Cross-Layer Rate Allocation for Multimedia Applications in Pervasive Computing Environment

Liang Zhou, Benoit Geller, Anne Wei, Baoyu Zheng, Jingwu Cui, Shan Xu

HAL Id: hal-01225534
https://hal.science/hal-01225534

Submitted on 4 Dec 2015

Cross-Layer Rate Allocation for Multimedia Applications in Pervasive Computing Environment

Liang Zhou[†,‡], Benoit Geller[⋆,∗], Anne Wei[♭], Baoyu Zheng[‡], Jingwu Cui[‡], Shan Xu[‡]
† Electronic Engineering Department, Shanghai Jiao Tong University, Shanghai, China;
‡Institute of SP&T, Nanjing University of Posts and Telecommunications, Nanjing, China;
⋆Laboratory SATIE, Ecole Normale Supérieure (E.N.S.), Cachan, France;
∗LEI, Ecole Nationale Supérieure des Techniques Avancées, Paris, France
♭Laboratory CEDRIC, Conservation National des Arts et Métiers, Paris, France

Abstract— An important issue for supporting multimedia applications in multiple heterogeneous networks, a typical pervasive computing environment, is how to optimize the rate allocation by intelligently utilizing the available network resources while, at the same time, to meet each application's QoS (Quality of Service) requirement. In this work, we develop and evaluate a rate allocation scheme in terms of audio and video applications based on a cross-layer design framework. At first, we construct a general distortion model according to the observed parameters in each network, as well as each application's rate-distortion characteristic. Then, the rate allocation is formulated as a convex optimization problem that minimizes the sum of the expected distortion of all applications. Furthermore, the realization of the distributed rate allocation algorithm for achieving an optimal or close-to-optimal end-to-end QoS under the overall limited resource budget is the highlight of this paper. Simulation results are provided which demonstrate the effectiveness of our proposed distributed rate allocation scheme.

I. INTRODUCTION

Pervasive computing environment provides seamless access to multiple heterogeneous networks at any time from any location for any user [1]. In this environment, the service providers have interests in operating kinds of multimedia applications, and integrating the characteristics of heterogeneous networks and multimedia applications to provide pervasive service.

Inevitably, there are huge and different kinds of application data streaming from different users which may influence each other and thus, it is essential to enforce a distributed rate allocation policy designed for suitable application metrics and efficient network utilization. Indeed, the problem of rate allocation over heterogeneous networks is, compared to traditional network, further complicated by the heterogeneity in both the application contents and the network conditions. This poses a serious challenge for providing, in this context, dependable and different applications that meet the users' requirements in terms of QoS, reliability and availability [2].

The issue of rate allocation among multiple flows over limited network resources is still an open problem and has received considerable attention recently. For media applications over UDP, TCP-friendly rate control is widely used [3], and several modifications have been proposed to improve its media adequation [4], [5]. The problem of efficient utilization of multiple networks via a suitable allocation of media flows

has also been explored from different aspects. [6] addresses a game-theoretic framework which allocates bandwidth for elastic services with fixed capacities. In addition, [7] proposes a cost price mechanism, which enables a mobile device to split its traffic among several 802.11 access points based on the throughput obtained and the price charged. However, [7] does not take into account the existence of heterogeneous networks with different traffic characteristics, nor does it specify an operational method to split the traffic. In [8], a specific rate allocation in heterogeneous networks is proposed, by taking into account the impact of both packet loss rates and available bandwidth over each link. To the best of our knowledge, the current literatures consider the multimedia application and the heterogeneous networks characteristics separately and independently. In order to provide a satisfying QoS for any application in the context of heterogeneous networks, the above two factors are jointly considered in this paper.

The rest of paper is organized as follows. In the next section, we present the system distortion model of the application and network, respectively. In Section III, we propose a distributed rate allocation scheme for multimedia application streaming sessions in heterogeneous networks. Then, we provide some selected simulation results for the proposed rate allocation scheme in Section IV and make some concluding remarks in Section V.

