G. Anastasi, D. Grillo, and L. Lenzini

An Access Protocol for Speech/Data/Video Integration in TDMA-Based Advanced Mobile Systems

Paper accepted for publication in the IEEE Journal on Selected Areas in Communications, Special Issue on Personal Communications - Services, Architecture and Performance Issues

An Access Protocol for Speech/Data/Video Integration in TDMA-Based Advanced Mobile Systems

Giuseppe Anastasi*, Davide Grillo**, and Luciano Lenzini*

** Dept. of Information Engineering, University of Pisa, Via Diotisalvi 2, I-56126 Pisa (Italy),*
anastasi@iet.unipi.it, lenzini@iet.unipi.it
*** Fondazione Ugo Bordoni, Via Baldassarre Castiglione, 59, I-00142, Roma (Italy), grillo@fub.it*

Abstract **- One key requirement on the radio access design of advanced, third generation mobile systems is the ability to accommodate a variety of service types via flexible and efficient multiple access protocols. The paper introduces a multiple access protocol, SIR++ (Service Integration for Radio Access), which has the potential for meeting the service requirements of speech, video and bursty data traffic in an efficient way. SIR++ is evolved from and generalizes a former protocol, SIR, which only considered the requirements of speech and data. The paper also evaluates the buffer occupancy distribution associated with a (H.263) video connection by solving a worst-case model of the system.**

I. INTRODUCTION

The scope of wireless communications extends in various dimensions. These include: the serviced area, the mix of services to be supported (and related bitrates), the degree of mobility enabled, the operation environment - to name just a few. The combinations between the options possible for each dimension have given rise to a manifold of systems, each optimized for a particular combination. Advanced, third generation systems like FPLMTS/IMT-2000, [1], and UMTS, [2], aim to cover a large set of the possible combinations under one system design. Given the many, diverse operating conditions and requirements involved, [3], and in an attempt to accommodate them in an optimal way, such systems are being conceived with a high degree of adaptivity for techniques such as channel coding, interleaving, power control, error recovery, etc. In addition, the multiple access techniques for advanced mobile systems should be flexible enough to accommodate the needs of different - and to some extent not yet specified - services while guaranteeing efficient use of radio resources. In the current vision of a radio interface for advanced mobile systems, the bandwidth allocation mechanisms play a very significant role and are viewed as an essential "radio-dependent" building block which should help to achieve universal service, [4].

On the TDMA front, a family of access protocols has evolved from the initial proposal of the PRMA protocol, [5] - [6], for exploiting pauses of traffic sources characterized by intermittent activity. With PRMA, the base station informs the mobiles about the occupancy status of time slots in a regularly repeating frame used for bandwidth apportionment. Traffic sources associated with mobiles contend for bandwidth at the beginning of each of their "activity" periods by attempting transmission of information segments on slots perceived as free. PRMA++, [7], [8], [9], is a variation of PRMA whereby stations contend to notify their bandwidth requirement via "request packets" and dedicated bandwidth. With PRMA++ bandwidth is centrally allocated by the base station to queued requests. Although both PRMA and PRMA++ can be operated in a range of variations, they have been mainly studied for speech traffic. Moreover, both protocols are entirely based on contention of traffic sources which, if not properly controlled, may lead to instability of operation and inefficient use of bandwidth.

A combined contention/reservation protocol, SIR (Service Integration for Radio Access) has been introduced specifically for mixtures of speech and data traffic, [10] - [11]. With SIR, speech is handled by making use of the basic PRMA++ scheme, whereas data applications share bandwidth in a collision-free manner both for notifying transmission requests and for transmitting user data.

This paper analyzes the performance of an enhancement of SIR, SIR++, designed to meet the requirements of a generic mixture of services. Basic concepts underlying SIR++ are: i) services are allocated a guaranteed amount of bandwidth plus an amount on a "best effort" basis from the bandwidth not currently used by other services; ii) services are associated with priority classes for managing the allocation of the "best effort" bandwidth. Depending on the service requirements and characteristics, guaranteed and "best effort" bandwidth have not necessarily to be both given. Similarly as with SIR, the new protocol uses Round-Robin mechanisms within each service class for both distributing "best effort" bandwidth and handle related signalling, thus achieving fair and controlled sharing of radio resource. The potential of SIR++ is demonstrated with reference to a mixture of speech, data and low bitrate video sources. Although not expressly proposed for wireless communication, the H.263 standard, [12], is assumed in this paper as the video coding technique due to its potential and the availability of statistics useful for modelling traffic sources.

The focus of the paper is primarily on bandwidth allocation mechanisms, implementation of physical details of the protocol is beyond the scope of the paper.

