This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2008 proceedings.

Multi-Source Streaming in Next Generation Mobile Communication Systems

Tarik Taleb*, Tomoyuki Nakamura, and Kazuo Hashimoto
Graduate School of Information Sciences, Tohoku University
*talebtarik@ieee.org

***Abstract*— Despite recent advances in networking technologies, video streaming applications still suffer from limited bandwidth and highly varying network conditions. As a solution, this paper proposes a novel multi-source streaming strategy specifically tailored for next generation mobile networks to deliver multimedia services to mobile users. The proposed solution consists of a set of mechanisms that aim at ensuring seamless, continuous, and smooth playback of video for users, guaranteeing an efficient and fair utilization of network resources while meeting the playback and buffer constraints of the clients, preventing redundant transmissions and minimizing packet reordering. A set of computer simulations is conducted to evaluate the performance of the proposed strategy and encouraging results are obtained.**

I. Introduction

For multicast-non-based streaming, four approaches can be envisioned, namely single-source single-path streaming, single-source multi-path streaming [1], multi-source single-path streaming [2], and multi-source multi-path streaming. The former represents a typical streaming approach whereby a single server provides streaming contents to its subscribers via single connections. To speed up the start up delay and ensure backup links in case of congestion, a server may allocate a number of connections via different paths to users (second approach).

The system capacity can be increased by reducing the effective request arrival rate to a single server via the deployment of a number of servers with replicated data. This forms the basic idea behind the multi-source single-path (or multi-path) streaming concept. The multi-source concept ensures service continuity, minimizes the effective loss rate, enhances the system's reliability, and provides users with the flexibility of choosing the most appropriate servers, a fact that may remarkably shorten the start-up delay. Additionally, as users will be receiving data from nearby servers, this can clearly achieve further reduction in the backbone bandwidth requirement.

In this paper, we design a multi-source single-path streaming approach for next generation mobile communication systems. The proposed architecture aims at alleviating network congestion via an efficient utilization of network resources via, in turn, an efficient rate allocation mechanism. It also prevents redundant transmissions from the servers via an efficient packet partitioning scheme. The proposed scheme also ensures fast data playback for users via appropriate selection of servers. To cope with the energy constraints of mobile terminals, most operations of the proposed approach are handled at newly-defined entities along the communication path. The performance of the proposed approach is evaluated and is compared to that of a recent multi-source streaming approache proposed in [2]. Interesting results are obtained.

The remainder of this paper is structured as follows. Section II highlights some research work pertaining to multi-path and multi-source streaming. Section III describes the proposed streaming approach. Section IV discusses the simulation results. The paper concludes in Section V.

II. Related Work

As discussed earlier, different streaming approaches can be envisioned. The traditional concept consists in the streaming of video contents from a single server to a client via a single route. To optimize the use of network resources and to ensure high reliability, some researches have investigated the use of multiple independent routes for communication [3], [4]. The work in [1] develops some analytical models for multi-path live streaming via TCP. It considers the case of static multi-path streaming where the bandwidth of each path is fixed, and the case of dynamic multi-path streaming where the transmission capacity of each path is computed based on the condition of receiving buffer. In both cases, the work demonstrates the superiority of multi-path TCP streaming to its single-path counterpart. In [5], a server-based packet scheduling mechanism is proposed for streaming over multiple paths. The scheduling algorithm aims at reducing the playback delay experienced by the client.

The work in [2] proposes a receiver-driven protocol for simultaneous video streaming from multiple senders with forward error correction (FEC) for packet-switched networks. The approach employs a rate allocation algorithm that determines the sending rate for each server and a packet partitioning algorithm that fragments the video stream into several sub-streams, each delivered by different server via different path. Whilst the approach considered the case of wired clients, its use in mobile communication systems is not practical due to a number of reasons related to the operations incorporated in the approach. Firstly, the approach assumes constant Round Trip Time (RTT) between the endpoints. This assumption does not hold in the case of mobile communication systems as users are on move. Secondly, the approach is receiver-driven, in other words, most operations of the approach are initiated by the receiver side. This incurs some computation load at the receiver, the exchange of a number of signaling messages with the servers and their processing load. The receiver also monitors the dynamics of each route to each server in order to reallocate the data transmission rates among servers. Applying such approach to mobile users will definitely drain up the