II. SYSTEM DISTORTION MODEL

A. Multimedia Applications

Assume multiple users $\mathbf{S} = \{1, ...s..., S\}$ that simultaneously access any of two different multimedia applications via a server, namely real-time video streaming ($\mathcal{A}_1$), and audio conversation ($\mathcal{A}_2$). Let user $s \in \mathbf{S}$ access one of the available applications $\mathcal{A}_i$ ($i = 1, 2$) at one transmission period. The server decides the average allocated rate to user s that has chosen application $\mathcal{A}_i$. We assume that the server can scalably adapt the transmission process to the channel conditions for user s. To this end, for each application $\mathcal{A}_i$, the server can choose the right transmission parameters, from a predefined set of available parameters $R_{\mathcal{A}_i}$. We assume that $N_{\mathcal{A}_1}$ encoded video layers and $N_{\mathcal{A}_2}$ audio transcoders are available at the server. Each video layer $l \leq N_{\mathcal{A}_1}$ is characterized by its

average encoding rate ρ_l and each transcoder $v \leq N_{\mathcal{A}_2}$ is characterized by its encoding rate ρ_v. We define $R_{\mathcal{A}_1} = \{\rho_l : 1 \leq l \leq N_{\mathcal{A}_1}\}$, $R_{\mathcal{A}_2} = \{\rho_v : 1 \leq v \leq N_{\mathcal{A}_2}\}$ as the available parameter sets for the video and audio applications, respectively.

In general, all the video and audio sources should be compressed conveniently for transmission and storage; however, this leads to information loss. According to [9], the distortion caused by source compression can be approximated by:

$$D_{comp} = \frac{\theta}{R - R^0} + D^0 \tag{1}$$

where R is the rate of the video/audio stream, which is equal to $R_{\mathcal{A}_i}$ (i=1,2) as described above; θ, R^0 and D^0 are the parameters of the distortion model which depend on the encoded video/audio sequence as well as on the encoding structure. Using nonlinear regression techniques, these parameters can be estimated from empirical rate-distortion curves obtained by encoding a sequence at different rates.

B. Multiple Heterogeneous Networks

Similarly to D_{comp}, the distortion caused by packet loss in heterogeneous networks can be modeled by a linear model related to the packet loss rate P_{loss}:

$$D_{loss} = \kappa P_{loss} \tag{2}$$

where κ depends on parameters related to the compressed sequence, such as the proportion of intra-coded macro-blocks and the effectiveness of error concealment at the decoder. The packet loss rate P_{loss} combines the rate of random losses and late arrivals of packets. In a bandwidth-limited network, this combined loss rate can be further modeled based on the M/M/1 queuing model. In this case, the delay distribution of packets over a single link is exponential [9]. Note that, since the end-to-end delay of packet delivery in wireless network is dominated by the queuing delay at the bottleneck link, the empirical delay distribution for realistic traffic patterns can still be modeled by an exponential formulation function:

$$Pr\{Delay > T\} = e^{-\omega T} \tag{3}$$

where T reflects the delay constraint, and ω is the arriving rate which is determined by the average delay:

$$\omega = \frac{1}{E\{Delay\}} \tag{4}$$

Generally, ω needs to be determined empirically from end-to-end delay statistics over the network. In order to present a general solution for online operation, here we employ a model to approximate the average packet delay.

Consider multiple networks $\mathbf{N} = \{1, 2, ..., N\}$ simultaneously available to multiple users $\mathbf{S} = \{1, 2, ..., S\}$. Each network $n \in \mathbf{N}$ is characterized by its Available Bit Rate ABR^n and Round Trip Time RTT^n, which are measured and updated periodically. It should be noted that as channel conditions in wireless environments change on very short time scales (e.g., up to a few tens of ms), we assume that ABR^n and RTT^n represent average values computed on larger time

scale (e.g., one to a few seconds), and represent the average channel conditions for user $s \in \mathbf{S}$ on the given period.

