMAX+Internet) environment

To evaluate the performance of the newly proposed cost function in the real network conditions, we implement a test-bed based on both the Oversim-based simulation platform and a real mobile WiMAX network with two real WiMAX mobile terminals. The parameters of the test-bed are explained in Table 2. The convergent network is set up as illustrated in Fig.9. The real WiMAX mobile terminals connect to the simulated platform by using the real mobile WiMAX access network and core network provided by the French National Pole de Competitivite System@tic POSEIDON project [1]. The Oversim-based simulation platform is the same as the one from the previous simulation scenario. The mobile WiMAX access network comprises of an Acatel-Lucent extended Base Station (ALU xBS: 9710 C-WBS). The core network comprises of the Operations and Maintenance Centre (OMC), the Wireless Area Controller (WAC), the AAA (Authentication, Authorization, and Accounting), the HA (Home Agent), the DHCP (Dynamic Host Configuration Protocol), FTP and video servers. The first type of WiMAX mobile terminal is an Alcatel-Lucent 9799 PCMCIA card. The second type of WiMAX mobile terminal is a Sequans USB card. The IEEE 802.16e 2005 release technology (also called the "Mobile WiMAX" release) is used here.

Figure 10 illustrates the integration between the Oversim-based simulation platform and the real mobile WiMAX access network. This simulation scenario emulates a video conferencing service built on top of the ALM network. The participants can be divided into two groups. The first group comprises of simulated peers participating to the ALM group from the INET[26] underlay network (a simulation topology of the Internet). We use 1 or 2 peer(s) participating into the ALM group from the mobile WiMAX network using the OMNET++ single host underlay[27]. A tunnelling interface is set up to connect between the main ALM group and the external WiMAX mobile peer(s). Figure 11 shows the performance evaluation platform of SVC transmission on the overlay network built over a convergent underlay network of a simulated Internet topology and a real mobile WiMAX network. The Application Layer Multicast tree is constructed using the newly proposed cross-layer multi-variable cost function.

In Figures 12 and 13, the link stress, end-to-end delay and their average values are calculated at the mobile WiMAX access link and within the overlay network. In these two figures, the "Distance WiMAX" and "Cost WiMAX" labels show the performance of the conventional NICE's distance function and our

newly proposed cost function on the mobile WiMAX access links, respectively. The "Distance-average" and "Cost-average" labels show the performance of the conventional NICE's distance function and our newly proposed cost function in average of all the links in the overlay network, respectively. In this case, the average values of both link stress and end-to-end delay are made over the entire overlay network including the simulated Internet topology and the mobile WiMAX underlay network.

Figures 12 and 13 show that, on the real WiMAX mobile link, our newly proposed cross-layer multi-variable cost function outperforms the conventional distance function in both the link stress and the end-to-end delay. On the overlay network, our newly proposed cross-layer multi-variable cost function also outperforms the conventional distance function in both the average link stress and the average end-to-end delay.

In comparison between the mobile WiMAX link and the entire overlay network built over the convergent underlay network, Figure 12 shows that the mobile WiMAX link has a smaller link stress value than the average link stress on the entire overlay network. The reason is because the wireless link has less available bandwidth and a higher delay than other link within the overlay network. Therefore, the cost (calculated by our newly proposed cost function) for a data packet to travel through the mobile WiMAX links is high. The overlay routing algorithm will try to avoid as much as possible the real mobile WiMAX links and that is why they are placed at lower layers of the multicast tree. At the lower layers, the mobile WiMAX links and nodes do not have to do too much data forwarding for the ALM. Thus, they will have less link stress than other links in the overlay network. Our newly proposed cost function manage this automatically. Figure 13 shows that the end-to-end delay on the mobile WiMAX link is higher than the average end-to-end delay on the overlay network. It is normal since the wireless links generally have a higher delay than the fixed networks. Moreover, the wireless link here is a real WiMAX radio environment while the overlay network only takes simulation results.

6 Conclusion

In this research, a new cross-layer multi-variable cost function has been proposed. It gives a better optimum cost and provides a bigger set of suitable routes for the ALM algorithms to choose the optimal route from. The mathematical derivation process has also been described in details so that one can apply it to obtain other cross-layer multi-variable cost functions according to their specific requirements, in other contexts and in more general use cases. In this research work, delay and bandwidth are used as an example but we also show how to form the cost function with more QoS parameters ($N \geq 3$) such as PER (Packet Error Rate), BER (Bit Error Rate), jitter... Theoretical analysis has shown that the newly proposed cross-layer multi-variable cost function can simultaneously consider various requirements from the application as well as the possible resource from the network. It performs a cross layer joint optimization between the Application and Network layers. For the scalable video multicast and conferencing services running on the simulated overlay network, the new cost function outperforms the conventional distance function on both the average link stress and average end-to-end delay. Intensive simulation results built over the convergent underlay network of a simulated Internet topology and a real mobile WiMAX network have shown that, for a video conferencing service running on both the real mobile WiMAX link and the overlay network, the newly proposed cross-layer multi-variable cost function outperforms the conventional distance function in all four metrics of link stress, average link stress, end-to-end delay and average end-to-end delay. For future work, we can compare our newly proposed cost function with other conventional cost functions in more convergent contexts of both the terminals and the underlay networks.

References

1. Gillet and Franck, *Rapport du project POSEIDON*, Tech. Rep, 2001.

2. 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. 85110, 1990.

3. 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. 7888, 2000.

4. R. Boivie, N. Feldman, Y. Imai, W. Livens, D. Ooms, and O. Paridaens, *Explicit multicast (Xcast) concepts and options*, IEEE Network, vol. 14, no. 1, pp. 7888, 2000.

5. J. N. Hwang, *Multimedia Networking: From Theory to Practice*, 2000.

6. J. 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. 5874, 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. 180184.

8. B. M. Waxman, *Routing of multipoint connections*, Selected Areas in Communications, IEEE Journal on, vol. 6, no. 9, pp. 16171622, 2002.

9. B. Fortz and M. Thorup, *Internet traffic engineering by optimizing OSPF weights*, in INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE. IEEE, 2002, vol. 2, pp. 519528.

10. L. H. Sahasrabuddhe and B. Mukherjee, *Multicast routing algorithms and protocols: A tutorial*, Network, IEEE, vol. 14, no. 1, pp. 90102, 2002.

11. C. Fang, C. Feng, and X. Chen, *A heuristic algorithm for minimum cost multicast routing in OTN network*, in Wireless and Optical Communications Conference (WOCC), 2010 19th Annual. IEEE, 2010, pp. 15.

12. I. Matta and L. Guo, *On routing real-time multicast connections*, in IEEE International Symposium on Computers and Communications, 1999. Proceedings, 1999, pp. 6571.

13. 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. 16701679.

14. 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. 102114, 2002.

15. R. Widyono, *The design and evaluation of routing algorithms for real-time channels*, International Computer Science Institute, TR-94-024, 1994.

16. 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.

17. 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.

18. 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. 16.

19. I. Baumgart, B. Heep, and S. Krause, *OverSim: A flexible overlay network simulation framework*, in Proceedings of 10th IEEE Global Internet Symposium (GI07) in conjunction with IEEE INFOCOM. Citeseer, 2007, vol. 7, pp. 7984.

20. K. Calvert and E. Zegura, *GT internetwork topology models (GT-ITM)*, 1997.

21. E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, *How to model an internetwork*, in Proceedings IEEE INFO COM96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation, 1996, vol. 2.

22. D. Constantinescu, *Overlay multicast networks: elements, architectures and performance*, Department of Telecommunication Systems, School of Engineering, Blekinge Institute of Technology.

23. 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.

24. 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.

25. R. Itu-T and I. Recommend, *G. 114*, One-way transmission time, vol. 18, 2.

26. A. Varga, *INET Framework http://inet.omnetpp.org/*, 2007.

27. A. Varga, *OMNeT++ http://www.omnetpp.org/*, IEEE Network Interactive, vol. 16, no. 4, 2002
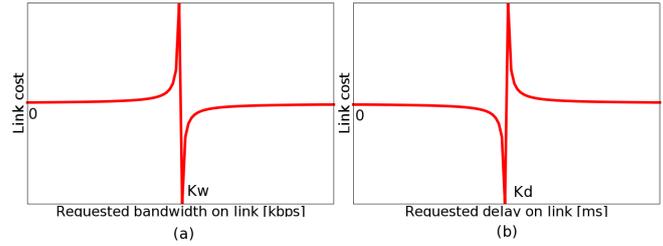
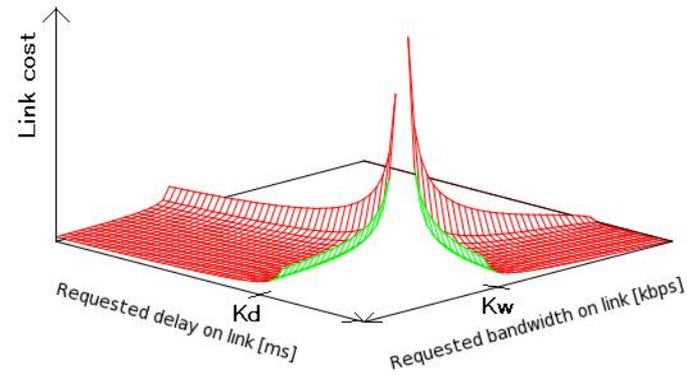**Fig. 1** Bandwidth-type (a) and delay-type (b) cost functions according to Equ.11 and Equ.13.