978-1-4244-2075-9/08/$25.00 ©2008 IEEE

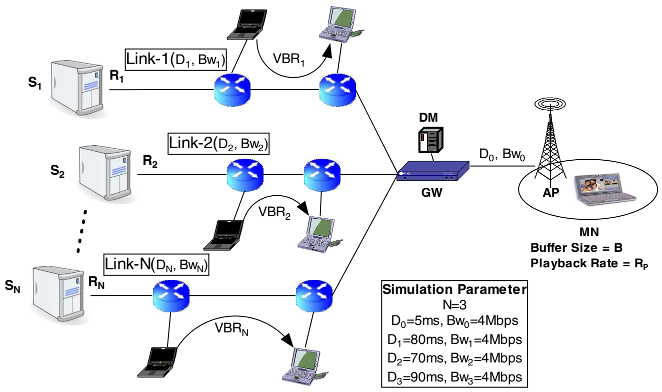

Fig. 1. The envisioned architecture for multi-source video streaming to mobile users.

scarce energy of mobile terminals. Furthermore, a receiver-driven approach usually makes selfish decisions as receivers do not take into account the efficiency of the overall network resources and fairness in their usage. Another issue pertains to the number of signaling messages. Indeed, by exchanging signaling messages between each client and servers, a storm of signaling messages will be generated when the number of subscribers increases. This shall affect the scalability of servers and waste the network resources

III. Proposed Scheme

A. Architecture Description

The envisioned network consists of a number of access points, forming a wireless domain administrated by a domain manager (DM), interconnected to a number of streaming servers (with replicated video contents) via a backbone network (Fig. 1). In our proposed system, for each mobile node the domain manager selects a number N of appropriate servers (from among M available servers). Similar in spirit to many previous research works [2], [5], this work assumes that the domain manager can predict accurately the bandwidth and delay of each path to each of the M servers. The monitoring of the bandwidth and delay of each path is performed periodically every δ time. Throughout this paper, taking into account all servers connected to the studied domain manager, δ is set to a multiple of the propagation delay to the farthest server (i.e., $\delta = \alpha \cdot max_{i=1}^{M} D_i$ where α is an integer).

Using estimates of the link bandwidth availability, the delay D_i and the bandwidth Bw_i of a path i can be computed as the aggregate delays of all links forming the path and the capacity of the link that has the minimum bandwidth along the path, respectively. In Fig. 1, D_0 and Bw_0 denote the delay and the bandwidth of the link from the gateway to the mobile node, respectively. The client side is modeled by a buffer length and a playback rate denoted as B and R_p, respectively.

B. Server Selection

Using a cross layer design similar to that proposed by the authors in [6], the application layer of a mobile node can refer to a set of tools to sort out the access points which the mobile node is most likely going to be connected to during the streaming service. The mobile node then informs the domain manager of this list of access points.

Using the mobility pattern of the mobile node, the domain manager selects the most appropriate servers for the mobile user in a way that guarantees that the video streaming to the mobile user traverses the minimum average number of hops during the entire service time. This would prevent the delay variations and jitter, alleviate network congestion, and ensure seamless and prompt service to the user.

C. Rate Computation to Ensure Seamless Play and Efficient Use of Network Resources

Let R_i denote the streaming rate of server S_i. First, to ensure that the aggregate streaming rate meets the terminal's playback rate, we have the following condition.

$$\sum_{i=1}^{N} R_i = R_p \tag{1}$$

Intuitively, the streaming rates should not exceed the available bandwidth on each path.

$$R_i \leq Bw_i \qquad 1 \leq i \leq N \tag{2}$$

To meet the stringent buffer constraint of the mobile terminal and to ensure that packets are not dropped due to buffer overflow, the following bandwidth delay product should not exceed the buffer size.

$$\max_{i=1}^{N}(D_i) \cdot \sum_{i=1}^{N} R_i \leq B \tag{3}$$

To guarantee an efficient and fair utilization of the network resources, the streaming rate of a given server S_j should be proportional to the ratio of the available bandwidth of its respective path to the aggregate bandwidth of all considered paths. This condition can be formulated as follows.