II. SIR++ ACCESS PROTOCOL FOR SERVICE INTEGRATION

SIR++ is designed for supporting an arbitrary number of service classes, with associated service requirements such as delay and throughput. Classes are identified by an integer $Class$, ranging from 0 to a maximum value $Class\_Max$. Users specify the required service class for the actual session during the connection set-up phase. $Class = 0$ class is equivalent to the speech service provided by PRMA++. $Class = 1$ class provides each individual user with a minimum guaranteed bandwidth while additional bandwidth, when requested and available, is provided on a best effort basis while ensuring equitable and fair handling of the sessions.

This service class has been envisaged to manage VBR sources (typically video), where the amount of bandwidth guaranteed is below the peak rate (but above the average). Finally, service classes 2 to $Class\_Max$ are intended to accommodate various types of data traffic and behave similarly as class 1, with two differences: i) the bandwidth guarantee is optional; ii) the guaranteed bandwidth is collectively allocated to the group of data sessions requiring the same service class as opposed to being allocated on an individual basis.

With SIR++ the bandwidth in excess of the guaranteed bandwidth is obtained according to a priority scheme based on the $Class$ value, i.e., in achieving extra bandwidth, the service class characterized by $Class \in [1, Class\_Max - 1]$ has priority over the $(Class + 1)$ class. Similarly as with PRMA++ and SIR, bandwidth is centrally allocated in SIR++ by the base station which maintains information on slot status and usage.

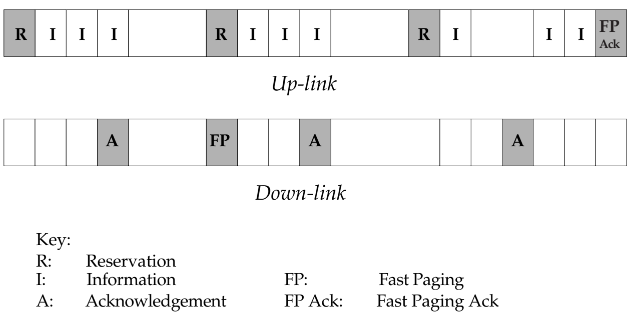

Figure 1: SIR++ frame structure (key elements).

A. Frame Structure and Slot Usage

To service the above classes, SIR++ adopts the physical frame structure shown in Fig. 1 (this structure may be viewed as a generalization of the PRMA++ frame). Uplink and downlink slots and frames are synchronized. The slots on the uplink are separated into reservation slots $(R-\text{slots})$ and user information slots $(I-\text{slots})$. The slots on the downlink are separated into acknowledgement slots $(A-\text{slots})$, paired with uplink $R-\text{slots}$, and user information slots $(I-\text{slots})$.

$I-$ and $R-\text{slots}$ in an uplink frame are partitioned into subsets associated with service classes, $I_{Class}-\text{slots}$ and $R_{Class}-\text{slots}$ respectively. The $R_{Class}-\text{slots}$ provide each class with dedicated bandwidth for making reservations and the number of such slots in each class is an operation parameter. The mechanisms for accessing the $R-\text{slots}$ are differentiated: while the $R_0-\text{slots}$ are accessed on a contention basis, access for all other $R-\text{slots}$ is ruled according to a Round-Robin discipline. In the following it is assumed that the $R_{Class}-\text{slots}$ in each class are set once for all, i. e. they do not change during the observation of the system.

The mechanism for allocating the $I_{Class}-\text{slots}$ is based on the distinction between Available and Reserved status. An $I-\text{slot}$ is said to be Reserved when its allocation to an individual session (class 0 and 1), or to a group of sessions (classes 2 to $Class\_Max$) implies the repeated use of the slot for a time which may either correspond to an activity spurt (typically a talkspurt) or to the whole duration of a session. An Available $I-\text{slot}$ is one whose use in a particular frame does not imply its repeated use by the same session, or group of sessions, in subsequent frames. (When allocated on an activity spurt basis, the status of a slot alternates between Reserved and Available).

Service class 0 uses only Reserved slots, whereas service classes 1 to $Class\_Max$ may use a hybrid of Reserved and Available slots. A further distinction in the protocol is that while class 1 uses Reserved slots for providing a session with bandwidth necessary to meet a minimum bandwidth requirement and Available slots are used for accommodating (possible) excess bandwidth demand, with classes 2 to $Class\_Max$ the use of Reserved slots is optional. Finally, to provide fairness in bandwidth allocation, Round Robin discipline is used for $I-\text{slot}$ allocation with classes 1 to $Class\_Max$.

