Perception-based Application Layer Multicast Algorithm for scalable video conferencing

Tien Anh Le, Hang Nguyen

Perception-based Application Layer Multicast Algorithm for scalable video conferencing

Tien Anh Le, Hang Nguyen

***Abstract*—In this research work, a new application layer multicast algorithm using distributed service architecture and scalable video coding is proposed for scalable video conferencing services. The proposed algorithm considers the limitations of the human perception while participating in a video conference so as to minimize traffic that is not necessary for the communication session. A theoretical analysis was conducted to compare the total waiting time between a centralized and a distributed queues. The result shows that distributed service architecture can, by applying the newly proposed multicast algorithm, provide a much smaller delay than the centralized service architecture.**

***Index Terms*—application layer multicast; architecture analysis; video conference; service architecture;**

I. Introduction

Video communications is foreseen as the next popular digital communication method after voice communications[1]. Video conferencing, the most complex type of video communications, has been used for a number of years in business, and more recently, in everyday life. A very common conventional conferencing method is to use a Multipoint Control Unit (MCU) as the central point for receiving, transcoding, trans-rating, mixing, and then sending back to participants (centralized service architecture). However, the cost of conventional conferencing method increases exponentially with an increasing number of participants. Moreover, it cannot support satisfactorily users with different types of terminals due to scalability problems.

In[2], the centralized service architecture of conventional centralized conferencing services was combined with Scalable Video Coding (SVC) to provide more scalability. A new MCU architecture (Scalable Video Conferencing Server-SVCS) was proposed. Similar to the conventional centralized architecture, SVCS received SVC bit-streams from all participants. However, it directly routes bit-streams to their desired destinations without any mixing, or transcoding/rating. All of that workload is processed at the participants' terminals. To date, this is the most advanced method of centralized scalable video conferencing service and the one which is widely deployed because of its scalability and advanced performance compared with the conventional centralized methods.

When more participants want to participate in a conference (e.g. for large events), the cost of centralized architecture increases sharply. Therefore, distributed architectures have been proposed. Naturally, when a multi-point communication service is to be provided on a distributed scheme, multicast is involved. IP-multicast[3] is currently the most efficient type of multicast. However, deployment issues are preventing it from being widely applied[4]. In[5], a distributed architecture was proposed for video conferencing service based on an Application Layer Multicast (ALM) algorithm. However, that proposal did not consider the scalability problems associated with a variety of terminals. Moreover, the cost function used to construct the media distribution tree has not yet been investigated. Scalability is considered in[6],[7], and SVC was applied to support various types of participant terminals. Unfortunately, the proposed algorithms can only support a limited number of participants (e.g.10-15) since they do not have a mechanism to limit unnecessary traffic on the multicast tree.

To the best of our knowledge, none of the proposed distributed architecture systems has compared their performance with the conventional centralized architecture. Thus, no conclusion can be made on whether a particular distributed architecture is really better than the centralized architecture and therefore, if service users should change their current centralized conferencing system for a new distributed system. The first attempt to compare the two scalable conferencing architectures using its proposed simulation platform[8] is in [9]. Their results show that, even when using a much higher computational load, the MCU of a centralized architecture can only provide a bit-rate that is similar to that of the distributed architecture at the trade-off of a much higher delay. The results are interesting but must be validated by theoretical analysis before they can be applied in more general conditions.

In the research work presented here, we make the following contributions:

- **An ALM algorithm is proposed** in which the limitations of human perception when participating in a conference are considered with the objective of reducing the unnecessary traffic on the multicast tree,
- **Theoretical queuing models** for centralized and for the proposed distributed architectures are constructed and analyzed, and the total waiting time in each type of queue compared. The analysis has applied the real source model of SVC video published in[10].