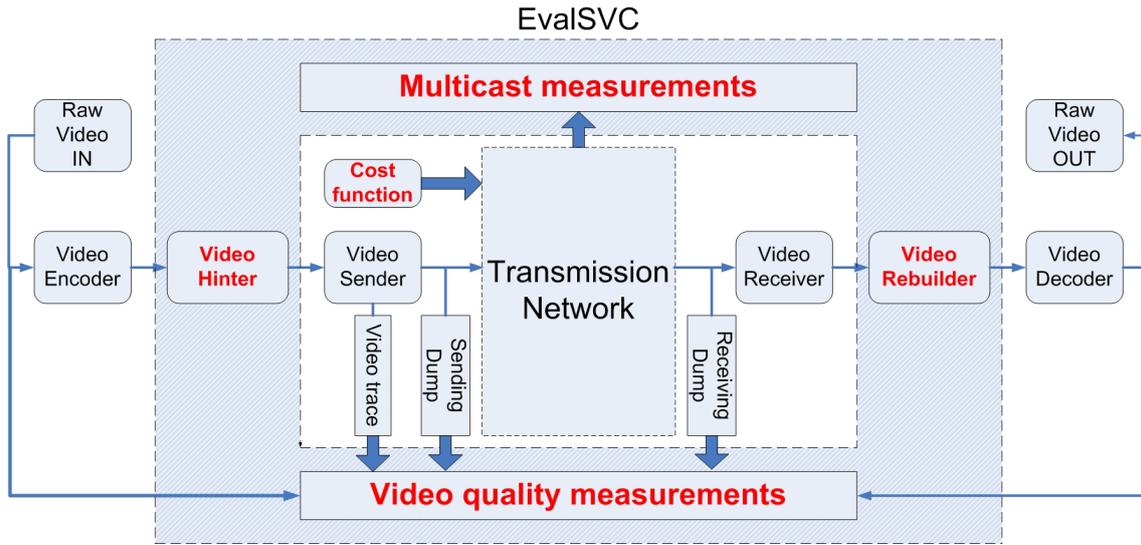**Fig. 2** Cross-layer multi-variable cost function according to Equ.31.

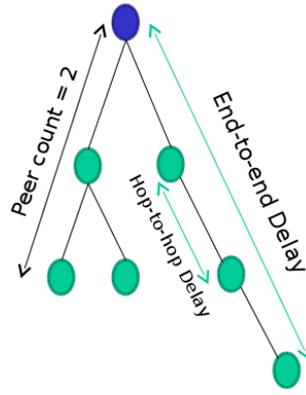**Fig. 3** EvalSVC: An evaluation platform of SVC-based services.
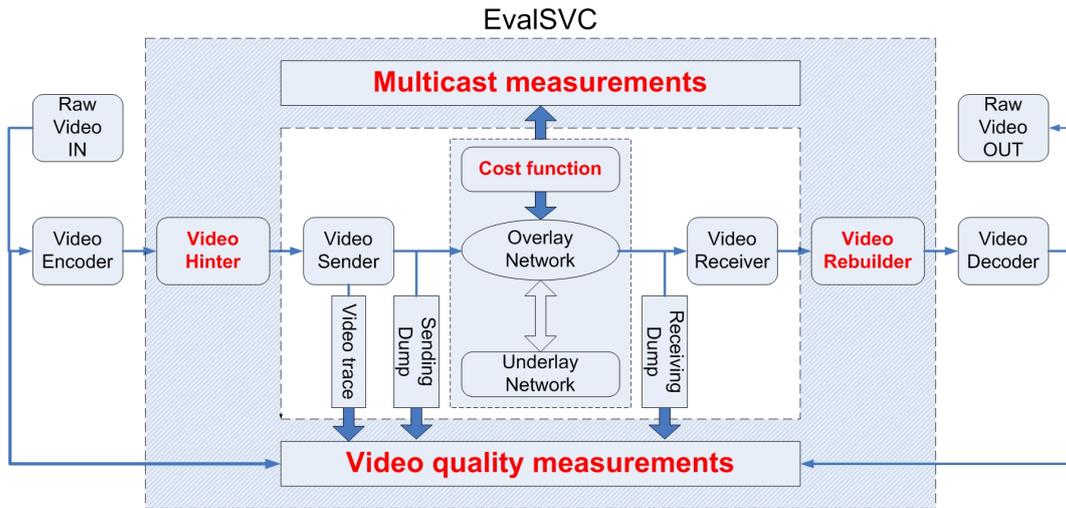
Fig. 4 End to end delay.

Fig. 5 Performance evaluation platform of SVC-based multicast service on an overlay network constructed from our newly proposed cross-layer multi-variable cost function.
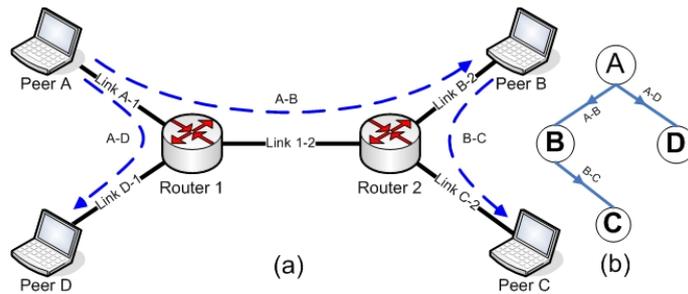

Fig. 6 A sample ALM data delivery tree, (a) Overlay and underlay diagram, (b) Tree diagram.
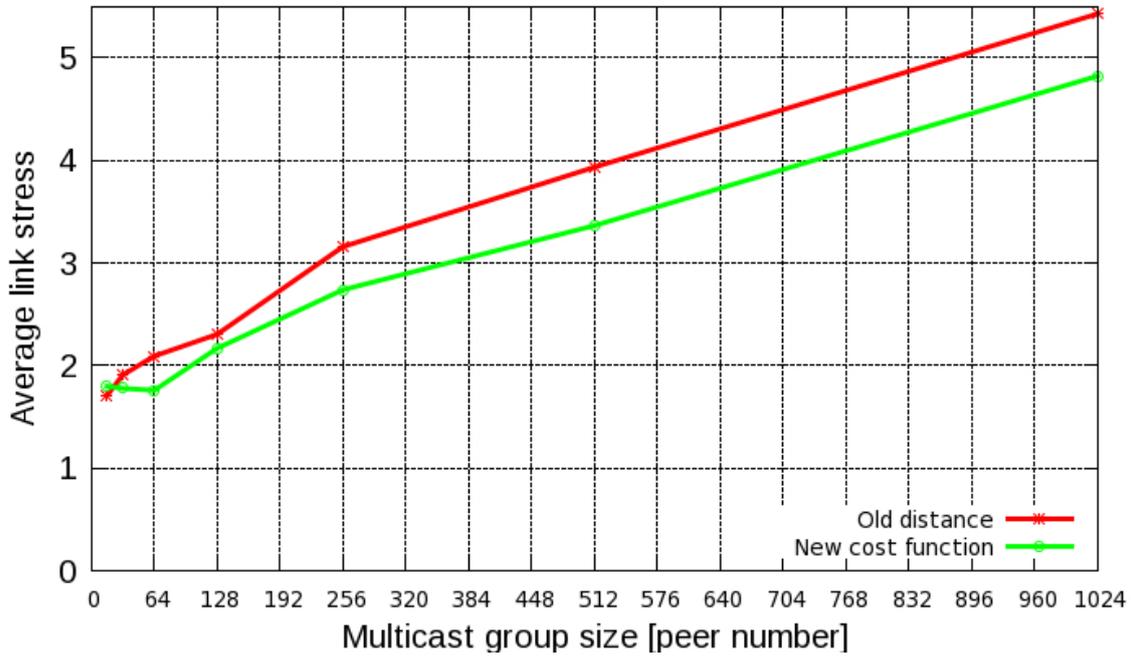
Fig. 7 Average link stress comparison for the NICE data-plan using the old and new cost functions. Transmitting data is obtained from a real SVC multicast session.
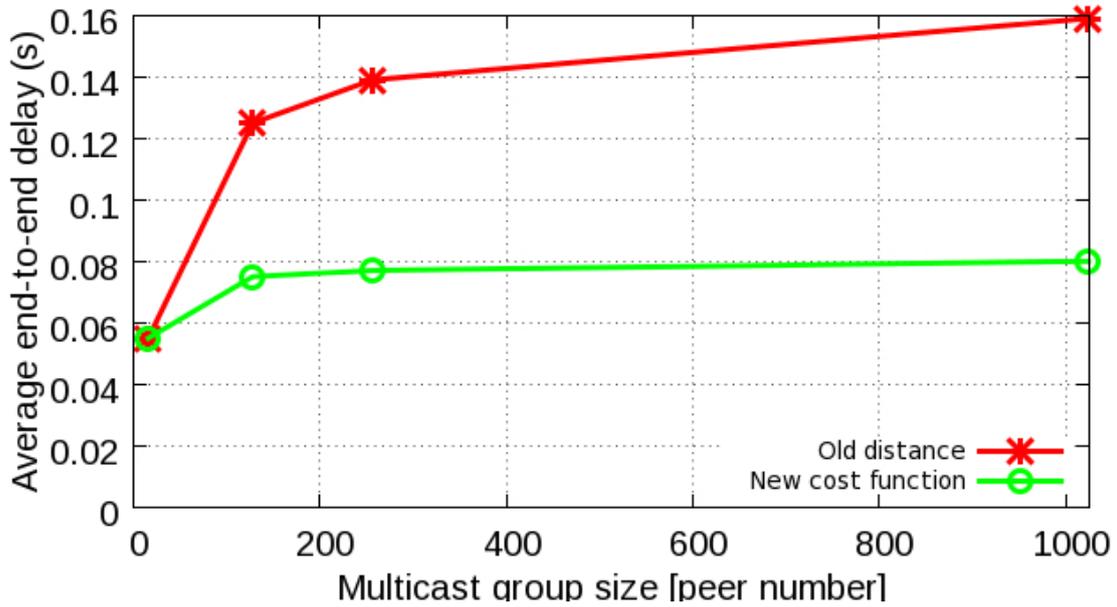
Fig. 8 Average end-to-end delay comparison for the NICE data-plan using the old and new cost functions. Transmitting data is obtained from a real SVC multicast session.

