MS^2: A New Real-Time Multi-Source Mobile-Streaming Architecture

Tarik Taleb, *Senior Member, IEEE*, and Kazuo Hashimoto, *Member, IEEE*

***Abstract*—Video streaming to mobile users is gaining a tremendous momentum within the communities of both industrial and academic researchers. While the networking technologies continue to evolve, video streaming applications are still prone to constrained bandwidth and also to highly varying network dynamics. This paper addresses these issues and attempts to provide a unique solution by proposing a new multi-source streaming strategy specifically tailored for next generation mobile networks for delivering multimedia services to mobile users. The proposed strategy is dubbed MS^2, Multi-Source Mobile Streaming, which comprises a set of mechanisms that aim at ensuring seamless, continuous, and smooth playback of video for roaming 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. Through various computer simulations, the performance of MS^2 is then validated. The simulation results are encouraging and exhibit the effectiveness of the adopted MS^2 architecture.**

***Index Terms*—Mobile video streaming, mobility management, multi-source streaming, packet scheduling.**

I. Introduction

MOBILE-STREAMING services are expected to expedite the recognition and popularity of 3GPP networks, and will lead to an auspicious market for service providers and network operators alike. Different approaches and architectures have been proposed for multimedia streaming [1]. Some of them considered the use of multicast technology [2]. While multicast makes efficient use of network resources, its major limitation pertains to the fact that it is only appropriate for scenarios with a single sender and multiple receivers interested in the same contents.

To facilitate streaming non based on multicast, three approaches have come into use over the years, namely single-source single-path, single-source multi-path [3], and multi-source single-path [4] streaming schemes. As its name suggests, the former represents a typical streaming mechanism in which a single server caters streaming contents to each of its subscribers over a single connection. In such an approach, the use of a single connection per user introduces some start-up delay at the receiver side. The delay and the quality of the service may deteriorate even further during congestion events.

To resort to supplemental links in case of congestion, a server may assign to its subscribers a number of connections over different paths. This shall enhance the perceived quality by increasing the overall connection throughput. However, it makes packets of the same application experience different latencies, resulting in delay jitter and out-of-order packet delivery to the final destination, a fact that impacts largely the perceived Quality of Service (QoS).

On the other hand, the basic concept of multi-source single/multi path streaming technologies aims at increasing the overall system capacity whereby the effective request arrival rate to a single server may be reduced by deploying a number of servers comprising replicated contents. The use of multiple servers for streaming enhances the service continuity. Indeed, by an appropriate splitting of data packets among different servers, the packet drop rate that is experienced on each path would be smaller even during congestion events and this shall ensure service continuity and reliability. Certainly, the likelihood of having all paths experience bursty packet losses all at the same time is usually weak. The multi-source concept also 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. These merits do not come without shortcomings. The prime one pertains to packet reordering. Indeed, in contrast to single-server multi-path streaming techniques (whereby the packet reordering issue may be addressed by effectively computing the optimum streaming rates over each path), multi-server streaming concepts additionally require an efficient packet partitioning mechanism.

In this paper, we propose MS^2, a multi-source single-path streaming approach for next generation mobile communication systems. While the proposed MS^2 scheme may be easily extended to the case of multi-source multi-path streaming, we focus in this paper on multi-source single-path streaming. The MS^2 architecture aims at alleviating network congestion by efficiently utilizing the network resources through an effective rate allocation scheme. In addition, an efficient packet partitioning scheme allows MS^2 to avoid redundant transmissions from the respective servers. By appropriately selecting the servers, the MS^2 scheme also ensures that the users may experience fast data playback. MS^2 introduces a number of newly-defined entities along the communication path which handle most of the required operations. Thus, MS^2 prevents the mobile terminals from consuming valuable energy for computing resource intensive tasks.

Manuscript received June 08, 2010; revised March 08, 2011; accepted May 18, 2011. Date of publication July 18, 2011; date of current version August 24, 2011.

T. Taleb is with the NEC Laboratories Europe, Heidelberg 69115, Germany (e-mail: talebtarik@ieee.org).

K. Hashimoto is with the Tohoku University, Miyagi 980-8579, Japan (e-mail: kh@aiet.ecei.tohoku.ac.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBC.2011.2159529

The performance of MS^2 is evaluated and is also compared to existing related streaming systems. The results, obtained through various computer-based simulations, should be of considerable interest to researchers in both industry and academia. To our best knowledge, although few research work have been conducted to facilitate multi-path streaming in WLANs [5], no previous research has yet taken into account the concept of multi-source streaming in the context of next-generation mobile communication systems.

The remainder of this paper is structured as follows. Section II highlights some research work pertaining to multi-path and multi-source streaming techniques. Section III presents the key components of the envisioned network architecture, describes the proposed streaming approach (i.e., MS^2), and highlights the distinct operations that it incorporates. Section IV portrays the simulation philosophy and discusses the simulation results. Finally, the paper concludes in Section V.

II. Related Work

As per earlier discussion, although various streaming concepts exist in literature, the traditional approach allows a single server to stream video contents to a client over a single route. Some researchers have attempted to utilize multiple independent routes for communication in order to optimize the network resources usage and to also ensure high reliability. Wang *et al.* [3] introduced a number of analytical models for multi-path live streaming over the conventional TCP by taking into account two scenarios. First, they considered a static multi-path streaming approach in which the bandwidth of each segment (path) is fixed *a priori*. Second, they considered the case of dynamic multi-path streaming whereby the transmission capacity of every segment is computed based upon the condition of the receiving buffer. The work conducted by Wang *et al.* clearly demonstrated that multi-path TCP streaming was far superior in both envisioned scenarios, compared to its single-path counterpart. Furthermore, in [6], a server-based packet scheduling algorithm is proposed for streaming over multiple paths in order to reduce the playback delay experienced by the concerned client.

Even though the multi-path scheme leads to increased throughput, it fails to adequately address the startup latency problem. This prompted Content Delivery Network (CDN) operators (e.g., Akamai Inc.) to consider deploying servers at the edges of the target network in order to serve users from the nearest possible servers. Consequently, shorter service latency is achieved; a fact which minimizes the overall congestion in the considered network. This approach does not, however, provide an adequate solution in terms of stability since it requires multiple edge servers for the streaming purpose. In the latter, a client obviously needs to connect to more than one video/streaming servers to receive packets (which are different yet complementary) over different paths with uncorrelated loss processes. To facilitate this type of simultaneous video streaming from multi-sources, Nguyen *et al.* [4] envisaged a receiver-driven protocol with Forward Error Correction (FEC) for packet-switched networks that employs a rate allocation algorithm and a packet partitioning algorithm for determining the sending rate for each server and for fragmenting the video stream into several sub-streams, respectively. Each sub-stream is delivered by a different server along a different path. This solution, although deemed adequate for wired clients, is rather impractical for mobile communications owing to the following reasons. First, this approach heralds the assumption that the Round Trip Time (RTT) between a pair of end-points remains constant. This naive assumption becomes invalid in the case of mobile communication systems as the users/subscribers are always roaming. Second, most of the operations involved in this approach are initiated on the receiver end-terminal which results in various overheads, namely computation loads on the receiver, number of signaling messages exchanged between the servers and their clients, and also processing loads on the server-ends. In addition, in order to be able to reallocate the data transmission rates among multiple servers, the receiver needs to monitor various network dynamics along each route to each server. Hence, implementing such a cumbersome technique on the mobile terminals will drain up their scarce power. This type of receiver-driven scheme is also subject to making rather selfish decisions since the receivers generally tend to overlook in their usage vital factors such as the efficient utilization of the overall network resources as well as fairness issues. Finally, the scalability of this approach is also put under a serious question mark when the number of mobile subscribers increases which leads to an overwhelming volume of signaling messages exchanged between individual clients and their corresponding servers.

In the remainder of this paper, we describe a multi-source streaming approach that is specifically tailored to mobile communication systems. As a remedy to the above-mentioned issues, we substitute the receiver-driven approach by the introduction of newly defined network elements (decision makers) that compute the sending rates of each server on behalf of all clients while maintaining an efficient and fair utilization of network resources.

