Multi-objective Optimization For Peer-to-Peer Multipoint Video Conferencing Using Layered Video

Istemi Ekin Akkuş*, Öznur Özkasap*, M. Reha Civanlar†
*Koç University, Department of Computer Engineering, Istanbul, Turkey
†DoCoMo Communications Laboratories, Palo Alto, CA, USA

***Abstract*— We present a multi-objective approach and corresponding formulations for the optimal operation of a peer-to-peer multipoint video conferencing system. The system aims end-points with low bandwidth connections (i.e., single full-quality video in and out) and makes use of layered video to achieve that each participant can view any other participant's video at anytime. This may cause some of the peers receive lower quality video. Moreover, since the peers may have to forward the video they receive, this may cause larger delays for the peers that receive the video after it is forwarded by several peers. Objective formulations to determine the number of lower quality video receiving peers and the delay experienced by the peers are derived. A multi-objective optimization approach for minimizing both simultaneously is described. An extension that allows multiple video requests from the participants with sufficient bandwidth is proposed. Formulations to minimize the number of lower quality video receivers while maximizing the number of additional video requests are presented. A multi-objective optimization technique assigning importance weights to each of these objectives and its sensitivity to changes in the weights are shown. The use of multi-objective optimization techniques within a system is demonstrated through example scenarios. The effects of our optimization approach on the percentage of base quality receiving peers are examined through simulations.**

I. Introduction

Several instant messaging (e.g., Microsoft Messenger^{TM}) and voice over IP applications (e.g., Skype^{TM}) allow pair-wise video communications; however, multipoint (MP) video conferencing is still not popular mostly because of the bandwidth demands of video transportation. Low bandwidth connections like wireless GPRS are barely enough for point-to-point video communications let alone supporting multipoint video. Moreover, to obtain a better quality video, users tend to use as much bandwidth as it becomes available.

An alternative approach for MP video conferencing was presented in [1]. This system is based on a distributed peer-to-peer (P2P) approach, where each participant could see any other participant in most cases. The extension in [2] makes use of layered video, to guarantee that each participant can see any other participant in all cases. Although some users may have to view lower quality video (i.e., base layer), it is shown that this is a small percentage of the participant count and tends to decrease as the participant count increases; thus, making the system scalable.

There are a large number of P2P streaming systems reported in the literature. In [3], a layered P2P streaming scheme for on-demand media distribution is proposed. The asynchrony of user requests and heterogeneity of peer network bandwidth together with cache-and-relay and layer-encoded streaming techniques are utilized to efficiently use peers' bandwidth, maximize streaming quality of peers and save server bandwidth, making the system scalable. In ZigZag [4], a P2P architecture for distributing media from a single source is described and analyzed. In [5], an implementation of video conferencing through an end system multicast has been presented. This system assumes that the peers have large upstream bandwidths and there is only a single source. Another P2P approach for MP video conferencing is presented in [6]. However, the approach is based on a centralized architecture.

Our approach does not assume high bandwidth connections; it makes use of layers of video with low bandwidth requirements, but with acceptable quality [2]. The approach is fully distributed and does not require a centralized server, thus, preventing single-point of failures. The system targets small groups of participants (i.e., in the order of tens of participants), where all peers can act as sources of video without interrupting the others.

Although previous work described an optimization approach to minimize the number of base layer receivers, the delays between the peers and their connection diversity were not taken into account. In this work, we present a multi-objective optimization technique that can be employed within the system described in [1] and [2]. Next section gives an overview of the problem, describing our assumptions and considerations. Section III defines the optimization objectives. Section IV explains the details of the multi-objective optimization approach. Section V presents the simulation results. Section VI outline overall discussions. Section VII suggests possible future work.

II. Overview

Before continuing to explain each of the objectives separately, we first describe our assumptions. We give the general problem description and the solution we propose.

[1]**This work is supported in part by TUBITAK (The Scientific and Technical Research Council of Turkey) under CAREER Award Grant 104E064.**

We also present the computational complexity of our solution and show that it can be applied within our system.

A. Assumptions

Each peer is assumed to have a connection bandwidth that can support *at least* one full quality video (i.e., base *and* enhancement layers) to be received and sent at the same time. Some peers may have download bandwidths to enable them to receive more than one full quality video simultaneously. Likewise, some peers may have upload bandwidths that can support more than one full quality video at the same time. These will be addressed in the multi objective optimization formulation as well.

We assume that each video source peer, called a "chain head", obtains the packet delay to other peers receiving its video, i.e., its "chain mambers", by gathering the round-trip-time (RTT) values (i.e., the time a packet takes to go from one peer to another and back) during the session. RTT values are assumed to be symmetric, so that $d_{i,j}$, (i.e., the delay in the direction from peer i to peer j) is the same as $d_{j,i}$ (i.e., the delay in the direction from peer j to peer i) and one-way delays can be taken as RTT/2. These delay values are stored in a table by the chain head and updated periodically. A time synchronized algorithm [1] may be used for measuring one-way delays, however, the RTT information seems to be sufficient without complicating the system.

The computational burden of the system is shown to be small [1]. Delays other than the end-to-end delays (e.g., processing, switching, forwarding) are assumed to be negligible compared to the network delays.

We also assume that a participant can make at most two video requests, even if it has a bandwidth that is sufficient to view more than two video signals. The reason behind this assumption is that the focus of a participant cannot cover interactions with more than two peers at the same time. Although this may look like an artificial assumption, the situation is similar to whenever a group of people interact with each other in a face-to-face conference. In [6], it is argued that in a video conference *mostly* two persons are in a participant's view.

