Human perception-based distributed architecture for scalable video conferencing services: theoretical models and performance

Tien Anh Le, Hang Nguyen

Enriched human perception-based distributed architecture for scalable video conferencing services: theoretical models and performance

Tien Anh LE, *Student Member, IEEE,* and Hang NGUYEN, *Member, IEEE*

Abstract

This research work proposes an enriched human perception-based distributed architecture for the multi-party video conferencing services. Rich theoretical models of the three different architectures: the proposed perception-based distributed architecture, the conventional centralized architecture and perception-based centralized architecture have been constructed by using queuing theory to reflect the traffic generated, transmitted and processed on all three architectures. The performance of these three different architectures has been considered in different aspects from the total waiting time, the point-to-point delay and the required service rates to the total throughput. Together, the modelling tools, the analysis, and the numerical results help to answer the common concern about advantages and disadvantages between the centralized and distributed architectures for the video conferencing service architecture.

Index Terms

Service architecture, Distributed architecture, Centralized architecture, Scalable video coding, Video-conference Service

Tien Anh LE, and Hang NGUYEN are with the Department of Wireless Networks and Mobile Multimedia Services, Telecom Sud Paris, Evry, Ile-de-France, 91011 FRA. E-mails: {Tien_Anh.Le, Hang.Nguyen}@it-sudparis.eu.

I. INTRODUCTION

Video conferencing service, the most complex type of video communications is foreseen to be the next popular digital communication after voice communications. In general, centralized architectures are mainly used to build a video conference service and distributed architectures are finding their ways to be widely recognized. Centralized and distributed architectures have their own advantages and questions are usually asked about which architecture is preferable in certain network and usage conditions. Main players in the centralized video conferencing architectures are often special equipment-based solutions [2], [3], web-based services [4] or IMS-based architectures [5]. When more participants want to join the conference (e.g. at big events), the cost of the centralized architecture increases sharply. Therefore, distributed architectures have been proposed in [6], [7]. Both IP-Multicast and Application Layer Multicast have been applied to build the distributed video conferencing architecture [8], [9]. In[10], [11] scalability has been considered and Scalable Video Coding (SVC) has been applied to support varied types of participants' terminals. From the best of our knowledge, none of the proposed distributed architectures has compared its performance with the conventional centralized architecture. Thus, no conclusion can be made on whether a distributed architecture is really better than the centralized architecture and whether service users should bother changing their current centralized conferencing systems with a new distributed system. So far,[12] is an early attempt to compare the two types of conferencing architectures using its proposed evaluation platform[13]. The experimental results show that, when using a much higher computational capacity, the MCU of the centralized architecture can only provide a similar bit-rate with the distributed architecture at the trade-off of a much higher delay. These results were interesting but they need to be validated by theoretical analysis in order to be applied in more general conditions.

In section II of this paper, we propose a new enriched human perception-based distributed video conferencing architecture in which limitations of human perception when participating

into the conference will be considered in order to reduce the unnecessary traffic on the overlay network. In section III, theoretical queuing models are constructed and analysed for four main performance criteria: total waiting time, end-to-end delay, computational requirements and total throughput of three different architectures: centralized, perception-based centralized and the proposed perception-based distributed architectures. The performance of the three architectures have been considered and compared in terms of the four performance criteria. Mathematical analysis and expression for these parameters have also been constructed. The waiting time performance compares the processing and queuing mechanisms in the three architectures. The end-to-end delay comparison considers the delay of a packet including the processing time in the queues and also the transmission time in the underlay network. The required service rates explain how much computation capacity the important nodes have to equip in order to support steady states of the queues formed and processed by each architecture. Finally, the total throughput performance answers very common concerns of whether the distributed architecture usually has much more traffic than the centralized architecture or not. By approaching the performance from different aspects, a completed view of the proposal is archived. Numerical computation of the analysing models for four main performance criteria shows that the perception-based distributed architecture performs better than the other two in all four main aspects of performance.

II. Proposed Architecture of Human Perception-based Distributed Scalable Video Conferencing Services

In our first proposal [14] and in this paper, the application-aware multi-variable cost function proposed in [15] can be used as the optimal cost function to build the media distribution tree for the perception-based distributed architecture. The following model and analysis can be generally applied to a media distribution tree built from any cost function. Figure 1 displays the main characteristics of our proposal. A cluster is a group of k peers which have the nearest distance to each other according to the optimization of the applied cost function. When a peer wants to

join the overlay group, it will first try to explore its nearest cluster to join into by measuring the costs to reach the leaders of all clusters. A cluster has the maximum size of **k** peers depending on the network conditions. A group's leader is the one who has the minimum total cost to reach to all other peers in the cluster. Here we assume that all the important criteria such as processing capacity, available memory, bandwidth of the group's leader have been fully considered in the applying multi-variable cost function before calculating the cost. All leaders, from the first layer, will then use the same cost function to calculate its costs to reach to all other leaders at layer 1. These costs will then be applied to form clusters and a second layer. The calculations are made until the maximum number of layers (l_{max}) is reached. A leader l will receive bit-streams from its cluster's peers j with a throughput of $\lambda_{j\rightarrow l}$ and a traffic variation represented by $C^2_{j\rightarrow l}$. The leader makes (k-1) duplications of the arriving bit-stream before multicasting the traffic back to the other peers at a variation of $C^2_{\psi\rightarrow l}$ and to the upper layer's leaders. At the same time, it receives the bit-streams from upper layer's leader and forwards them to its cluster's members. The proposed architecture is applied when SVC (or any multilayer video coding) is used. In a conference session, at any given time, there are normally one or a few active speakers (the participants who are giving the speech or participating into the discussion). They can be automatically found by comparing the participant microphones' output power. A simple reason is that, if all participants are to be displayed with full quality in a conference session, a human-being and his terminal will not have enough perception and displaying 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 by comparing its input audio powers[16]. We call $r^b_{hl}(t), r^{e_i}_{hl}(t)$ the instantaneous values of the traffic rectification coefficients on base and i^{th} enhancement video layers from h^{th} participant to a cluster leader l. Let $r^{e_i}_{hl}(t) = 1$ if the member wants to receive i^{th} enhancement video layer from h^{th} participant, it equals to 0 otherwise. All peers send their base video layer (at a bit-rate of $\gamma^b_{hl}(t)$) to its

cluster's leader. Active speakers also send their i^{th} enhancement layers to the cluster's leader (at a bit-rate of $\gamma_{hl}^{ei}(t)$). Enhancement layer bit-stream from an inactive but interesting users may be desirable for some peers. Those particular peers can inform its cluster's leader about the interesting user(s) they want to receive enhancement video layers from. Through a network of leaders, this information will be notified to the interesting users and to all the cluster's leaders. After receiving the notification, the interesting users send their enhancement video layers to the group as if they are active speakers. Each leader maintains a Conferee Preference Table (CPT) and to all the cluster's leaders. This is a record table of N rows and N columns (N is the number of participants). The AASD at each participant can also update its cluster leader whether it is an active speaker so that the cluster's leader can update to the CPT. Therefore, the default value of the CPT can be determined by the AASD at each participant. Another option is that each conferee can also select the interesting participant(s) by updating 1 to the corresponding place(s) (CPT[h,h']) of the table. Each participant can also select whether or not it wants to have the CPT automatically updated by its AASD. Thus, the CPT can be dynamically updated by the AASD or it can be manually maintained by the users' selections. The CPT's content is synchronized among all the leaders of the perception-based distributed architecture. As a result, $r_{hl}^{ei}(t) = 1$ if it is an active speaker detected by the AASD or it is an interesting user registered by at least one other participant, it equals to 0 otherwise. A participant sends its enhancement video layers if its corresponding traffic rectification coefficient $r_{hl}^{e_i}(t) = 1$ and does not send it otherwise. On the other hand, after receiving the enhancement video layers from its upper layer leader, each leader decides whether or not it should forward an enhancement video layer to its cluster's members based on its CPT. A private point to point video chat session can be established and maintained using the same CPT's mechanism.

III. THEORETICAL ANALYSIS

In order to compare the queuing delay of the three different architectures, three queuing models are constructed with the notations in Table I, Fig.1 and Fig.2. According to[17], 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:

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

where:

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

C_B^2 is a fixed value determined by the type of hardware used at the queuing nodes. The SCV value of the distribution of arriving traffic at each node (C_{Ai}^2) can be calculated by:

$$C_{Ai}^2 = \frac{1}{\lambda_i}\left(\gamma_i C_{Oi}^2 + \sum_{j=1}^{k}\lambda_j C_{ji}^2\right) \tag{3}$$

The SCV value of the distribution of departing traffic from each node (C_{Di}^2) can be calculated by:

$$C_{Di}^2 = \rho_i^2 C_{Bi}^2 + (1-\rho_i^2)C_{Ai}^2 \tag{4}$$

$$C_{ij}^2 = \frac{\lambda_{i\to j}}{\lambda_{Di}}C_{Di}^2 + \left(1-\frac{\lambda_{i\to j}}{\lambda_{Di}}\right) \tag{5}$$

Replace Equ.4 into Equ.5, we have the final form of the SCV of service rate for traffic generated by node i departing for node j:

$$C_{ij}^2 = \frac{\lambda_{i\to j}}{\lambda_{Di}}\left[\rho_i^2 C_{Bi}^2 + (1-\rho_i^2)C_{Ai}^2\right] + \left(1-\frac{\lambda_{i\to j}}{\lambda_{Di}}\right) \tag{6}$$

For all of the following models, we use an abstract concept of traffic arriving from the "multicast duplicator". This is due to the fact that, in a multicast session, each transmitting node has to duplicate the arriving traffic before forwarding them to several multicast receivers. We model that extra traffic as the traffic from an abstract "multicast duplicator" (ψ) at the multicast transmitter.