Table 1 indicates the use of slots in SIR++. From the above, it is apparent that class 0 is tailored to meet the requirements of speech as done with PRMA++. In fact, once reservations (handled on a contention basis) are successfully processed by the base station, guaranteed bandwidth is allocated for the residual duration of a talkspurt (via Reserved $I-\text{slots}$) as with PRMA++. When class 1 is associated with VBR video sources, Reserved $I-\text{slots}$ sufficient for meeting the guaranteed bandwidth requirements are allocated to the video sessions already at session set-up. The need for excess bandwidth is notified via $R-\text{slots}$ and accommodated on a best-effort basis. To provide fairness both in notifying the transmission needs and in sharing the extra bandwidth, $R-\text{slots}$ and Available $I-\text{slots}$ are accessed following a Round-Robin algorithm.

Classes 2 to $Class\_Max$ are typically associated with data sessions having different throughput and delay requirements. As anticipated, SIR++ can be operated so as to choose how many Reserved slots should be associated with classes 2 to $Class\_Max$

Table 1: Use of slots with the SIR++ protocol.

SIR phase	$Class=0$ Service Class	$Class=1$ Service Class	$Class>1$ Service Class
Connection set-up	R_0-slots, access contention	R_0-slots, access contention	R_0-slots, access contention
Bandwidth request notification	R_0-slots, access contention	R_1-slots, collision free (Round-Robin)	R_{Class}-slots collision free (Round-Robin)
Information Transfer	Reserved I_0-slots, collision free (STDM)	Reserved I_1-slots collision free (STDM, allocation on an individual basis) + Available I_1-slots collision free (Round-Robin, collectively used by all sessions)	Reserved (optional) and Available I_{Class}-slots collision free (Round-Robin, both Reserved and Available slots allocated collectively)

B. Priority Handling and Protocol Features

The association between Available $I-\text{slots}$ and service classes is established by the base station by means of a set of thresholds $Thres_n$ for $n \in \{1,2,\ldots,Class\_Max\}$, with

$$0 \le Thres_1 < \cdots < Thres_n < Thres_{n+1} < \cdots < Thres_{Class\_Max},$$

where $Thres_{Class\_Max}$ is bounded above by the maximum number of $I-\text{slots}$ in a SIR++ frame. If S denotes the number of Available $I-\text{slots}$ in a frame, the priority algorithm works as follows: the number Δ_{Class} of Available $I-\text{slots}$ which will be associated by the base station to the $Class$ service class is given by

$$\Delta_{Class} = \begin{cases} \min(S, Thres_{Class}) & \text{if } Class = 1 \\ \left(\min(S, Thres_{Class}) - Thres_{Class-1}\right)^+ & \text{if } Class > 1 \end{cases}$$

where

$$(x)^+ = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \ge 0 \end{cases}$$

In the following we focus on $Class \ge 1$ service classes and, specifically, we emphasize the description of the SIR++ protocol for the provision of excess bandwidth.

C. Reservation and Bandwidth Allocation Mechanisms

In this section, details about the procedures for notification of the transmission need and for bandwidth allocation are given. For a description of these procedures in terms of FSM see [11].

The concepts of "open connection" and "active connection" are recurrently used in the following description. An open connection is a relationship established between a mobile terminal and the base station for the exchange of user data. This relationship may originate from set-up of calls or successful handling of handover attempts.

During the time an open connection is maintained:

- traffic sources requesting $Class=1$ service classes (e.g., video) may need additional slots other than the reserved ones;
- traffic sources requesting $Class>1$ service classes (e.g., data) may alternate "activity" and "silence" periods.

In both cases, when Available slots will be requested by the mobile to the base station the connection is said to become "active". (During the time an open connection is maintained, $Class=0$ service classes also alternate between "activity" and "silence" periods. However, as already explained, the related bandwidth request is met by only using Reserved slots). An active connection remains in this state as long as Available slots are needed and then it switches back to the inactive state. Hence, during the lifetime of a connection, active and inactive states alternate.

Since, for each service class, the SIR++ protocol adopts the same procedure for managing best-effort bandwidth, in the following, we describe the Available I-slots acquisition procedure with reference to a generic service class. Hence, the $Class$ subindex is omitted during the description of SIR++.

C.1 Connection Set-Up and Tear-Down

Independently of the class of the connection to be set-up, mobile originated connections make use of R slots associated with $Class=0$ service class for notifying the base station. These slots are handled as ordinary $Class=0$ R slots. Once successful, the request control packet is acknowledged by the base station, in the paired A-slots, with another control packet containing:

- an identifier, say {i}, assigned to the connection just opened, which is unique within the cell controlled by the base station;
- the value of a variable indicating the number of open connections (NOC) of the same class. Note that the value of NOC includes the connection just opened and is broadcast by the base station.