II. Perception-based Application Layer Multicast Algorithm for scalable video conferencing services

The proposed ALM algorithm is used to build an ALM media distribution tree for Scalable Video Coding contents. The distance between any pair of peers is calculated based on the multi-variable cost function (Equ.1) proposed in[11]:

$$u(x_w, x_d) = \sqrt{\frac{x_w}{\kappa_w - x_w} . \frac{\kappa_d}{x_d - \kappa_d}} \quad (1)$$

In which κ_w and κ_d are the maximum available bandwidth and the minimum delay, respectively, that the network can

provide to the service using all available resources, and x_w, x_d are the required bandwidth and required delay, respectively, from the SVC application. A cluster is a group of peers that have the shortest distance to each other. When a peer wants to join an ALM, it will first try to explore its nearest clusters using a cost function to measure the distance to reach the leader of each cluster. A cluster has the maximum size of **k** peers depending on the network conditions. A leader is the peer that has the minimum distances to all other peers in a cluster. All leaders together form a first layer and then use the same multi-variable cost function to calculate their distance from all other leaders at layer 1, to form clusters and to form the second layer. A leader will receive bit-streams from its cluster's peers and forward them to the other peers in its cluster and to the upper layer's leaders (each peer within a cluster will receive bit-streams from all other peers in the same cluster but not its own bit- stream). At the same time, layer-one leaders receive bit- streams from the upper layer's leaders and forwards them to its cluster's members.

The proposed ALM algorithm is applied when SVC (or other kinds of multilayer video coding) is used on the video conferencing service. The main advantage of SVC is that the video can be encoded into a base layer bit-stream and several enhancement layer bit-streams. In a video conferencing session, at any given time there are only one or a few active speakers (active speakers are the conference participants who are giving the speech or actively participating in the argument/ discussion). Active speakers can be found automatically by comparing the participants microphones' output power levels. One simple justification is that, if there are too many active speakers in a conference session, other participants will not have enough perception capacity to follow all of them. From the multicast tree's point of view, an Auto Active Speaker Detector (AASD) can easily reduce the unnecessary traffic for the entire distributed system. The AASD is a functional block placed at each peer to automatically detect whether the peer is an active speaker or not by comparing its output audio power with that of the others. The AASD will then notify its cluster's leader as to which peer is an active speaker so that the leader can update its active speaker database.

All peers will send their base layer SVC stream (with its lower bit-rate) to its cluster's leader. In addition to sending their base video layer, active users also send their enhancement layers to the cluster's leader (with enhanced bit-rates). Apparently, enhancement layer bit-streams from inactive but interested users may be desirable for some peers. In this case, those particular peers must inform their cluster's leader about which interested user(s) they want to receive enhancement video layers from. This information will then be transmitted to the interested users via the leader network. After receiving notification, the interested users will send an enhancement video layer to the group as if they are active speakers. A very important point to note is that each peer in the multicast group will contribute only a portion of its computational capacity to support the conference, according to the number

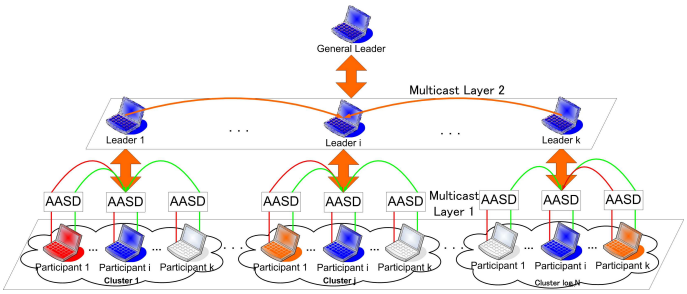

Fig. 1. Perception-based ALM for scalable video conferencing services

of enhancement video layers required by the conference to maintain a steady-state of the multicast system (otherwise, there will be congestion for the peers). However, this portion of contributing computational capacity is flexible and should remain variable in situations where more enhancement video layers are required. Figure 1 demonstrates a use case of the proposed algorithm. The blue peers (participants j in each cluster) are the clusters' leaders. The red peer (participant 1 of cluster 1) is the active speaker, while the orange peers (participants 1 of cluster j and participant k of cluster $log_k N$ - the last cluster) are interested users. The AASD placed at each user will detect whether or not that user is an active speaker in order to allow or block its enhancement video layers (red lines). Meanwhile, the base video layers (green lines) are always passed through the AASD to its cluster's leader. All cluster leaders then forms an upper multicast layer in order to transmit data among leaders from the first multicast layer. Similarly, the upper multicast layers are built until a general leader is found.