Therefore, we can use a matrix $\mathbf{R} = \{R_s^n\}_{S \times N}$ to denote the rate allocation, where each element R_s^n corresponds to the allocated rate of user $s \in \mathbf{S}$ over the network $n \in \mathbf{N}$. Therefore, the total allocated rate over network n is $R^n = \sum_{s \in \mathbf{S}} R_s^n$, and the total allocated rate for user s is $R_s = \sum_{n \in \mathbf{N}} R_s^n$. We express the Residual Bandwidth (RB) over network n as:

$$RB^n = ABR^n - \sum_{s \in \mathbf{S}} R_s^n \tag{5}$$

For user s in network n, the observed available bandwidth ABR_s^n is :

$$ABR_s^n = ABR^n - \sum_{s' \neq s, s' \in \mathbf{S}} R_{s'}^n \tag{6}$$

In fact, the average packet delay typically increases due to the network congestion. Here, we employ a simple fractional function to approximate the non-linear increase of the packet delay in $n \in \mathbf{N}$ based on M/M/1 model [10]:

$$E\{Delay\} = \frac{\beta^n}{RB^n} = \frac{\beta^n}{ABR^n - \sum_{s \in \mathbf{S}} R_s^n} \tag{7}$$

Assuming equal delay on both directions, the value of β^n can be estimated from the last past observations of RTT'^n and RB'^n:

$$\beta^n = \frac{RB'^n \cdot RTT'^n}{2} \tag{8}$$

More specifically, if the current residual bandwidth is equal to the past observation value for network $n \in \mathbf{N}$ ($RB'^n = RB^n$), the average current delay is $RTT'^n/2$. Therefore, for each network $n \in \mathbf{N}$:

$$Pr\{Delay > T\} = e^{-\omega T} = e^{-\frac{2(ABR^n - \sum_{s \in \mathbf{S}} R_s^n)}{RB'^n RTT'^n}T} \tag{9}$$

Taking into account P_B^n, the random packet loss rate in network $n \in \mathbf{N}$ due to transmission errors, the total packet loss rate in network $n \in \mathbf{N}$ is then:

$$\begin{aligned}
P_{loss}^n &= P_B^n + (1 - P_B^n)Pr\{Delay > T\} \\
&= P_B^n + (1 - P_B^n)e^{-\frac{2(ABR^n - \sum_{s \in \mathbf{S}} R_s^n)}{RB'^n RTT'^n}T} \tag{10}
\end{aligned}$$

The overall distortion from packet loss in network $n \in \mathbf{N}$ can be expressed as:

$$D_{loss}^n = \kappa P_{loss}^n = \kappa \left(P_B^n + (1 - P_B^n)e^{-\frac{2(ABR^n - \sum_{s \in \mathbf{S}} R_s^n)}{RB'^n RTT'^n}T} \right) \tag{11}$$

III. DISTRIBUTED RATE ALLOCATION SCHEME

In general, the reconstructed video/audio quality over network is affected by both the source compression from the encoder and the packet loss during the transmission. Here, we assume that the two forms of the induced distortion are independent and additive. Thus, we can calculate the overall distortion D_{all} as

$$D_{all} = D_{comp} + D_{loss} \tag{12}$$

Based on the previous discussion, we seek to minimize the sum of the total distortion D_{all} as follows:

$$\min_{s \in \mathbf{S}, n \in \mathbf{N}} \left\{ D_{all}(R_s^n) = \sum_{s \in \mathbf{S}} \left(\frac{\theta_s}{\sum_{n \in \mathbf{N}} R_s^n - R_s^0} + D_s^0 \right) + \right.$$

$$\left. \sum_{n \in \mathbf{N}} \kappa \left(P_B^n + (1 - P_B^n) e^{-\frac{2(ABR^n - \sum_{s \in \mathbf{S}} R_s^n)}{RB'^n RTT'^n} T} \right) \right\} \quad (13)$$