III. Proposed Architecture: MS^2

Our proposed MS^2 scheme incorporates a set of approaches from different perspectives. This section presents a detailed description of each of these mechanisms. First is a description of the considered network model.

A. Network Architecture

Fig. 1 depicts the envisioned network architecture of MS^2 and its principal components. The envisioned network architecture of MS^2 comprises several wireless network domains, which are inter-linked with a number of streaming servers through a backbone network (e.g., Internet). Wireless Access Points (APs) are deployed in these wireless domains and are linked with the backbone networks via GateWays (GWs). There is an Authentication, Authorization, and Accounting (AAA) server per each wireless domain, which verifies if the mobile subscribers are indeed authorized to access the corresponding wireless network. In addition to the AAA server, each wireless domain also consists of a Domain Manager or a Decision Maker (DM) that carries out the overall service management for that particular domain. We assume that DMs have adequate knowledge about the overall network topology and servers' contents. The major operations of DMs include the following: *i)* the selection of adequate servers for each mobile user moving in their domains, *ii)* estimation of RTT and real time assessment

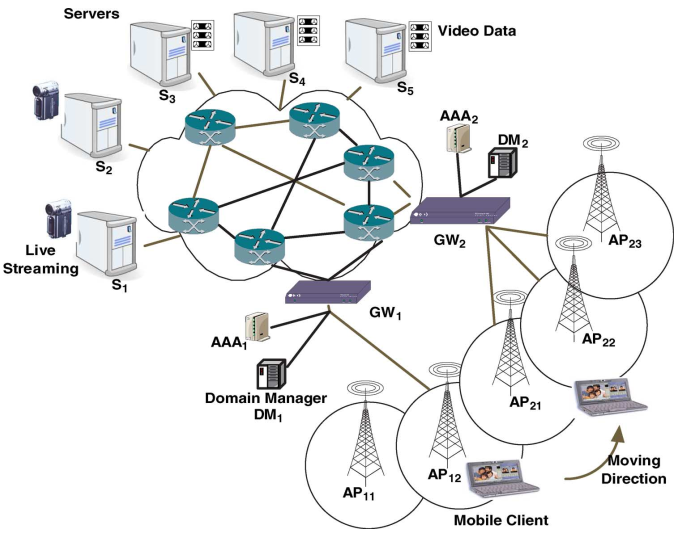

Fig. 1. Envisioned architecture for multi-source video streaming to mobile users.

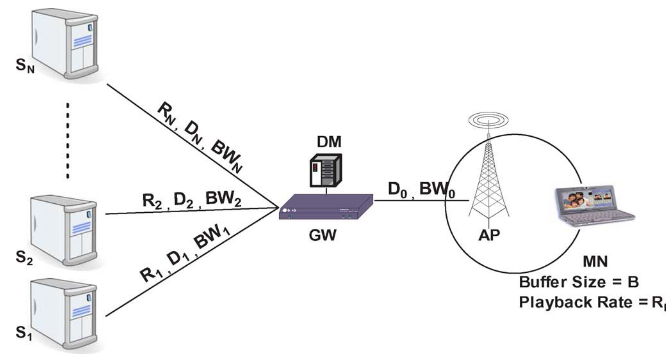

Fig. 2. Simplified model of the network topology.

of bandwidth availability of each path from each server, *iii)* computation of the streaming rate of each server for each mobile user based on the estimated RTT and path bandwidth, and *iv)* notification of these streaming rates to the servers. These functionalities of DMs will be detailed in the next subsections.

In order to assure fast inter-domain handoff management for mobile users, exchange of information on users (i.e., users' profiles) between neighboring DMs is also envisioned. For the sake of simplicity, the paths linking a specific wireless network domain to each streaming server are considered to be independent of uncorrelated packet loss processes[1]. Similar to major CDN incorporations (e.g., Akamai and Bandai networks), a number of servers with replicated video contents are deployed on the wired side of the MS^2 architecture. The placement of the servers is of strategic importance. They are deployed at the edges of the considered backbone network to ensure lower service startup delay and to reduce congestion events.

The "mirror" servers supposedly have the capability of streaming temporarily scattered video packets [7]. To this end, the servers employ the information, which are dispatched via the control packets from DMs, and execute the same program for partitioning the video contents. From the mobile users' perspectives, their terminals are assumed to possess a sufficient memory for buffering the packets to handle two issues, namely packet reordering and the associated jitter [8]. In addition, the mobile terminals are assumed to have the ability to receive the video contents from multiple servers at the same time. For ensuring efficient streaming of video data to users, it is assumed that there is no bandwidth bottleneck at the last hop. This assumption still holds given the recent advances in wireless communication technologies. Indeed, 4G mobile users will be able to download data at rates as high as 100 Mbps till 1 Gbps (i.e., 1 Gbps for terminals with limited mobility). Finally, the video playback rate at the end user is assumed to be known to domain managers and to be coverable by the aggregate bandwidth of all involved servers.

[1]This assumption does not compromise the rudimentary observations that are made regarding the proposed MS^2 approach.

While our studies can be easily extended to the case of multiple domains administrated by different operators, for the sake of simplicity we first consider the case of a single domain administrated by a single operator. The above-described network architecture can be thus modeled as shown in Fig. 2. In MS^2, the DM has, in its disposal, M servers. The DM may select a number of appropriate servers, N, out of these M available ones, for every mobile node while assuring two aspects regarding the paths from the mobile node to the selected servers, namely that these paths *i)* are independent and *ii)* have uncorrelated packet losses. Our envisioned MS^2 assumes that the DM entities can predict (with a certain level of accuracy) the network dynamics (e.g., bandwidth and delay of each path) to each of these M servers [4], [6]. The network conditions along each path are monitored periodically (i.e., in every δ time-interval). Throughout this paper, for determining the value of δ, we have taken into consideration all the servers connected to the concerned DM and set δ to a multiple of the propagation delay to the farthest server (i.e., $\delta = \gamma \cdot \max_{i=1}^{M} D_i$ where γ is an integer). The rationale behind this setting is to make the value of δ universal for all the participating servers linked to the domain in question. It shall also ease the packet scheduling operation as will be explained in Section III-E. The setting of γ will be also explained later.

For bandwidth probing, different methods can be employed. Traditional approaches are based on the use of the Simple Network Management Protocol (SNMP). To avoid the signaling traffic that may be associated with SNMP, we suggest the use of information on packet size and the inter-arrival time of consecutive packets to estimate the bandwidth availability similar to many previous approaches such as TCP Westwood [9] and TCP New Jersey [10]. Indeed, 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. 2, 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

On one hand, without a cross layer design, DMs are compelled to select the servers in an arbitrary and heuristic manner, following any existing server selection mechanism. On the other hand, if we adopt a cross layer design similar to the one that we proposed in an earlier work [11], it will assist DMs to select the most appropriate servers in a systematic way. In this cross layer methodology, the application layer of a mobile node employs specific tools (e.g., history of the concerned user's mobility patterns comprising spatial conceptual map, user's personal information, user's current location, and velocity vector [12]) to predict, as accurately as possible, the AP(s) with which it is most likely to bind itself during the course of streaming. In order to more precisely project the next most plausible AP, *a priori* erudition regarding the topology of the wireless network [13] and also the type of the application [14] may be exploited. The DM is then apprised of the list of the specific APs that the mobile node is most likely going to connect to during the streaming service. It should be emphasized that this particular operation does not contribute to any substantial overhead as far as the mobile node is concerned since it is performed only either at the commencement of the streaming operation or when the node switches to another wireless domain.