III. Theoretical analysis

In order to compare the queuing delay of the centralized and distributed architecture, two queuing models are constructed with the following notations:

- N: Total number of participating peers,
- r_{ij}: Transition probability of a packet from leaders of layer l or MCU to each peer,
- r_{ji}: Transition probability of a packet from each peer to leaders of layer l or MCU,
- μ_j, μ_l, μ_m: Service rates at each peer, leaders of layer l, and MCU, respectively,
- $\gamma_j, \gamma_l, \gamma_m$: External arrival rates to each peer, leaders of layer l, and MCU, respectively,
- $C^2_{Bj}, C^2_{Bl}, C^2_{Bm}$: Squared coefficient of variation (SCV) of service distributions at each peer, leaders of layer l, and MCU, respectively,
- $C^2_{Oj}, C^2_{Ol}, C^2_{Om}$: SCV of external inter-arrival distributions to each peer, leaders of layer l, and MCU, respectively,
- $\lambda_i, \lambda_l, \lambda_m$: Throughput at peers, leaders of layer l, and MCU, respectively,
- ρ_i, ρ_l, ρ_m: Traffic intensity (traffic congestion) at each peer, at leaders of layer l, at MCU, respectively ($\rho = \frac{\lambda}{\mu}$)

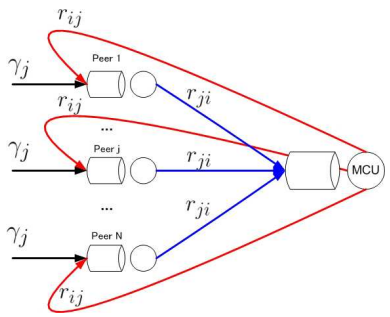

Fig. 2. Queuing model of the MCU-based video conference service architecture

According to[12], the approximated waiting time of each service architecture (G/G/1 queue of General distribution of inter-arrival time, General distribution of service time, 1 parallel server) is calculated by:

$$W_q \approx \left(\frac{\rho}{1-\rho}\right)\left(\frac{C_A^2+C_B^2}{2}\right)\left(\frac{1}{\mu}\right)g(\rho,C_A^2,C_B^2) \quad (2)$$

where:

$$g(\rho,C_A^2,C_B^2)=\begin{cases} exp\left[-\frac{2(1-\rho)}{3\rho}\cdot\frac{(1-C_A^2)^2}{C_A^2+C_B^2}\right], & \text{if } C_A^2<1 \\ 1, & \text{if } C_A^2\geq 1 \end{cases} \quad (3)$$

Formula (2) has a nice "product form" of three terms: (i) a traffic-intensity factor; (ii) a variability factor; and (iii) a time-scale factor (each packet requires $\frac{1}{\mu}$[unit time] of service).
For calculating (2), we must obtain C_A^2 from (4):

$$C_{Ai}^2=\frac{\gamma_i}{\lambda_i}C_{Oi}^2+\sum_{j=1}^{k}\frac{\lambda_j r_{ji}}{\lambda_i}\left\{r_{ji}\left[\rho_j^2 C_{Bj}^2+(1-\rho_j^2)C_{Aj}^2\right]+1-r_{ji}\right\} \quad (4)$$

In which, λ_i is calculated by:

$$\lambda_i=\gamma_i+\sum_{j=1}^{k}\lambda_j.r_{ji} \quad (5)$$

Since the end-to-end delay of each architecture is proportional to the queuing time, if we succeed in building queuing models and then calculate the approximated waiting time for the two architectures, end-to-end delay can be compared based on these results.