$$subject \ to \quad R_s^n = \frac{ABR_s^n}{\sum_{n \in \mathbf{N}} ABR_s^n} R_s, \forall n \in \mathbf{N}$$

$$R_s^n \leq ABR_s^n, \forall n \in \mathbf{N}$$

where θ_s, R_s^0 and D_s^0 are the corresponding parameters for user $s \in \mathbf{S}$. Intuitively, the reconstructed quality is limited by coarse quantization at low rates, whereas at high rates, the application stream will cause more network congestion. This, in turn, leads to higher loss rates and reduces the reconstructed quality. For multiple applications in bandwidth-limited environments, we therefore expect to achieve optimal QoS for some intermediate rate.

A. Piecewise Approximate Theorem

In order to get an optimal or close-to-optimal result with a fast convergence adapting to the online operation, here we propose a Continuous Piecewise-Linear (CPL) approach for solving the rate allocation optimization based on the utility framework introduced in [11], which iteratively takes a locally approximate optimal decision on each user in each network. The foundation of our proposed piecewise approximate method follows the fact that, the CPL function used to approximate the original goal function is convex in the convex union of many small hypercubes, and that an approximately globally optimal solution of the original problem confined in this union can be found in the set of locally solutions. In many cases, the number of such unions may be much less than that of all smaller hypercubes partitioned. Hence, the CPL approach can decrease the computational effort greatly.

Since D_{all} is the sum of the univariate functions D_{comp} for each user s and D_{loss} for each network n, CPL approximation can be obtained based on each function by a univariate CPL function. This can be achieved by partitioning the interested region of each univariate function into sufficiently many non-overlapping small intervals. Let g be an arbitrary univariate function whose interested region is $[\alpha, \alpha'] \subset \mathbb{R}$. Let m breakpoints $\alpha < \alpha_1 < \alpha_2 < \ldots < \alpha_m < \alpha'$ be suitably chosen so that g can be approximated by the linear function $\hat{l}_k = A_k x + B_k$ in each small interval $I_k = [\alpha_{k-1}, \alpha_k]$ for any $1 \leq k \leq m + 1$, where $\alpha_0 = \alpha$, $\alpha_{m+1} = \alpha'$; A_k and B_k are determined by the linear equations $g(\alpha_{k-1}) = \hat{l}_k(\alpha_{k-1})$, $g(\alpha_k) = \hat{l}_k(\alpha_k)$. An approximation function φ of g on $[\alpha, \alpha']$ can be obtained by connecting these segments.

For any $1 \leq k \leq m$, we call α_k an inflection point if $A_k > A_{k+1}$. Denote by $\alpha_{\tau(1)} < \alpha_{\tau(2)} < \ldots < \alpha_{\tau(q)}$ all inflection points among the breakpoints α_k ($q \leq m$). Let $\tau(0) = 0$, $\tau(q + 1) = m + 1$, and define $\hat{I}_t = [\alpha_{\tau(t-1)}, \alpha_{\tau(t)}]$ for any $0 \leq t \leq q+1$. It can be seen that $\hat{I}_t$ is the union of intervals I_k, $\tau(t-1) < k < \tau(t)$. Based on the above partition we can get a piecewise-convex expression of the function φ, which is very useful for the global optimization of separable programming problems. This is given by the following theorem:

Theorem: For any $1 \leq t \leq q + 1$,

$$\varphi(\lambda) = \max_{k \in \hat{s}_t} \hat{l}_k(\lambda), \quad \forall \lambda \in \hat{I}_t \quad (14)$$

where $\hat{s}_t = k, \tau(t - 1) < k \leq \tau(t)$.