$$\frac{R_1}{Bw_1} = \frac{R_2}{Bw_2} = \cdots = \frac{R_N}{Bw_N} = \frac{R_p}{\sum_{i=1}^{N} Bw_i} \tag{4}$$

From the above equations, the streaming rate R_i of server S_i can be computed as follows.

$$R_i = min\left(\frac{Bw_i \cdot R_p}{\sum_{k=1}^{N} Bw_k}, \frac{Bw_i \cdot B}{max_{i=1}^{N}(D_i) \cdot \sum_{k=1}^{N} Bw_k}\right) \tag{5}$$

The domain manager constantly monitors variations of network traffic over each path to each available server. Upon a noticeable change in traffic dynamics, the domain manager recomputes the streaming rates for each server and notifies them via identical control messages. The control messages contain information on the session (mobile node's Home Address (HoA), its playback rate, and the streaming video title), servers IDs, their respective streaming rates, and the estimated values of the delays and bandwidths of their respective paths. While the monitoring of network traffic is performed every δ time, the control messages are sent only when the domain manager decides that the streaming rates should be readjusted.

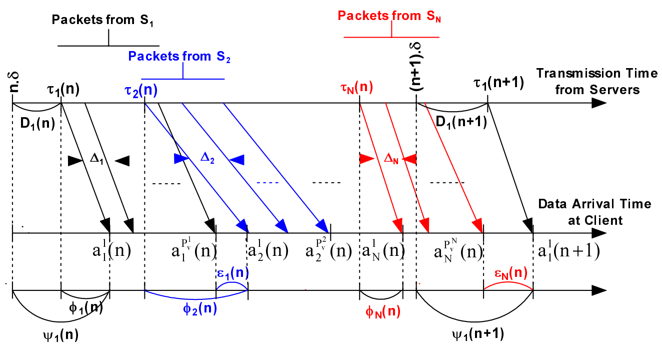

Fig. 2. Packet delivery during a single monitoring interval time.

D. Video Fragmentation to Prevent Packet Redundancy

Upon receiving the control packets, each sender runs the same video fragmentation algorithm to determine the fragment of packets that it will be sending during the current monitoring time interval. Knowing the video playback rate R_p of the video, the average packet size L_p, and the universal monitoring interval δ, the number of video packets P_v that should be delivered during the monitoring time interval can be expressed as follows.

$$P_v = \frac{\delta \cdot R_p}{L_p} \tag{6}$$

Using the computed streaming rate, and in order to ensure fairness in data transmission among servers, a server S_i will be responsible for transmitting P_v^i video packets out of the P_v portion of video data as follows.

$$P_v^i = P_v \cdot \frac{R_i}{\sum_{j=1}^{N} R_j} = P_v \cdot \frac{R_i}{R_p} = R_i \cdot \frac{\delta}{L_p} \tag{7}$$

E. Video Packet Scheduling to Minimize Packet Reordering and Ensure Smooth Playback

While in [5] a packet video selection and scheduling mechanism is proposed based on the recently proposed Earliest Delivery Path First (EDPF) scheme [7], our proposed scheduling mechanism is based on an enhanced version of the EDPF scheme proposed by the authors in [8] that significantly outperforms the original EDPF scheme. In the enhanced EDPF algorithm, a fixed slot-time based policy enforcement strategy is added to enable a more accurate estimation of the delivery time of packets through each path. Our enhanced version of EDPF is dubbed Time-Slotted EDPF (TS-EDPF) and more details on its working are available at [8].

To best explain our proposed scheduling mechanism, let's consider the monitoring time interval $[n \cdot \delta, (n+1) \cdot \delta]$. Throughout this paper, for any parameter f, $f(n)$ denotes the value of f during the monitoring interval $[n \cdot \delta, (n+1) \cdot \delta]$. As discussed earlier, P_v packets should be transmitted during this time interval. Among these packets, each server S_i should deliver P_v^i packets. Let Δ_i denote the time between two consecutive packet transmissions made by server S_i. Let $\tau_i(n)$ denote the time at which server S_i transmits its first packet during the time interval $[n \cdot \delta, (n+1) \cdot \delta]$.