A. Centralized architecture

Figure 2 shows the queuing model for the MCU-based architecture. Here, all N peers are generating a media stream with a mean data rate of γ_j[Mbps]. Each peer sends this encoded video to the MCU at a transition probability of $r_{ji}=1$ (e.g. peer's output is all sent to the common MCU). The MCU then routes back N media streams to N participating peers, each contains data from (N-1) other peers (assuming that each peer will not receive its own stream). Since the transition probability r_{ij} is defined by the probability at which a packet who completes service at node i transits to node j, we have:

$$r_{ij}=\frac{1}{N(N-1)} \quad (6)$$

From (5), we have:

$$\begin{cases} \lambda_m=\gamma_m+N.\lambda_j \\ \lambda_j=\gamma_j+\lambda_m.r_{ij} \end{cases} \quad (7)$$

Since there is no external input other than from participating peers to the MCU then $\gamma_m=0$. Because all peers are generating aggregated SVC bit-streams, then they seem to "receive" external inputs at the same rate of $\gamma_j=\gamma_a$[Mbps] in which γ_a is the mean bit-rate of the aggregated SVC bit-stream. According to[10], the mean rate of an aggregated (base QCIF + enhancement CIF) spatial traffic is $\gamma_a=0.383$Mbps, the mean rate of an enhancement layer's spatial traffic is $\gamma_e=0.231$Mbps, and the mean rate of a base layer's spatial traffic is $\gamma_b=0.152$Mbps. From the above values, by replacing (6) into the system of equations (7) and solve it, we have:

$$\begin{cases} \lambda_m=\frac{N(N-1)^2}{(N-2)}\gamma_a \\ \lambda_j=\frac{(2N-3)}{(N-2)}\gamma_a \end{cases} \quad (8)$$

The traffic intensity at the MCU is $\rho_m=\frac{\lambda_m}{\mu_m}$ (assuming that only one server is used as the MCU). In order for the queue at the MCU to be in steady-state conditions, we must have $\rho_m<1$ or $\mu_m>\lambda_m$:

$$\mu_m>\frac{N(N-1)^2}{N-2}.\gamma_a \quad (9)$$

Assuming that we have designed a MCU to support of up to N_{max} participants, then the maximum throughput to be managed at the MCU is:

$$M_m=\left(\frac{N_{max}(N_{max}-1)^2}{N_{max}-2}+1\right)\lambda_a \quad (10)$$

We have $\rho_m=\frac{\lambda_m}{M_m}$
Similarly, the traffic intensity at each peer is $\rho_j=\frac{\lambda_j}{\mu_j}$. Assuming that all peers have a similar computational capacity, the service rates at all peers are equal ($\mu_j=\mu_p$). Let:

$$M_j=\left(\frac{2N_{max}-3}{N_{max}-2}+1\right)\gamma_a \quad (11)$$

We have $\rho_j=\frac{\lambda_j}{M_j}$.

SCV values are defined by ratio of the standard deviation to the mean value ($SCV=\frac{Var}{E^2}$). From the statistical data reported in[10], we obtained the SCV value of the aggregated traffic of the spatial scalability bit-streams arriving at each peer: $C_{Oj}^2=46$[kbit]. We have:

$$\begin{cases} C_{Om}^2=0; C_{Oj}^2=46[kbit] \\ C_{Bm}^2=10[kbit]; C_{Bj}^2=30[kbit] \end{cases} \quad (12)$$

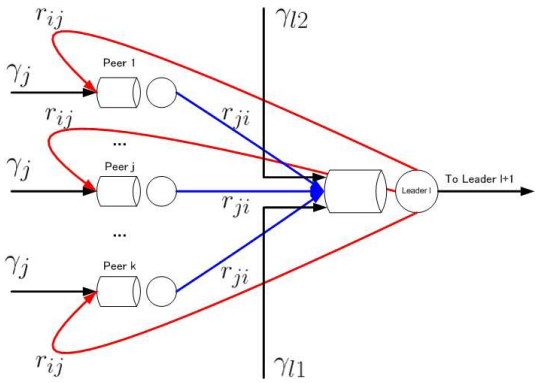

Fig. 3. Queuing model of the ALM-based video conference service architecture in cluster of the l^{th} layer