Besides, not allowing a peer p make more than two video requests, guarantees that each video request it receives can be granted. Suppose that peers were allowed to make three video requests and p already watches the videos of $r1$ and $r2$ that it forwards to other peers. p's third request can be granted by another peer $r3$ by adding it to the end of its chain where p would not have to relay it to another peer. However, if another peer s requests p's own video signal, the request would have to be denied, because p's uploading bandwidth can only support two base layers at a time and is already occupied fully (i.e., sending $r1$'s and $r2$'s video signals). Limiting the number of video requests to two prevents this situation and guarantees that each *first* request, even in the case of multiple requests, are to be granted.

B. Problem Definition

Each chain head that receives a video request needs to configure its chain accordingly to grant the request *and* to achieve the defined objectives. This is done using *only* local information, and brings the advantage of not setting up a global information exchange mechanism to be used whenever a request is made. Omitting this overhead also allows the chain head to decide fast, since the chain of a peer needs to be updated dynamically, whenever a video request is received by the chain head.

Optimal operation of such a system has several objectives such as best possible video quality, minimum delay and support for additional requests. One can optimize each objective separately, but as we will show in Section IV, these objectives conflict, so that one solution for an objective would cause another objective to fail. Therefore, we need an approach that will optimize *all* objectives simultaneously. One can choose one objective to be achieved and set constraints for the others (e.g., the number of base layer receivers should be less than c_b where c_b is an integer less than the number of participants in the session). However, this approach has the disadvantage of having to choose the main objective to be achieved. Another disadvantage is to decide for the values of the constraints.

We claim that any comparison between the objectives to decide their order of importance cannot be justified. Instead, we combine these objectives by making use of the preferences of the participants. Since they are the ones who would get affected by the chain configuration that is going to be employed by the chain head, this is a reasonable solution.

In order to formulate the objectives, we define the following variables.

- i**:** id of the peer
- c**:** a *possible* chain configuration
- l_i**:** the length of the chain headed by peer i (i.e., the cardinality of the set consisting of the peers that receive peer i's video signal)

Let $f_{v,c}(o)$ be an integer valued function of the positions of peers that return the id of a peer given its position, o, in a *possible* chain configuration c headed by peer v.

Suppose that, there are three peers receiving the video of peer 1, namely peer 3, 4 and 5. One possible chain configuration of peer 1 is given by: $<4, 3, 5>$. In this particular chain configuration, $f_{1,<4,3,5>}(3) = 5$. This configuration is illustrated in Figure 1. The arrows indicate the direction of video transmission. F stands for full video quality and H stands for base layer quality.

The number of possible chain configurations headed by peer v is given by the factorial of the number of the peers receiving the video of v, namely l_1, the chain length of peer 1. In this example, l_1 = 3, so the number of possible chain configurations is 3! = 6. Since the chain head would always be in the 0'th order, it is omitted in the chain representation.

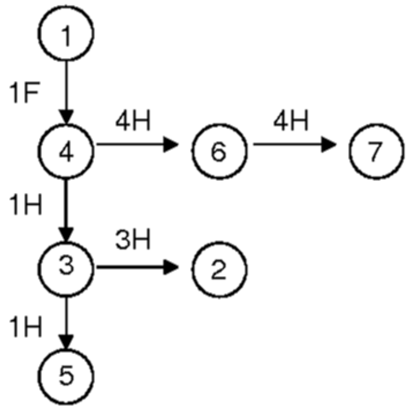

Fig. 1. A possible chain configuration of peer 1 with the set of receivers 3, 4, 5. The chain is represented as <4, 3, 5>. Peer 4 and peer 3 are also heads for the chains, <6, 7> and <2>, respectively. Arrows indicate the direction of video transmission. F stands for full video quality and H stands for base layer quality.

C. Computational Complexity Considerations

The solution space consists of the possible chain configurations headed by a peer v. The use of combinatorial optimization by exploring all the possible solution space gives a computational complexity of $O(n!)$. This would be problematic when the chain of a peer becomes very long (i.e., $l_v > 10$). Since our target group size for this application is small (i.e., participant count < 10), the enumeration of possible chains is not as costly as applying special optimization formulations or running a special optimization software. This way, each chain head can update its chain configuration dynamically and in real-time.

III. Optimization Objectives

Although previous work [2] considered the number of base layer receivers in the system as the optimization objective, the delay experienced by the peers is not considered as a parameter in the optimization process. Furthermore, it assumed that end-points had low bandwidth connections that can support only one video signal to be sent and received, although some peers may have higher bandwidth and asymmetric (i.e., download rate $>$ upload rate) Internet connections. In this section, we define the optimization objectives to consider the delays experienced by the peers and to support peers that have asymmetric bandwidth connections.

A. Objective 1: Minimize the number of base layer receivers

By using layered video, the system allows each conference participant to see any other participant at any given time under *all* multipoint configurations of any number of users, with a caveat that some participants may have to receive only the base layer video. An objective of the system is to maximize the quality of the video each participant receives, that is, both the base and the enhancement layers of the video of a requested peer should be delivered, as long as this does not require the denial of another participant's request. Therefore, the system's objective is *to minimize the number of participants receiving only the base layer video in a given configuration.*

The following variable is defined to formulate the objective function.