Proof: For any $1 \leq t \leq q + 1$, since $A_{\tau(t-1)+1} < A_{\tau(t-1)+2} < \cdots < A_{\tau(t)}$, according to [12], function $\varphi(\hat{\lambda})$ is convex on $\hat{I}_t$, where $\hat{\lambda}$ is arbitrarily chosen in $\hat{I}_t$. There should be an integer $k \in \hat{s}_t$ such that $\hat{\lambda} \in I_k$ and $\varphi(\hat{\lambda}) = \hat{l}_k(\hat{\lambda})$. Then, the following relation must be satisfied,

$$\hat{l}_{k'}(\hat{\lambda}) \leq \hat{l}_k(\hat{\lambda}), \forall k' \in \hat{s}_t - \{k\} \quad (15)$$

Otherwise, there is a $k' \in \hat{s}_t - \{k\}$ for which $\hat{l}_{k'}(\hat{\lambda}) > \hat{l}_k(\hat{\lambda})$, and we can choose a $\bar{\lambda} \in \hat{I}_{k'} \subset \hat{I}_t$ and a sufficiently small positive number ε such that $\lambda' = \varepsilon \hat{\lambda} + (1 - \varepsilon) \bar{\lambda} \in I_{k'}$. Because φ is convex on $\hat{I}_t$, we have

$$\varphi(\lambda') \leq \varepsilon \varphi(\hat{\lambda}) + (1 - \varepsilon) \varphi(\bar{\lambda}) = \varepsilon \hat{l}_k(\hat{\lambda}) + (1 - \varepsilon) \hat{l}_{k'}(\bar{\lambda})$$
$$< \varepsilon \hat{l}_{k'}(\hat{\lambda}) + (1 - \varepsilon) \hat{l}_{k'}(\bar{\lambda}) = \hat{l}_{k'}(\lambda') \quad (16)$$

which contradicts the known relation $\varphi(\lambda') = \hat{l}_{k'}(\lambda')$. As the above $\hat{\lambda}$ is arbitrarily chosen, the theorem is proved. ∎

According to the above theorem, in order to get an approximate optimal value for the rate allocation problem ((13) is a convex function), we can partition the original interested region into a number of smaller non-overlapping hypercubes and approximate the goal function on every hypercube by a convex CPL function. In fact, the goal function D_{all} corresponds to the arbitrary univariate function g and any potential rate allocation is the breakpoint in the above theorem. Therefore, finding the appropriate breakpoints and judging whether they are inflection points or not are the key issues for implementing the piecewise approximate theorem to our rate allocation problem. In order to get around this difficulty, we employ a utility-based function described in the following subsection.

B. Rate Allocation Algorithm

We define $\overline{R_s^n} \to R_s^n$ as the transition of the next allocation rate for user $s \in \mathbf{S}$ in network $n \in \mathbf{N}$, and $\overline{R_s^n}$ is selected in the set of $R_{\mathcal{A}_i}$ (i=1,2). $\overline{R_s^n} = R_s^n + \triangle R_s^n$ where $\triangle R_s^n$ is the rate improvement varied at each iteration[1]. The utility of this transition can be calculated as:

$$U_s^n = \frac{\varphi(\overline{R_s^n}) - \varphi(R_s^n)}{\overline{R_s^n} - R_s^n} \quad (17)$$

where φ is the approximate linear function for D_{all} in the interval of $[R_s^n, \overline{R_s^n}]$. The total utility matrix is $\mathbf{U} = \{U_s^n\}_{S \times N}$. At each iteration, the proposed algorithm finds $\mathbf{R} = \{R_s^n\}_{S \times N}$ that brings the highest utility $\mathbf{U}^* = \{U_s^n\}_{S \times N}$ to the overall system by its transition:

$$\mathbf{U}^* = \arg \max_{\mathbf{R}} \mathbf{U} \quad (18)$$

---

[1]In theory, the initial $\triangle R_s^n$ can be chosen at random as long as it is less than ABR_s^n, here the initial value $\triangle R_s^n$ is set to $ABR_s^n/2$.