Using the analysis of the authors' previous work in [8], a packet (with size L_p) transmitted by server S_i will take $\phi_i(n)$ time till it reaches the destination.

$$\phi_i(n) = D_i(n) + \frac{L_p}{Bw_i(n)} + D_0(n) + \frac{L_p}{Bw_0(n)} \tag{8}$$

Taking into account the transmission time of control packets from the domain manager to the servers (equal to $D_i(n)$) and in order to speed up the data playback, the server with $min_{i=1}^{N}(D_i(n)+\phi_i(n))$ should initiate the video packet transmission. Without any loss of generality, we order the servers according to the value of their respective value $(\psi_i(n) = D_i(n) + \phi_i(n))$ [1].

$$j \leq l \Leftrightarrow \psi_j(n) \leq \psi_l(n) \tag{9}$$

Let $a_i^k(n)$ denote the arrival time of the k^{th} packet transmitted by server S_i during the considered monitoring interval time $(1 \leq k \leq P_v^i)$. $a_i^k(n)$ can be formulated as follows.

$$a_i^k(n) = \tau_i(n) + \phi_i(n) + (k-1) \cdot \Delta_i \tag{10}$$

Fig. 2 portrays the delivery sequence of packets during a single monitoring interval time. From the figure, server S_i transmits P_v^i packets during a period of time equal to $\{(P_v^i - 1) \cdot \Delta_i\}$. To ensure smooth playback of data at the client display, the data inter-transmission time of each server Δ_i should be set as follows.

$$R_p = \frac{P_v^i \cdot L_p}{(P_v^i - 1) \cdot \Delta_i} \Leftrightarrow \Delta_i = \frac{P_v^i \cdot L_p}{(P_v^i - 1) \cdot R_p} \quad 1 \leq i \leq N \tag{11}$$

To ensure prompt startup at the client display and to prevent packet reordering, server S_1 should start delivering its portion of video packets right after the arrival of the control packet from the domain manager $\big($at time $t = n \cdot \delta + D_1(n)\big)$. $\tau_1(n)$ is thus

$$\tau_1(n) = n \cdot \delta + D_1(n) \tag{12}$$

To ensure smooth playback, the first packet from a server S_i $(1 < i \leq N)$ should reach the client right after the arrival of the last packet transmitted by server S_{i-1}. We denote the elapsed time since the arrival of the last packet from server S_{i-1} till the arrival of the first packet from server S_i as $\epsilon_{i-1}(n)$ (Fig. 2). We obtain the following expression.

$$a_i^1(n) = a_{i-1}^{P_v^{i-1}}(n) + \epsilon_{i-1}(n) \quad 1 < i \leq N \tag{13}$$

From Equation 10, we have

$$\tau_i(n) + \phi_i(n) = \tau_{i-1}(n) + \phi_{i-1}(n) + (P_v^{i-1} - 1) \cdot \Delta_{i-1} + \epsilon_{i-1}(n) \quad 1 < i \leq N \tag{14}$$

From Equations 11 and 7, we have

$$(P_v^j - 1) \cdot \Delta_j = \delta \cdot \frac{R_j}{R_p} \tag{15}$$

The transmission start time $\tau_i(n)$ of server S_i can be thus expressed in the following recursive way.

$$\tau_i(n) + \phi_i(n) = \tau_{i-1}(n) + \phi_{i-1}(n) + \delta \cdot \frac{R_{i-1}}{R_p} + \epsilon_{i-1}(n) \quad 1 < i \leq N \tag{16}$$

[1] In case of a number of servers with the same value of $\psi_i(n)$, the domain manager can re-sort them according to the value of their respective $D_i(n)$, $Bw_i(n)$, or R_i.