$k_{v,c}$: the number of lower quality video receivers in a *possible* chain configuration c headed by peer v.

Consider the example chain configuration of peer 1 in Figure 1. Peers 3 and 4 in peer 1's chain are also chain heads. Let $f^{-1}_{v,c}(j)$ be the inverse function of $f_{v,c}(o)$, so that it gives the position of the peer with the id j in a *possible* chain configuration c. In this example, $f^{-1}_{1,<4,3,5>}(3) = 2$ and $f^{-1}_{1,<4,3,5>}(4) = 1$. Let o be the smallest of these values, namely 1. Also, suppose that peer 3 has a chain length of 1 and peer 4 has a chain length of 2 as depicted in Figure 1. So $k_{v,c}$, the number of lower quality video receivers in a possible chain configuration c headed by peer v, is calculated as

$$k_{v,c} = \begin{cases} 0 \quad \text{if } o = l_v \textbf{ or } \textit{no such peer} \\ l_{f^{-1}_{v,c}(o)} + 1 + \sum\limits_{j=o+1}^{l_v-1} \left(l_{f^{-1}_{v,c}(j)} + 1 \right) \quad \text{otherwise} \end{cases}$$

The number of base layer receivers in a possible chain configuration c is 0, if $o = l_v$ (i.e., the corresponding peer is at the end of the chain) or there is no such peer that is acting as the head of a chain and a relay simultaneously, so that every peer receives full quality video. Otherwise, it is calculated as given above.

Remember that peers 3 and 4 are chain heads and relays at the same time. This means that the peers in the chain of peer 3 (i.e., peer 2) and the peers in the chain of 4 (i.e., peer 6 and peer 7) receive only the base layer, like peers 3 and 5. In this particular chain configuration of peer 1, the total number of base layer receivers is 2+1+1+1 = 5. (i.e., the chain length of peer 4 + peer 3 + the chain length of peer 3 + peer 5). Note that, if peer 5 was also a chain head; since it would not relay, its chain would receive full quality video and thus, would not be included in the sum.

Consequently, the first objective function $g_v(c)$ is defined as

$$g_v(c) = k_{v,c} \tag{1}$$

B. Objective 2: Minimize the maximum delay experienced by a peer

Since a video signal might have to be forwarded from a peer to another, this may cause the peers located towards the end of a chain experience large delays. Another objective of the system is *to minimize the delays experienced by the peers in a chain.* Since the maximum delay in a chain will be experienced by the peer at the end of the chain, we aim to minimize the delay of that peer and do our calculations based on this. We define the delay of a chain as the delay experienced by the peer at the end of that chain.

In order to calculate the delay of a chain, the end-to-end delay information between the peers are needed by the chain head. To do this, we re-define some of the messages described in [1]. Whenever a peer makes a video request, it piggybacks the RTT information between itself and the other peers. Thus the requested peer has information about the end-to-end delays between the requesting peer and other peers, and it can calculate the delay of any possible chain configuration.

Besides the request message, the keep-alive message is also re-defined. Keep-alive messages are sent by the members of a chain to the chain head periodically, so that the chain head can find out whether a peer has crashed or lost its connection and, then rearrange its chain accordingly. These messages are also used to keep the head informed of peers' current status and chain lengths (if they have any), so that the chain head can evaluate a possible chain configuration without needing to exchange any additional messages. With the modification, each member piggybacks the RTT information between itself and other peers in the conference to the keep-alive messages. This way, the chain head has all the information it needs to calculate the delay of any possible chain.

Since the maximum delay in a chain configuration is experienced by the peer at the end of the configuration c, the following holds.

$d_{v,c}$**:** delay experienced by the peer at the end of a *possible* chain configuration c headed by peer v.

The chain head receiving a video request calculates the delay value of each possible chain configuration by adding the one-way delay values between the peers.

$$d_{v,c} = \sum_{j=1}^{l_v} d_{f_{v,c}(j-1), f_{v,c}(j)}$$

The second objective function $h_v(c)$ is defined as

$$h_v(c) = d_{v,c} \quad (2)$$

The possible chain configurations and their delay values for the example given in Figure 1 are given below.

$$h_1(<3,4,5>) = d_{1,3} + d_{3,4} + d_{4,5}$$
$$h_1(<3,5,4>) = d_{1,3} + d_{3,5} + d_{5,4}$$
$$h_1(<4,3,5>) = d_{1,4} + d_{4,3} + d_{3,5}$$
$$h_1(<4,5,3>) = d_{1,4} + d_{4,5} + d_{5,3}$$
$$h_1(<5,3,4>) = d_{1,5} + d_{5,3} + d_{3,4}$$
$$h_1(<5,4,3>) = d_{1,5} + d_{5,4} + d_{4,3}$$

C. Objective 3: Maximize the number of additional requests granted

The P2P approach we presented in [1] and [2] assumed that the peers have a connection bandwidth that is enough to send and receive only one video signal. However, the diversity and the asymmetry of the Internet connections connections lead to the fact that peers may have spare bandwidth for sending and receiving more than one video signal.