18
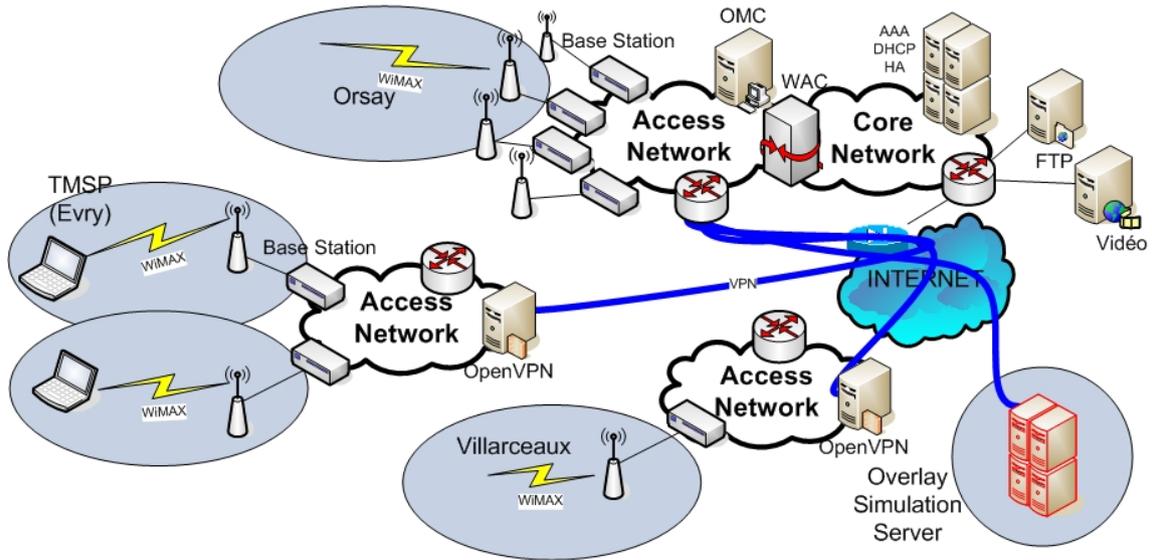
Fig. 9 Convergent underlay network composed of real POSEIDON's Alcatel-Lucent mobile WiMAX network and an overlay simulation platform [1].
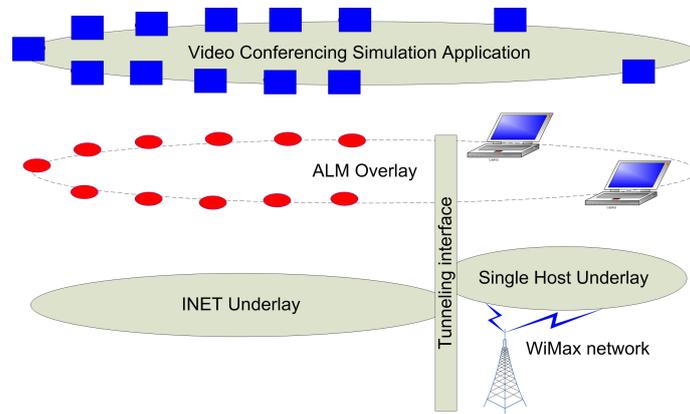
Fig. 10 Integration between the Oversim-based simulation platform and the real mobile WiMAX access network.
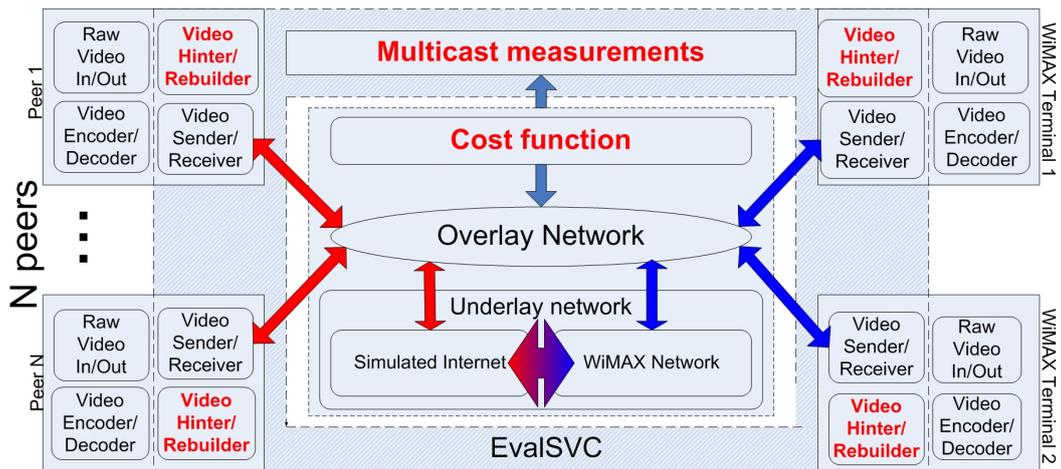
Fig. 11 Performance evaluation platform of SVC-based conferencing services on the overlay network built over a convergent underlay network comprised of a simulated Internet topology and a real mobile WiMAX network. The Application Layer Multicast tree is constructed using the newly proposed cross-layer multi-variable cost function.