Using the values of $\gamma_m, \lambda_m, \lambda_j, \rho_j^2, \rho_m^2, r_{ij}$ and (12), the approximated waiting times at the MCU is $W_q m$. Then, we obtain the total waiting time of the centralized architecture:

$$W_{qc} \approx N W_{qm} \tag{13}$$

B. Distributed architecture

Figure 3 shows the queuing model of a random cluster on layer l in the ALM-based architecture. Each cluster has k peers, one of them is elected to be the cluster's leader. All peer is generating its base layer bit-stream at a mean data rate of $\gamma_j = \gamma_b$ [Mbps] and then sending that base-layer to the leader at the transition probability of $r_{ji} = 1$. Some of them ($n_1 \leq k$) are also sending an enhancement layer to the leader (according to the layer registration information). All of the enhancement layers form an external arrival rate of $\gamma_{l1} = n_1.\gamma_e$[Mbps] to the cluster's leader in which γ_e is the arrival data rate created by the enhancement layers ($\gamma_p = \gamma_b + p\gamma_e$). Here, we use a mean value of p to represent the percentage that an enhancement video layer is sent from a peer ($0 < p < 1$). The leader is also receiving another external data rate of γ_{l2}[Mbps] from the leader on a higher layer. The leader then distributes the data to k members of its cluster at the transition probability of r_{ij}.

In layer l, a leader will receive k bit-streams from its cluster, each with a data rate of ($\gamma_j = k^{(l-1)}\gamma_p$)[Mbps]. We have $\gamma_l = (N - k^l)\gamma_p + \gamma_p(k-3)k^l$.

$$\begin{cases} \gamma_l = N\gamma_p + \gamma_p(k-3)k^l \\ \gamma_j = k^{(l-1)}\gamma_p \end{cases} \tag{14}$$

Since the transition probability r_{ij} is defined by the probability at which a packet who completes service at node i transits to node j, we have:

$$r_{ij} = \frac{1}{k(k-1)+k} = \frac{1}{k^2} \tag{15}$$

From (5), we have:

$$\begin{cases} \lambda_l = \gamma_l + k\lambda_j \\ \lambda_j = \gamma_j + \lambda_l r_{ij} \end{cases} \tag{16}$$

Replace (14) and (15) into (16) and solve the equation system, we have:

$$\begin{cases} \lambda_l = \frac{k\gamma_p}{k-1}N + \frac{\gamma_p k(k-2)k^l}{k-1} \\ \lambda_j = \frac{\gamma_p}{k(k-1)}N + \frac{(2k-3)\gamma_p}{k(k-1)}k^l \end{cases} \tag{17}$$

The traffic intensity at a leader of layer l is $\rho_l = \frac{\lambda_l}{\mu_l}$. In order for the queue at the leader to be in steady-state conditions, we must have $\rho_l < 1$ or $\mu_l > \lambda_l$, with (17), we have:

$$\mu_l > \frac{k\gamma_p}{k-1}N + \frac{\gamma_p k(k-2)k^l}{k-1} \tag{18}$$

Assume that the top leader has been designed to support the maximum throughput of maxλ and the system can support of up to N_{max} participants, at the top leader, $k^l = N$, therefore we have:

$$M_l = (k+1)N_{max}\gamma_p, \tag{19}$$

Thus, we have $\rho_l = \frac{\lambda_l}{M_l}$.

$$\begin{cases} C_{Ol}^2 = 16[kbit]; C_{Oj}^2 = 16[kbit] \\ C_{Bl}^2 = 30[kbit]; C_{Bj}^2 = 30[kbit] \end{cases} \tag{20}$$