Mobile initiated connection tear-down is performed by transmitting a control packet to the base station in the related R-slots (see below). Upon receiving this control packet, the base station updates NOC whose use by each station is described below.

C.2 Acquisition of Available Slots

Upon arriving at station *{i}* [1], a message is segmented into packets, whose size fits the payload transmitted in one slot. Packets then join a local queue, L_Q(i), which is left when packets are scheduled for transmission after station *{i}* receives authorization from the base station[2]. To obtain this authorization a station executes the *Request* algorithm which basically consists in sending to the base station a control packet (using dedicated R-slots) carrying the current number of packets waiting in the local queue. The station then waits for a control packet from the base (in the A-slot paired with the R-slot) informing on the current number of active connections, NAC. At the end of this information exchange the open data connection becomes *active* and remains such until the last authorized packet has been transmitted. When this occurs, the data connection becomes *inactive* if the local queue is empty, otherwise another request procedure is immediately initiated.

When a data connection becomes active, it shares Available I-slots with all the other active connections. To achieve this, all the stations execute the Transmission algorithm which makes use of NAC. The number of active connections may change at the end of each slot since an open connection may become active or an active connection may become inactive. For this reason, a variable local to each station, denoted by Q, is initialized to the NAC value when the connection becomes active, and it is updated to keep track of the changes before the beginning of the subsequent slots on the basis of information broadcast by the base station.

The NOC data connections in each class are then partitioned into two sets, one including the Q active connections, and the other containing the NOC-Q inactive connections.

To simplify the protocol description, we assume that each mobile is associated with only one connection.

C.3 User Information Transport

The processes associated with information transport are: i) a periodic scan of the open connections to collect bandwidth requests (Request algorithm); ii) allocation of the available bandwidth to meet the collected requests (Transmission algorithm). By design, both processes are managed so as to avoid contention and are implemented in a distributed fashion. With the Request algorithm, R-slots are cyclically offered for dedicated use to each of the open connections for collecting the accumulated bandwidth request since last notification. With the Transmission algorithm, Available slots are used cyclically by the active connections: i.e., one Available slot is offered for use to each of the active connections for packet transmission.

To access R-slots, station{i}, maintains a counter, denoted R_POS(i), which is initialized to the NOC value at the time the connection is opened. This counter is decreased by one every time station{i} observes an R-slot of the related service class. When R_POS(i) reaches the value of one, the next R-slot can be used by station{i} for transmission. After completion of transmission R_POS(i) is initialized again to the current NOC value. This process is repeatedly executed during the time a connection remains open.

Depending on the current value of NOC and the number of R-slots in a frame, a complete scan of all open connections may take one or more frames. Clearly, the price paid for collision-free request collection is that not all R-slots will necessarily be used at each scan, and the trade-off between bandwidth utilization and timeliness in identifying bandwidth requests is due to an operation decision.

With the Transmission algorithm, Available I-slots are offered for use of active connections. These slots are allocated in a cyclic fashion. Specifically, when the open connection at station{i} becomes active, station{i} updates a counter, T_POS(i), to the Q value. This counter is decreased by one every time station{i} observes Available I-slots of the related class (all stations observe slots at the same time). When T-POS(i) reaches the value of one, the next Available I-slot can be used by station{i} for packet transmission. After completion of transmission T-POS(i) is loaded with the value of Q which equals the current NAC. This process is repeatedly executed during the time a connection remains active.

The Transmission algorithm is transparent to the frame begin/end demarcation, and basically allows use of any residual capacity. The key parameters for SIR^{++} operation are summarized in Table 2. The technicalities of the Request and Transmission algorithms are described in [11].

Table 2: Key parameters for SIR^{++} operation.

NOC$_{Class}$	Number of open connections belonging to class *Class*
NAC$_{Class}$	Number of active connections belonging to class *Class*
Q$_{Class}$	Number of active connections (i. e. with at least one packet in the transmission buffer) of class *Class*
R_BKG$_{Class}$	Number of packets in the request buffer of class *Class*
T_BKG$_{Class}$	Number of packets in the transmission buffer of class *Class*
R_POS$_{Class}$	Counter for controlling transmission of requests for best-effort bandwidth on R-slots of class *Class*
T_POS$_{Class}$	Counter for controlling transmission of data packets on Available I-slots of class *Class*

III. WORST-CASE MODEL DESCRIPTION

One particular case of system operation for a mixture of speech, video and data traffic, is to meet the requirement of excess demand for VBR video sources and bandwidth demand for data traffic exclusively via bandwidth occasionally left unused by speech traffic. This case will be considered in the following to assess the protocol performance.