19
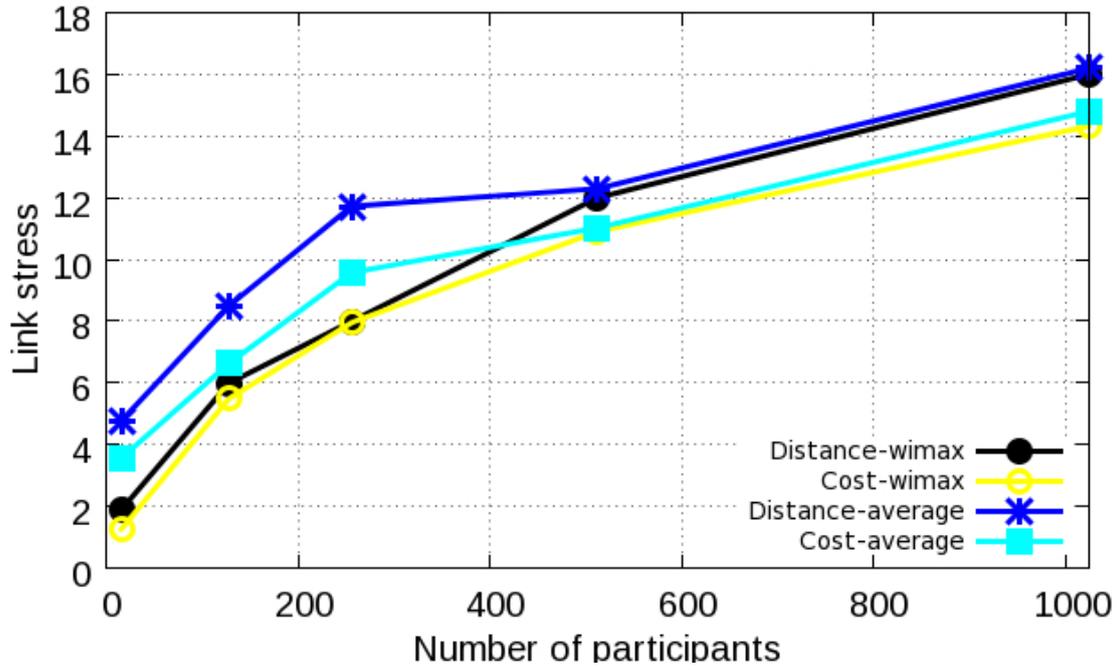
Fig. 12 Link stress performance of the video conferencing service built over a convergent underlay network comprised of a simulated Internet topology and a real mobile WiMAX network.
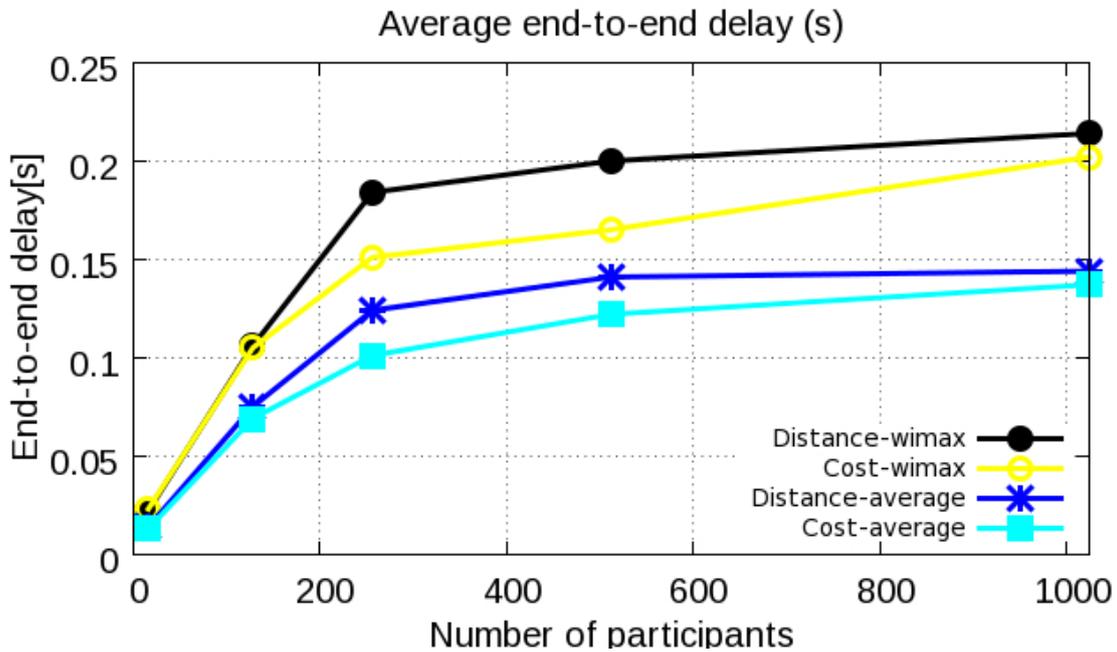
Fig. 13 End to end delay performance of the video conferencing service built over a convergent underlay network comprised of a simulated Internet topology and a real mobile WiMAX network.

20

Table 1 Simulation parameters of the SVC transmission on overlay network constructed from the cross-layer multi-variable and conventional cost function.

Parameters	Values
Purpose	Evaluation of the new cross-layer 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	− New cross-layer multi-variable cost function, − NICE's popular cost function
Overlay measurements	Average link stress, average end-to-end delay

Table 2 Simulation parameters of the SVC video conferencing service on the overlay network built over a convergent underlay network of simulated Internet topology and real mobile WiMAX network.

Parameters	Values
Purpose	Evaluation of the newly proposed cross-layer multi-variable cost function for Scalable Video Conferencing service on convergent network (simulated Internet + real WiMAX).
Video encoding	SNR SVC
Video size	CIF
Multicast	Overlay network
Transmission network	Simulated Internet topology and mobile WiMAX network
Number of WiMAX mobile terminals	2
Service	Application Layer Multicast for scalable video conferencing service
Network simulation tool	Oversim
Number of peer	1-1024 peers
Underlay network	Internet topology generated by GT-ITM
Cost functions	− New cross-layer multi-variable cost function, − NICE's popular cost function
Overlay measurements	− Average link stress, − Average end-to-end delay