TABLE I
SIMULATION PARAMETERS

Factor	Parameters and range of value
Buffer size B (pkts)	100
Packet size Lp (kB)	1
Playback rate Rp (pkts/s)	100
Monitoring interval δ (s)	0.3
Video size (pkts)	950
Simulation time (s)	10.0

From the requirement that all video packets P_v should be transmitted during the entire monitoring interval, we have

$$a_N^{P_v^N}(n) + \epsilon_N(n) = (n+1)\cdot\delta + \psi_1(n+1) \quad (17)$$

In order to avoid the arrival of packets in monitoring intervals subsequent to $\{(n+2)\cdot\delta\}$ (to prevent too much delayed arrival of packets), the monitoring interval δ should be set as follows.

$$\delta \gg max_{i=1}^{M}(D_i(n)+\phi_i(n)) \Leftrightarrow \alpha \gg \frac{max_{i=1}^{M}(D_i(n)+\phi_i(n))}{max_{i=1}^{M}(D_i(n))} \quad (18)$$

From the observations that ($\phi_i(n) \simeq D_i(n)+D_0(n); D_0(n) \ll D_i(n) \;\; \forall i \in [1, M]$), the parameter α should be set to a value higher than three ($\alpha \gg 3$).

From Equations 10 and 15, we have

$$\begin{aligned} a_N^{P_v^N}(n)+\epsilon_N(n) &= \tau_N(n)+\phi_N(n)+\delta\cdot\frac{R_N}{R_p}+\epsilon_N(n) && (19)\\ &= (n+1)\cdot\delta+\psi_1(n+1) && (20)\end{aligned}$$

Using the recursive Equation 16, we obtain

$$\tau_1(n) + \phi_1(n) + \delta + \sum_{i=1}^{N}\epsilon_i(n) = (n+1)\cdot\delta + \psi_1(n+1) \quad (21)$$

From the setting of $\tau_1(n)$ in Equation 12, we obtain the following equation.

$$\sum_{i=1}^{N}\epsilon_i(n) = \psi_1(n+1) - \psi_1(n) \quad (22)$$

As the value of $\psi_1(n+1)$ is not known during the monitoring interval $[n\cdot\delta, (n+1)\cdot\delta]$, and in order to ensure continuous playback of the video data during the entire time of the streaming service, we slide the time difference $\big(\psi_1(n+1) - \psi_1(n)\big)$ forward to the next monitoring interval $[(n+1)\cdot\delta, (n+2)\cdot\delta]$. We obtain the following equation.

$$\sum_{i=1}^{N}\epsilon_i(n) = \psi_1(n) - \psi_1(n-1) \qquad 1 < n \quad (23)$$

Considering a uniform distribution, $\epsilon_i(n)$ can be computed as follows.

$$\epsilon_i(n)=\frac{\psi_i(n)}{\sum_{j=1}^{N}\psi_j(n)}\cdot\Big\{\psi_1(n)-\psi_1(n-1)\Big\} \quad \begin{cases}1 \le i \le N\\ 1 < n\end{cases} \quad (24)$$

Substituting values of $\epsilon_i(n)$ in Equation 16, the transmission start time of each server can be computed.

IV. PERFORMANCE EVALUATION

The design of the simulation setup relies on Network Simulator (NS)[9]. The considered simulation topology is shown in Fig. 1. The bandwidth and delay of each link are as shown in the figure. Other simulation parameters are summarized in Table. I. To investigate the resiliency of the proposed scheme to traffic dynamics, different Variable Bit Rate (VBR) flows are used as background traffic on each path. The sending rate of each flow is varying throughout the course of the simulation and is randomly chosen every 1s from within the range $[0.1\cdot BW_i, \beta\cdot BW_i]$ $(0.1 < \beta \le 1)$. Simulations were all run for 10s.

As a comparison term, we use the multi-source streaming concept proposed in [2]; referred to as PPA (Packet Partitioning Algorithm) throughout the remainder of this paper. As quantifying parameters, we use the following measures:

- Packet reordering delay defined as the difference between the arrival time of a packet and the arrival time of its preceding one.
- Playback ratio defined as the number of played packets over the number of packets that should have been received and played every 100 ms.
- Client's average queue occupancy during the streaming service time measured every 300ms.
- Average path utilization measured every 300ms and computed as the ratio of the transmitted traffic over the total path bandwidth.