In order to analyze speech, data and video integration by SIR++ we assume that speech, video and data are associated with classes 0, 1 and 2, respectively. Furthermore, we focus our analysis on a single cell and assume that the number of

[1]The same identifier i is used to address a mobile or the connection associated with it.

[2]For users requiring a $Class = 1$ service class with reserved bandwidth, e.g., video, only packets exceeding the number of reserved slots are queued in L_Q queue.

speech, data and video users remains unchanged, i.e., we do not consider variations in the number of NOC due to incoming and terminating calls and handovers.

Since in our analysis the prioritized-available-slots are used by data and video sessions, to ease the presentation we make reference to low- ($Class = 2$, data) and high- ($Class = 1$, video) priority-available-slots. Furthermore, to make reference to slots (e.g., I and R-slots) and quantities (e.g., number of sessions) associated with speech, video, and data, we use sub-indices s, v, and d, respectively.

Under the assumption of a fixed number of speech, video, and data sessions, the total number, N, of slots in a SIR++ uplink frame will be partitioned into three sets. The first set includes N_d I_2-slots and N_{d_sig} R_2-slots. I_2-slots (Reserved) are allocated for exclusive use of data sessions. As stated above their number (N_d) may be zero. The second set comprises N_v I_1-slots each of which is allocated to a specific video session (i.e., these slots are in the Reserved state) and N_{v_sig} R_1-slots used by video sessions to require extra bandwidth. The third set is made up of N_s I_0-slots and N_{s_sig} R_0-slots. According to PRMA++, I_0-slots are allocated by the base station to speech sessions and R_0-slots are used by speech sessions to require the use of these I_0-slots when a talkspurt comes up. The portion of the N_s I_0-slots not currently allocated to any speech session (i.e., in the Available state) are referred to as high-/low-priority-available slots. In the following we denote by $S_{\max}$ the maximum number of high-priority-available-slots (i.e., $Thres_1 = S_{\max}$). $S_{\max}$ is assumed to be a portion of the total maximum number[3] of slots unused by speech traffic. Available slots exceeding $S_{\max}$ are considered as low-priority-available-slots (i. e., $Thres_2 = N_s$) and hence, they can be accessed by data sessions.

In our analysis we assume that N_d is equal to zero. Hence, to run data sessions, SIR++ makes use of low-priority-available-slots only. Therefore, $N = N_s + N_v + N_{sig}$, where N_{sig} represents the total number of $R-$slots per SIR++ uplink frame, plus the FP-Ack-slot (i. e., $N_{sig} = N_{d\_sig} + N_{v\_sig} + N_{s\_sig} + 1$).

The SIR++ system (or simply *system* thereafter) under consideration comprises T terminals (speech, video and data) which access a common uplink channel. Since throughout the paper it is assumed that a terminal is associated with only one session at time, when it does not generate confusion, the terms "terminal" and "session" will be used interchangeably in the following. We denote with T_d, T_s, and T_v the number of data, speech and video terminals respectively. Clearly $T = T_d + T_s + T_v$.

[3] In principle, this number is equal to N_s.

The performance evaluation of SIR++ when loaded with a mixture of speech and data traffics can be found in [10] and [11]. Hence, to evaluate the full SIR++ service integration capability, it remains to analyze the Quality of Service (QoS) achieved by a specific video terminal (tagged video terminal thereafter) for different (fixed) numbers of speech and data terminals . The number of slots accessible to the tagged video terminal (on a frame basis) obviously depends on the amont of traffic associated with speech terminals. Furthermore, although data traffic belongs to a service class with lower priority than video, only a portion of the number of slots (in a SIR++ frame) unused by speech terminals can be allocated to video traffic when it is decided to provide a minimum of service to data.

Unfortunately, this type of analysis is very complex and computational demanding due to the SIR++ protocol, to the number and workload characterization of data, speech and video sources. To overcome these difficulties we introduce a simplified model which can be analytically solved and yet still provides useful information on the QoS achieved by video sessions. In this model all the video sessions, apart from the tagged one, are assumed to be constantly in the need of extra bandwidth. We solve this model by using the matrix analytic technique [13].

Specifically, the model developed bases on the following assumptions:

- uplink and downlink channel transmissions are error-free;
- base station receiver has no packet capture capability;
- each video source generates a video-frame every N_f SIR++ frames;
- each video session uses its Reserved I_1-slots the number of which, for the tagged terminal per SIR++ frame, will be denoted by S_g ($1 \le S_g \le N_v$);
- all the video sources but the tagged one, in addition to using their Reserved I_1-slots, have always enough packets in the queue to be transmitted in the high-priority-available-slots;

From the bandwidth allocation standpoint, due to the latter assumption, the above model belongs to the class of the so called "worst case" models and will be accordingly referred to in the following.