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

A. Theoretical expression of the waiting time in queues

Each peer encodes a scalable video stream with a base layer and several enhancement layers. n^e_{max} is the pre-defined number of enhancement video layers from each participant. The instantaneous value of the traffic rate departing from the h^{th} participant to the leader l at the 1^{st} layer is:

$$\gamma^p_{hl}(t) = r^b_{hl}(t).\gamma^b_{hl}(t) + \sum_{i=1}^{n^e_{max}} r^{ei}_{hl}(t).\gamma^{ei}_{hl}(t) \tag{7}$$

To calculate the waiting time in the three architectures, we need to properly calculate the value of CoV^2 in these three cases. Let $X^b_{hl}(t), X^{ei}_{hl}(t), X^p_{hl}$(t) are respectively the instantaneous random inter-arrival time of the traffic generated by the base layer, the i^{th} enhancement layer, and the aggregated video layer from participant h to the leader l of its cluster. Since a SVC stream comprises of one base layer and i enhancement layers, we have:

$$X^p_{hl}(t) = X^b_{hl}(t).r^b_{hl}(t) + \sum_{i=1}^{n^e_{max}} X^{ei}_{hl}(t).r^{ei}_{hl}(t) \tag{8}$$

The CoV value of the aggregated video can be calculated from Equ.8 as:

$$CoV^2(X^p_{hl}(t)) = \frac{Var(X^p_{hl}(t))}{E^2(X^p_{hl}(t))} \tag{9}$$

In which, $Var(X^p_{hl}(t)), E^2(X^p_{hl}(t))$ can be calculated from Equ.8:

$$\begin{cases} Var(X^p_{hl}(t)) = (r^b_{hl}(t))^2.CoV^2(X^b_{hl}(t)).E^2(X^b_{hl}(t)) + \sum_{i=1}^{n^e_{max}} (r^{ei}_{hl}(t))^2.CoV^2(X^{ei}_{hl}(t)).E^2(X^{ei}_{hl}(t)) \\ E(X^p_{hl}(t)) = r^b_{hl}(t).E(X^b_{hl}(t)) + \sum_{i=1}^{n^e_{max}} r^{ei}_{hl}(t).E(X^{ei}_{hl}(t)) \end{cases} \tag{10}$$

In the case of the perception-based centralized architecture, the final form of $CoV^2(X^p_{hp}(t))$ is:

$$CoV^2(X^p_{hp}(t)) = \frac{(r^b_{hp}(t))^2.CoV^2(X^b_{hp}(t)).E^2(X^b_{hp}(t)) + \sum_{i=1}^{n^e_{max}} (r^{ei}_{hp}(t))^2.CoV^2(X^{ei}_{hp}(t)).E^2(X^{ei}_{hp}(t))}{\left(r^b_{hp}(t).E(X^b_{hp}(t)) + \sum_{i=1}^{n^e_{max}} .r^{ei}_{hp}(t).E(X^{ei}_{hp}(t)) \right)^2} \tag{11}$$

In case of the MCU (centralized architecture), the $CoV^2(X^a_{hl}(t))$ can be directly calculated by:

$$CoV^2(X^a_{hS}(t)) = \frac{Var(X^a_{hS}(t))}{E^2(X^a_{hS}(t))} \tag{12}$$

1) Perception-based distributed architecture: The distributed model for this architecture is as shown in Fig.1. The throughput arriving to a leader at layer l (λ_{Al}) is comprised of: the throughput arriving from all peers j of the cluster to the leader l ($\lambda_{j\to l}$), and the throughput from the "multicast duplicator" ψ_l to the leader l ($\gamma_{\psi\to l}$), considered as the external traffic at the leader l.

The throughput arriving from all peer j of the cluster to the leader l ($\lambda_{j\to l}$) is comprised of the throughput from all the video bit-streams from all the conferees belonging to the sub-trees whose root is the leader l to the leader l:

$$\lambda_{j\to l} = \sum_{h=1+(j-1)k^{l-1}}^{jk^{l-1}} \gamma^p_{hl}(t) \tag{13}$$

The throughput from the "multicast duplicator" ψ_l to the leader l ($\gamma_{\psi\to l}$) comprises of (k-1) copies of throughput from all the peers:

$$\gamma_{\psi\to l} = (k-1)\sum_{h=1}^{N} \gamma^p_{hl}(t) \tag{14}$$

From Equations (13) and (14), we can calculate the overall throughput arriving to the leader l:

$$\lambda_{Al} = k\sum_{h=1}^{N} \gamma^p_{hl}(t) \tag{15}$$

The purpose is to form an equation for calculating the SCV of times between arrivals to the leader l because it will determine the waiting time of the traffic queue at the leader l. The general

form of C_{Al}^2 is:

$$C_{Al}^2 = \frac{v_{Al}}{\lambda_{Al}} = \frac{1}{\lambda_{Al}}.\left(\gamma_{\psi\to l}.C_{o\psi l}^2 + \sum_{j=1}^{N}\lambda_{j\to l}.C_{j\to l}^2\right) \tag{16}$$

We need to calculate the SCV of times between departures from peer j to the leader l ($C_{j\to l}^2$) and from the abstract "multicast duplicator" to the leader l ($C_{o\psi l}^2$).

At the first layer, since all the peers are directly the video conferees, we have:

$$C_{j\to l=1}^2 = C_{h=j\to l}^2 = CoV^2(X_{hl}^p(t)) \tag{17}$$

The value of $CoV^2(X_{hl}^p(t))$ is given by Equ.9. Now we can calculate the SCV of times between departures from the abstract "multicast duplicator" to the leader l as the combination of the SCV of times between departures from (k-1) peers j to the leader l:

$$C_{o\psi l=1}^2 = \sum_{j=1}^{k-1} C_{j\to l=1}^2 \tag{18}$$

In Equ.16, for the first layer, all of the parameters have been given in Equ.15 (λ_{Al}), Equ.14 ($\gamma_{\psi\to l}$), Equ.13 ($\lambda_{j\to l}$), Equ.18 ($C_{o\psi l}^2$), and Equ.17 ($C_{j\to l}^2$). Therefore, $C_{Al=1}^2$ is known for the first layer.

At the second layer, according to Equ.4 and Equ.5, we have:

$$C_{j\to l=2}^2 = \frac{\lambda_{j\to l}}{\lambda_{Dj}}\left[\rho_j^2.C_{Bj}^2 + (1-\rho_j^2).C_{Aj}^2\right] + (1 - \frac{\lambda_{j\to l}}{\lambda_{Dj}}) \tag{19}$$

The traffic intensity (traffic congestion) at each peer (ρ_j) is calculated by:

$$\rho_j = \frac{\lambda_j}{M_l} = \frac{k\sum_{h=1}^{N}\gamma_{hl}^p(t)}{M_l} \tag{20}$$

λ_{Dj} is the throughput departing from a peer j, calculated by:

$$\lambda_{Dj} = k.\sum_{h=1}^{N}\gamma_{hl}^p(t) \tag{21}$$

For the second layer, the SCV of service rate for the traffic arriving to the peer j (C_{Aj}^2) at layer 2 can be calculated from $C_{Al=1}^2$ at the first layer as:

$$C_{Aj}^2 = C_{Al=1}^2 \tag{22}$$

In Equ.19, all of the parameters are known ($\lambda_{j\to l}$ in Equ.13, λ_{Dj} in Equ.21, ρ_j in Equ.20, C^2_{Aj} in Equ.22). Therefore, $C^2_{j\to l=2}$ is known. Now we can calculate the SCV of times between departures from the abstract "multicast duplicator" to the leader l of the second layer as the combination of the SCV of times between departures from (k-1) peers j to the leader l at the second layer:

$$C^2_{o\psi l=2} = \sum_{j=1}^{k-1} C^2_{j\to l=2} \tag{23}$$

In Equ.16, for the second layer, all of the parameters have been given in Equ.15 (λ_{Al}), Equ.14 ($\gamma_{\psi\to l}$), Equ.13 ($\lambda_{j\to l}$), Equ.23 ($C^2_{o\psi l=2}$), and Equ.19 ($C^2_{j\to l=2}$). Then, we would be able to calculate the SCV of service rate for the traffic arriving to the leader l at layer 2 ($C^2_{Al=2}$). Recursively, we can calculate the values of C^2_{Al} on all the upper layers by applying the same method of calculating it on the first and the second layers and using Equ.16.

From the value of C^2_{Al} of all layers obtained from Equ.16 we can calculate the waiting time at a leader in layer l according to Equ.1, we have:

$$W_{ql} \approx \left(\frac{\rho_l}{1-\rho_l}\right)\left(\frac{C^2_{Al}+C^2_{Bl}}{2}\right)\left(\frac{1}{\mu_l}\right) \tag{24}$$

The required service rate and the congestion rate at each leader l (μ_l, ρ_l) are explained and calculated in more details in section III-C and Equ.39. Since k members in a cluster have to wait in sequence to be served by the cluster's leader, the total waiting time of the perception-based distributed architecture is calculated by:

$$W_{qd} = \sum_{l=1}^{l_{max}} kW_{ql} \tag{25}$$

In which l_{max} is the maximum number of layers in the perception-based distributed architecture.

2) Centralized architecture: The centralized model is as shown in Fig.2. The throughput arriving to the centralized MCU (λ_{AS}) is comprised of: the throughput arriving from each peer $j = h$ to the MCU ($\lambda_{j\to S}$), and the throughput generated by the "multicast duplicator" ψ_S for the MCU ($\gamma_{\psi\to S}$) considered as the external traffic at the MCU. The overall throughput arriving

to the centralized MCU is therefore:

$$\lambda_{AS} = \gamma_{\psi \to S} + \sum_{j=1}^{N} \lambda_{j \to S} \tag{26}$$

Applying Equ.3, we can obtain the SCV of the service rate at the MCU server:

$$C_{AS}^2 = \frac{1}{\lambda_{AS}}. \left(\gamma_{\psi \to S}. C_{o\psi S}^2 + \sum_{j=1}^{N} \lambda_{j \to S}. C_{j \to S}^2 \right) \tag{27}$$

In which:

$$\begin{cases} C_{o\psi S}^2 = \sum_{j=1}^{N-1} C_{j \to S}^2 \text{ and } C_{j \to S}^2 = CoV^2(X_{h=j,l=S}^a(t)) = CoV^2(X_{hS}^a(t)) \\ \lambda_{j \to S} = \gamma_{h=j}^a(t) \text{ and } \lambda_{AS} = (N-1) \sum_{h=1}^{N} \gamma_{hS}^a(t) \\ \gamma_{\psi \to S} = (N-2) \sum_{h=1}^{N} \gamma_{hS}^a(t) \end{cases} \tag{28}$$

From the value of C_{AS}^2 obtained from Equ.27 and Equ.28 we can calculate the waiting time at the MCU according to Equ.1, we have:

$$W_{qm} \approx \left(\frac{\rho_S}{1 - \rho_S} \right) \left(\frac{C_{AS}^2 + C_{BS}^2}{2} \right) \left(\frac{1}{\mu_m} \right) \tag{29}$$

The required service rate and the congestion rate at the MCU (μ_m, ρ_m) are explained and calculated in more details in section III-C and Equ.40.