Having an upload bandwidth that can support more than one video signal is always beneficial. Suppose a peer *R* receives a video signal (i.e., watches another peer *H*) and forwards it to another peer *E* in a chain. Whenever another peer *A* requests the video of peer *R*, peer *R* may need to drop the quality of the forwarded video to base layer, in order to be able to send its own video with the remaining bandwidth (also in base layer). If peer *R* has an upload bandwidth to support more than one video signal at the same time; however, it does not need to drop the quality of the forwarded video, instead it uses its spare bandwidth to grant the request. So, peer *E* and peer *A* would both receive full quality video (i.e., base + enhancement layers), instead of only the base layer. This applies to all similar situations whenever a peer needs to be a chain head and relay at the same time. Therefore, we do not have to consider the cases where peers have spare bandwidths for upload in optimal system operation. Rather we concentrate on peers having spare bandwidth to receive more than one video signal (i.e., watching more than one peer).

Considering these, another objective of the system is *to maximize the number of such additional requests that are granted*. Thus, we formulate the objective so that the value of the objective function is 1 if the additional request can be granted, whereas it is -1 if the request is to be declined. Every chain head investigates each possible chain configuration c whenever a peer makes an additional video request to the corresponding chain head.

$$s_{v,c} = \begin{cases} -1 & \text{if the request is not granted} \\ 1 & \text{if the request is granted} \end{cases}$$

The third objective function $m_v(c)$ is defined as

$$m_v(c) = s_{v,c} \quad (3)$$

IV. Multi-objective Optimization

In this section, we will describe our multi-objective optimization approach. We will first present the mechanism to assign importance weights to each objective using the preferences of the peers. Then, we will demonstrate how the technique is applied to the system through example scenarios.

A. Formulation

In order to determine the best solution, we use the weighted sum method [7]. The issue of determining importance weights to be assigned to each objective is overcome by employing a preference mechanism. Peers being aware of the optimization objectives choose one of the objectives as their preference. These will be exchanged during the initialization of the conference. Peers may change their preferences during the conference, but they need to inform the others. The assigned importance weights of each objective function f_i is defined as

$$w_{f_i} = \frac{p_{f_i}}{n} \quad (4)$$

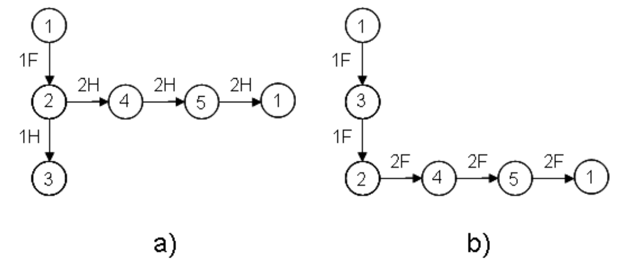

Fig. 2. a) If the importance weights are assigned only with respect to the preferences of the peers in a chain. b) If the importance weights are assigned according to the preferences of all peers. The majority of the peers get what they want: maximum video quality.

where p_{f_i} represents the number of peers that prefer f_i to the other optimization objectives. n is the number of participants in the conference.

The importance weights are determined using the number of participants in the *entire* conference (i.e., n), rather than the number of participants in a corresponding chain. The reason is that a peer's preference (e.g., a chain head) may affect other peers' (e.g., the peers in its chain) received video quality. Consider the conference in Figure 2 with 5 participants. Suppose that peers 2 and 3 prefer minimum delay and peers 1, 4 and 5 prefer maximum video quality. Peers 1 and 2 are geographically closer, so the chain configuration in Figure 2a will be employed by the chain head (i.e., peer 1) to minimize the delay. This will force the other chain head (i.e., peer 2) to send its own video signal in base layer quality, although *all* peers in its chain prefer maximum video quality. Since they can receive only the base layer, their preferences will have no effect, even if they constitute the majority in the conference. Therefore, rather than using only the preferences in the corresponding chain, all peers' preferences are taken into account while determining the importance weights. As a consequence, the configuration in Figure 2b should be used to satisfy the majority of the peers (i.e., 3 out of 5 peers).

Each chain head v would calculate the scaled versions of the optimization functions while determining which chain configuration they are going to employ whenever they receive a video request message. The formula for that is

$$f_{i,v,scaled}(c) = w_{f_i} \frac{f_{i,v}(c) - f_{i,v,min}(c)}{f_{i,v,max}(c) + f_{i,v,min}(c)} \quad (5)$$

where $f_{i,v,min}$ and $f_{i,v,max}$ represent the minimum and the maximum value of that optimization function, respectively. The combined objective function would be $u_v(c)$ as given below.

$$u_v(c) = min(\sum_{m} f_{i,v,scaled}) \quad (6)$$

where m is the number of optimization objectives that are used in the multi-objective solution. The chain head v would employ c^*, the configuration optimizing the objective function.

TABLE I
LATENCY TABLE

peer id	1	2	3	4	5	6	7
1	0	26	**89**	**24**	**95**	66	70
2	26	0	104	78	108	73	75
3	**89**	104	0	**85**	**20**	55	42
4	**24**	78	**85**	0	**98**	65	71
5	**95**	108	**20**	**98**	0	53	47
6	66	73	55	65	53	0	12
7	70	75	42	71	47	12	0

B. Example scenario: Minimize the number of base layer receivers and the maximum delay in a chain

In this part, we will illustrate the multi-objective optimization technique with an example scenario. For simplicity and ease of understanding, we assume that all peers have sufficient connection bandwidth only for one full-quality video. Thus, we do not take the additional requests into account; they will be further investigated in the second example.

Each chain head v should try *to minimize the number of base layer receivers in its chain* and *minimize the maximum delay experienced in that chain*, at the same time. First, we will show that these optimization objectives may be conflicting with each other.