IV. CHARACTERIZATION OF THE AVAILABLE SLOTS COUNTING PROCESS

From the above section, it clearly emerges that the solution of the proposed model requires the knowledge of the distribution of the number (S) of Available $I-$slots, on a frame by frame bases. It can be shown that the value of S in any frame depends upon the value of S in the previous frame in addition to the value of other state variables (e.g., the number of sessions for which the talkspurt comes up) [14]. However, to simplify the analysis we assume that the process which counts the number of Available $I-$slots (hereafter Counting Process) can be modeled by a discrete-time, discrete-state Markov chain $\{S_n, n \in \mathrm{N}\}$ with probability transition matrix $\|S_{hk}\|$ where $0 \le h,k \le N_s$. This assumption simplifies the tractability of the problem with a small impact on the accuracy of the results. Furthermore, let us denote with y_i, with $i \in \{0,1,...,N_S\}$, the generic component of the stationary probability row vector of $\|S_{hk}\|$

Since in the system we analyze the maximum number $(S_{\max})$ of high-priority-available-slots used by video sessions is much lower than N_s (see Section IX) the characterization of the counting process (for our purposes) could be greatly simplified if there would exist an integer value $S_{\dim}$, $S_{\max} \le S_{\dim} < N_s$, such that the probability transition matrix $\|S_{hk}\|$ would be *exactly lumpable* [15] with respect to the partition $\{0,1,...,S_{\dim}-1,\{S_{\dim},....,N_s\}\}$ into $S_{\dim}+1$ mutually exclusive and exhaustive sets of the discrete-time, discrete-state Markov chain $\{S_n, n \in \mathrm{N}\}$ states. In this case, states $\{0,1,...,S_{\dim}-1\}$ remain unchanged while states $\{S_{\dim},....,N_S\}$ will be aggregated into a *macrostate* that will be denoted by $\{\hat{S}_{\dim}\}$.

For the system configurations analyzed in Section IX, we verified that there exists an $S_{\dim}$ such that the sum $\sum_{k=0}^{S_{\dim}-1} S_{hk}$ is approximately the same for any $h \in \{S_{\dim},...,N_s\}$, and hence the necessary and sufficient condition for exact lumpability are met, [15] . Hence, the transition probability matrix of the lumped Markov chain, of size $S_{\dim}+1$, denoted throughout by $\|\tilde{S}_{ij}\|$, is

$$\tilde{S}_{ij} = \begin{cases} S_{ij} & 0 \le i,j \le S_{\dim}-1 \\ \sum_{k=S_{\dim}}^{N_s} S_{ik} & 0 \le i \le N_S, j = \hat{S}_{\dim} \\ \sum_{h=S_{\dim}}^{N_s} \left(\dfrac{y_h S_{hj}}{\sum_{n=S_{\dim}}^{S_{\max}} y_n} \right) & i = \hat{S}_{\dim}, 0 \le j \le S_{\dim}-1 \\ \sum_{k=S_{\dim}}^{N_s} \sum_{h=S_{\dim}}^{N_s} \left(\dfrac{y_h S_{hj}}{\sum_{n=S_{\dim}}^{S_{\max}} y_n} \right) & i = \hat{S}_{\dim}, j = \hat{S}_{\dim} \end{cases} \tag{1}$$

Moreover, if $\hat{\mathbf{y}}$ is the stationary probability row vector of $\tilde{S}_{ij}$, then $\hat{y}_i = y_i$ for $0 \le i \le S_{\dim}-1$, while $\hat{y}_{\hat{S}_{\dim}} = \sum_{h=S_{\dim}}^{S_{\max}} y_h$.

V. VIDEO ARRIVAL PROCESS CHARACTERIZATION

In our analysis we represent the behaviour of a video source by means of a discrete-time discrete-state Markov chain, denoted by $\{C_n, n \in \mathrm{N}\}$, with $(V_{\max}+1)$ states and transition probability matrix $\|C_{ij}\|$, with $0 \le i,j \le V_{\max}$. This Markov chain was derived through a fitting procedure by using a real video sequence coded according to the H.263 standard [12], and available at a server ([16].). H.263 is a new standard for Very Low Bit Rate Video coding which, even if designed for fixed networks (one key application will be videotelephony on normal analogue telephone lines) will be likely enhanced to be used on third generation mobile networks [17].