One starts to allocate resources to user s in network n. Once the resources of the network n are depleted, the algorithm will find a different user that can free the required resources for user s in the other network, by allocating part of its rate. This operation is performed as long as the overall utility of the system is still improved, and as long as free network resources still exist in the overall system. The algorithm stops when there are no more free resources in the network system, or when no other possible user transition can bring any improvement in the overall system utility.

TABLE I

DISTRIBUTED RATE ALLOCATION ALGORITHM

01: **Input:**
02: P_B^n, ABR_s^n, RTT^n $\forall$ user $s \in \mathbf{S}$ in network $n \in \mathbf{N}$;
03: $R_s^n = 0$, $\triangle R_s^n = ABR_s^n/2$, $\forall$ user $s \in \mathbf{S}$ in network $n \in \mathbf{N}$;
04: **Output:**
05: Global Rate Allocation $\mathbf{R}$
06: **Procedure RateAllocation**
07: while (true)
08: **for** s=1 to S **do**
09:     **for** n=1 to N **do**
10:       compute the utility of $\overline{R_s^n} \to R_s^n$:
11:       $U_s^n = \frac{\varphi(\overline{R_s^n}) - \varphi(R_s^n)}{\overline{R_s^n} - R_s^n}$;
12:       $\triangle R_s^n = \triangle R_s^n / U_s^n$;
13:       $\overline{R_s^n} = R_s^n + \triangle R_s^n$;
14:       update the approximate function φ;
15:     **end for**
16: **end for**
17: find $\mathbf{U}^* = \arg\max_{\mathbf{R}} \mathbf{U}$;
18: IntraNet($\mathbf{R}$, $\mathbf{U}^*$, n)
19: **Procedure** IntraNet($\mathbf{R}$, $\mathbf{U}^*$, n)
20: **if** network n has enough free resources **then**
21:     $\overline{R_s^n} \to R_s^n$;
22:     update free resources on network n;
23: **else**
24:     InterNet($\mathbf{R}$, $\mathbf{U}^*$, n);
25: **end if**
26: **Procedure** InterNet($\mathbf{R}$, $\mathbf{U}^*$, n)
27: find other user that can transfer part of its allocated to network
28: $n' \neq n \in \mathbf{N}$ with maximum transition utility improvement $\triangle \mathbf{U}$;
29: **if** $\triangle \mathbf{U} > 0$ **then**
30:     perform the resource free up:
31:     $\overline{R_s^n} \to R_s^n$;
32:     update free resources on network n and n';
33: **else**
34:     break;
35: **end if**

The proposed Distributed Rate Allocation (DRA) algorithm (see Table.I) represents a sketch of the proposed algorithm. In this algorithm, the IntraNet procedure always attempts to increase the system's utility by allocating the resource in the network $n \in \mathbf{N}$ to the best user. If the free resources are not sufficient, the InterNet procedure tries to find a new user that can provide enough resources by allocating parts of its allocated rate through another network $n' \neq n \in \mathbf{N}$. As long as the network resources allow it, the whole procedure is operated until no extra utility improvement can be brought to the overall system.

In order to adapt source rates at the transport layer according to network states reported from the network layer, some cross-layer information exchange is needed. At the network layer, the distributed allocation scheme would require track the

TABLE II

STATISTICS OF MEASURED ABR AND RTT

Network	Parameter	ABR (Mbps)	RTT (ms)
Ethernet	Avg.	30.3	202.0
	Std. Dev.	1.9	4.9
802.11b	Avg.	4.4	224.0
	Std. Dev.	1.3	8.7
802.11g	Avg.	15.8	297.0
	Std. Dev.	4.9	12.8

observations of ABR^n and RTT^n over all available access networks. It also records the intended rate allocation R_s^n advertised by each user, and calculates the values of D_{all} and U_s^n accordingly. At the transport layer, the rate controller at the source advertises its neighbors of its intended rate allocation R_s^n. The network state monitor traversed by the stream then calculates the relevant parameters based on its local cache of ABR^n, RTT^n and RB^n within the same access network. The destination node extracts such information from the packet header and reports back to the sender with acknowledgment packets, so that the rate controller can re-optimize its intended rate R_s^n based on the proposed DRA algorithm, with updated network state information.