Suppose there is a conference session with 7 participants. Peers 1 and 4 are located in the USA, peers 3 and 5 are located in Turkey, peers 6 and 7 are located in Germany and peer 2 is located in Canada. Typical one-way delay values between the peers are given in Table I.

Let the video request configuration of this conference be the following: Peers 3, 4 and 5 request to view peer 1's video, peer 2 requests peer 3's video and peers 6 and 7 request peer 4's video. So peers 1, 3 and 4 have chains with lengths 3, 1 and 2, respectively.

Suppose that peer 1 needs to configure its chain, so that the number of base layer receivers and the maximum delay experienced by a peer are minimized. The minimum number of base layer receivers is achieved by the chain configuration <5, 3, 4> (i.e., peer 4 has the longest chain length, and thus, it should be at the end of the chain, giving a total of 2 base layer receivers). The minimized maximum delay is achieved by the chain configuration <4, 3, 5> and yields a maximum delay of 129 ms as calculated from Table I. These possible configurations can be seen in Figure 3.

$g_1(< 4, 3, 5 >)$ = 5 and $h_1(< 4, 3, 5 >)$ = **129 ms**
$g_1(< 5, 3, 4 >)$ = **2** and $h_1(< 5, 3, 4 >)$ = 200 ms

If the chain head (i.e., peer 1) *were* to minimize *only* the number of base layer receivers, then the chain configuration it should be employing would be <5, 3, 4>. On the other hand, if it *were* to minimize *only* the maximum delay in a chain, then the chain configuration <4, 3, 5> should be employed. Clearly, these two objectives conflict with each other. Figure 4 shows details of this example.

In our scenario, peers 5 and 6 prefer maximum video

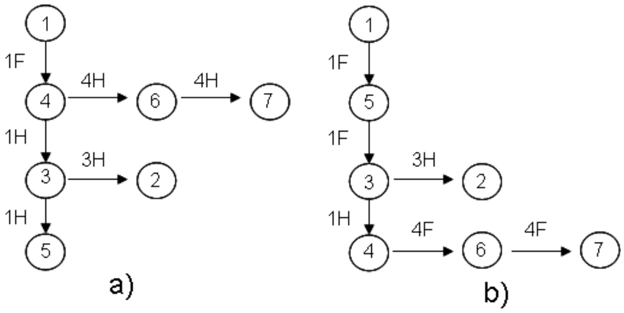

Fig. 3. a) Chain configuration <4, 3, 5> yielding 5 as the number of base layer receivers (Peers 6, 7, 3, 2 and 5). b) Chain configuration <3, 5, 4> yielding 2 as the number of base layer receivers (Peers 2 and 4).

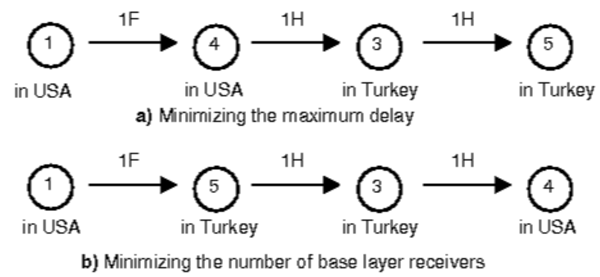

Fig. 4. a) Correct chain order to minimize the maximum delay. b) Correct chain to minimize the number of base layer receivers.

quality and peers 1, 2, 3, 4 and 7 prefer minimum delay. So, p_g = 2 and p_h = 5. Then the weights would be w_g = 0.29 and w_h = 0.71. $g_v(c)$ and $h_v(c)$ values are scaled according to the equation (5). The best solution is determined according to the optimization function $u_v(c)$. The entire set of the possible chain configurations and their $g_v(c)$ and $h_v(c)$ values are given in Table II.

According to Table II, the chain configuration <4, 5, 3> gives the minimum value of the objective function. This chain is employed by the chain head. Remember that, 4 of 6 peers had told that they would prefer minimum delay over high quality of video. We can see the trade-off between minimizing the delay and minimizing the number of base layer receivers. Peer 1 employs a chain that has the *second* best minimized delay and *third best* minimized number of base layer receivers.

Table III gives $g_v(c)$ and $h_v(c)$ values calculated with the preference values $w_{pref,g}$ = 0.71 and $w_{pref,h}$ = 0.29, so that now 2 of 7 peers prefer high video quality over minimum delay.

Since the preference values have changed, peer 1 has to change its chain to the best configuration determined by the objective function. The chain configuration <5, 3, 4> has the minimum $g_v(c)$ value and has the *third best* delay. The compromise is done to match the preferences of the peers.

TABLE II

$g_1(c)$ AND $h_1(c)$ VALUES WITH w_g = 0.29 AND w_h = 0.71.

c	$g_1(c)$	$h_1(c)$	$g_{1,scaled}(c)$	$h_{1,scaled}(c)$	$u_1(c)$
<3, 4, 5>	5	272	0.29	0.68	0.97
<3, 5, 4>	3	207	0.10	0.37	0.47
<4, 3, 5>	5	129	0.29	0.00	0.29
<4, 5, 3>	**4**	**142**	**0.19**	**0.06**	**0.25**
<5, 3, 4>	2	200	0.00	0.34	0.34
<5, 4, 3>	3	278	0.10	0.71	0.81

TABLE III

$g_1(c)$ AND $h_1(c)$ VALUES WITH w_g = 0.71 AND w_h = 0.29.

c	$g_1(c)$	$h_1(c)$	$g_{1,scaled}(c)$	$h_{1,scaled}(c)$	$u_1(c)$
<3, 4, 5>	5	272	0.71	0.28	0.99
<3, 5, 4>	3	207	0.24	0.15	0.39
<4, 3, 5>	5	129	0.71	0.00	0.71
<4, 5, 3>	4	142	0.47	0.03	0.50
<5, 3, 4>	**2**	**200**	**0.00**	**0.14**	**0.14**
<5, 4, 3>	3	278	0.24	0.29	0.53

C. Example scenario: Minimize the number of base layer receivers and maximize the number of additional requests granted

In this example, we assume that the peers are geographically close to each other and thus, do not take the delays into account. The peers prefer either maximum video quality or additional video requests.