According to (4), remembering that $r_{ji} = 1$, we have:

$$\begin{cases} C_{Al}^2 = \frac{\gamma_l}{\lambda_l}C_{Ol}^2 + k\frac{\lambda_j}{\lambda_l}\rho_j^2 C_{Bj}^2 + k\frac{\lambda_j(1-\rho_j^2)}{\lambda_l}C_{Aj}^2 \\ C_{Aj}^2 = \frac{\gamma_j}{\lambda_j}C_{Oj}^2 + \frac{\lambda_l r_{ij}}{\lambda_j}(r_{ij}\rho_l^2 C_{Bl}^2 + 1 - r_{ij}) + ... \\ ... + \frac{\lambda_l r_{ij}^2}{\lambda_j}(1-\rho_l^2)C_{Al}^2 \end{cases} \tag{21}$$

Let:

$$\begin{cases} A_1 = \frac{\gamma_l}{\lambda_l}C_{Ol}^2 + k\frac{\lambda_j}{\lambda_l}\rho_j^2 C_{Bj}^2 \\ B_1 = k\frac{\lambda_j(1-\rho_j^2)}{\lambda_l}; B_2 = \frac{\lambda_l.r_{ij}^2}{\lambda_j}(1-\rho_l^2) \\ A_2 = \frac{\gamma_j}{\lambda_j}C_{Oj}^2 + \frac{\lambda_l.r_{ij}}{\lambda_j}(r_{ij}\rho_l^2 C_{Bl}^2 + 1 - r_{ij}) \end{cases} \tag{22}$$

From (21) and (22), we have:

$$C_{Al}^2 = \frac{A_1 + A_2.B_1}{1 - (B_1.B_2)} \tag{23}$$

Replace the values of (14), ρ_j, ρ_l, (15) and (20) into (22) to obtain A_1, A_2, B_1, B_2, and then use them to calculate (23) Replace (23) into (2) we have W_{ql}. Thus, the total waiting time in the distributed architecture is:

$$W_{qd} \approx k\sum_{l=1}^{l_{max}} W_{ql}$$

IV. Result Analysis

Figure 4 shows the comparison between the total waiting time for centralized (13) and distributed (24) queues:

- Total waiting time in a centralized queue,
- Total waiting time in a distributed queue when the clusters' sizes are $k = 3$, $k = 5$, and $k = 7$, respectively.

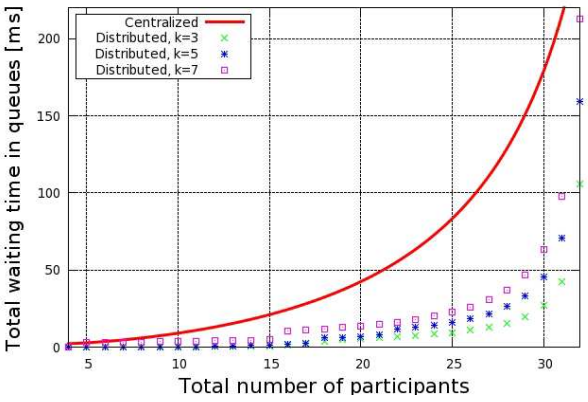

Fig. 4. Comparison of queuing waiting time between centralized (13) and distributed (24) architectures.

In the centralized architecture, we assume that the MCU can support of up to N_{max} participants at the same time, all participant is sending its base video layer. A fracture of p peers will send their enhancement video layers. Meanwhile in the distributed architecture, all peers are sending both base layers and each leader can support at least k peers in its cluster. From the result in Fig.4 we can find that, the total waiting time at the MCU increases rapidly with the increasing number of participants. Meanwhile the total waiting time of the distributed architecture increases linearly but at a much lower speed. When the cluster size increases, the total waiting time of the distributed architecture increases. There is an interesting conclusions which we can withdraw after analyzing the results in Fig.4: When the number of peers in a cluster (k) increases (from $k = 3$ to $k = 7$), the total waiting time in the distributed queue increases. Therefore, for a certain number of participants, **it is recommended to use a smaller cluster size** to maintain a lower total waiting time.

V. Related work

In[13], Kuehn has applied an approximate method for building the analysis model of general queuing networks. The queuing network is of the open network type, having N single server queuing stations with arbitrary interconnections. In[14], NICE has been proposed as an ALM algorithm in which the idea of clustering and layering has been built. However, from the best of our knowledge, none of the conventional research has either considered the perception limitation of all participants in order to reduce unnecessary traffic or built an analysis model for evaluating the delay in a distributed conference.