The DM, in its selection of the most adequate servers for a particular mobile user, takes into account the mobility patterns of the user and whether the chosen servers will assure the video streaming to the user with a minimum number of server relocations during the course of the entire service time. By so doing, DMs ensure against probable delay jitter and network congestion events, thus servicing the user in a smooth, continuous, and swift fashion. Fig. 1 elucidates a simple example to qualitatively comprehend the servers selection mechanism for the mobile client that moves in the specified direction. In the illustrated scenario, even though the mobile user currently lies in geographical proximity with Servers S_1 and S_2, it should be linked up with Servers S_4 and S_5 to smoothen later its transition to the new wireless domain.

Another important feature that needs to be carefully taken into account in the server selection operation pertains to the number of servers that need to be selected for each mobile terminal. Indeed, as discussed earlier, the use of multiple servers can minimize the effective loss rate and hence enhances the service's reliability, as some servers can provide the same data in case others become unavailable (e.g., if a route between the latter and the receiver becomes congested while streaming). However, the number of servers per mobile terminal shall be set by the service provider in a way that it does not exceed a certain value to control load and complexity at DMs. An appropriate value of this metric shall be a tradeoff between the service reliability and system complexity at DMs.

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

In this section, we attempt to find the most appropriate streaming rate for each server. Our objectives are twofold: to ensure seamless play of video data by meeting the stringent playback delay and buffer constraints at the user side, and to ensure fair and efficient utilization of the entire network resources. We define R_i to be the streaming rate of server S_i, where $(1 \leq i \leq N(\leq M))$. The following condition is then defined in order to ensure that the mobile terminal's play rate, denoted by R_p, is met by the overall streaming rate of all the servers:

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

For practical reasons, the streaming rates should not surpass the accessible bandwidth along each of the considered paths.

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

The next condition aims at *i)* satisfying the limited buffer of the mobile terminal and *ii)* averting packet drops owing to buffer overflow. In this vein, the bandwidth delay product, as stated on the left hand side of Inequality (3), should not surpass the buffer size, B.

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

In order to assure an efficient and fair utilization of the network resources, the streaming rate of a particular server S_j ought to be proportional to the ratio of the bandwidth, which is available along its corresponding path, to the entire bandwidth of all the considered paths. Equation (4) represents this specific condition, which ties the streaming rates of the selected servers with the bandwidths of their respective paths:

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

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

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

In the MS^2 framework, one of the roles of the DM is to consistently monitor disparity in traffic dynamics along all the available paths up to each streaming server. Once the DM detects any conspicuous change in the network traffic conditions, it initiates the re-computation of the streaming rates for every server. The servers are then informed by the DM regarding these updated values of the streaming rates via identical control messages comprising session information and servers information. The former consists of the title of the streaming video along with the Home Address (HoA) and playback rate of the mobile terminal while the latter include the servers' IDs, streaming rates, and estimated values of the delays and bandwidths of the corresponding paths of all the available servers. A sample control message format is depicted in Fig. 3. It should be noted that the transmission of control messages do not contribute to flooding in the network since instead of being issued periodically, they are dispatched to the respective servers only when the DM deems it necessary to recompute their streaming rates. In fact, this further diminishes the overheads involved with the

Mobile Node Home Address		Video Title	Playback Rate
Servers ID	Streaming Rate	Path Bandwidth	Path Delay

Fig. 3. Typical format of the control message.

transmission of signaling messages. In the event that the control message to a particular server is dropped/lost, owing to congestion in the network, the server obviously fails to send feedback (in terms of an acknowledgement) to DM. To circumvent this issue, DM sets a timeout threshold as $(2 \cdot \max_{i=1}^{N} D_i)$ when it issues the control message. The control message is reissued to the server if it fails to acknowledge the DM within this threshold period.

D. Video Fragmentation to Prevent Packet Redundancy

In this section, we delineate the MS^2 video fragmentation algorithm, role of which is to make sure that each packet is transmitted by only one of the streaming servers, thereby avoiding redundant transmission of duplicate packets to ensure efficient utilization of both network's and servers' resources. When the servers receive the aforementioned control packets from the DM, they execute the same video fragmentation algorithm for finding out which packet fragments they are supposed to send over the current monitoring time interval, δ. The number of video packets, P_v, which ought to be delivered during δ can be expressed as follows

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

where R_p and L_p denote the video playback rate and the average packet size, respectively.

The manner in which the servers transmit their respective data to the same mobile client is crucial. As a result, out of the above calculated P_v video packets, a server S_i is responsible to transmit P_v^i packets by using the most recently computed streaming rate as follows.

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

At this point, the servers know only which video packets to deliver, without any knowledge regarding "when to transmit which packet". This absence of scheduling knowledge may lead to a high number of unordered packets as well as high startup latency at the mobile terminal. Therefore, we realize the crucial need to envision a new packet scheduling algorithm, which is delineated in the remainder of this section.

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

In the remainder of this section, we elucidate the packet scheduling mechanism at servers to address the previously stated issues, namely to minimize the packet reordering and the playback delay at the end-user. In addition, the scheduling scheme also ensures that all servers share the video streaming responsibility in a fair manner. In contrast with the packet video selection and scheduling scheme introduced by Jurca *et al.* [6], which is based on the recently proposed Earliest Delivery Path First (EDPF) approach [15], our envisioned scheduling mechanism is based upon an enhanced edition of the EDPF scheme [16], which substantially outperforms the original version. We call our enhanced EDPF algorithm Time-Slotted EDPF (TS-EDPF) since it incorporates a fixed time-slot based policy enforcement strategy for more precisely estimating the delivery time of packets through each path. Further details of TS-EDPF are available at [16].

In order to best explain our proposed TS-EDPF-oriented scheduling scheme, at first we consider the monitoring time interval, defined by $[n \cdot \delta, (n+1) \cdot \delta]$. In the remainder of this paper, for any parameter f, $f(n)$ indicates the value of f over the monitoring interval $[n \cdot \delta, (n+1) \cdot \delta]$. According to the analysis presented earlier, P_v video packets should be transmitted to the mobile terminal in course of this time interval. The sequence numbers of these packets are denoted by $(P_1, P_2, \ldots, P_{P_v})$. A server S_i is supposed to deliver P_v^i out of these P_v packets in a steady stream. A server may not transmit its respective packets in a single burst for two reasons. First, a bursty transmission by a server may contribute to substantial packet drops. Second, this will affect the continuity of the playback of the video at the end-user.

Let Δ_i denote the time between two consecutive packet transmissions initiated by Server S_i. Let $\tau_i(n)$ indicate the time instant when Server S_i transmits its first packet during the time interval $[n \cdot \delta, (n+1) \cdot \delta]$. We are then interested in finding out the time taken by a packet (of size L_p) to reach its corresponding destination after it was transmitted by Server S_i. This time is denoted by $\phi_i(n)$. As per the analysis in our earlier work [16], $\phi_i(n)$ is evaluated as follows.

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

The next task is to determine which server ought to initiate the video packet transmission. To this end, we need to consider the time it takes for the control packet issued by DM to arrive at a server S_i. Ignoring queuing delays, this time can be roughly estimated as equal to $D_i(n)$. Keeping in mind the important objective of speeding up the data playback at the end-user as much as possible, the server with $\min_{i=1}^{N}(D_i(n) + \phi_i(n))$ is the most suitable candidate to commence the transmission of video packets. Without any loss of generality, throughout this paper the servers are ordered as per their respective values of: $(\psi_i(n) = D_i(n) + \phi_i(n))$[2].

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

Let $a_i^k(n)$ denote the arrival time (at the end-terminal) of the kth packet transmitted by Server S_i during the considered monitoring interval time $(1 \le k \le P_v^i)$. $a_i^k(n)$ can be formulated as follows.

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

[2]In the event that a number of servers have the same value of $\psi_i(n)$, DM can re-arrange them according to the values of their respective $D_i(n)$, $Bw_i(n)$, and/or R_i.