Since N participants have to wait in sequence to be served by the MCU, the total waiting time of the centralized architecture is calculated by:

$$W_{qc} = N W_{qm} \tag{30}$$

3) Perception-based centralized architecture: is a special case of the centralized architecture with the newly proposed perception-based function (replacing $\gamma_{hS}^a(t)$ by $\gamma_{hp}^p(t)$):

$$C_{Ap}^2 = \frac{1}{\lambda_{Ap}}. \left(\gamma_{\psi \to p}. C_{o\psi p}^2 + \sum_{j=1}^{N} \lambda_{j \to p}. C_{j \to p}^2 \right) \tag{31}$$

$$\begin{cases} C_{o\psi p}^2 = \sum_{j=1}^{N-1} C_{j \to p}^2 \text{ and } C_{j \to p}^2 = CoV^2(X_{h=j,l=p}^p(t)) = CoV^2(X_{hp}^p(t)) \\ \lambda_{j \to p} = \gamma_{h=j}^p(t) \text{ and } \lambda_{Ap} = (N-1) \sum_{h=1}^{N} \gamma_{hp}^p(t) \\ \gamma_{\psi \to p} = (N-2) \sum_{h=1}^{N} \gamma_{hp}^p(t) \end{cases} \tag{32}$$

$$W_{qp} \approx N\left(\frac{\rho_p}{1-\rho_p}\right)\left(\frac{C_{Ap}^2 + C_{Bp}^2}{2}\right)\left(\frac{1}{\mu_p}\right) \tag{33}$$

The required service rate and congestion rate at the perception-based centralized server p (μ_p, ρ_p) are explained and calculated in more details in section III-C and Equ.41.

4) Numerical calculation of the waiting time in queues: The newly proposed theoretical models and mathematical expressions formed in the previous subsections allow us to evaluate the performance in the four different performance criteria for the three different architectures in real-time, with an arbitrary number of participants, and with very heterogeneous contexts of peers such as peers with terminals of different screen sizes, different available bandwidth, different computational capacities and a variety of users' preferences. The proposed theoretical models and mathematical expressions can be applied to model a very complex enriched video conferencing services. Our first step here is to have a first comparison of the three different architectures. Therefore, we apply the available traffic models of the SVC video streams which are currently limited to the mean values of a video session presented in [18], [19] for a simple case of the distributed scalable video conference service. To provide first numerical results of the waiting time in the three different architectures, we apply an averaging of some video conferencing sessions' values presented in the previous subsections so that we can apply the data provided in [18], [19]. The specifications of the spatial video traffic are shown in Table I. Figures 3 and 4 show the comparison among the total waiting time of the three architectures when the clusters' sizes are $k = 3$, $k = 5$, $k = 7$. The video streams are encoded with spatial SVC. In the centralized architecture, we assume that the MCU can support of up to N_{max} participants at the same time, all participants are sending both their base and enhancement video layers. In the two figures, there are in average 3 enhancement video layers which are transmitted from all the participants ($n_e = 3$). In the perception-based centralized architecture and the perception-based distributed architecture, all peers send their base layers and some peers send their enhancement video layers. In Figure 3, the base video layer and the enhancement

video layers from 10% of the total number of participants are sent ($r_b = 1, r_e = 0.1$). In Figure 4, 50% of the peers send their enhancement video traffic ($r_b = 1, r_e = 0.5$). Each leader can support at least k peers in its cluster. We are going to analyze the results to show the effect of three main aspects on the total waiting time performance: (i) comparison between the distributed and the centralized architectures, (ii) impacts of the cluster size on the performance, and (iii) the newly proposed perception-based function's performance. The total waiting time of both the centralized and perception-based centralized architectures increases exponentially with the increasing number of participants. Meanwhile the total waiting time of the perception-based distributed architecture increases at a much lower logarithmic speed. The centralized architecture has the highest value followed by the perception-based centralized architecture. The perception-based distributed architecture outperforms the other two centralized architectures. Especially, when the number of participants increases. The cluster size also plays a role in the waiting time. When the number of peers in a cluster (k) increases, the total waiting time in the distributed queue increases. Therefore, for a certain number of participants (N=50), it is recommended to use a smaller cluster size to maintain a lower total waiting time. When we make comparisons among the total waiting time of the two different figures ($r_e = 0.1$, $r_e = 0.5$), the more unnecessary traffic is reduced by applying our newly proposed perception-based function (both in the perception-based centralized and perception-based distributed architectures), the lower the total waiting time is. The newly proposed perception-based function can actually limit the unnecessary traffic and reduce the total waiting time.

B. Point-to-point delay

We apply the point-to-point delay analysis proposed in[20] to analyse the delay caused by packet transmission on the underlay network. This end-to-end delay comprises of a fixed delay

and a variable delay:

$$d_{point-to-point} = d^{fix} + d^{var} = p.\sum_{i=1}^{h}(\frac{1}{C_i}) + \sum_{i=1}^{h}\delta_i + d^{var} = p.\alpha + \beta + d^{var} \quad (34)$$

In which, h is the number of hops, C_i is the capacity of each link, δ_i is the propagation delay on each link, and p is the packet's size. Through real measurement data: α is found to be in the range of $\{27.10^{-6} : 6.10^{-5}\}$ and β is found to be in the range of $\{28 : 35\}$. There are two value ranges for the d^{var}: if the link utilization $\leq 90\%$ then $d^{var} \leq 1ms$, otherwise if the link utilization $> 90\%$ then d^{var} is tens of ms, we choose d^{var} to be $20ms$. With a packet size ranging from $\{40 : 1500\}$ bytes (most common packet size on the Internet), we obtain d^{fix} from Equ.34.

1) Perception-based distributed architecture: A cost function considering the cost to join each cluster has been applied. We manage to group participants which are in the same local area or with the lowest possible cost into one cluster. Therefore, it is likely that a packet has to travel across less hops before reaching its cluster's leader. The parameter set for calculating the point-to-point delay in the perception-based distributed architecture is: $\alpha_{ALM} = 3.10^{-5}, \beta_{ALM} = 28ms, d^{var}_{ALM} = 1ms$. From Equ.34, applying $p = 1500$(bytes) we can obtain the total delay of the perception-based distributed architectures which is equal to the sum of the total waiting time at all nodes (as calculated by W_{qd}) and its correspondent point-to-point delay:

$$D_{ALM} = W_{qd} + d_{ALM} = W_{qd} + (3.10^{-5} + 28).\log_k N = W_{qd} + 28\log_k N \quad (35)$$

2) Centralized architecture: All participants have to connect to the same MCU to be served. Thus the probability that these participants are located in different countries or even different continents is high. In such cases, the number of underlay hops that a packet has to travel in order to reach the MCU is also high. Therefore, we choose a value set which has the highest value to calculate the point-to-point delay of the centralized architecture. The parameter set for calculating the point-to-point delay in the centralized architecture is $\alpha_{MCU} = 6.10^{-5}, \beta_{MCU} = 35ms, d^{var}_{MCU} = 20ms$. From Equ.34, applying $p = 1500$(bytes) we can obtain the total delay of

the centralized architecture which is equal to the sum of the total waiting time at all nodes (as calculated by W_{qc}) and its correspondent point-to-point delay.

$$D_{MCU} = W_{qc} + d_{MCU} = W_{qc} + 6.10^{-5}p + 55 = W_{qc} + 55 \tag{36}$$

3) Perception-based centralized architecture:

$$D_{PerMCU} = W_{qp} + 55 \tag{37}$$

4) Result Analysis: Figures 5 and 6 show the point-to-point delay of the three different architectures considering both total waiting time in queues and point-to-point transverse time at the underlay network. It is clear in all figures that, the point-to-point delay of the perception-based distributed architecture is far better than the centralized and perception-based centralized architectures. Regarding the perception-based distributed architecture, the larger the cluster size (k) is, the higher the delay is. Therefore, it is better to keep it small. When the newly proposed perception-based function is applied to reduce the unnecessary traffic, the point-to-point delay performance is improved. The more unnecessary traffic is reduced by applying our newly proposed perception-based function, the lower the point-to-point delay is.

C. Required service rate

In general, a required service rate is necessary to maintain stability in a queue. According to the queuing theory, this required service rate can be calculated by the following condition:

$$\rho = \frac{\lambda}{\mu} < 1 \tag{38}$$

We will calculate and analyze these requirements in details for each architecture. If an architecture can fulfill its required service rate, it can maintain the stability of the service queues. There are required services rates to maintain steady-states at the queues processed by each architecture (M_l, M_p, M_S).

1) Perception-based distributed architecture: The traffic intensity at a leader of layer l is $\rho_l = \frac{\lambda_{Al}}{\mu_l}$. In order for the queue at the leader to be in steady-state conditions, we must have $\rho_l < 1$ or $\mu_l > \lambda_{Al}$. Assuming that the perception-based leader l has been designed to support the maximum throughput $Max(\lambda_{Al})$ and the system can support of up to N_{max} participants, we have:

$$\rho_l = \frac{\lambda_{Al}}{M_l} \text{ and } \mu_l = M_l = (k.N_{max} + 1).Max(\gamma^p_{hl}(t)) \tag{39}$$

$Max(\gamma^p_{hl}(t))$ is the peak value of the instantaneous $\gamma^p_{hl}(t)$.

2) Centralized architecture: The traffic intensity at the MCU is $\rho_m = \frac{\lambda_{AS}}{\mu_m}$ (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 $\mu_m > \lambda_{AS}$. Assuming that we have designed a MCU to support of up to N_{max} participants, and a peak value of the instantaneous $Max(\lambda_{AS})$ then the maximum throughput to be managed at the MCU is:

$$\rho_m = \frac{\lambda_{AS}}{M_m} \text{ and } \mu_m = M_m = (N^2_{max} - N_{max} + 1).Max(\gamma^a_{hS}(t)) \tag{40}$$

3) Perception-based centralized architecture:

$$\rho_p = \frac{\lambda_{Ap}}{M_p} \text{ and } \mu_p = M_p = (N^2_{max} - N_{max} + 1).Max(\gamma^p_{hp}(t)) \tag{41}$$

4) Result Analysis: Figures 7 and 8 show the required service rates among the three architectures for two different traffic configurations ($r_e = 0.1, r_e = 0.5$). If a steady state is maintained in a queue, the waiting time can be high but never be infinitive meaning that all of the traffic will be definitely processed. Otherwise, if the steady state is not maintained, the queues will enter a blocked state in which no more traffic can be processed and congestion happens. It is clear from the two figures that the required service rate at the perception-based distributed architecture is much smaller than it is at the centralized and perception-based centralized architectures. With the same perception-based distributed architecture, and with the total number of 50 participants, the required service rate increases when the cluster size increases. Therefore, for this configuration of video conference, it is recommended to use a smaller cluster size. All two figures

show that, when the newly proposed perception-based function is applied, the perception-based centralized architecture can reduce the required service rate in comparison with the centralized architecture. The more unnecessary traffic is reduced by applying our newly proposed perception-based function, the lower the required service rate is. There is an obvious relation between the required service rate and the price of the solution built from each architecture. This relation can be exponential. We can conclude that, the distributed architecture and the newly proposed perception-based function, when applied, can reduce the required service rate and therefore the cost of the video conferencing service.