Average end-to-end hop count for distance and new cost function

Average end-to-end delay for distance and bandwidth-type cost function

Average link stress for distance and bandwidth-type cost function

Old distance
Bandwidth-type cost function
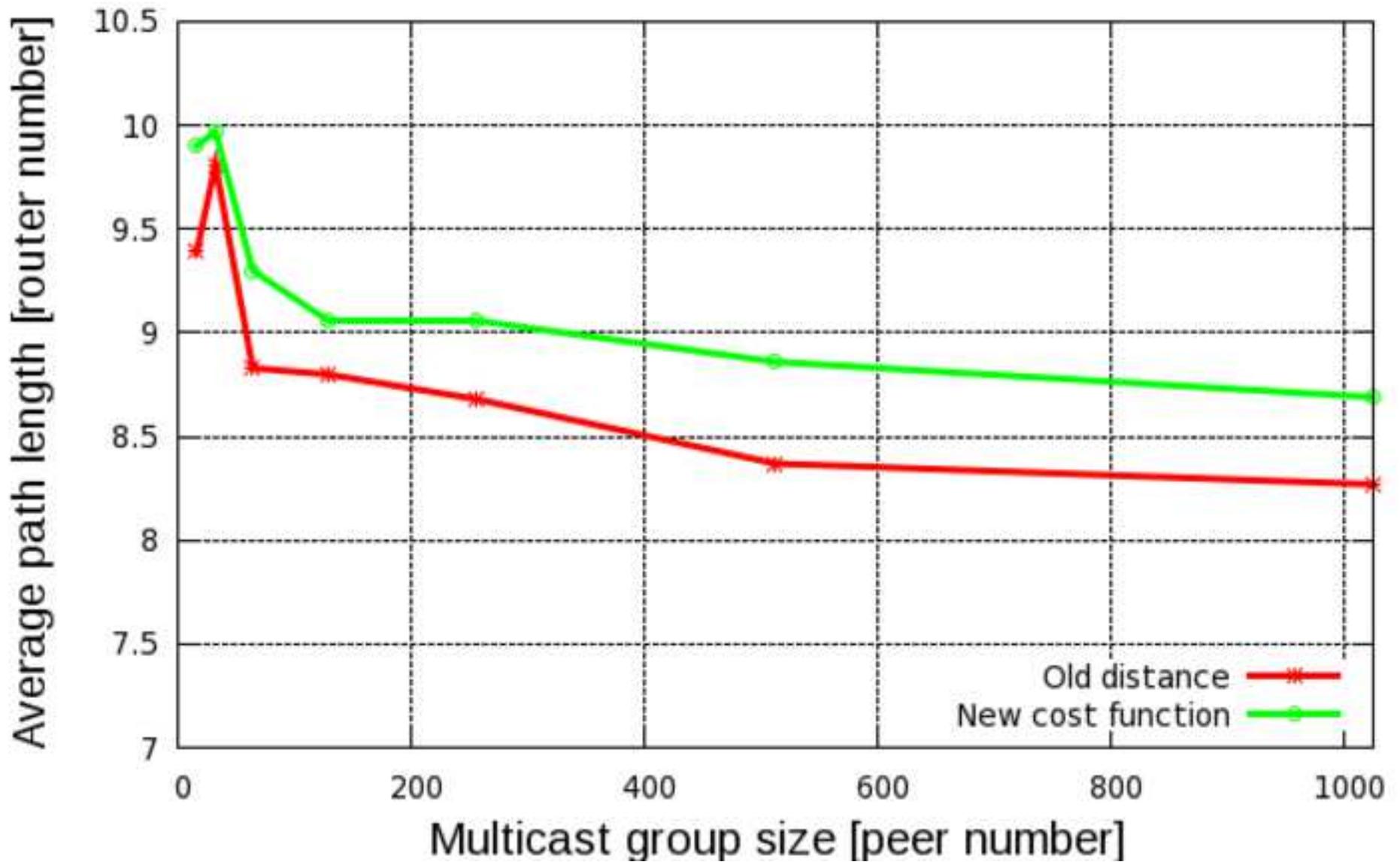
Average link stress
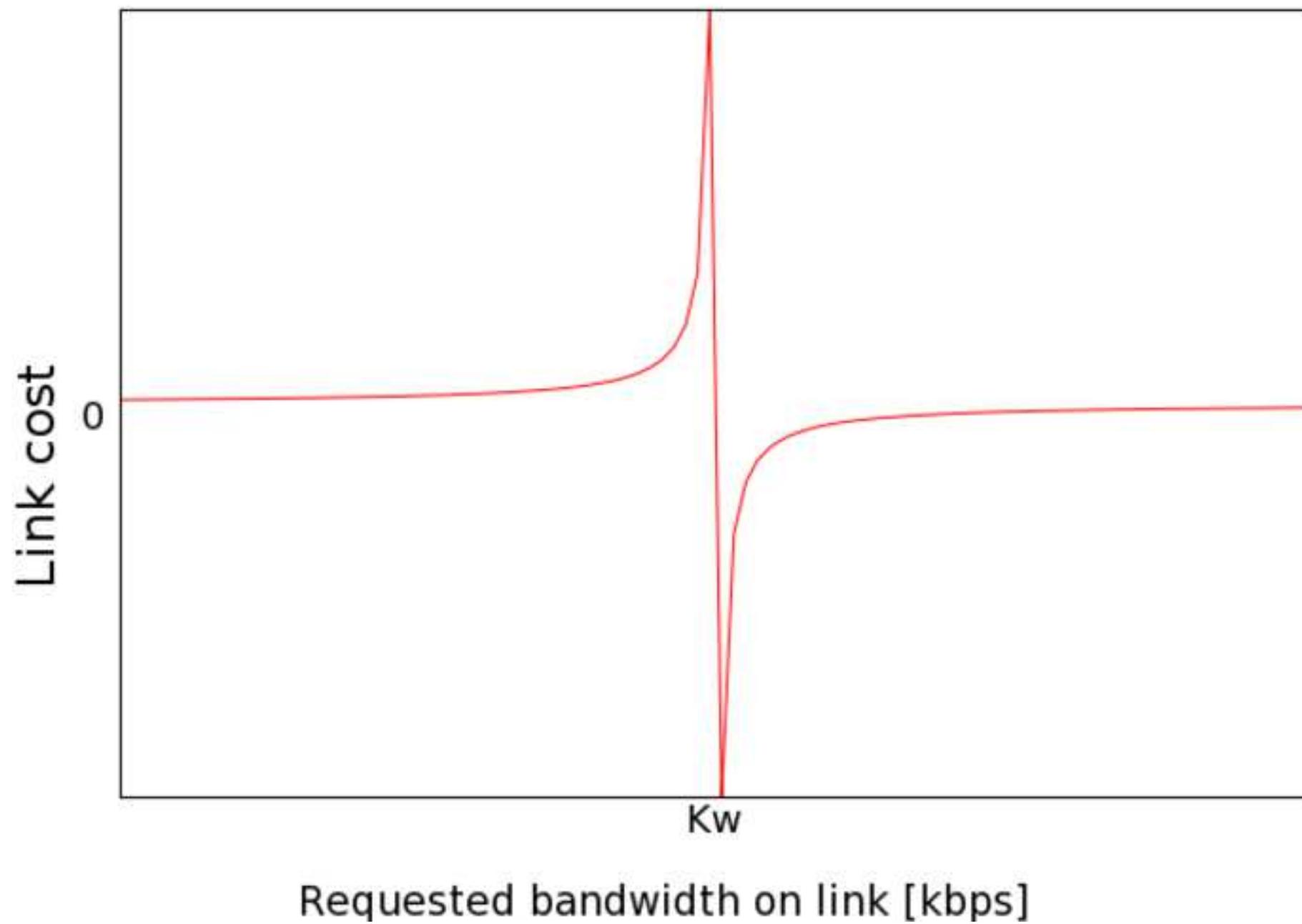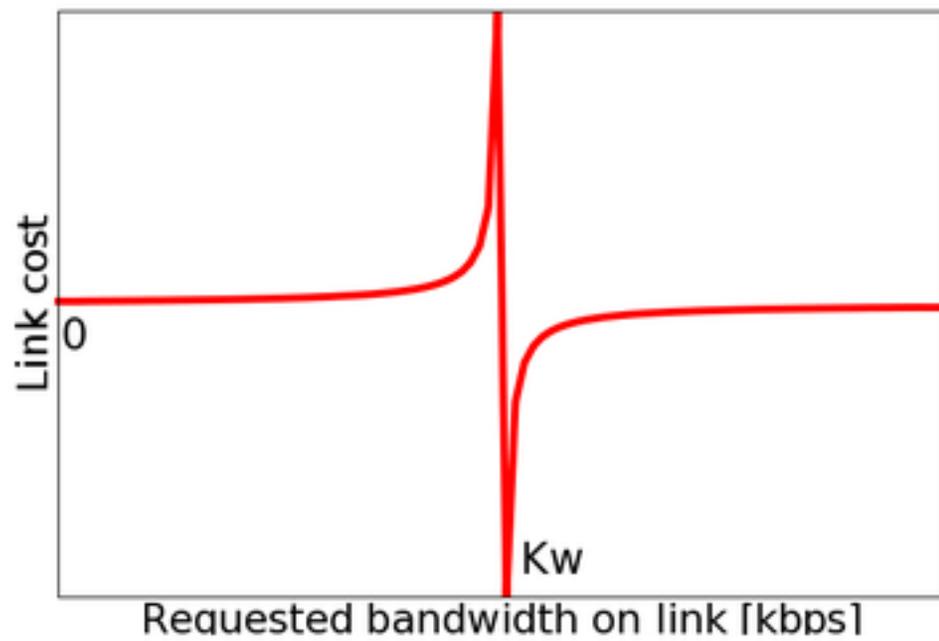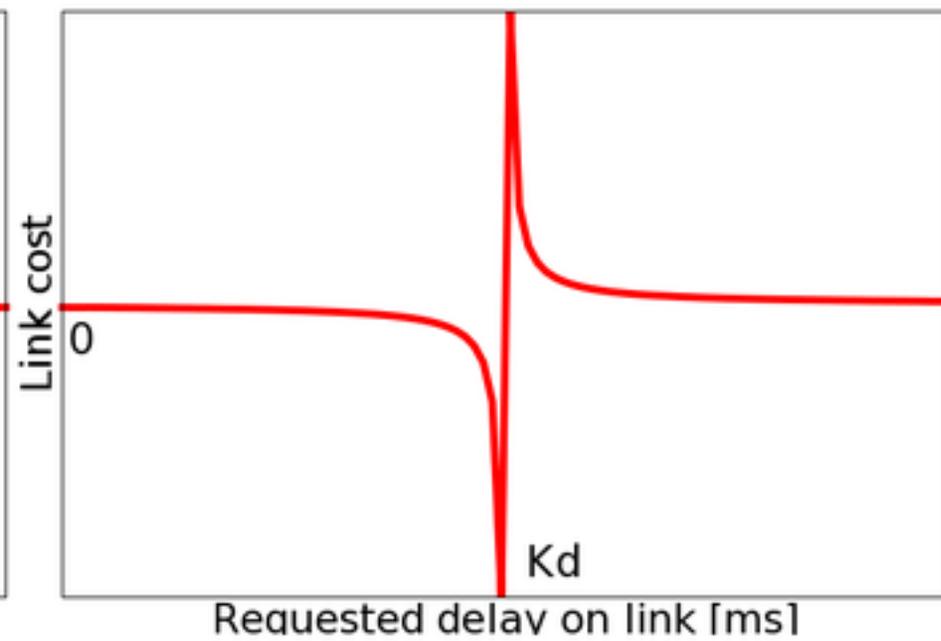
Multicast group size [peer number]
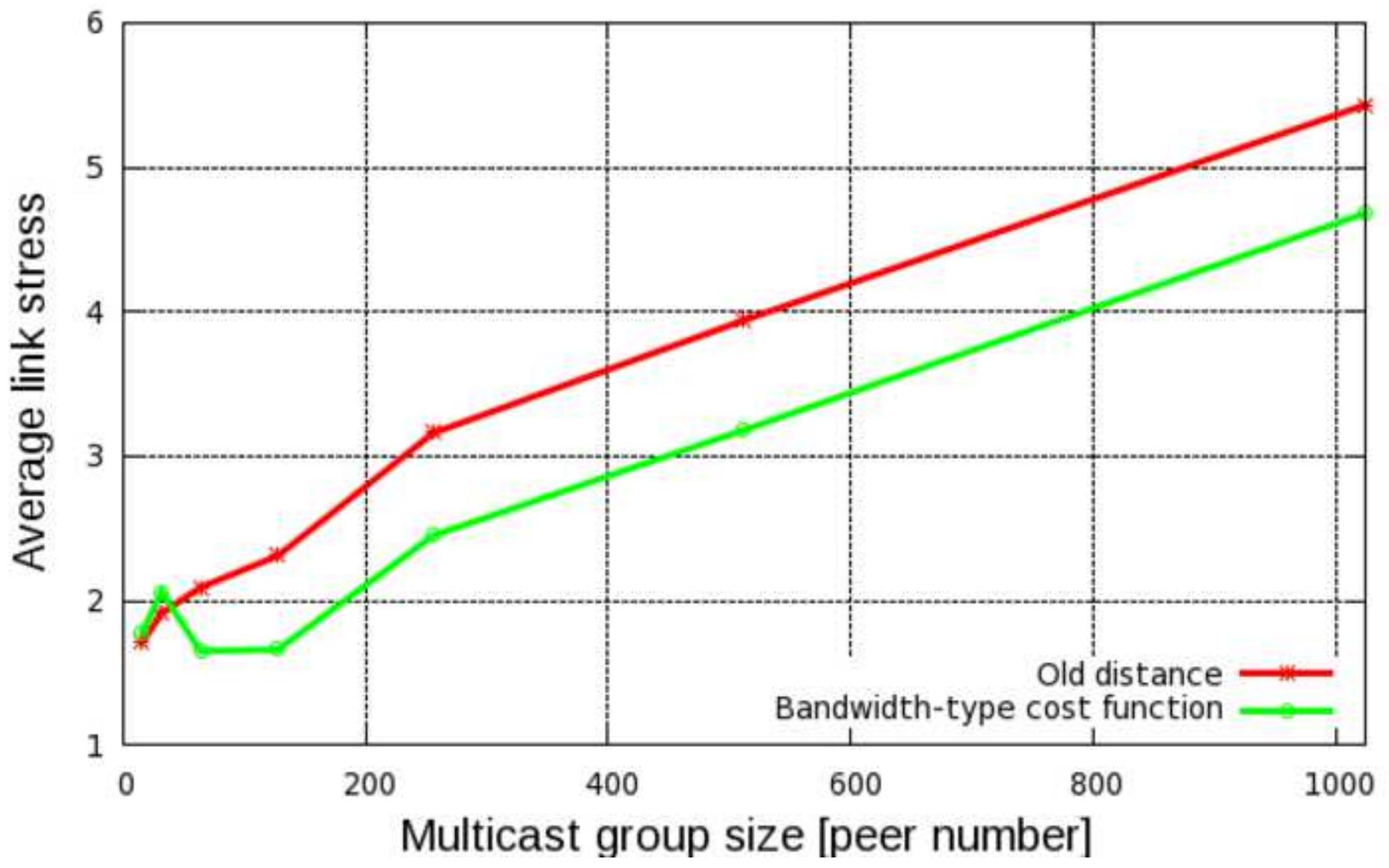
Average path length for distance and new cost function