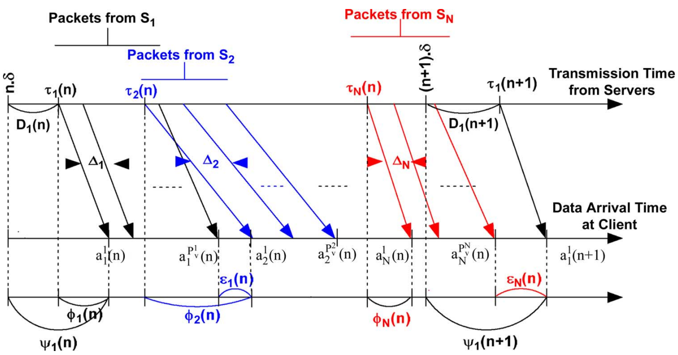

Fig. 4. Packet delivery during a single monitoring interval time.

In the remainder of this section, we aim at determining the values of Δ_i and $\tau_i(n)$ for every server while ensuring smooth playback at the client display.

The sequence in which the packets are delivered is depicted in Fig. 4, covering only an individual monitoring time interval. As shown in this figure, Server S_i dispatches P_v^i packets during a time period of $\{(P_v^i - 1) \cdot \Delta_i\}$. In order to make sure that the client displays are able to play back the data in a smooth fashion, the data inter-transmission time of each server Δ_i is to be set according to Equation (11).

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

It is worth stressing out that the Δ_i values are uniform and not vastly skewed. The reason behind this choice is the fact that extremely skewed inter-transmission times may cause some of the selected servers to transmit a huge number of packets in a short burst which will lead to buffer overflow at the mobile terminal. Since our design computes the streaming rate of every server while taking into consideration the buffer constraints of the concerned end-user, we avoid such a drastic situation.

In order to start up the video playback at the client's display as promptly as possible (and also to minimize packet reordering events), Server S_1 ought to begin delivering its share of video packets immediately followed by the arrival of the control packet from the DM (at time $t = n \cdot \delta + D_1(n)$). The term $\tau_1(n)$ can, therefore, be expressed as:

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

For smooth playback, the first packet transmitted by a particular server S_i $(1 < i \leq N)$ should arrive at the client, immediately following the arrival of the last packet obtained from Server S_{i-1}. As illustrated in Fig. 4, this inter-arrival time is denoted as $\epsilon_{i-1}(n)$. Thus, we obtain the following expression.

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

From Equation (10), we obtain

$$\begin{aligned}\tau_i(n) + \phi_i(n) = \tau_{i-1}(n) + \phi_{i-1}(n) + \left(P_v^{i-1} - 1\right) \\ \cdot \Delta_{i-1} + \epsilon_{i-1}(n) \qquad 1 < i \leq N \end{aligned}\tag{14}$$

From Equations (7) and (11), we derive

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

The time to commence the transmission (denoted by $\tau_i(n)$) of server S_i can be, thus, expressed in the following recursive manner.

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

It should be noted that once we find values for ϵ_i, the transmission start times of all servers can be easily computed from the above equation. For this purpose, we consider the requirement that all video packets P_v should be transmitted during the entire monitoring interval. Taking into account the time a control packet takes to reach the farthest server and the time till a packet from the same server (server S_N in Fig. 4) reaches the mobile user, this condition can be expressed as follows.

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

Upon determining the values of ϵ_i, the task of calculating the transmission start times of all servers by using Equation (16) is straightforward. 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.

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

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 γ should be set to a value higher than three $(\gamma \gg 3)$. From Equations (10) and (15), we derive

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

By employing the recursive Equation (16), we acquire the following expression.

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

Furthermore, from the setting of $\tau_1(n)$ in Equation (12), we attain

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

Due to the fact that the value of $\psi_1(n+1)$ is not known during the monitoring time interval $[n \cdot \delta, (n+1) \cdot \delta]$, and also to ensure continuous playback of the video data during the entire course of streaming, we slide the time difference $(\psi_1(n+1) - \psi_1(n))$

forward to the next monitoring interval $[(n+1)\cdot\delta, (n+2)\cdot\delta]$. As a consequence, we obtain the following equation.

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

By taking into account a uniform distribution, $\epsilon_i(n)$ can be calculated in the following manner.

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

By substituting values of $\epsilon_i(n)$ in Equation (16), it is then possible to calculate the transmission start time of each server.

It is worth emphasizing that even in the event of slightly inaccurate estimations of the delays and bandwidths associated with the paths, each server should be able to reach the same scheduling decision. This is due to the fact that all servers employ the same parameters, which are provided by identical control messages issued by the DM. Furthermore, the DM can, of course, make the same decision and assist the system to recover from packet drops scenarios, where the mobile client/terminal has no knowledge on which server to get in touch with to re-request for the missing packets. Therefore, the client notifies the DM regarding the missing packets. DM, in turn, first uses the aforementioned scheduling scheme to determine the server which was in charge of delivering the missing packets in the first place, and then requests that the corresponding server to retransmit the same.

F. Service Continuity Support

To the best of our knowledge, no previous research work has considered the multi-source streaming concept in the context of mobile communications systems. As a consequence, none of the handoff management mechanisms available in literature to-date can be tailored to support multi-source streaming in mobile systems [17]. Therefore, in the rest of this section, we turn to the problem of mobile users performing handoff operations while moving within the envisioned network topology, and attempt to solve the problem by introducing an adequate handoff management scheme. Before providing further insights into the mobility management method, we first delineate a number of operations to assist DMs: *i)* to reduce the frequency of server relocations and *ii)* to predict the handoff occurrence time (i.e., particularly inter-domain handoffs as will be explained later).

1) Preliminaries: When a mobile user enters the service area of a DM, the DM can adopt a context-aware server selection strategy (as described in Section III-B) to mitigate the influence of server relocations on the streaming service continuity. Furthermore, the system needs to precisely assess the network conditions (e.g., via intelligent sensing of MAC QoS conditions, monitoring of the PHY-layer, and so forth [18]) as well as the user's preferences (e.g., by referring to users' profiles and attributing adequate intelligence to DMs) to make meticulous handoff decision and accurately estimate the handoff occurrence time. We broadly envision two types of mobile end-hosts, namely technology-embedded terminals (e.g., GPS-grafted vehicles and smart phones capable of tracking their current locations and velocity bearings) and other terminals lacking such

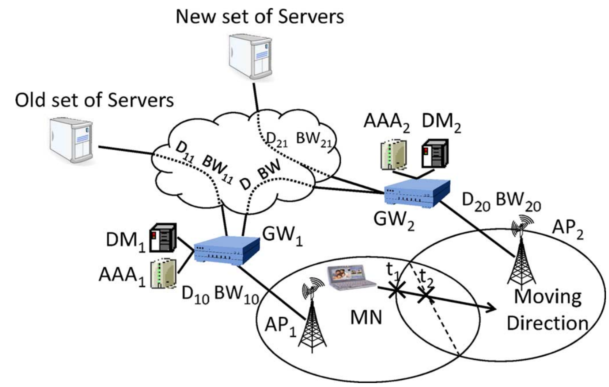

Fig. 5. Schematic model of the handoff operation.

technological features (e.g., simple cellular phones). In case of the technology-embedded devices, even if the user is driving a vehicle, it is possible to transmit, via short-range technologies such as Bluetooth, information pertaining to the speed, current location, and moving direction of the vehicle to the user's mobile phone. With *a priori* knowledge on the layout of their service areas, DMs can predict the handoff occurrence time for mobile terminals capable of retrieving their current locations and of computing their moving speeds. In case of terminals without such attributes (i.e., a simple mobile user walking while receiving a video stream on his/her mobile phone), estimation of handoff occurrence time may be made by adding context-awareness to the respective DMs. To this end, DMs incrementally build a statistical profile of users' behavior over time at particular locations, such as near the cross roads or at the zone overlapped by two or more collocated APs. For example, when a user enters the overlapping coverage areas of two neighboring APs at a specific location (e.g., bus/train stations), the handoff occurrence time during rush hours is more likely to differ from that in late night when the streets are not as busy with traffic. Through profiling such spatial and temporal awareness pertaining to the behavior of users, DMs can approximately predict when and where a user may perform handoff.