Time instants at which Markov chain transitions occur correspond to the video frame arrival instants. Furthermore, video frame size distribution only depends on the current state of the Markov chain and for a specific state it is deterministic. Thus, if the Markov chain makes a transition to state $\{i\}$, with $0 \le i \le V_{\max}$ a video-frame with a constant number of packets, thereafter $arr(i)$, is generated according to the following relation

$$arr(i) = iF + F/2 \tag{2}$$

where F is a constant factor. Both F and the elements of $\|C_{ij}\|$ are estimated from the video sequence histogram. Specifically, we followed a methodology similar to that described in [18]. F was simply obtained by dividing the maximum frame size in the sequence by the number of states $(N_{\max}+1)$ of the Markov chain $\{C_n, n \in \mathrm{N}\}$. Furthermore, if D_n, $n \in \mathrm{N}$, indicates the number of SIR++ packets in video frame n then the corresponding state in the Markov chain $\{C_n, n \in \mathrm{N}\}$ is given by

$$C_n = \left\lfloor \frac{D_n}{F} \right\rfloor$$

Transition probabilities C_{ij} ($0 \le i,j \le V_{\max}$) are estimated in the usual way [19]

$$C_{ij} = \frac{number\ \ of\ \ transitions\ \ from\ \ i\ \ to\ \ j}{number\ \ of\ \ transitions\ \ out\ \ of\ \ j} \tag{3}$$

for $0 \le i \le N_{\max}$ and for those $0 \le j \le V_{\max}$ for which the denominator in (3) is greater than zero.

The resulting Markov chain model, approximates the real video source due to the Markovian structure of the model and the limited number of states. The latter reason is due to the fact that an increase in the number of states causes an increase in the size of the state space of the SIR++ model (see next section). The above approximations in the arrival process is not critical for our analysis since we compare the

SIR++ performances to different video bandwidth allocation schemes.

VI. MARKOV CHAIN DESCRIPTION OF THE SIR++ SYSTEM

Transition epochs of the Markov chain $\{C_n, n \in \mathrm{N}\}$, representing the (tagged) video arrival process, occur immediately before the beginning of an R_1-slot, that, for simplicity, is assumed to be located at the end of a SIR++ frame[4]. For ease of presentation we introduce the notion of superframe as the number (N_f) of SIR++ frames included between two consecutive video frame arrivals (see **Fig. 2**).

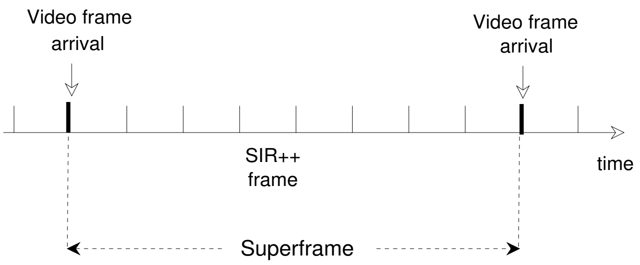

Figure 2: Definition of superframe.

With the above assumptions and definitions we observe the system immediately after the beginning of each SIR++ frame (embedding point). To characterize the state of the system at the n-th embedding point the following tuple of random variables is used: $[Q, \Phi, C, S, \Sigma]_{(n)}$ where

- Φ: identifies each SIR++ frame inside a superframe, i.e., $0 \le \Phi \le N_f - 1$. The r.v. Φ evolves according to the following equation

$$\Phi_{(n)} = \left(\Phi_{(n-1)} + 1\right) \bmod \left(N_f\right)$$

where mod indicates the modulus operator.

- C: represents the state of the video arrival process. As shown in Section V, this process is represented by a discrete-state, discrete-time Markov chain $\{C_n, n \in \mathrm{N}\}$ with transition probability matrix $\|C_{ij}\|$.

- S: represents the number of slots left free by speech sources in the current frame. As shown in Section IV, $\{S_n, n \in \mathrm{N}\}$ is assumed to be a discrete-state, discrete-time Markov chain with lumped transition probability matrix $\|\tilde{S}_{ij}\|$.

- Σ: represents the video session which will use the last high-priority-available-slot in the current SIR++ frame. As these slots are accessed by video terminals according to the Round Robin discipline, it can be verified that

$$\Sigma_n = \left[\Sigma_{n-1} + \min\left(S_n, S_{\max}\right)\right] \bmod \left(T_v\right).$$

- Q: represents the number of queued packets in the tagged video terminal. Q evolves according to the following equation

$$Q_n = \max\left[0, Q_{(n-1)} - U_n\right] + C_n I_{\{\phi_{(n)}=0\}}$$

where U_n indicates the maximum number of (guaranteed + Available) I-slots which can be accessed by the tagged video terminal in the current frame. It can be easily verified that the following relation holds

$$U_n = S_g + \left\lfloor \frac{\Sigma_{n-1} + \min\left(S_n, S_{\max}\right)}{T_v} \right\rfloor \qquad (4)$$

The discrete-time, discrete-state process $X = \left\{[Q, \Phi, C, S, \Sigma]_{(n)}, n \in \mathrm{N}\right\}$ is an homogeneous Markov chain.