IV. SIMULATION RESULTS AND DISCUSSIONS

In this simulation, we allow each user to stream video or audio applications via three access networks (Ethernet, 802.11b and 802.11g) with a maximum allowable total delay $T = 350$ milliseconds. Each network is simulated as a link with varying available bandwidth and delay, according to the traces collected from the actual access networks using the ABR and RTT measurements[2]. Table II summarizes the statistics of the collected ABR ad RTT of each network tracing over 200 GOP (Group Of Pictures) periods. For audio application, we use 4 audio transcoders, namely G.723.1B, iLBC, SPEEX and G.711 with average encoding rates of 6.4, 15.2, 24.6 and 64kbps, respectively. For video application, we use a HD (High-Definition) sequence, which has a spatial resolution of 1280×720 pixels, and a frame rate of $F = 60$fps. Each video stream is encoded using a fast implementation of H.264/AVC codec at various quantization step sizes, with GOP length of 30 and $IBBP...$ structure similar to that often used in MPEG-2 bitstreams.

To demonstrate the effectiveness of our proposed DRA algorithm, we use the Additive-Increase-Multiplicative-Decrease (AIMD)-based rate allocation method which is used by TCP congestion control [13] for comparison. AIMD-based scheme probes the network for available bandwidth and reduces the rate allocation after congestion occurs. In the simulations, each network rate allocation is increased proportion to each average ABR. In addition, congestion over network n is indicated upon detection of some lost packets, or when the observed RTT exceeds a specified threshold, based on the playout deadline of the audio/video stream.

---

[2]Forward and backward trip delays are both simulated as half of the measured RTT
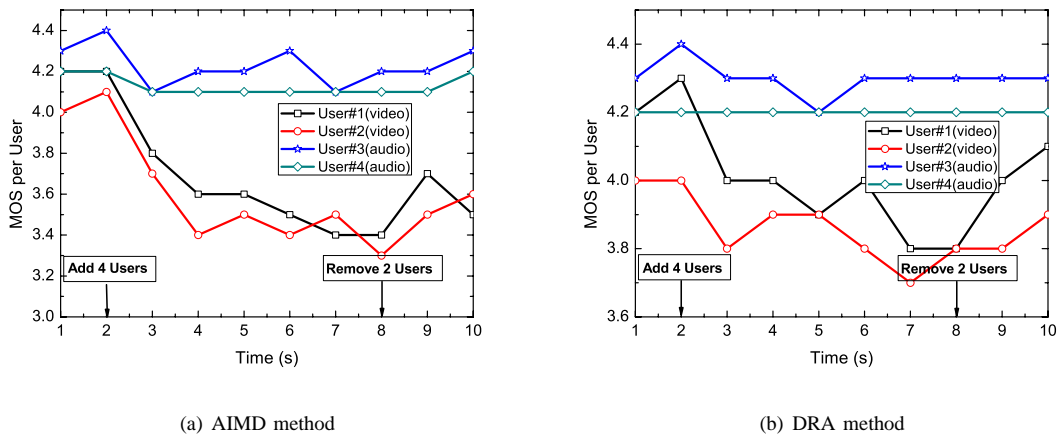
(a) AIMD method

(b) DRA method

Fig. 1. Average performance per user in case users join/leave the network.