D. Estimation of global throughput in the network

In this section, we will estimate the global throughput generated by the three architectures. The purpose is to prove that the newly proposed perception-based distributed architecture does not actually increase the total traffic transmitted on the overlay network. It can even reduce the total traffic when the perception-base function is applied to throttle the enhancement video layer traffic.

1) Perception-based distributed architecture: Considering a peer j at layer l of the perception-based distributed architecture, we have the throughput from a peer j to a leader l and the throughput from the leader l to the peer j are $\lambda_{j\to l}$ and $\lambda_{l\to j}$, respectively. The throughput from a peer j to a leader l ($\lambda_{j\to l}$) can be calculated by adding the traffic arriving from each participant:

$$\lambda_{j\to l} = \sum_{h=1+(j-1)k^{(l-1)}}^{jk^{l-1}} \gamma_{hl}^{p}(t) \tag{42}$$

$$\lambda_{l\to j} = \sum_{h=k^l+1}^{N} \gamma_{hl}^{p}(t) + \left[\sum_{h=1}^{k^l} \gamma_{hl}^{p}(t) - \sum_{h=1+(j-1)k^{(l-1)}}^{jk^{(l-1)}} \gamma_{hl}^{p}(t)\right] \tag{43}$$

The total throughput in a cluster is calculated by the throughput from all k participants and the multicasting throughput from the leader l to k participants.

$$T_{cluster}^{l} = \sum_{h=1}^{k^l} \gamma_{h}^{p}(t) + k \sum_{h=k^l+1}^{N} \gamma_{h}^{p}(t) + (k-1) \sum_{h=1}^{k^l} \gamma_{h}^{p}(t) = k \sum_{h=1}^{N} \gamma_{h}^{p}(t) \tag{44}$$

There are totally $\frac{N}{k^l}$ clusters at layer l. Thus, the total throughput at layer l is:

$$T^l = \frac{N}{k^{(l-1)}} \sum_{h=1}^{N} \gamma_h^p(t) \tag{45}$$

Since there are $l_{max} = \log_k N$ layers, the total throughput is:

$$T_d = \sum_{l=1}^{l_{max}} T^l = kN \sum_{h=1}^{N} \gamma_h^p(t) \left(\sum_{l=1}^{l_{max}} \frac{1}{k^l} \right) = kN \sum_{h=1}^{N} \gamma_h^p(t) \left[\frac{(N-1)}{(k-1)N} \right] = \frac{k(N-1)}{(k-1)} \sum_{h=1}^{N} \gamma_h^p(t) \tag{46}$$

2) Perception-based centralized architecture: We mainly have the throughputs from participant j to the perception-based centralized leader L ($\lambda_{j \to L} = \gamma_{h=j}^p(t)$) and vice versa from the perception-based centralized leader L to the participant j ($\lambda_{L \to j} = \sum_{h=1}^{N} \gamma_{hl}^p(t) - \gamma_{h=j}^p(t)$). Therefore, the total throughput is:

$$T_p = N \sum_{h=1}^{N} \gamma_h^p(t) \tag{47}$$

3) Centralized architecture: By replacing $\gamma_h^p(t)$ with $\gamma_h^a(t)$ in Equ.47, we have:

$$T_m = N \sum_{h=1}^{N} \gamma_h^a(t) \tag{48}$$

4) Result analysis: From Figures 10 and 11 we can see the total throughputs in the three architectures. In fact, the total throughput of the perception-based distributed architecture is equivalent to the total throughput of both the centralized and perception-based centralized architectures when 50% ($r_e = 0.5$) and all ($r_e = 1$) participants send their enhancement video layers. We can hardly realize the difference between these throughputs in a logarithm plot. When the newly proposed perception-based function is fully applied in Fig.9 so that only 10% of the participants send their enhancement video layers ($r_e = 0.1$), the total throughput of the perception-based centralized and perception-based distributed architectures can be even far lower than the total throughput of the centralized architecture when no perception-based function is applied. This will answer many concerns about whether the perception-based distributed architecture has to manage a higher total throughput than the two conventional centralized architectures or not. The answer is clearly no. It can even reduce the total traffic when the perception-base architecture is applied to throttle the enhancement video layer traffic.

IV. CONCLUSION

In this research, a new enriched distributed video conferencing architecture considering the limitation of human's perception has been proposed. Mathematical analysis and models have been built and compared for the three architectures using queuing theory in terms of total waiting time, point-to-point delay, required service rates, and total throughput. It is worth noticing that all the enriched features of the proposed video conferencing architecture have been modelled in details and included in the mathematical analysis and expressions. The theoretical models and mathematical expressions allow us to evaluate the performance in all four different criteria for all three different architectures in real-time, with an arbitrary number of participants and with very heterogeneous contexts of peers. Our mathematical models and expressions can be used to determine the optimal cluster size for a given number of participants of the distributed video conference. Numerical simulations obtained from the theoretical analysis models and the off-line statistical data have been done in the context of a multi-party multi-layer video conferencing service. The results show that the newly proposed perception-based distributed architecture outperforms over all four performance criteria the centralized and perception-based centralized architectures. Regarding the original question we try to solve from the beginning of the paper for whether the distributed architecture is better than the centralized architecture or not, we can conclude that, from three different criteria (total waiting time, end-to-end delay and required service rate) that we have chosen to compare, the distributed architecture outperforms the two conventional centralized architectures. Especially when the total number of participants is large. In terms of the total throughput, it is also shown that the distributed architecture has an equivalent performance with the centralized architecture. The newly proposed perception-based function when applied can reduce the total waiting time, point-to-point delay, the required service rate. It can maintain an equivalent or even can reduce the total throughput of the perception-based distributed architecture in comparison with the centralized architecture. In our future work, more

simulations exploiting all the power and details of our mathematical models and expressions should be done. The instantaneous values of γ and λ can be applied for evaluating a global performance in real-time of the three architectures with heterogeneous contexts of peers.

REFERENCES

[1] K. Tirasoontorn, S. Kamolphiwong, and S. Sae-Wong, "Distributed P2P-SIP conference construction," in *Proceedings of the International Conference on Mobile Technology, Applications, and Systems*. ACM, 2008, p. 20.

[2] T. M. O'Neil, "Quality of experience and quality of service for IP video conferencing," *Polycom Video Communications, Milpitas, CA, USA, White paper*, 2002.

[3] Y. Lu, Y. Zhao, F. Kuipers, and P. Van Mieghem, "Measurement study of multi-party video conferencing," *NETWORKING 2010*, pp. 96–108, 2010.

[4] M. S. Silver, "Browser-based applications: popular but flawed?" *Information Systems and E-Business Management*, vol. 4, no. 4, pp. 361–393, 2006.

[5] R. Spiers and N. Ventura, "An Evaluation of Architectures for IMS Based Video Conferencing," *University of Cape Town, Rondebosch South Africa*, 2009.

[6] L. De Cicco, S. Mascolo, and V. Palmisano, "Skype video responsiveness to bandwidth variations," in *Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video*. ACM, 2008, pp. 81–86.

[7] S. A. Baset and H. Schulzrinne, "An analysis of the skype peer-to-peer internet telephony protocol," *Arxiv preprint cs/0412017*, 2004.

[8] S. E. Deering, "Multicast routing in a datagram internetwork," 1991.

[9] 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.

[10] F. Pereira, L. Torres, C. Guillemot, T. Ebrahimi, R. Leonardi, and S. Klomp, "Distributed Video Coding: Selecting the most promising application scenarios," *Signal Processing: Image Communication*, vol. 23, no. 5, pp. 339–352, 2008.

[11] H. Jeong, J. Abuan, J. Normile, R. Salsbury, and B. S. Tung, "Heterogeneous video conferencing," May 2011.

[12] T. A. Le and H. 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, Jun. 2010, pp. 394–399.

[13] T. A. L. H. N. H. Zhang, "EvalSVC - an evaluation platform for scalable video coding transmission," in *14th International Symposium on Consumer Electronics (ISCE 2010)*, Braunschweig, Germany, Jun. 2010, pp. 85–90.

[14] T. A. Le and H. Nguyen, "Perception-based Application Layer Multicast Algorithm for scalable video conferencing," in *IEEE GLOBECOM 2011 - Communication Software, Services, and Multimedia Applications Symposium (GC'11 - CSWS)*, Houston, Texas, USA, Dec. 2011.

[15] T. A. Le, H. Nguyen, and H. 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.

[16] K. Boakye, B. Trueba-Hornero, O. Vinyals, and G. Friedland, "Overlapped speech detection for improved speaker diarization in multiparty meetings," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 4353–4356.

[17] D. Gross, *Fundamentals of queueing theory*. Wiley-India, 2008.

[18] G. Van der Auwera, P. T. David, and M. Reisslein, "Traffic and quality characterization of single-layer video streams encoded with the H. 264/MPEG-4 advanced video coding standard and scalable video coding extension," *IEEE Transactions on Broadcasting*, vol. 54, no. 3 part 2, pp. 698–718, 2008.

[19] G. Van der Auwera, P. T. David, M. Reisslein, and L. 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. [Online]. Available: http://dx.doi.org/http://dx.doi.org.gate6.inist.fr/10.1155/2008/164027

[20] B. Y. Choi, S. Moon, Z. L. Zhang, K. Papagiannaki, and C. Diot, "Analysis of point-to-point packet delay in an operational network," *Computer Networks*, vol. 51, no. 13, pp. 3812–3827, 2007.

TABLE I: Scalable video codec parameters and notations for reference of the queuing model.