Fig. 3 plots the playback ratio achieved by the mobile node in case of the two schemes and for different values of β; different congestion levels. The figure demonstrates that the proposed concept outperforms the PPA scheme for all considered values of β. The good performance of the proposed scheme in case of high values of β indicates the robustness of the proposed scheme to traffic dynamics (simulated via VBR flows). It also reflects the resiliency of the proposed scheme to slight errors in the assessment of the bandwidth availability and the delay of each path.

Fig. 4 graphs the reordering delay experienced by the first 200 packets. The figure shows that the mobile node experiences frequent reordering delays when the PPA scheme is in use. In case of the proposed scheme, the reordering delay is maintained in the vicinity of zero. The credit of this good performance goes to the packet scheduling mechanism of the proposed streaming strategy. From the figure, we notice that some packets experience a reordering delay equal to 200ms in case of the PPA scheme. Such long reordering delay usually results in the arrival of a number of packets later than their playback time. Such packets are ultimately discarded at the client side, an operation that should degrade the perceived quality of the streaming service. Fig. 5 supports further this observation as it shows that in case of the proposed scheme the client buffer has always data to play whereas in case of the PPA scheme the queue occupancy decreases. In the presence of such underflows, along with the delayed arrival of packets at the client's device, the user will notice ruptures in

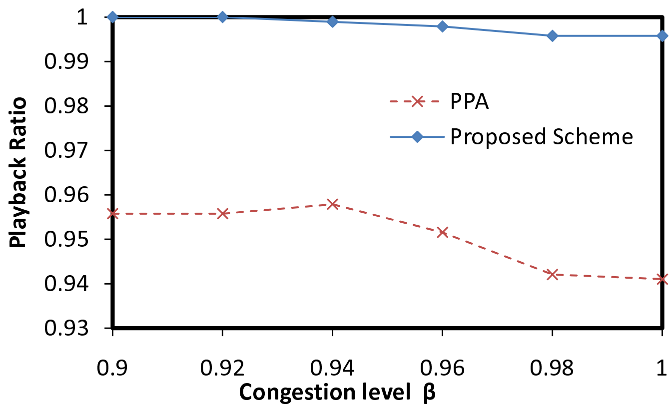

Fig. 3. Playback ratio for different congestion levels in case of the proposed scheme and the PPA scheme.

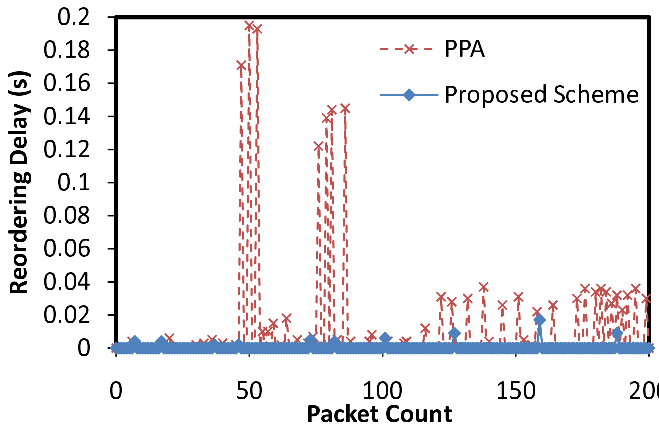

Fig. 4. Reordering delay experienced by the first 200 packets ($\beta = 0.9$).

the streaming service, a fact that impacts the perceived quality.

While the above results demonstrate the better performance of the proposed scheme in terms of reduced packet reordering delay and increased playback ratio, two metrics important from the client's perspectives, the performance of the proposed scheme is also better from the network's perspectives. Indeed, our proposed scheme ensures fair and efficient use of the network resources. It also reduces the number of signaling messages as it centralizes the network monitoring operation at the domain managers, and saves the scarce energy of mobile receivers as it frees them from the burden of the monitoring operation.