2) DM-Driven Handoff Management Scheme: We broadly categorize the possible handoff scenarios into two categories, namely intra-domain and inter-domain handoff events. The former indicates a handoff between two APs within a specific wireless domain. The latter, on the other hand, refers to a handoff operation between two adjacent APs, belonging to different wireless domains.

Various mobility management techniques that are available in the recent literature ([18]–[20]) are able to deal with intra-domain handoffs. However, for handling inter-domain handoff, we resort to the model depicted in Fig. 5, where the considered Mobile Node (MN) enters the coverage area of the new AP at time t_1. The new AP's coverage area implies the overlapping zone between the two APs in Fig. 5. The actual handoff occurrence time, t_2, may depend on both the MAC and physical layers [18]. In our model, t_2 is simply set to the time instant at which the MN crosses the middle line of the overlapping area and switches to AP_2.

The sequence of signaling packets that are exchanged during the handoff operation is illustrated in Fig. 6. Upon the entrance of a MN into the overlapping zone between the coverage areas

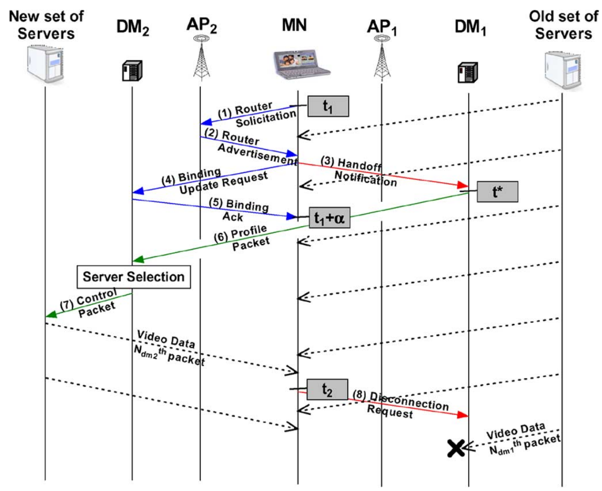

Fig. 6. Sequence of packets exchanged during the handoff operation.

of two APs, it receives a router advertisement message from the new AP(s). Consequently, MN informs DM_1 through AP_1 that a handoff to AP_2 is imminent. Meanwhile, in order to bind itself with the new AP, MN starts performing handoff registration with AP_2 and this operation takes α time. The value of α largely depends on two factors, namely the underlying mobility management protocol (e.g., Hierarchical MIPv6) and the adopted handoff management protocol. Generally speaking, if α is not lower bounded by the difference $(t_2 - t_1)$, communication disruption may become prevalent in the considered network. Indeed, this difference of t_2 and t_1 is attributed to the surface of the overlapping area and the velocity of MN. In our envisaged network topology, α may be estimated as the Round Trip Time (RTT) from AP_2 to DM_2 by ignoring the delay at the MAC layer (which is in the order of micro seconds, thus negligible). It is also possible to employ the history records of the MN's mobility to obtain the average values of α by using the Exponentially Weighted Moving Average (EWMA) method in order to derive a more precise estimate of α.

Upon receiving the "handoff notification" message regarding a MN, DM_1 dispatches a "profile packet" to DM_2 that consists of the current time denoted by t^*, the session ID (i.e., video title), the home address and playback rate R_p of MN, and IDs of the involving servers. In addition, DM_1 also informs DM_2 of an estimated sequence number N_{dm2} of the packet, which is supposed to arrive at MN by the time the handoff operation is successfully registered, i.e., $(t_1 + \alpha)$. Then N_{dm2} can be estimated as follows, provided that the sequence number of the packet to be forwarded at time t^*, denoted by N^*_{dm1}, is known:

$$N_{dm2} = N^*_{dm1} + \left\lfloor \frac{\alpha \cdot R_p}{L_p} \right\rfloor \tag{25}$$

On the other hand, the actual handoff occurrence time can be roughly determined for MNs, which are coupled with *i) a priori* knowledge regarding the network topology such as coordinates and transmission ranges of APs, and *ii)* GPS and/or similar technologies that can track their geographical locations and velocity vectors. In case of such MNs, the sequence number N_{dm2} may be calculated by using the following expression:

$$N_{dm2} = N^*_{dm1} + \left\lfloor \frac{MAX\{(t^*_2 - t_1); \alpha\} \cdot R_p}{L_p} \right\rfloor \tag{26}$$

where t^*_2 refers to the estimated time at which the handoff takes place. Upon receiving the "profile packet" from DM_1, DM_2 attempts to look for a new pool of adequate servers for streaming, conducts the rate allocation procedure, and finally transmits control packets to the newly selected servers. It should be noted that the new pool of servers, thus formulated, may include all, a few, or none of the old servers. When the new servers receive these control packets, they perform the scheduling operation and commences the video streaming (starting from the packet having the sequence number N_{dm2}) to the new location of MN.

When MN desires to stop the streaming via DM_1, say at a later time t_2, it sends a "disconnection request" to DM_1 through AP_1. This prompts DM_1 to verify the sequence number of the first packet (denoted by N_{dm1}) which ought to be discarded. One of the following three scenarios holds as DM_1 compares between the two sequence numbers N_{dm2} and N_{dm1}:

- $(N_{dm1} - N_{dm2}) < 0$: This scenario leads to $(N_{dm2} - N_{dm1})$ packets to be missed that are, at once, encapsulated and forwarded by DM_1 to DM_2, which transmits the same to MN via AP_2. This procedure is consistent with the store and forward concept of contemporary handoff management schemes such as [19], [20].
- $0 \leq (N_{dm1} - N_{dm2}) \leq \epsilon$: ϵ, the maximum number of duplicate packets which the system is able to cope with, is computed as follows:

$$\epsilon = \frac{(t_2 - t_1 - \alpha + D) \cdot R_p}{L_p} \tag{27}$$

- Equation (27) provides an estimate of the number of packets to be delivered to MN since the beginning of the handoff registration at time $(t_1 + \alpha)$ up to the actual handoff occurrence at time t_2 in addition to the time required for DM_1 to inform DM_2 regarding the handoff event. This implies that in the case of setting ϵ as in Equation (27), even when DM_1 informs DM_2 of the probable number of duplicate packets, it will be too late since the packets would have been already dispatched. As a consequence, in this particular scenario, DM_1 takes no action at all.
- $\epsilon < (N_{dm1} - N_{dm2})$: This case portrays a high level of redundancy since a significant number of duplicate packets will be (or have been already) transmitted. To counter this scenario, DM_1 at once sends a notification regarding the event to DM_2, which eventually forwards only packets having sequence numbers that exceed N_{dm1}.

It is worth noting that a number of issues need to be addressed for practical implementation of the proposed mobility management mechanism. For example, the ability of DMs to read up to the packets' sequence numbers violates the IP-Security (IPSec) semantics, which stresses that packets are to be handled by end-hosts alone, rather than any third-party or intermediate entity. Researchers have, however, felt the need to consider violating such clauses for the sake of having this kind

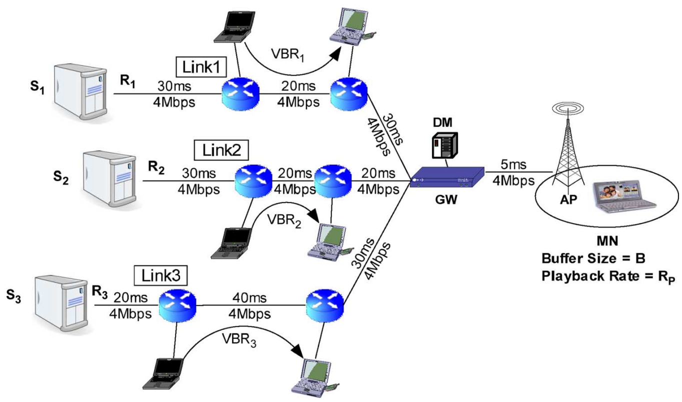

Fig. 7. Simulation environment.