We test the proposed rate allocation method in a dynamic environment where users can join or leave the networks. To give a reasonable evaluation for both audio and video applications, we exemplify a concrete quality metric based on MOS (Mean Opinion Score) value. MOS reflects the average user satisfaction on a scale from 1 to 4.5. The minimum value reflects an unacceptable application quality, and the maximum value refers to an excellent QoS. We start with 4 users (two video applications and two audio applications). At time $t = 2s$, we add 4 users (two video applications and two audio applications), and at time $t = 8s$, we randomly remove 2 users (one video application and one audio application). Fig.1(a) and Fig.1(b) present the average MOS for the original 4 users obtained by the AIMD method and the proposed DRA method, respectively. It is observed that proposed DRA method outperforms AIMD method on the aspect of constant performance. That is because our proposed DRA method manages to keep a rather constant application quality for all active users by redistributing parts of the network resources to the new users. In addition, the proposed DRA method further takes advantage of explicit knowledge of the application distortion-rate characteristics, and can achieve more better performance than the AIMD method. It should also be noted from Fig.1 that the network influence on audio application is much less than that on video application. This is because the video application usually requires more network resources than audio application.

V. CONCLUDING REMARKS

In this paper, we develop and evaluate a framework for rate allocation in pervasive computing environment, which contains multiple heterogeneous networks, based on observed each access network's parameters, as well as each application's characteristics. The simulation results demonstrate the effectiveness of our proposed rate allocation scheme for multimedia applications in heterogeneous networks.

REFERENCES

[1] K. Ren, W. Lou, R. H. Deng, and K. Kim, "A Novel Privacy Preserving Authentication and Access Control Scheme in Pervasive Computing Environments",IEEE Transactions on Vehicular Technology, vol. 55, no. 4, pp.1373-1384, July 2006

[2] A.H. Zahran , B. Liang, "A Generic Framework for Mobility Modeling and Performance Analysis in Next-Generation Heterogeneous Wireless Networks", IEEE Communications Magazine, vol. 45, no. 9, pp. 92-99, 2007.

[3] M. Handley, S. Floyd, J. Pahdye, and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 3448, Jan. 2003.

[4] J. Yan, K. Katrinis, M. May, B. Plattner, "Media- and TCP-friendly congestion control for scalable video streams", IEEE Trans. on Multimedia, vol. 8, no. 2, pp. 196-206, 2006.

[5] M. Chen and A. Zakhor, "Rate control for streaming video over wireless", in Proc. INFOCOM'04, vol. 2, pp. 1181-1190, Mar. 2004.

[6] T. Alpcan and T. Basar, "A utility-based congestion control scheme for Internet-style networks with delay", IEEE Trans. on Networking, vol. 13, no. 6, pp. 1261-1274, 2005.

[7] S. Shakkottai, E. Altman, and A. Kumar, "The case for non-cooperative multihoming of users to access points in IEEE 802.11 WLANs", in Proc. IEEE INFOCOM06, Barcelona, Spain, pp. 1-12, Apr. 2006.

[8] D. Jurca and P. Frossard, "Media-specific rate allocation in heterogeneous wireless networks", in Proc. 15th International Packet Video Workshop, (PV06), Hangzhou, China, vol. 7, pp. 713-726, May 2006.

[9] X. Zhu, E. Setton, and B. Girod, "Congestion-distortion optimized video transmission over ad hoc networks", Signal Processing: Image Communication, vol. 20, no. 8, pp. 773-783, Sep. 2005.

[10] S.C. Draper, M.D. Trott, G.W. Wornell, "A universal approach to queuing with distortion control", IEEE Trans. on Automatic Control, vol.50, no.4, pp. 532-537, 2005.

[11] F. Kelly and T.Voice, "Stability of End-to-End Algorithm for Joint Routing and Rate Control", ACM SIGCOMM Computer Communications Review, vol.32, no.2, pp.5-12, 2005.

[12] G.B. Dantzig, "Linear Programming and Extensions", Princeton University Press, Princeton, NJ, 1963.

[13] E. Altman, K. Avrachenkov, et al., "Performance analysis of AIMD mechanisms over a multi-state Markovian path", Computer Networks, vol.47, no.3, pp.307-326, 2005.