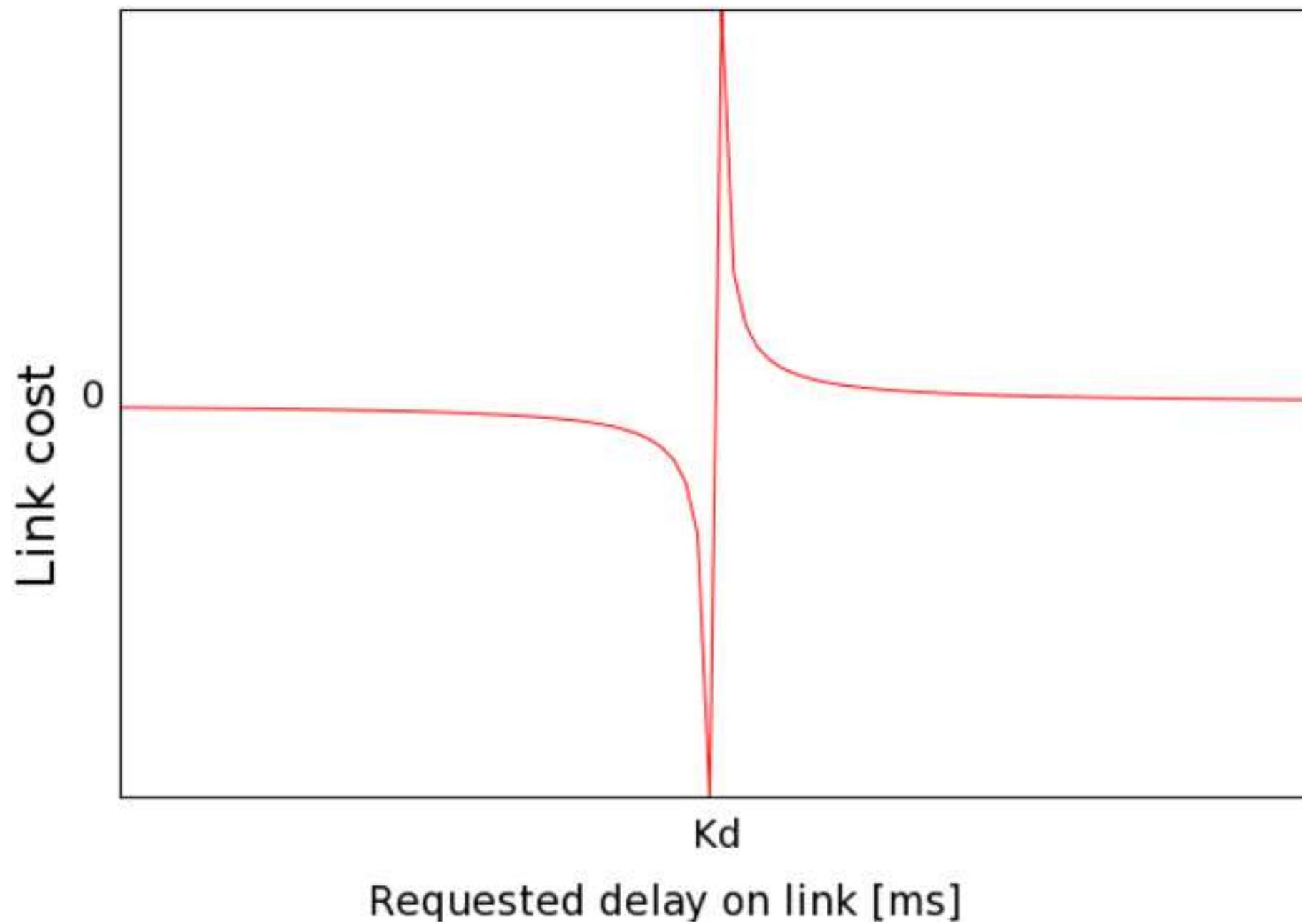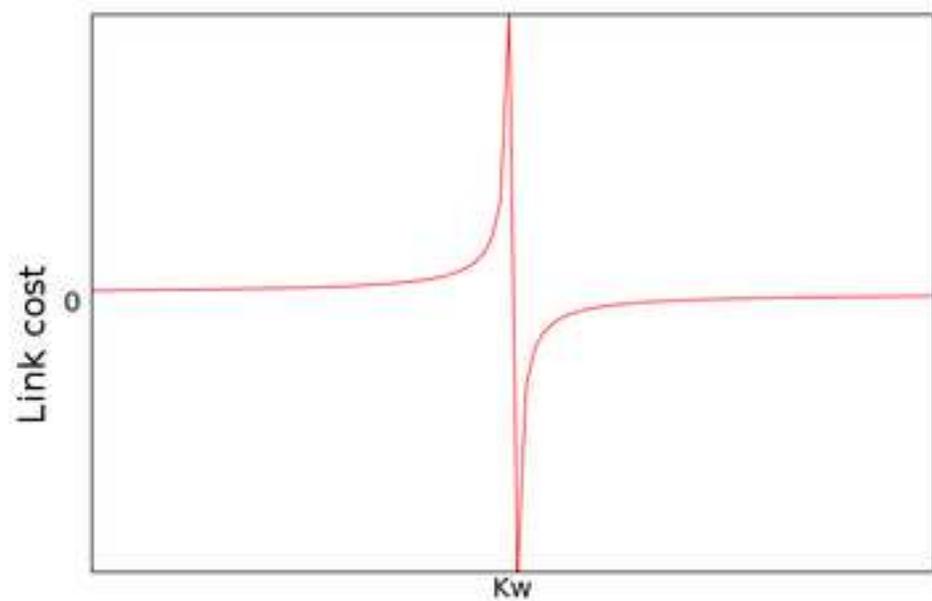Average path length for distance and new cost function
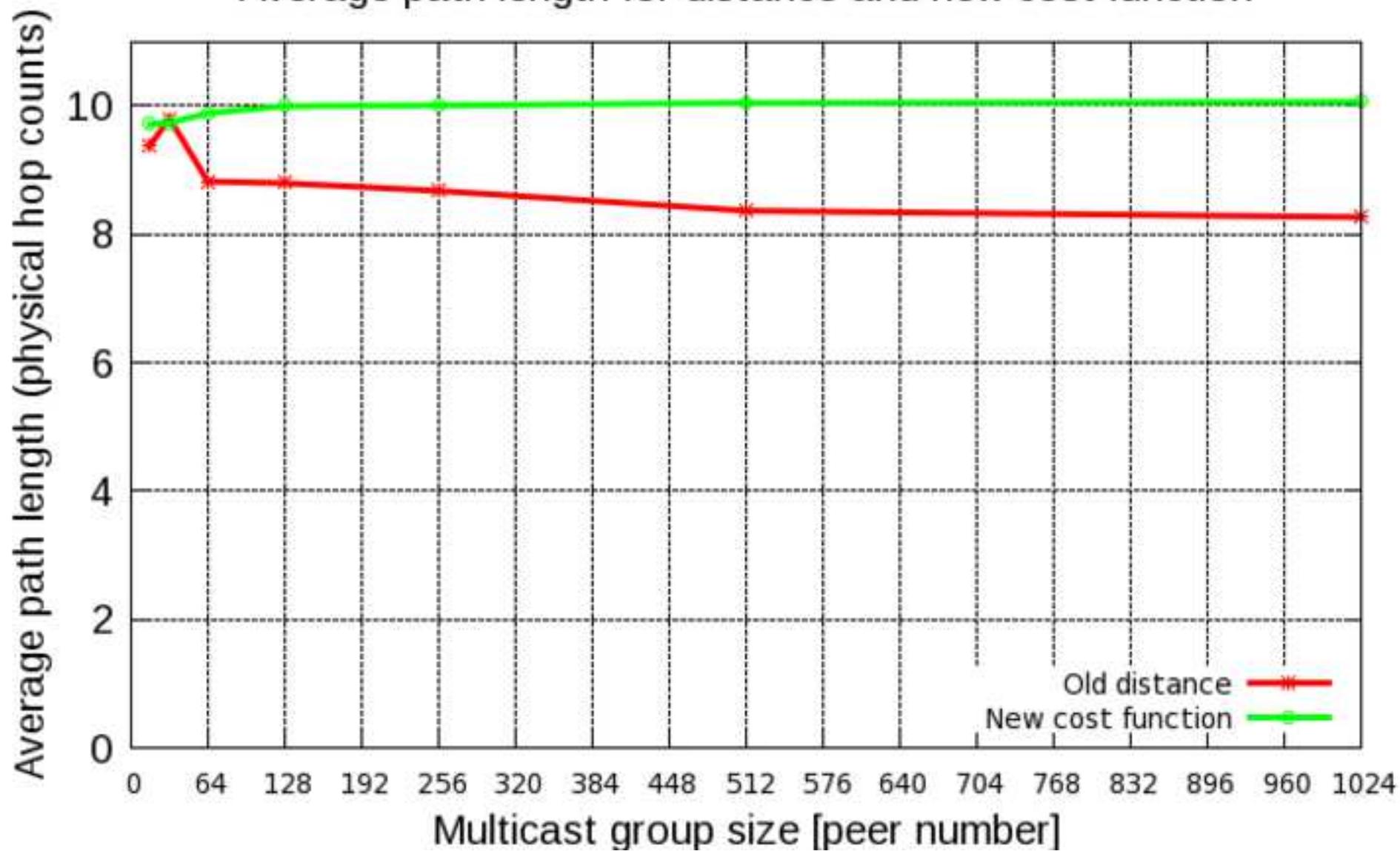
Link cost