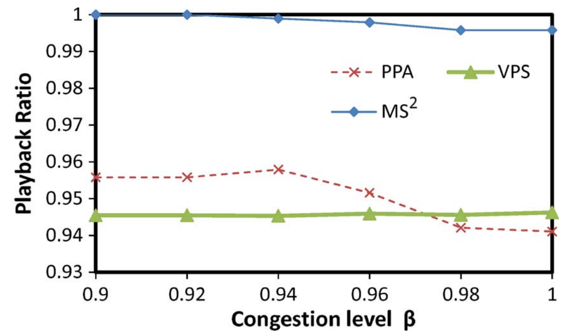

Fig. 8. Playback ratio for different congestion levels in case of the proposed MS^2, PPA, and VPS schemes.

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

of ability in many of the contemporary mobility management schemes [20].

IV. Performance Evaluation

In this section, we direct our focus to the evaluation of our proposed MS^2. We have dedicated our efforts to designing an accurate and realistic simulation environment using Network Simulator (NS) [21].

A. *Multi-Source Mobile Streaming via a Single Access Point*

Fig. 7 depicts the considered simulation topology, which includes three servers connected to an AP via a gateway. The topology consists of three uncorrelated paths from the servers to the gateway traversing a number of routers. The bandwidth and delay of each of these links are also shown in Fig. 7. Wireless links are simulated with packet error rates equal to 10^{-5}. Table. I lists the rest of the simulation parameters. For the sake of simplicity, we first restrict our studies to the case of a single mobile node. Indeed, the simulation of multiple users is useful to investigate the impact of the network congestion on the working of the system. To simulate this influence, we employ different Variable Bit Rate (VBR) flows that serve as background traffic on each path. This background traffic are employed as a means to verify the resiliency of MS^2 to traffic dynamics. The transmission rate of each individual flow is varied during the whole course of each simulation run and is randomly chosen every one second from within the range $[0.1 \cdot BW_i, \beta \cdot BW_i]$ $(0.2 < \beta \leq 1)$.

The performance of MS^2 is compared with two existing schemes, namely the multi-source streaming mechanism proposed in [4] and the video fragmentation algorithm introduced in [6]. In the remainder of the paper, we refer to these two schemes as PPA (Packet Partitioning Algorithm) and VPS, respectively. The following parameters are adopted as quantifiable measures.

- Packet reordering delay is defined as the inter-arrival time difference of two successive packets.
- Playback ratio is referred to as the ratio of the number of played packets over the number of packets, which should have been received and played every 100 ms.
- During the entire course of streaming, the average occupancy of the client's queue is measured every 300 ms.

First, we demonstrate, in Fig. 8, the playback ratio of the considered MN for varying congestion levels (denoted by β) in case of all three considered schemes, i.e., PPA, VPS, and the proposed MS^2. As evident from this figure, in terms of the playback ratio, MS^2 substantially outperforms the other two methods for the considered β values, ranging from 0.9 to one (i.e., heavily congested network). In addition, along with the increase in the level of network congestion, the playback quality of MN remains unaltered and deteriorates in cases of VPS and PPA, respectively. On the other hand, even though MS^2 exhibits some degradation in its playback ratio to some extent for β values higher than 0.96, the overall playback quality is still far superior in MS^2 in contrast with both PPA and VPS. It is worth stressing that as β values approach one for a particular VBR flow, the path associated with that VBR flow is no longer available for streaming. The excellent performance of MS^2 for elevated β values signifies its robustness to traffic dynamics, which are orchestrated by the VBR flows in our simulations. Furthermore, the simulation results presented in Fig. 8 also demonstrate that MS^2 is, indeed, tolerant to slender errors that might have been associated with the evaluation of the delay and bandwidth availability of each path. This resiliency of MS^2 can be attributed to its envisioned design whereby all servers employ identical control messages sent by DM for scheduling the transmission of video data in an effective and cooperative manner.

The next evaluation of MS^2 comprises the reordering delays, which are plotted in Fig. 9 for the first 200 packets. The congestion level is set to 0.9 throughout this particular experiment. This figure demonstrates that under the PPA and VPS schemes, MN experiences reordering delays quite often. MS^2, on the contrary, presents a contrasting performance altogether. The credit for the total absence of packets reordering delay (up to the considered number of packets) in our MS^2 scheme goes to the adopted packet scheduling technique for streaming video data. This result validates that the envisioned scheduling mechanism *i)* efficiently distributes the burden of video packet delivery

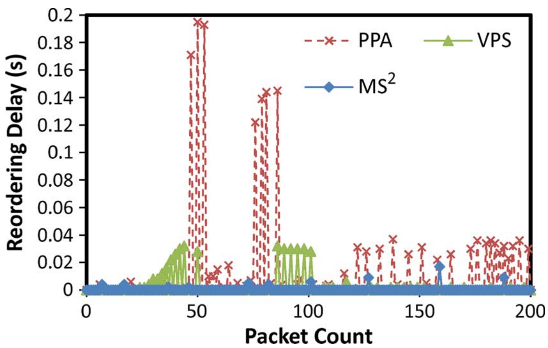

Fig. 9. Reordering delay experienced by the first 200 packets ($\beta = 0.9$).

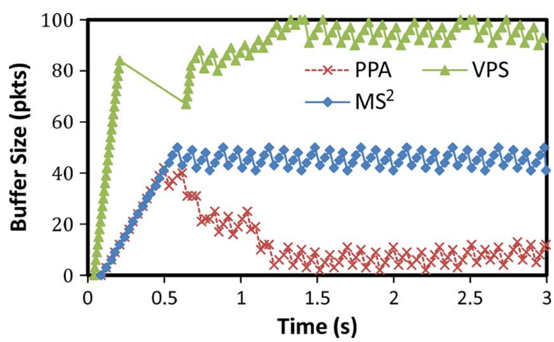

Fig. 10. Variation of the average occupancy of the client's buffer ($\beta = 0.9$).

among the simulated pool of servers and *ii)* ensures that MN receives the packets in the proper order. In particular, the first 50–100 packets in case of PPA experience reordering delays of high orders (exceeding 100ms). As a consequence, a number of packets are bound to reach MN much later than the actual playback time and hence, the MN has no choice but to discard these packets which seriously degrades the quality of perception for the video streaming. Fig. 10 provides additional evidence to this observation by demonstrating that the client buffer, when MS^2 is in use, unlike PPA, always has playable data, i.e., its queue occupancy is consistently high (however, never exceeds the buffer size). In schemes such as PPA, the presence of underflows coupled with the late arrival of packets at the MN will cause the client to notice serious disruption in the streaming service. In case of VPS scheme, however, the buffer reaches its maximum (100 packets) after 1.3s, following which packet drops become unavoidable. This actually justifies the previously demonstrated (i.e., in Fig. 8) rather low playback ratio of the VPS approach.

B. Coping With Users' Mobility

In this section, we attempt to evaluate how the handoff management strategy in MS^2 supports service continuity. For this purpose, the considered network topology is constructed, as shown in Fig. 5, which comprises the coverage areas of two APs, namely AP_1 and AP_2 that are managed by two different domain managers, DM_1 and DM_2, respectively. The distance between the two considered APs and the coverage radius of each AP are set in such a manner that the maximum overlapping distance is equal to $D_{\max}$.

In our simulations, we have considered a population of T_m MNs, which are arbitrarily scattered over the coverage area of AP_1 and are assumed to perform handoff to AP_2 at different time instants. The moving speed of each MN is obtained from a uniform distribution. To account for users in both urban and highway scenarios, the minimum and maximum values of this uniform distribution are set to a slow node moving speed, 18 Km/h, and a high node moving speed, 108 km/h, respectively. After the beginning of each simulation run, all MNs remain stationary for a short while till the video streaming from AP_1 stabilizes. For this particular experiment, no background traffic is simulated to avoid any possible confusion between service disruption owing to handoffs and that due to network congestions.