Video sequences	”Silence of the Lambs”
Video codec	H.264 Spatial SVC
Base video layer: QCIF	
Mean frame size ($\bar{X}_b$)	0.632 [kbyte]
Coefficient of variation of frame size (Cov_b)	1.757 [unit free]
Mean bit rate ($\bar{\gamma}_b$)	0.152 [Mbps]
SCV_b	16 [kbit]
Enhancement video layer: CIF	
Mean frame size ($\bar{X}_e$)	0.962 [kbyte]
Coefficient of variation of frame size (Cov_e)	2.043 [unit free]
Mean bit rate ($\bar{\gamma}_e$)	0.231 [Mbps]
Aggregated video: CIF	
Mean frame size ($\bar{X}_a$)	0.1.594 [kbyte]
Coefficient of variation of frame size (Cov_a)	1.9 [unit free]
Mean bit rate ($\bar{\gamma}_a$)	0.383 [Mbps]
SCV_a	46 [kbit]

Continued on next page

TABLE I – *Continued from previous page*

N, k, n^e_{max}	Total number of participating peers, maximum cluster size, and the average of maximum number of enhancement video layers, respectively
$X^b_{hl}(t), X^{ei}_{hl}(t), X^p_{hl}(t),$ $X^p_{hp}(t), X^a_{hS}(t)$	Instantaneous random inter-arrival time of the traffic generated by the base, i^{th} enhancement video layer, perception-based distributed aggregated, perception-based centralized aggregated, and centralized aggregated video layers from the participant h to the leader l of its cluster, to the centralized leader p, or to the centralized server S, respectively
$C^2_{Bl}, C^2_{Bp}, C^2_{Bs}$	Squared coefficient of variation (SCV) of service distributions at leader of layer l, at the top leader p of the perception-based centralized architecture, and MCU, respectively. These parameters depend on the hardware of processing nodes (peers, top leader, MCU)
$C^2_{Aj}, C^2_{Al}, C^2_{Ap}, C^2_{AS}$	Squared coefficient of variation (SCV) of service distributions at each peer, leaders of layer l, the top leader p, and the MCU, respectively. These parameters depend on the variation of the *arriving* traffic to the peers, leaders, and MCU

Continued on next page

TABLE I – *Continued from previous page*

$C^2_{Dj}, C^2_{Dl}, C^2_{Dp}, C^2_{Dm}$	Squared coefficient of variation (SCV) of service distributions at each peer, leaders of layer l, top leader p, and MCU, respectively. These parameters depend on the variation of the *departing* traffic from the peers, leaders, and MCU
$C^2_{ij}, C^2_{j\to S}, C^2_{j\to l}, C^2_{j\to p}, C^2_{\psi\to l}$	Squared coefficient of variation (SCV) of traffic from entity i to j, from peer j to the MCU, from peer j to a leader of layer l, from peer j to the top leader p, from the "multicast duplicator" to the leader of layer l, respectively
$C^2_{o\psi S}, C^2_{o\psi p}, C^2_{o\psi l}$	SCV of traffic from the "multicast duplicator" to the centralized MCU, the perception-based centralized leader, and the leader at layer l
$\gamma^a_{hl}(t)$, $\gamma^b_{hl}(t)$, $\gamma^{ei}_{hl}(t)$, $\gamma^p_{hl}(t)$, $\gamma^p_{hp}(t)$, $\gamma^a_{hS}(t)$, $\gamma^a_h(t)$, $\gamma^p_h(t)$	Instantaneous aggregated, base, i^{th} enhancement video traffic generated by the SVC video encoder at h^{th} participant, the instantaneous video traffic from the participant h to the leader l of its cluster, from the participant h to the perception-based centralized server, from the participant h to the centralized server (mcu), the instantaneous video traffic from the participant h, the instantaneous aggregated video traffic from the participant h, respectively
$\gamma_{\psi\to l}, \gamma_{\psi\to S}, \gamma_{\psi\to p}$	The external traffic from the "multicast duplicator" to the leader l, the centralized MCU, and the perception-based centralized leader, respectively

Continued on next page

TABLE I – *Continued from previous page*

$\bar{\gamma}_b = E_{h,t}(\gamma_{hl}^b(t))$, $\bar{\gamma}_e = E_{h,t}(\gamma_{hl}^{ei}(t))$, $\bar{\gamma}_p = E_{h,t}(\gamma_{hp}^p(t))$, $\bar{\gamma}_a = E_{h,t}(\gamma_{hS}^a(t))$	Mean value over time and for all of the conference participants of the base, enhancement, and aggregated video traffic from h^{th} participant to a leader in l^{th} layer and to the leader l of its cluster, respectively
$\lambda_{Al}, \lambda_{AS}, \lambda_{Ap}$	The throughput arriving to a leader at layer l, the centralized MCU, the perception-based centralized leader, respectively
$\lambda_{i\to j}, \lambda_{j\to S}, \lambda_{j\to l}, \lambda_{j\to p}$, $\lambda_{j\to L}, \lambda_{L\to j}$	Throughput from entity i to j, from peer j to the MCU, from peer j to a leader of layer l, from j to the perception-based centralized leader, from member j to the centralized leader L and from the centralized leader L to the member j in the perception-based centralized architecture, respectively
$\lambda_j(t), \lambda_l(t), \lambda_m(t)$	Instantaneous throughput at the peer j, the leader l, and the MCU, respectively
$r_{hl}^b(t), r_{hl}^{ei}(t), r_{hp}^b(t), r_{hp}^{ei}(t)$	Instantaneous values of the traffic rectification coefficients on base and i^{th} enhancement video layers from h^{th} participant to the leader l of its cluster, to the perception-based centralized server
$\bar{r}_b = E_{h,t}(r_{hl}^b(t))$, $\bar{r}_e = E_{h,t}(r_{hl}^{ei}(t))$	Mean values of the traffic rectification coefficients for base, enhancement video traffics from h^{th} participant to a leader in l^{th} layer and to the leader l of its cluster, respectively

Continued on next page

TABLE I – *Continued from previous page*

μ_l, μ_p, μ_m, M_l, M_p, M_m	Service rates and required service rates at the leaders on layer l of the perception-based distributed architecture, at the central leader of the perception-based centralized architecture and at the central MCU of the centralized architecture
ρ_l, ρ_p, ρ_S	Traffic intensity (traffic congestion) at leaders of layer l, at the top leader p, at the MCU, respectively ($\rho = \frac{\lambda}{\mu}$)

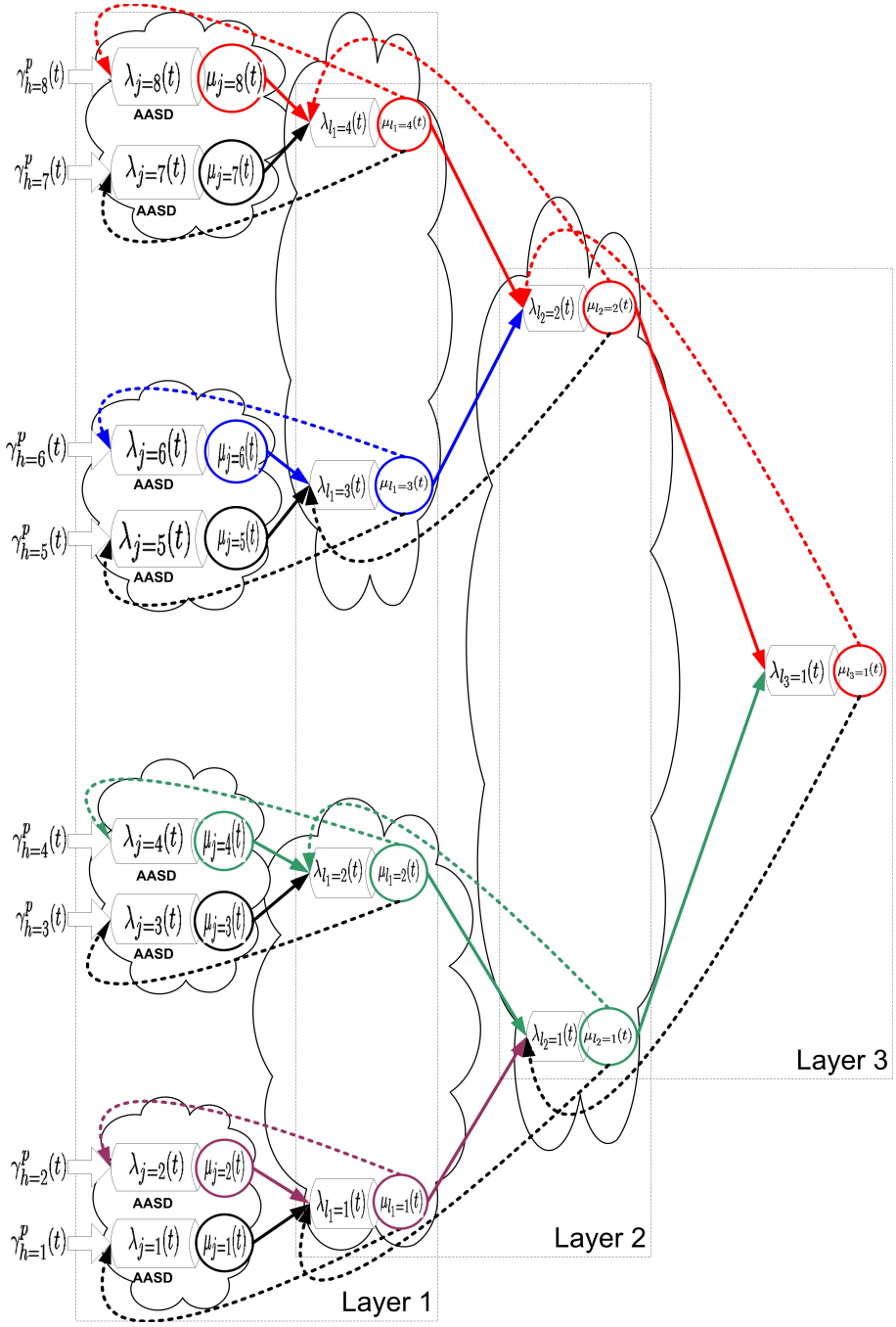

Fig. 1. General model analysis of the perception-based distributed video conference service architecture when N=8, k=2.