0

Kw

Requested bandwidth on link [kbps]

(a)

Requested bandwidth on link [kbps]

Link cost

0

Kw

(b)

Requested delay on link [ms]

Link cost

0

Kd

MCU

Participant

Participant

Participant

Participant

Participant

ALM

Participant

Participant
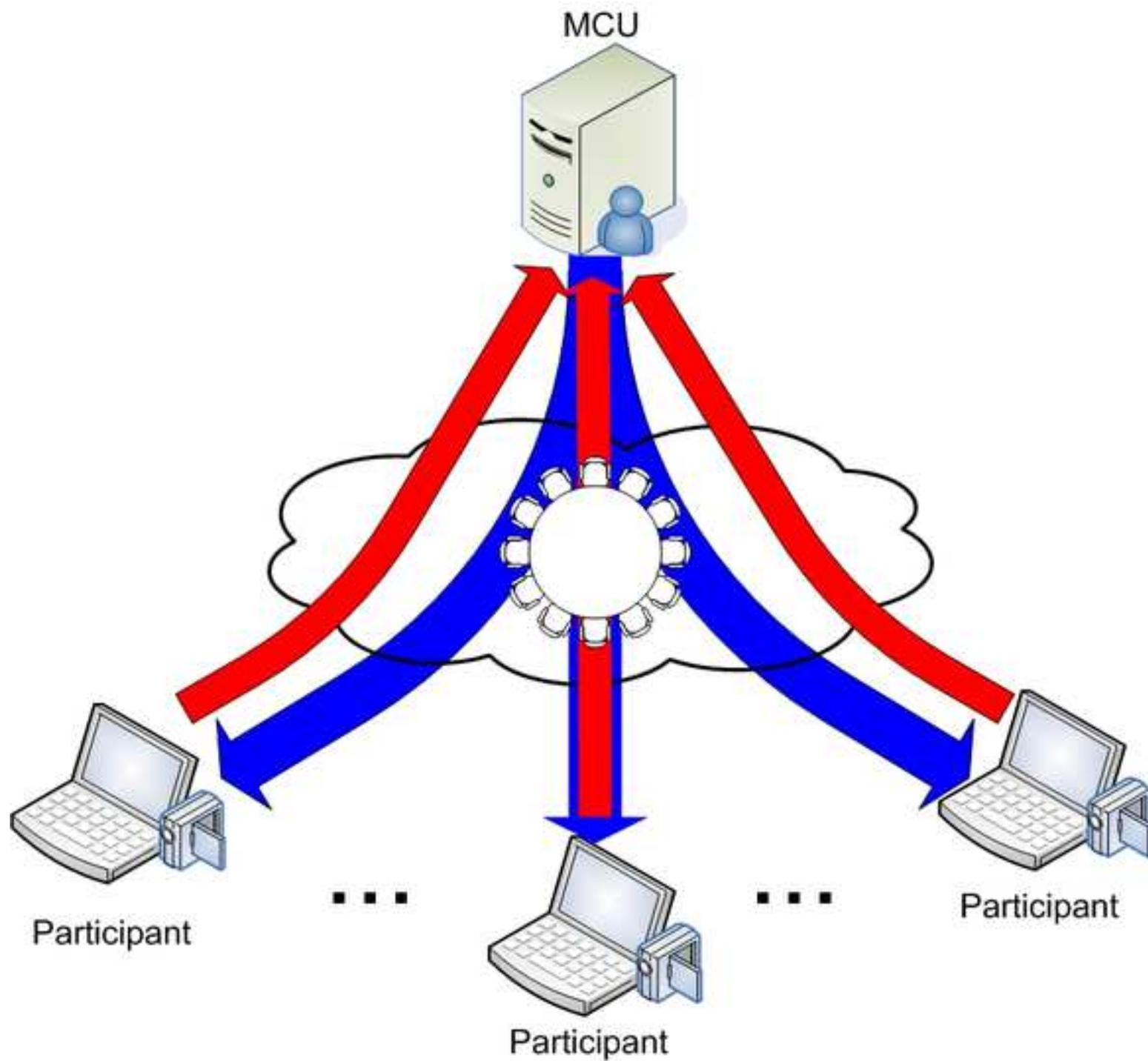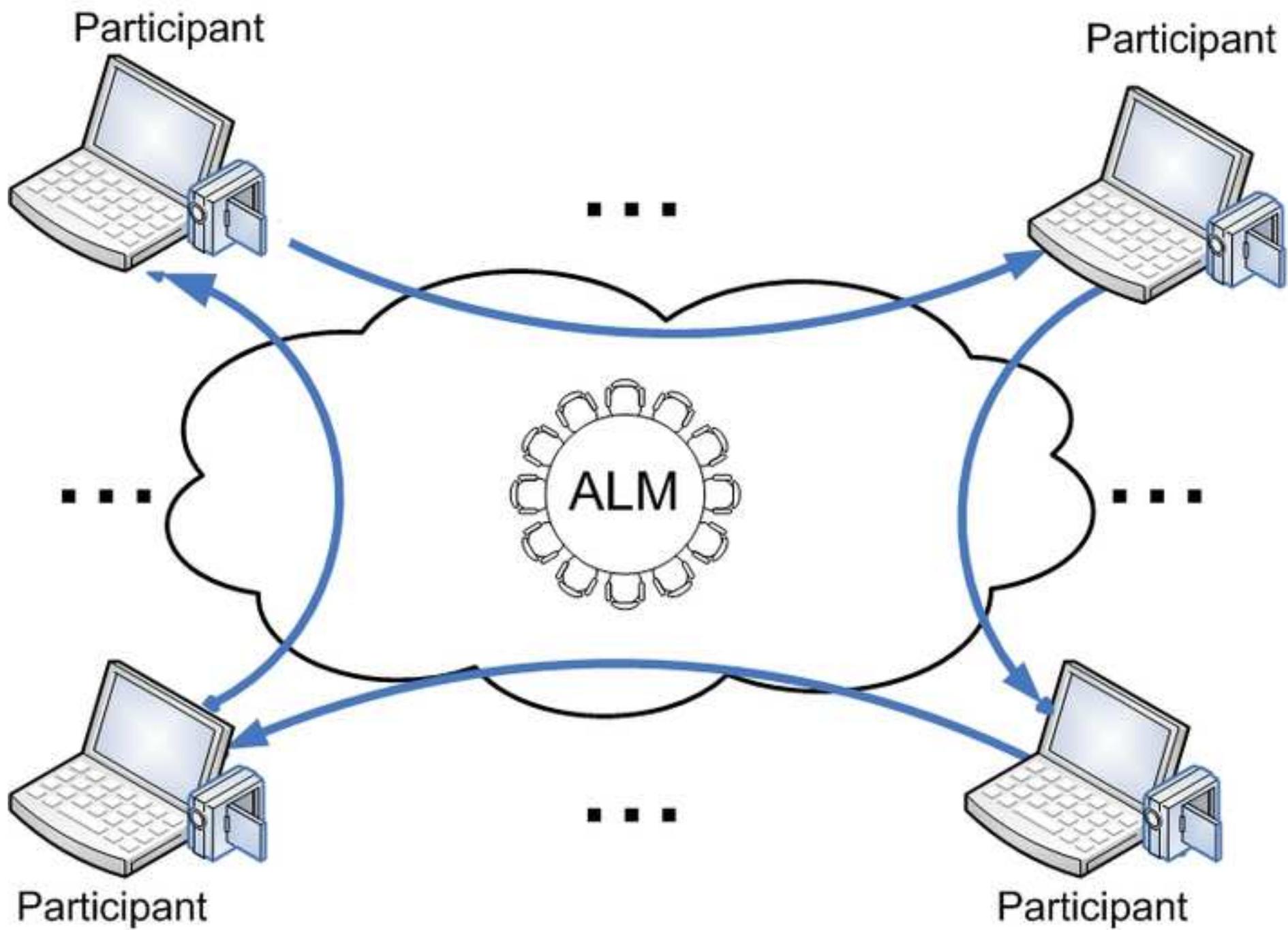
Requested bandwidth on link [kbps]