When a MN performs handoff, several statistics are collected including the time at which the MN enters the overlapping area (denoted by t_1), the actual time when the handoff occurs (t_2), the handoff time (α), the sequence number of the first packet received through AP_2, and its reception time. Seven simulation runs were conducted and the average values are used as results.

Upon a handoff, if all packets can be played in order at the display of the MN, we consider MS^2 to have been able to successfully deal with that particular handoff event. We define the following parameter for demonstrating the efficiency of the system in performing seamless handoffs.

$$\Theta = \frac{N_s}{N_h} \cdot 100 \tag{28}$$

where N_h and N_s refer to the total number of MNs that performed handoff to AP_2 during the course of the simulation and the number of successfully handled handoff operations, respectively.

In the envisioned inter-domain handoff procedure, the source DM informs the target DM of the range of frames, which may be required by the concerned MN. Inaccurate estimation of this range may contribute to duplicate packets' transmission. Therefore, there is a need to evaluate the efficiency of MS^2 in terms of the volume of redundant packets. To this end, we measure the ratio of the number of the no-redundant packets to the total number of dispatched packets averaged over the number of handoffs, N_h. This ratio is referred to as the transmission efficiency of the system, denoted by Λ. For every handoff event k, $\{k = 1, 2 \ldots, N_h\}$, we define two additional parameters: N^k_{total} and $N^k_{\text{duplicate}}$, which we refer to as the total number of packets received by the MN performing the handoff operation k and the total number of redundant packets received during the residual time of the same MN in the overlapping area, respectively. The system efficiency, Λ can then be expressed as follows:

$$\Lambda = \frac{1}{N_h} \cdot \sum_{k=1}^{N_h} \left(1 - \frac{N^k_{\text{duplicate}}}{N^k_{\text{total}}}\right) \cdot 100 \tag{29}$$

Figs. 11 and 12 plot the sequence numbers of data packets received during the handoff period, at two MNs that travel between domains DM_1 and DM_2 at two distinct speeds, 18 km/h and 46 km/h, respectively. The maximum overlapping distance of the APs (i.e., $D_{\max}$) is set to 10 m in both scenarios. Fig. 11 demonstrates the case of a smooth handoff operation whereby the concerned MN experienced no communication-disruption and only a few redundant packets. Fig. 12 presents a contrasting case whereby the comparatively higher speed of a MN leads

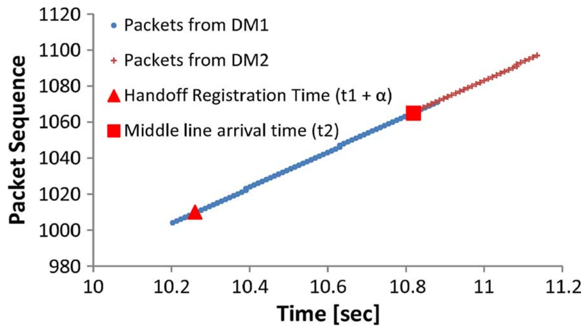

Fig. 11. Packet sequence number variation during handoff period for a mobile node traveling between DM_1 and DM_2 at a slow speed, 18 km/h ($D_{\max} = 10.0$ m).

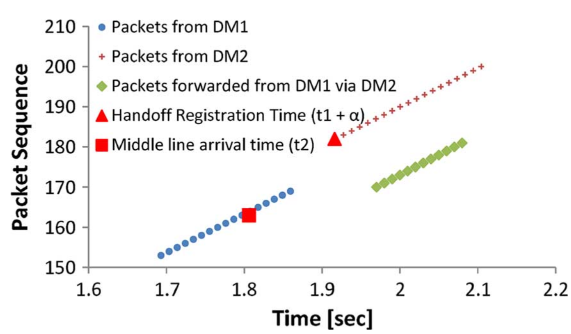

Fig. 12. Packet sequence number variation during handoff period for a mobile node travel ling between DM_1 and DM_2 at a relatively high speed, 46 km/h ($D_{max} = 10.0$ m).

to a substantially short residual time of the MN in the overlapping area, which consequently leads to the actual occurrence of the handoff before the handoff management procedure is completed. As a consequence, the MN misses a number of packets. At a later time, DM_1 forwards these missing packets to DM_2, which in its own turn, delivers them to the MN. The time difference between receiving the last packet from DM_1 and obtaining the first packet from DM_1 through DM_2, i.e., the disruption time, is 110 ms. Indeed, such a small order of disruption time can be easily accommodated by the MN's buffering mechanism. These results, thus, demonstrate a sound coordination between the deployed domain managers (i.e., DM_1 and DM_2) which assist the mobile clients to recover from missing packets.

The different values of the handoff success rate and the streaming efficiency, denoted by Θ and Λ, respectively, are plotted for different populations of MNs in Fig. 13. As per these results, the handoff success rate is close to 100% for every set of MNs population. The robustness of MS^2 in dealing with handoff situations is readily evident from this result, since it demonstrates the successful and smooth video playback at most of the user terminals, even when the population of mobile users is significantly increased. While the handoff success rate is encouragingly high, the system suffers from lower values of Λ. The most plausible reason behind this consists in the inaccuracy involved in assessing the handoff delay α and the actual handoff occurrence time t_2^*, which in turn impacts the computation of N_{dm2}.

To further investigate how the handoff management mechanism in MS^2 is influenced by the surface of the overlapping area, in Fig. 14, the values of Θ and Λ are plotted by varying

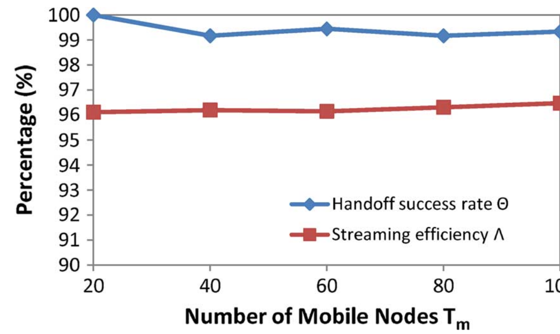

Fig. 13. Performance in terms of Θ and Λ for different sets of mobile nodes.

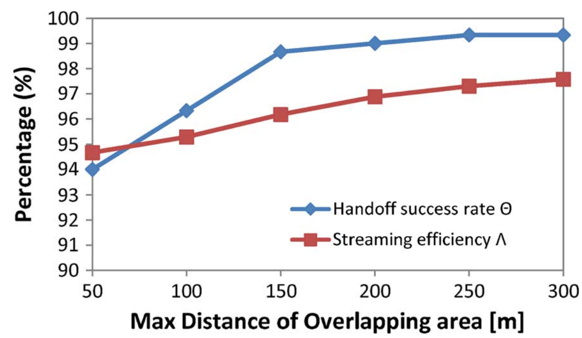

Fig. 14. Performance in terms of Θ and Λ for different values of $D_{\max}$.

$D_{\max}$. The population of mobile nodes is fixed to 100 in these experiments. The results demonstrate that the handoff success rate increases along with increase in the overlapping surface area. The fact that MNs have residual times long enough to recover from the handoff explains these results. Furthermore, the long residual times of MNs lead to the transmission of higher number of redundant packets which in turn degrades the streaming efficiency Λ. In this respect, it is worth mentioning that it is a better choice to secure smooth video playback at the mobile terminals even at the cost of generating duplicate packets to a certain extent. In case the system is capable of precisely predicting the exact handoff occurrence time t_2, the streaming efficiency (Λ) can be largely enhanced. This particular issue requires further attention and we leave this for future research work.

V. Conclusion

The MS^2 architecture, presented in this paper, employs multiple servers for real-time video streaming to mobile users and incorporates a set of mechanisms, each with a specific goal. The objective of the server selection mechanism is to ascertain smooth streaming for mobile users while minimizing the impact of their handoffs from one wireless domain to another. The purpose of the streaming rate computation scheme is to determine the streaming rate of each of the involved media servers in a way that the network resources are utilized in an efficient and fair manner while keeping in mind the limitations of mobile terminals (e.g., playback rate and buffer). The goal of the video fragmentation algorithm is to avoid redundant transmissions by ensuring that one and only one server transmits its allotted burden of packets. The objective of the scheduling operation is to enable fast playback of the video and to minimize packet reordering on the user-side. The paper also explained the influence of handoff events on MS^2 and devised an adequate handoff management scheme to support service continuity.