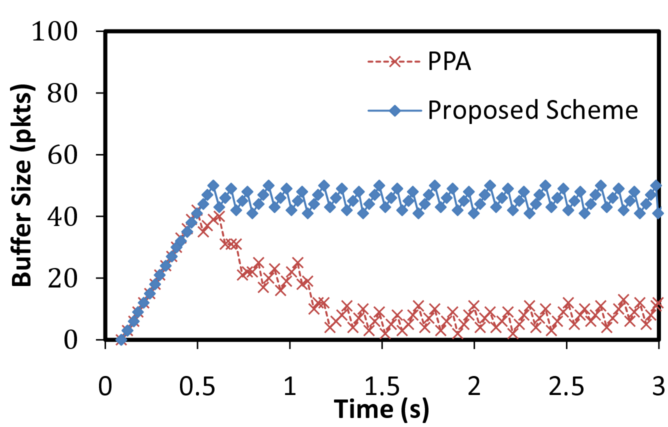

Fig. 5. Variation of the average occupancy of the client's buffer ($\beta = 0.9$).

V. Concluding Remarks

In this paper, we considered the use of multiple servers, distributed over the network, for real time streaming. A set of mechanisms is proposed; each mechanism designed for a particular goal. The goal of the server selection mechanism is to ensure seamless and continuous service for users, minimizing the affect of handoffs during the communication. The goal of the streaming rate computation algorithm is to determine the streaming rate for each server to ensure efficient and fair utilization of the network resources while meeting the playback and buffer constraints of the receiver. The goal of the video fragmentation algorithm is to ensure that packets are sent by one and only one server to avoid duplicate transmissions. The goal of the scheduling algorithm is to ensure fast playback of the video data and to minimize packet reordering. The performance evaluation of our proposed streaming scheme relied on computer simulations and a set of scenarios was considered. The obtained results were encouraging and elucidated the effectiveness of the proposed scheme in achieving its design goals.

Finally, it should be admitted that we did not investigate the interaction of the proposed scheme with the user's mobility. Indeed, the impact of handoff events should be taken into account in both the server selection algorithm and the rate computation mechanism. This forms the focus of the authors' future research work in this particular area of research.

References

[1] B. Wang, W. Wei, Z. Guo, and D. Towsley, "Multipath Live Streaming via TCP: Performance and Benefits", in Proc. IEEE INFOCOM, Anchorage, Alaska, USA, May. 2007.

[2] T. Nguyen and A. Zakhor, "Multiple Sender Distributed Video Streaming," IEEE Transactions on Multimedia, Vol. 6, No. 2, pp. 315-326, Apr. 2004.

[3] L. Golubchik, J. Lui, T. Tung, A. Chow, and W. Lee, "Multi-path continuous media streaming: What are the benefits?," ACM J. Perform. Eval., Vol. 49, No. 1-4, pp. 429-449, Sep. 2002.

[4] J. G. Apostolopoulos and M. D. Trott, "Path diversity for enhanced media streaming," IEEE Commun. Mag., Vol. 42, No. 8, pp. 80-87, Aug. 2004.

[5] D. Jurca and P. Frossard, "Video Packet Selection and Scheduling for Multipath Streaming," IEEE Transactions on Multimedia, Vol. 9, No. 3, pp. 629-641, Apr. 2007.

[6] T. Taleb, K. Kashibuchi, A. Leonardi, S. Palazzo, K. Hashimoto, N. Kato, and Y. Nemoto, "A Cross-Layer Approach for an Efficient Delivery of TCP/RTP-based Multimedia Applications in Heterogeneous Wireless Networks," to appear in IEEE Trans. on Vehicular Technology journal, Nov. 2008.

[7] K. Chebrolu and R. Rao, "Bandwidth Aggregation for Real-Time Applications in Heterogeneous Wireless Networks," IEEE Trans. on Mobile Computing, Vol. 5, No. 4, pp. 388-403, Apr. 2006.

[8] T. Taleb, J.C. Fernandez, K. Hashimoto, N. Kato, and Y. Nemoto, "A Bandwidth Aggregation-aware QoS Negotiation Mechanism for Next-Generation Wireless Networks," in Proc. IEEE Globecom'07, Washington DC, USA, Dec. 2007.

[9] UCB/LBNL/VINT: Network Simulator - ns (version 2). http://www.isi.edu/nsnam/ns/