The conference has 6 participants as illustrated in Figure 5. The chains of peer 1, peer 3 and peer 5 consist of peers 2 and 3, peers 4 and 5, and peer 6, respectively. Suppose that peer 5 has sufficient bandwidth to make an additional video request and that it requests peer 1's video. If peer 1 would arrange its chain just to minimize the number of base layer receivers, peer 5's additional video request would be rejected. Figure 5a depicts this situation. If the request were granted employing a chain configuration like in Figure 5b, that would mean that 3 peers would receive base layer video (i.e., 3 requests would be granted, but the requesters would receive base layer video; requesters 4, 5 *and* 5). However, the configuration shown in Figure 5c would allow a smaller number of requesters to receive base layer video (i.e., only 2; peers 2 and 6).

As can be seen, the two objectives conflict again, as one requires that the request is rejected and the other requires that some peers receive base layer video. Assuming that 4 out of 6 peers would prefer that additional requests are

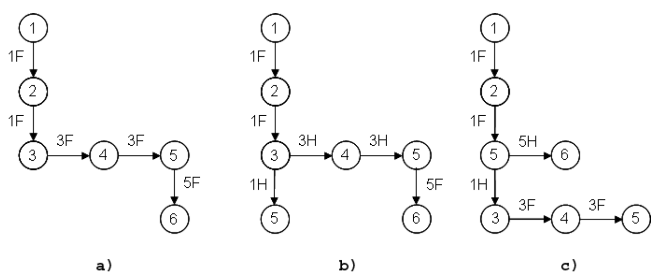

Fig. 5. a) The configuration after peer 5's additional request is rejected. b) A possible chain configuration of peer 1 after it granted peer 5's additional request. c) Another possible chain configuration of peer 1 with only 2 base layer receivers.

TABLE IV
$g_1(c)$ AND $m_1(c)$ VALUES WITH w_g = 0.33 AND w_m = 0.67.

c	$g_1(c)$	$m_1(c)$	$g_{1,scaled}(c)$	$m_{1,scaled}(c)$	$u_1(c)$
<2, 3, 5>	3	1	0.20	0.67	-0.47
<2, 5, 3>	**2**	**1**	**0.13**	**0.67**	**-0.53**
<5, 2, 3>	3	1	0.20	0.67	-0.47
<5, 3, 2>	5	1	0.33	0.67	-0.33
<3, 2, 5>	4	1	0.27	0.67	-0.40
<3, 5, 2>	5	1	0.33	0.67	-0.33
<2, 3>	0	-1	0.00	0.00	0.00

TABLE V
$g_1(c)$ AND $m_1(c)$ VALUES WITH w_g = 0.83 AND w_m = 0.17.

c	$g_1(c)$	$m_1(c)$	$g_{1,scaled}(c)$	$m_{1,scaled}(c)$	$u_1(c)$
<2, 3, 5>	3	1	0.50	0.17	0.33
<2, 5, 3>	2	1	0.33	0.17	0.16
<5, 2, 3>	3	1	0.50	0.17	0.33
<5, 3, 2>	5	1	0.83	0.17	0.66
<3, 2, 5>	4	1	0.67	0.17	0.50
<3, 5, 2>	5	1	0.83	0.17	0.66
<2, 3>	**0**	**-1**	**0.00**	**0.00**	**0.00**

granted over the video quality, the importance weights for the objective functions would be: $p_g = 2/6 = 0.33$ and $p_m = 4/6 = 0.67$.

The $u_v(c)$ values for all chain configurations are presented in Table IV. Since the second objective aims to *maximize* the number of additional requests granted, its value is negated to *minimize* the overall objective function $u_v(c)$. According to the Table IV, the chain configuration <2, 5, 3> is the best-compromise solution and thus, would be employed by the chain head (i.e., peer 1). Since 4 out of 6 peers prefer that additional requests are granted, the system has tried to maximize that number as well as to minimize the number of base layer receivers at the same time.

Table V shows how the objective function values change as the importance weights change when only one peer would prefer that additional requests are granted. Since the chain head would calculate the objective function values according to the new weights and employ the best-compromise chain configuration, the second request of peer 5 would be rejected.

V. Simulation results

In this section, we present simulation results covering possible scenarios with up to 10 participants. Each participant requests the video of another participant randomly. Participants with larger bandwidths make additional video requests. We again do not take the delays into account.