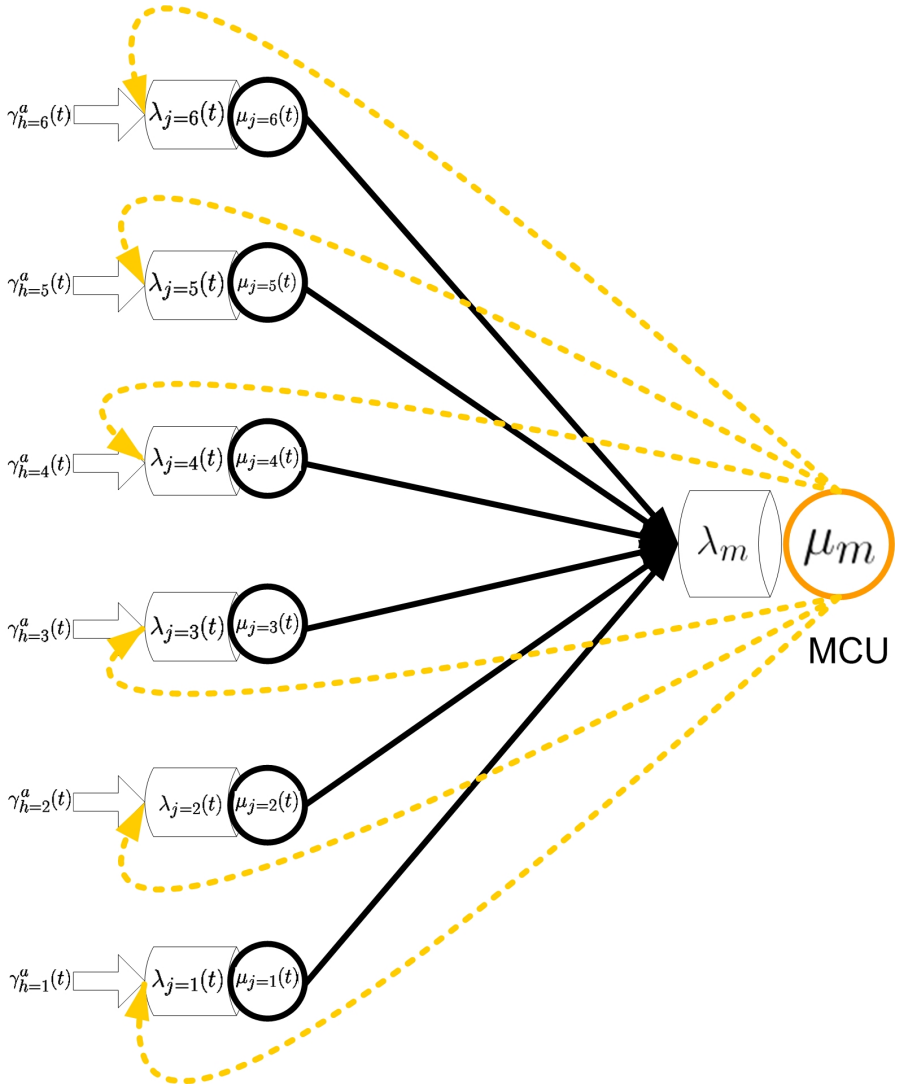

Fig. 2. General model analysis of the centralized video conference service architecture when N=6.

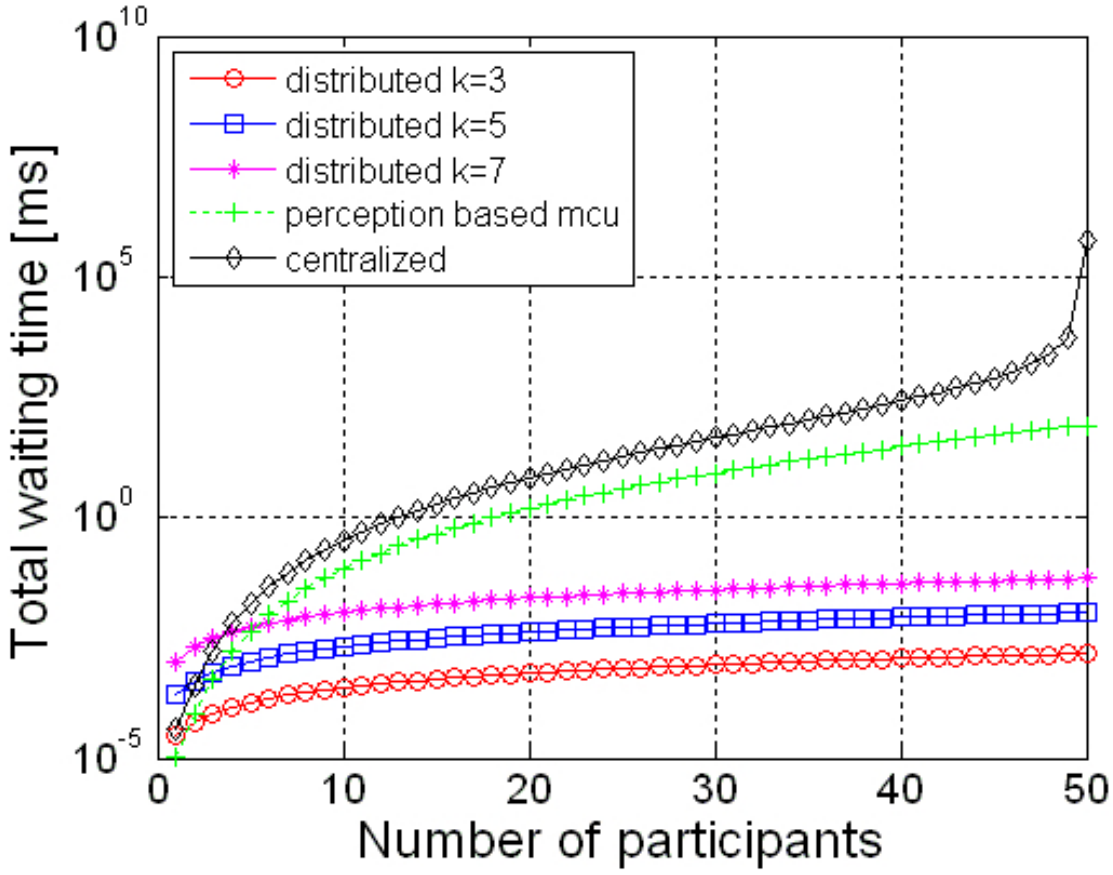

Fig. 3. Comparison of queuing waiting time among centralized, perception-based centralized and perception-based distributed architectures at minimum traffic when $r_b = 1, r_e = 0.1, n_e = 3$.

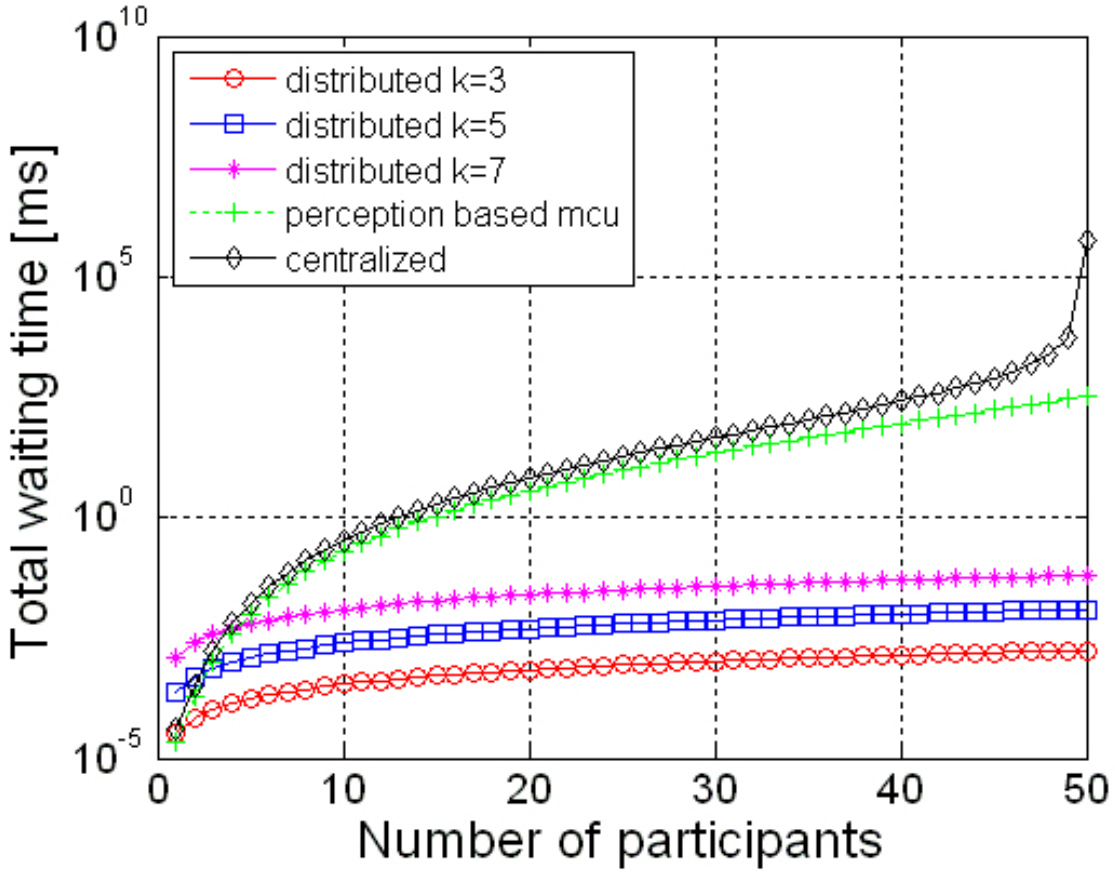

Fig. 4. Comparison of queuing waiting time among centralized, perception-based centralized and perception-based distributed architectures at reduced traffic when $r_b = 1, r_e = 0.5, n_e = 3$.

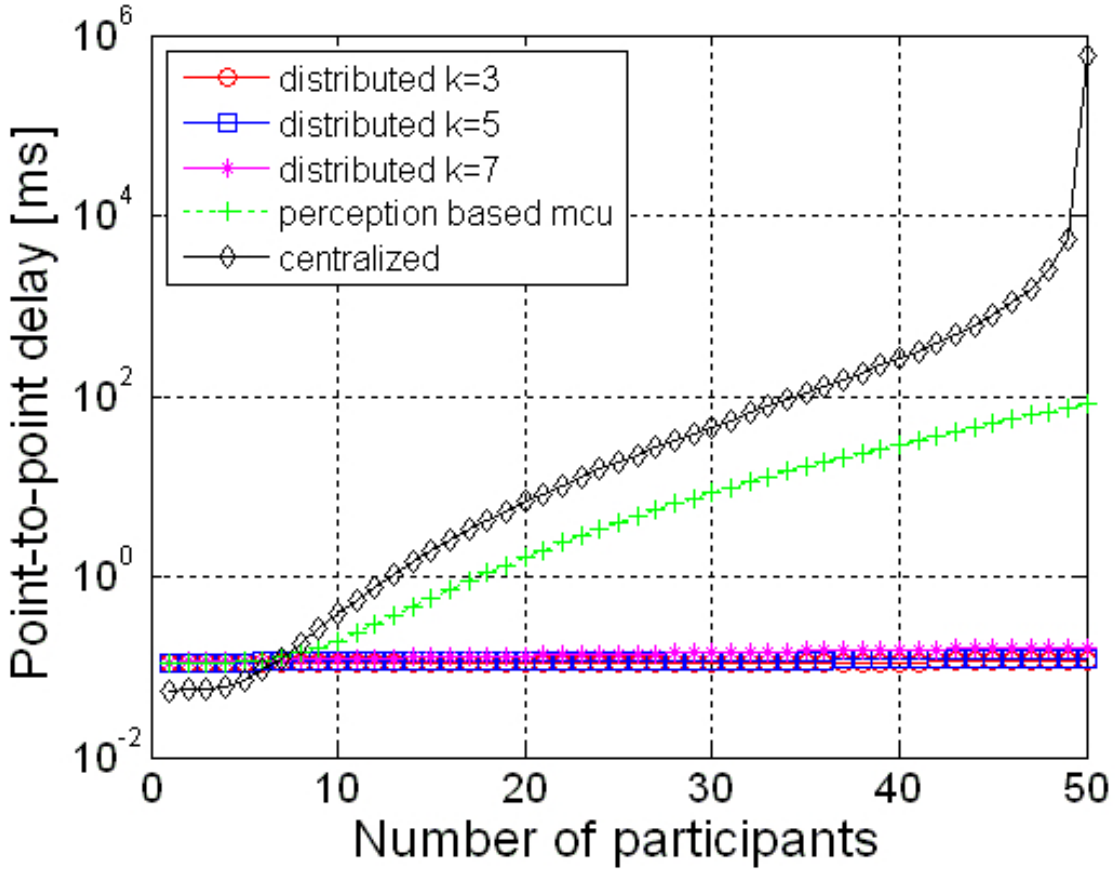

Fig. 5. Comparison of total point-to-point delay among centralized, perception-based centralized and perception-based distributed architectures at minimum traffic when $r_b = 1, r_e = 0.1, n_e = 3$.