Aided by computer simulations and a set of realistic scenarios, the performance of MS^2 was evaluated. The obtained results were encouraging and demonstrated the effectiveness of MS^2 in fulfilling its design objectives. We, however, admit that there is still a number of enhancements to be made to MS^2. For instance, the performance of the handoff management mechanism in MS^2 may be further improved by incorporating an algorithm to predict the actual handoff occurrence time with a high accuracy level. Envisioning such an effective mechanism forms our future goal in this particular research area.

References

[1] C. Xu, G. M. Muntean, E. Fallon, and A. Hanley, "Distributed storage-assisted data-driven overlay network for P2P VoD services," *IEEE Trans. Broadcast.*, vol. 55, no. 1, Mar. 2009.

[2] T. Taleb, N. Kato, and Y. Nemoto, "On-demand media streaming to hybrid wired/wireless networks over quasi-geo stationary satellite systems," *Elsevier J. Comput. Netw.*, vol. 47, no. 2, pp. 287–306, Feb. 2005.

[3] B. Wang, W. Wei, Z. Guo, and D. Towsley, "Multipath live streaming via TCP: Performance and benefits," in *Proc. IEEE INFOCOM*, Anchorage, Alaska, May 2007.

[4] T. Nguyen and A. Zakhor, "Multiple sender distributed video streaming," *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 315–326, Apr. 2004.

[5] H. Man and Y. Li, "A multipath video delivery scheme over diffserv wireless lans," in *Proc. SPIE-IS&T Electron. Imag.*, San Jose, Jan. 2004.

[6] D. Jurca and P. Frossard, "Video packet selection and scheduling for multipath streaming," *IEEE Trans. Multimedia*, vol. 9, no. 3, pp. 629–641, Apr. 2007.

[7] A. Nafaa, S. Murphy, and L. Murphy, "Analysis of a large-scale VOD architecture for broadband operators: A P2P-based solution," *IEEE Commun. Mag.*, vol. 46, no. 12, Dec. 2008.

[8] T. Kim and M. H. Ammar, "Receiver buffer requirement for video streaming over TCP," in *Proc. Vis. Commun. & Image Process. Cong.*, San Jose, CA, Jan. 2006.

[9] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "TCP Westwood: End-to-end congestion control for wired/wireless networks," *Wireless Netw.*, vol. 8, no. 5, pp. 467–479, Sep. 2002.

[10] K. Xu, Y. Tian, and N. Ansari, "Improving TCP performance in integrated wireless communications networks," *Comput. Netw.*, vol. 47, no. 2, pp. 219–237, Feb. 2005.

[11] 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," *IEEE Trans. Veh. Technol. J.*, vol. 57, no. 6, pp. 3801–3814, Nov. 2008.

[12] N. Samaan and A. Karmouch, "A mobility prediction architecture based on contextual knowledge and spatial conceptual maps," *IEEE Trans. Mobile Comput.*, vol. 4, no. 6, pp. 537–551, Nov./Dec. 2005.

[13] C. C. Tseng, L. H. Yen, H. H. Chang, and K. C. Hsu, "Topology-aided cross-layer fast handoff designs for IEEE 802.11/Mobile IP environments," *IEEE Commun. Mag.*, vol. 43, no. 12, pp. 156–163, Dec. 2005.

[14] S. Khan, Y. Peng, and E. Steinbach, "Application-driven cross-layer optimization for video streaming over wireless networks," *IEEE Commun. Mag.*, vol. 44, no. 1, pp. 122–130, Jan. 2006.

[15] K. Chebrolu and R. Rao, "Bandwidth aggregation for real-time applications in heterogeneous wireless networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 4, pp. 388–403, Apr. 2006.

[16] J. C. Fernandez, T. Taleb, M. Guizani, and N. Kato, "Bandwidth aggregation-aware dynamic QoS negotiation for real-time video applications in next-generation wireless networks," *IEEE Trans. Multimedia*, vol. 11, no. 6, pp. 1082–1093, Oct. 2009.

[17] B. Ciubotaru and G. M. Muntean, "SASHA—A quality-oriented handover algorithm for multimedia content delivery to mobile users," *Special Issue on IPTV, IEEE Trans. Broadcast.*, vol. 55, no. 2, pt. 2, Jun. 2009.

[18] C. Guo, Z. Guo, Q. Zhang, and W. Zhu, "A seamless and proactive end-to-end mobility solution for roaming across heterogeneous wireless networks," *IEEE JSAC*, vol. 22, no. 5, pp. 834–848, Jun. 2004.

[19] R. Koodli, "Fast handovers for mobile IPv6," Network Working Group, RFC 4068, Jul. 2005.

[20] A. A. Akintola, G. A. Aderounmu, and M. O. Adigun, "Performance analysis of double buffer technique (DBT) model for mobility support in wireless IP networks," *J. Netw. Comput. Appl.*, vol. 29, no. 4, pp. 217–234, Nov. 2006.

[21] "UCB/LBNL/VINT: Network simulator—Ns (version 2)," [Online]. Available: http://www.isi.edu/nsnam/ns/

Tarik Taleb (S'04–M'05–SM'10) received the B.E. degree in information engineering with distinction, and the M.Sc. and Ph.D. degrees in information sciences from GSIS, Tohoku Univ., in 2001, 2003, and 2005, respectively.

He is currently working as Senior Researcher and 3GPP Standards Expert at NEC Europe Ltd., Heidelberg, Germany. Prior to his current position and till March 2009, he worked as Assistant Professor at the Graduate School of Information Sciences, Tohoku University, Japan. From October 2005 till March 2006, he was working as research fellow with the Intelligent Cosmos Research Institute, Sendai, Japan. His research interests lie in the field of architectural enhancements to 3GPP networks, mobile multimedia streaming, wireless networking, inter-vehicular communications, satellite and space communications, congestion control protocols, handoff and mobility management, and network security. His recent research has also focused on on-demand media transmission in multicast environments.

Dr. Taleb is/was on the editorial board of the IEEE Wireless Communications Magazine, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE Communications Surveys & Tutorials, and a number of Wiley journals. He is serving as secretary of the Wireless Communications Technical Committee, the largest in IEEE Communication Society (ComSoC). He also served as Secretary and then as Vice Chair of the Satellite and Space Communications Technical Committee of IEEE ComSoc (2006–2010). He has been on the technical program committee of different IEEE conferences, including Globecom, ICC, and WCNC, and chaired some of their symposia. He is the recipient of the 2009 IEEE ComSoc Asia-Pacific Young Researcher award (June 2009), the 2008 TELECOM System Technology Award from the Telecommunications Advancement Foundation (March 2008), the 2007 Funai Foundation Science Promotion Award (April 2007), the 2006 IEEE Computer Society Japan Chapter Young Author Award (December 2006), the Niwa Yasujirou Memorial Award (February 2005), and the Young Researcher's Encouragement Award from the Japan chapter of the IEEE Vehicular Technology Society (VTS) (October 2003). Dr. Taleb is a Senior IEEE Member.

Kazuo Hashimoto (M'06) was born in Tokyo in 1953. He received the Ph.D. degree in Information Science from Tohoku University in 2001. He is currently a Professor at the Graduate School of Information Sciences of Tohoku University, directing the theoretical and practical study of web communication.

Dr. Hashimoto received the Incentive Award from the Japanese Society for Artificial Intelligence in 1999, the Best Paper Award in the field of Artificial Intelligence from the Institute of Electronics, Information and Communication Engineers in 2001, and the achievement award from the Association of Radio Industries and Businesses in 2002. He is a Member of IPSJ, IEICE, AAAI, IEEE.