For each conference case with different participant counts, we generated 100,000 cases randomly, in which the number of participants with additional bandwidths is increased from 1 up to the participant count for that case. For example, for a conference scenario with 6 participants, we generated 100,000 random cases with only one participant having additional bandwidth; we generated another 100,000 with two of the participants having additional bandwidth and so on.

```
1  if p is relay
2       if p can be moved to the end
3            move p to the end
4       else
5            move p as near to the end as possible

6  if r is chain head and relay
7       if p's last member is chain head and relay
8            reject request
9       else if p's last member is relay
10           reject request
11      else
12           add r to the end
13 else
14      generate possible chains of p with and without r
15      select best-compromise chain and employ it
```

Fig. 6. Pseudo code to handle a request.

During our simulations, we assumed that the participants with no additional bandwidth prefer maximum video quality and the rest prefers that additional requests are granted. This may be plausible, because a participant with additional bandwidth would be more likely willing that additional requests are granted. The importance weights for maximizing the video quality and maximizing the number of granted additional requests are calculated accordingly.

Whenever a request comes to a participant p from a participant r, the actions that p can take are given in the pseudo code (Figure 6). In the first part (lines 1-5), the status of the requested participant p is checked. If it is relaying video, the chain head of p tries to move p to the end of the chain, so that it would not forward the video anymore. This way, the number of base layer receivers in the chain is minimized [2]. If the last member of the chain has a shorter chain or no chain, the head of p can move p to the end. If not, then p is moved to a position where the number of base layer receivers is minimized according to our optimization formulations described in [2] (i.e., near the end of the chain).

The second part deals with the request. If the requesting participant r is a chain head and a relay at the same time, the status of the last member in p's chain (i.e., l) needs to be checked (lines 6-12). There can be three different status for l. These are explained in detail below.

l is a chain head and a relay: The last member l might have made two requests, for p's and o's video signals. Also, at least one other participant might have requested l's video, so l is a chain head. Suppose that o's chain requires that l must relay video. Since l is also a chain head, it will relay only the base layer of o's video. The remaining bandwidth will be used for l's own video. The only way that l receives p's video is that it is at the end of p's chain. Once another participant r, a chain head and a relay itself, requests p's video, the only position r could be is at the end, but since l cannot be in another position in p's chain and relay p's video to r, the request is rejected.

l is not a chain head, but is a relay: Similarly, l might have not been a chain head, but only a relay for o. Suppose that another participant m did not make a request yet and we allowed r relay two different video signals, so that l also relays p's video to r. When m makes its first request to receive l's video, this would be rejected, because l's uploading bandwidth is used for relaying two base layer video signals (i.e., p's and o's). However, our condition that each participant can see any other participant anytime contradicts with this. So, we do not allow that a participant relay two different video signals. Therefore, r's request is rejected. Remember that r had additional bandwidth and the request was its second. r has already a request granted, so our condition holds.

l **is neither a chain head, nor a relay:** The rest of the code is for when r is not a chain head and a relay at the same time. This means that it could also be relaying in p's chain, if the best-compromise chain requires it to do so. Participant p generates all possible chain configurations with r in its requester list. The best-compromise chain is selected by the head p using the calculations given in Section III regarding the importance levels. According to the best-compromise chain, the request is either granted or rejected.

The results of the scenarios for each participant count are averaged. Figure 7 shows the percentages of the rejected requests, granted base layer video receiving requests and granted full quality receiving requests. The percentage of the rejected requests does not exceed 15% and decreases as the number of participants increases. Although the percentage of the base layer video receiving requests increases with the participant count, this increase is asymptotic. Our system was able to grant at least 50% of the requests to receive full quality video.

Even when additional requests were not granted, there was a slight increase in the base layer receiving requests with the increase in the participant count. However, the ratio of the average number of base layer receiving requests to the total number of requests is decreasing as the participant count increases [2]. For the multi-objective case, simulations show that the increase in the participant count, increases the number of base layer video receiving requests as well. But this increase is also asymptotic and does not damage the system's scalability.

Instead of rejecting the requests, the system makes use of layered video and grants the additional requests according to the best-compromise chain found. Increased participant count increases the probability for a request to receive base quality layer video; however, in cases where base layer video is used, the average percentage of these requests to all requests stays below 45% percent and is generally about 39% (Figure 8).

In some cases, there may be rejected requests; however, all of these requests are additional requests, so that the requesters already receive a participant's video. Our system tries to maximize the number of granted additional requests, as long as this does not cause other participants' requests to be rejected.

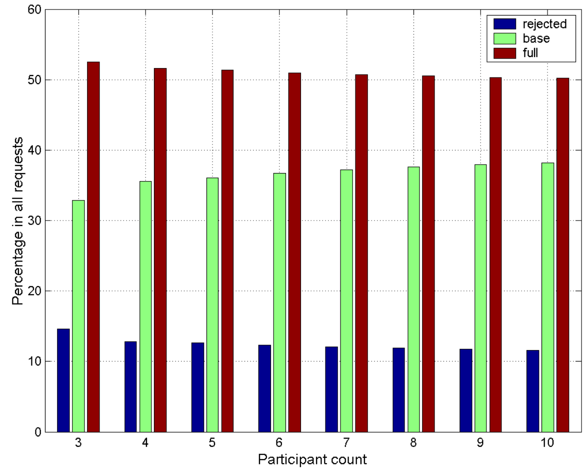

Fig. 7. Percentages of all requests.