(a)

Requested delay on link [ms]

(b)

Real ALU Core Network

Internet

Real ALU BS

Real WiMAX terminal

Real ALU BS

Real WiMAX terminal

Real ALU BSC and OMC

Overlay simulation server
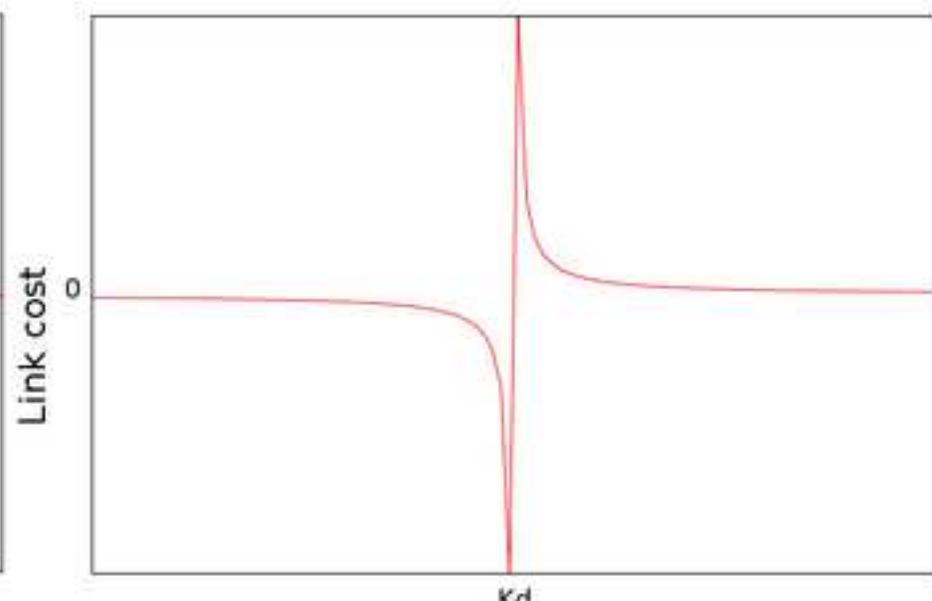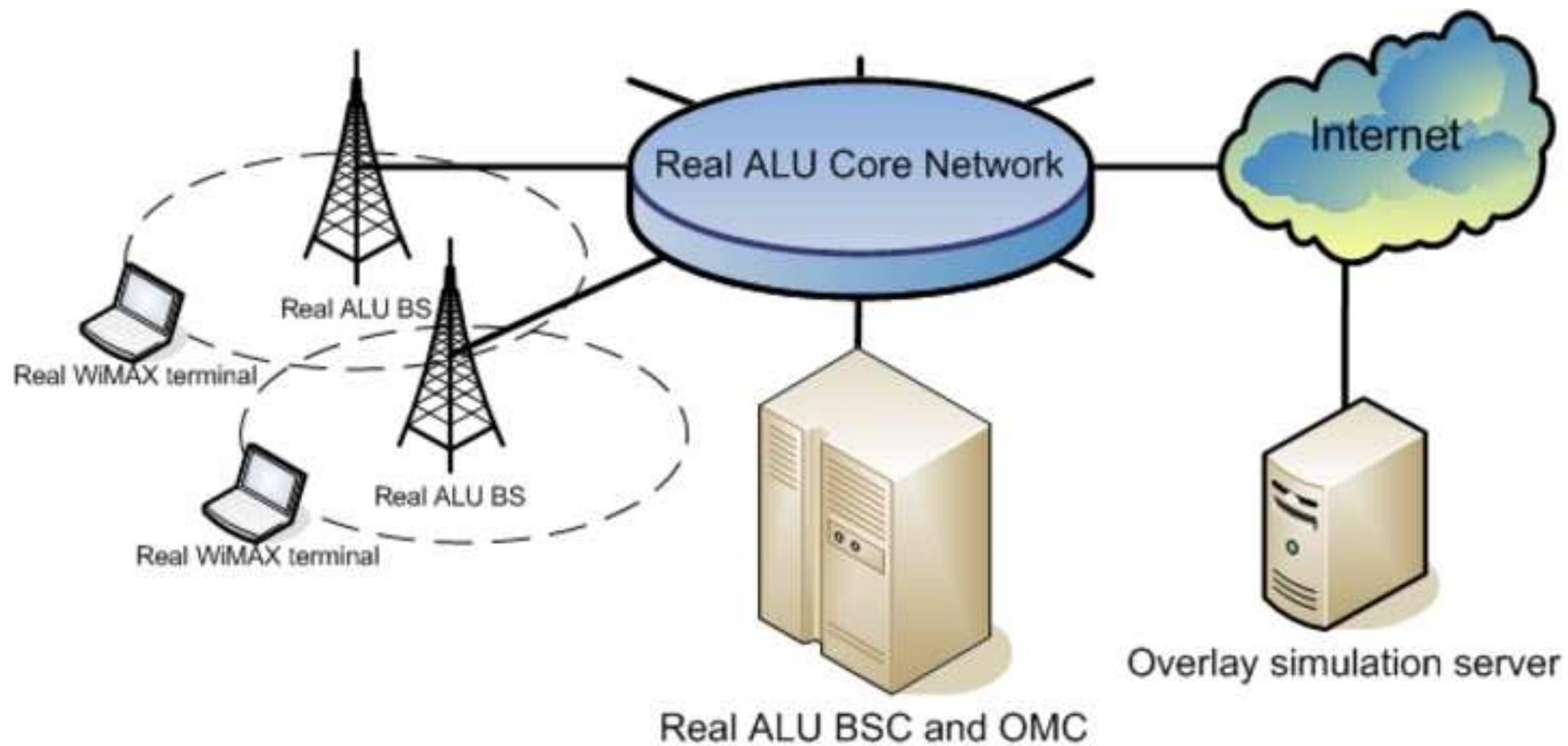
NICE's Average Link Stress
with different versions of cost function