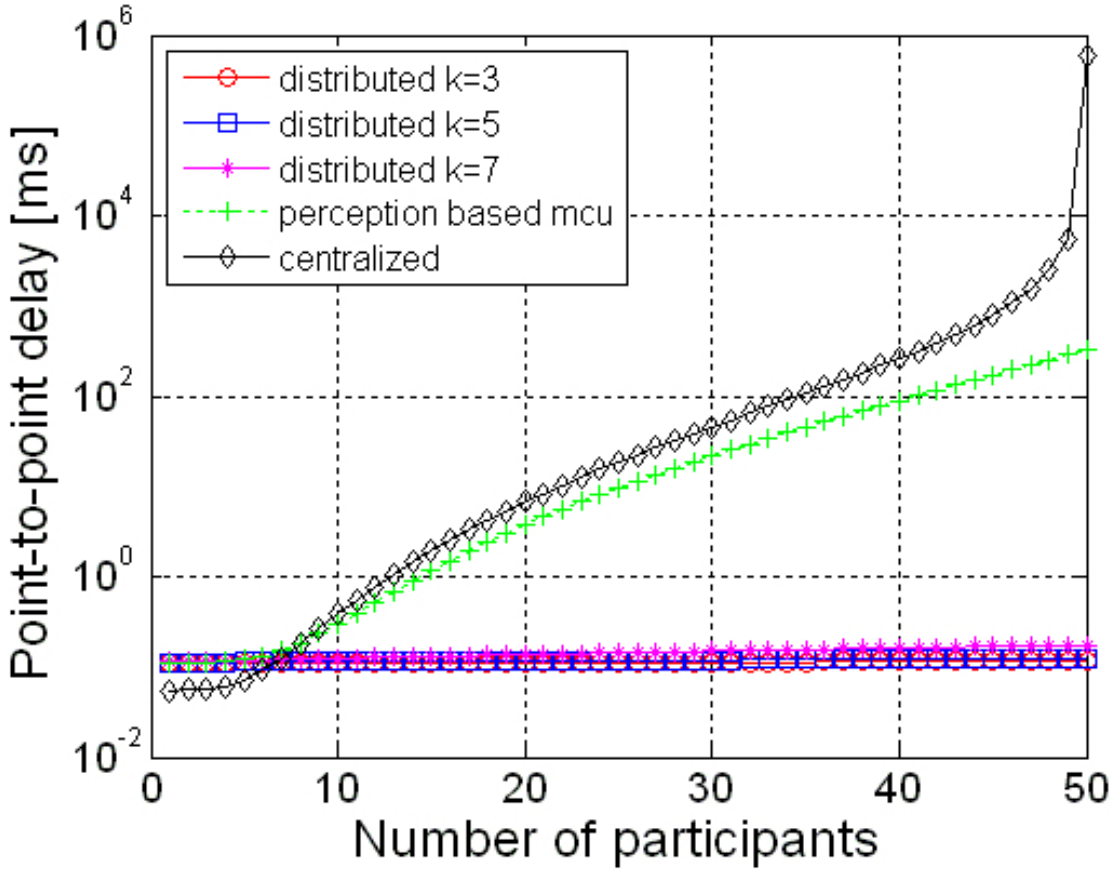

Fig. 6. Comparison of total point-to-point delay among centralized, perception-based centralized and perception-based distributed architectures at reduced traffic when $r_b = 1, r_e = 0.5, n_e = 3$.

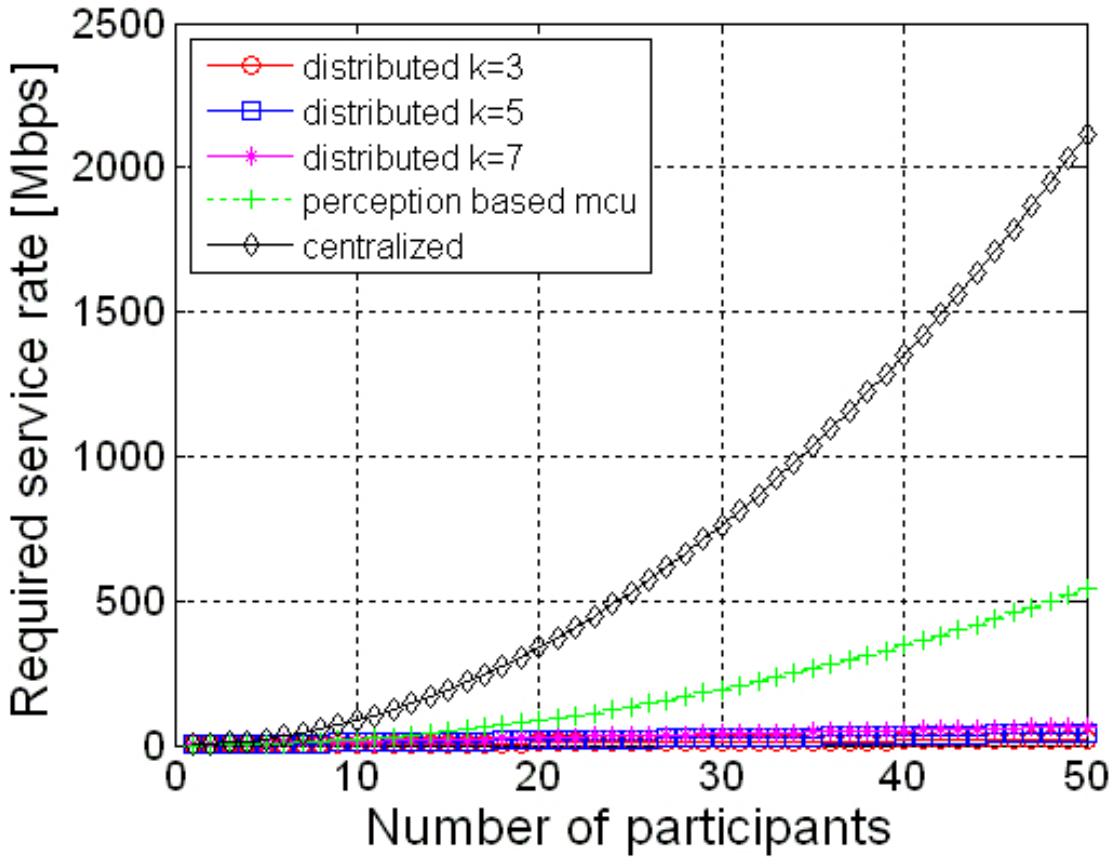

Fig. 7. Comparison of required service rates among centralized, perception-based centralized and perception-based distributed architectures at minimum traffic when $r_b = 1, r_e = 0.1, n_e = 3$.

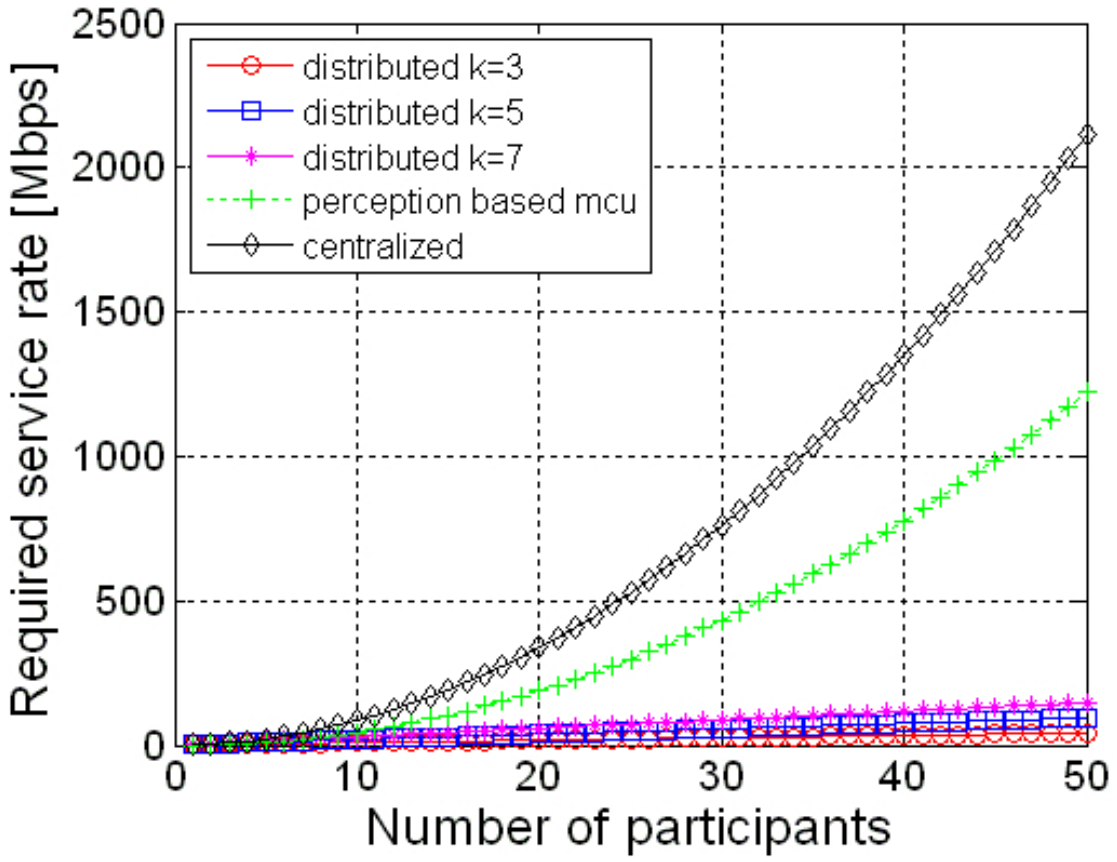

Fig. 8. Comparison of required service rates among centralized, perception-based centralized and perception-based distributed architectures at reduced traffic when $r_b = 1, r_e = 0.5, n_e = 3$.