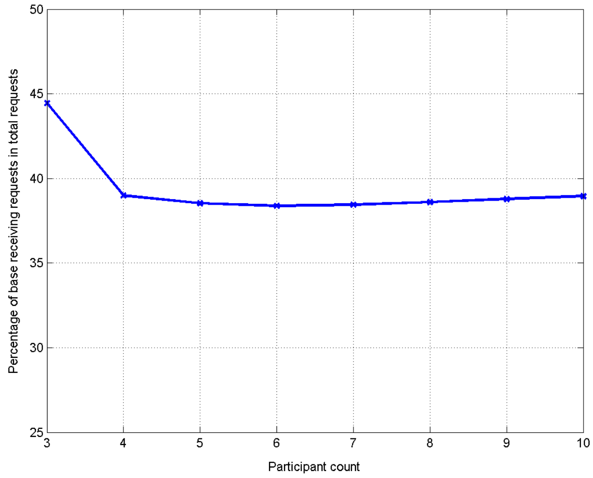

Fig. 8. Average percentage of base layer receiving requests to total requests in cases where base layer is used.

VI. Discussions

The best-compromise solution makes a trade-off between the $g_v(c)$ and $h_v(c)$ or $m_v(c)$ values. Each change in the number of base layer receivers may cause a change in the other objective value. The maximum delay experienced by the peers in the corresponding chain may increase or decrease; an additional request of a peer may be rejected or granted. However, one cannot make a clear statement saying that 'a change in the number of base layer receivers corresponds to a certain increase or decrease in the maximum delay of that chain' or 'a change in the number of base layer receivers corresponds to a certain number of additional requests to be granted or rejected'. The uniqueness of the video request set (i.e., which peers request the video of which other peers), the pair-wise delay values and which peers have sufficient bandwidth to make additional requests prevent this.

Since we cannot know how much delay is worth how many base layer receivers or how many full quality video receivers we can sacrifice to grant one more additional request, we need a way to determine the sensitivity of this trade-off. Therefore, we need to assign importance weights to the objective functions. This assignment is a hard task which is overcome by the preferences of the peers. There is no way to compare the number of base layer receivers with the delay values or additional requests to conclude one is more important than the other. The preferences of the peers are used to determine the importance weights of the optimization objectives. This is a plausible assumption since the peers will be the ones affected by the employed chain configuration.

The system makes use of layered video to guarantee that each video request at each participant is granted. With the presented extensions, the system makes compromises between the quality of the video peers receive, delays experienced by peers and additional requests of peers with sufficient bandwidth. Our multi-objective formulation successfully finds the best-compromise solution accordingly. We have shown with a counter-example that the solution should be calculated according to the preferences of *all* participants in the conference, not just in a corresponding chain.

The sensitivity analysis showed that our objective function reacts to the changes in the preferences of the participants and satisfies them in the best way. Although the employed solution causes more requesters to receive base layer video quality, simulation results show that this does not hurt the system's scalability. Since the intended group size is relatively small, generation of possible chain configurations does not bring much computational burden. This makes the system easy to implement.

VII. Future Work

Future work includes exploring of other multi-objective optimization techniques, in which greedy approaches may be employed. The number of additional granted requests may be increased until the number of base layer receivers exceeds some threshold. A possible threshold value is the half of the participant count. The number of base layer receivers does not exceed 50% of the participant count as shown in [2]; so this would be a good candidate. However, this may require a mechanism to exchange information about the number of base layer receivers between the chain heads.

Also, an algorithm for better estimation of the one-way delays between the participants may need to be implemented, instead of assuming symmetric delays.

Instead of denying an additional request when the combined objective function does not allow it to be granted, one may queue the requesting peer r. This way, this peer will have higher priority than a new requester n, when the chain head receives a new video request and tries to update its chain. However, this may cause the problem of having the peer n wait in queue a long time when the combined objective function never allows the request of r to be granted.

All three optimization objectives (minimize the number of base layer receivers, minimize the maximum delay experienced in a chain, maximize the number of additional requests granted) may be combined to achieve a more complete multi-objective optimization solution.

Acknowledgments

We would like to thank Sibel Salman, Serdar Tasiran, Murat Tekalp and the anonymous reviewers for their help in shaping the final version of this paper.

References

[1] M. R. Civanlar, O. Ozkasap, T. Celebi, "Peer-to-peer multipoint videoconferencing on the Internet," *Signal Processing: Image Communication*, Elsevier, 20 (8), pp. 743–754, 2005.

[2] I. E. Akkus, M. R. Civanlar, O. Ozkasap, "Peer-to-peer multipoint video conferencing using layered video," in *Proc. IEEE ICIP2006*, Atlanta, GA, 2006.

[3] Y. Cui, K. Nahrstedt, "Layered peer-to-peer streaming," in *Proc. NOSSDAV'03*, Monterey, CA, 2003.

[4] D. A. Tran, K. A. Hua, T. Do, "A peer-to-peer architecture for media streaming," in *IEEE Journal on Selected Areas in Communications*, 20(1), pp. 121–133, 2004.

[5] Y. Chu, S. G. Rao, S. Seshan and H. Zhang, "Enabling conferencing applications on the internet using an overlay multicast architecture," in *Proc. ACM SIGCOMM'01*, San Diego, CA, 2001.

[6] M. Hosseini, N. D. Georganas, "Design of a multi-sender 3D videoconferencing application over an end system multicast protocol," in *Proc. ACM Multimedia 2003*, Berkeley, CA, 2003.

[7] L. Zadeh, "Optimality and non-scalar-valued performance criteria," in *IEEE Trans. Automatica Control*, vol.8, pp. 59–60, 1963.