VI. Conclusion and future work

In this research work, a new Application Layer Multicast algorithm considering the limitation of human's perception has been proposed. The newly proposed algorithm can effectively reduce the total traffic load of the scalable video conferencing service. Analysis models have been built and compared for centralized and distributed architectures using queuing theory. The theoretical analysis result has shown a great advantage of the distributed architecture against the centralized architecture especially when the total number of conferencing participants increases. The analysis model has also proposed a method for calculating the necessary computational capacity of the MCU and leader nodes in the proposed ALM algorithm in order to provide a sufficient service rate for obtaining a steady-state in each service architecture.
Our future work will consider the delay of the underlay network, the possibilities of congestion that may happen when more participants are involved into the two concerning architectures, and the ability of controlling the number of enhancement video layers in the centralized architecture while building the analysis model.

VII. Acknowledgement

The research work is partly supported by Poseidon, a French national project on the evaluation of next generation wireless networks and advanced multimedia services over these infra-structures.

References

[1] J. R. Wilcox, *Videoconferencing & interactive multimedia: the whole picture*, Cmp, 2000.
[2] Eleftheriadis Alexandros, Civanlar M. Reha, and Shapiro Ofer, "Multipoint videoconferencing with scalable video coding," *Journal of Zhejiang University - Science A*, vol. 7, no. 5, pp. 696–705, 2006.
[3] S. E. Deering, "Multicast routing in a datagram internetwork," 1991.
[4] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for the IP multicast service and architecture," *IEEE Network*, vol. 14, no. 1, pp. 78–88, 2000.
[5] C. Luo, W. Wang, J. Tang, J. Sun, and J. Li, "A Multiparty Videoconferencing System Over an Application-Level Multicast Protocol," *IEEE Transactions on Multimedia*, vol. 9, no. 8, pp. 1621–1632, 2007.
[6] I. E. Akkus, M. R. Civanlar, and O. Ozkasap, "Peer-to-Peer Multipoint Video Conferencing using Layered Video," in *Image Processing, 2006 IEEE International Conference on*, Oct. 2006.
[7] M. Ponec, S. Sengupta, M. Chen, J. Li, and P. A. Chou, "Multi-rate peer-to-peer video conferencing: A distributed approach using scalable coding," in *Proceedings of the 2009 IEEE international conference on Multimedia and Expo*. IEEE Press, 2009, pp. 1406–1413.
[8] Tien Anh Le; Hang Nguyen; Hongguang Zhang, "EvalSVC - an evaluation platform for scalable video coding transmission," in *14th International Symposium on Consumer Electronics (ISCE 2010)*, Braunschweig, Germany, June 2010, pp. 85–90.
[9] Tien A. Le and Hang Nguyen, "Centralized and distributed architectures of scalable video conferencing services," in *The Second International Conference on Ubiquitous and Future Networks (ICUFN 2010)*, Jeju Island, Korea, June 2010, pp. 394–399.
[10] Geert Van der Auwera, Prasanth T. David, Martin Reisslein, and Lina J. Karam, "Traffic and quality characterization of the H.264/AVC scalable video coding extension," *Adv. MultiMedia*, vol. 2008, no. 2, pp. 1–27, 2008.
[11] Tien A. Le, Hang Nguyen, and Hongguang Zhang, "Multi-variable cost function for Application Layer Multicast routing," in *IEEE Globecom 2010 - Communications Software, Services and Multimedia Applications Symposium (GC10 - CSSMA)*, Miami, Florida, USA, Dec. 2010.
[12] D. Gross, *Fundamentals of queueing theory*, Wiley-India, 2008.
[13] P. Kuehn, "Approximate analysis of general queuing networks by decomposition," *Communications, IEEE Transactions on*, vol. 27, no. 1, pp. 113–126, 2002.
[14] 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.