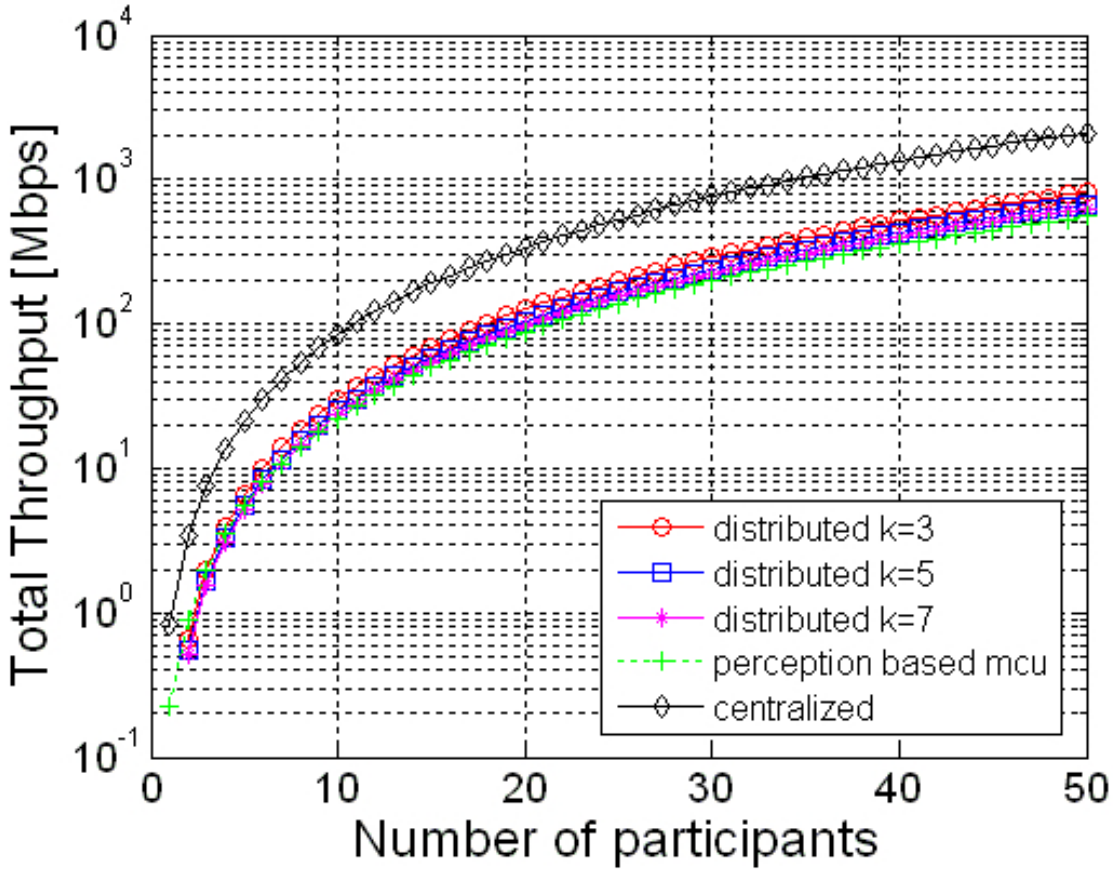

Fig. 9. Comparison of total throughput among centralized, perception-based centralized and perception-based distributed architectures at minimum traffic when $r_b = 1, r_e = 0.1, n_e = 3$.

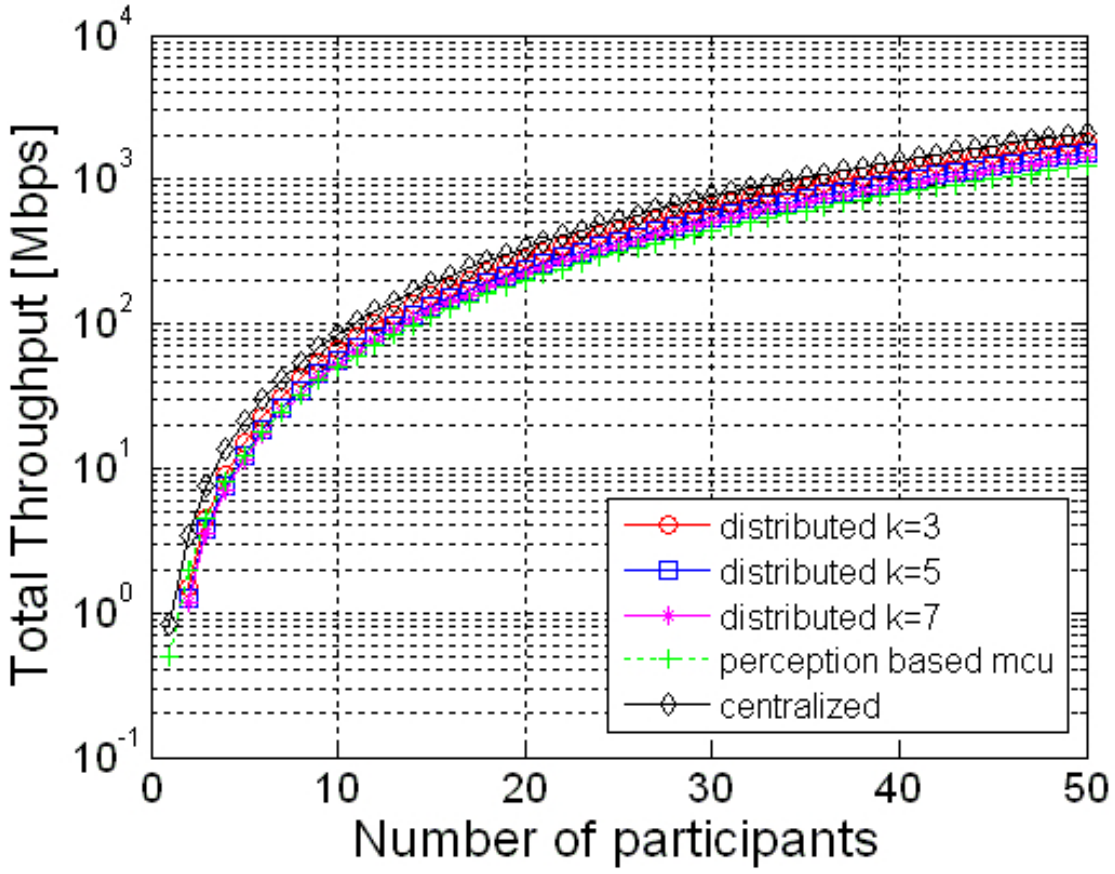

Fig. 10. Comparison of total throughput among centralized, perception-based centralized and perception-based distributed architectures at reduced traffic when $r_b = 1, r_e = 0.5, n_e = 3$.

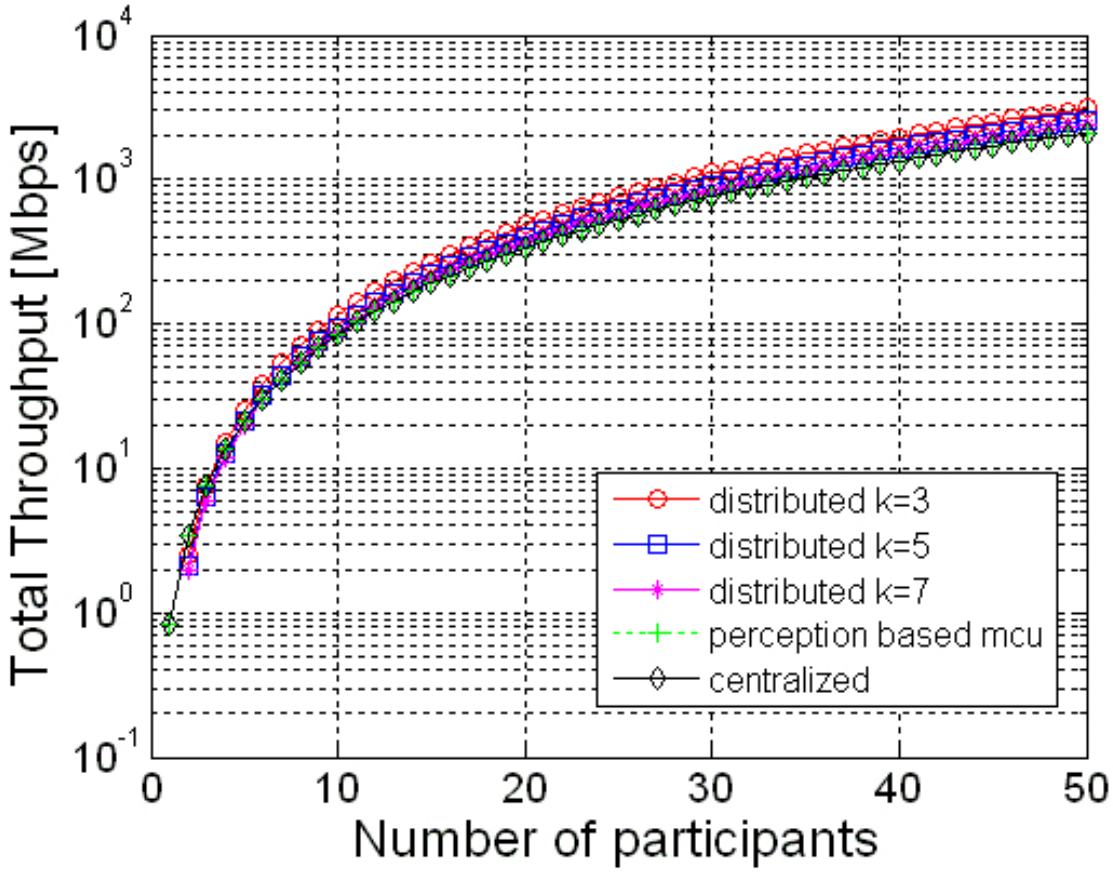

Fig. 11. Comparison of total throughput among centralized, perception-based centralized and perception-based distributed architectures at full traffic when $r_b = 1, r_e = 1, n_e = 3$.