By introducing two new random variables L and Z so that $Q = L \times P_{\max} + Z$ the system dynamic can be represented by Markov chain $Y = \left\{[L, Z, \Phi, C, S, \Sigma]_{(n)}, n \in \mathrm{N}\right\}$ which is an M/G/1-type Markov chain ([13]) with level L, phase (Z, Φ, C, S, Σ) and space state

$$E = \left[(l, z, \varphi, c, s, \sigma):\ l \in N,\ 0 \le z < P_{\max}, \\ 0 \le \varphi < N_f,\ 0 \le c \le V_{\max}, \\ 0 \le s \le \hat{S}_{\dim},\ 0 \le \sigma < N_v\ \right]$$

where $P_{\max}$ indicates the maximum number of (guaranteed + Available) I-slots which can be accessed by the tagged video terminal in a frame, i.e., $P_{\max} = \max_n (U_n)$.

The resulting transition probabilities matrix P of Y is as shown in **Fig. 3**.

$$P = \begin{array}{c} L_n = 0 \\ L_n = 1 \\ L_n = 2 \\ L_n = 3 \\ \cdots\cdots \end{array} \begin{pmatrix} B_0 & B_1 & B_2 & B_3 & B_4 & \cdots \\ A_0 & A_1 & A_2 & A_3 & A_4 & \cdots \\ 0 & A_0 & A_1 & A_2 & A_3 & \cdots \\ 0 & 0 & A_0 & A_1 & A_2 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix}$$

Figure 3: Transition probabilities matrix P.

[4] Obviously, this implies that the number of R_1-slots is such that this requirement is met.

Block matrices $B_k(k \ge 0)$, and $A_k(k \ge 0)$ are square matrices of $N_f \cdot (V_{\max}+1) \cdot (\hat{S}_{\dim}+1) \cdot N_V \cdot P_{\max}$ size with elements

$$B_k[(z,\varphi,c,s,\sigma),(z',\varphi',c',s',\sigma')] =
\begin{cases}
S_{ss'} & \text{if} \quad \{\varphi' = [(\varphi+1) \bmod N_f]\}, \\
& \{\varphi' \neq 0\},\ \{c' = c\}, \\
& \{\sigma' = [(\sigma + \min[s', S_{\max}]) \bmod T_v]\}, \\
& \{\max[0,(z-u)] = z'\} \\
S_{ss'}C_{cc'} & \text{if} \quad \{\varphi' = [(\varphi+1) \bmod N_f]\},\ \{\varphi' = 0\}, \\
& \{\sigma' = [(\sigma + \min[s', S_{\max}]) \bmod T_v]\}, \\
& \{\max[0,(z-u)] + arr(c') = k \cdot P_{\max} + z'\} \\
0 & \text{else}
\end{cases}$$

(5)

where

$$u = S_g + \left\lfloor \frac{\sigma + \min[s', S_{\max}]}{T_v} \right\rfloor \qquad (6)$$

$$arr(k) = kF + F/2 \qquad (7)$$

with F defined in Section V.

$$A_k[(z,\varphi,c,s,\sigma),(z',\varphi',c',s',\sigma')] =
\begin{cases}
S_{ss'} & \text{if} \quad \{\varphi' = [(\varphi+1) \bmod N_f]\}, \\
& \{\varphi' \neq 0\},\{c' = c\}, \\
& \{\sigma' = [(\sigma + \min[s', S_{\max}]) \bmod T_v]\}, \\
& \{z - u = (k-1)P_{\max} + z'\} \\
S_{ss'}C_{ii'} & \text{if} \quad \{\varphi' = [(\varphi+1) \bmod N_f]\},\{\varphi' = 0\}, \\
& \{\sigma' = [(\sigma + \min[s', S_{\max}]) \bmod T_v]\}, \\
& \{z - u + arr(c') = (k-1)P_{\max} + z'\} \\
0 & \text{else}
\end{cases}$$

(8)

It can be verified that each A_k and B_k, for $k \ge 0$, are irreducible matrices.

Transition probabilities in (5) and (8) can be justified by observing that embedding points are located immediately after the beginning of a SIR++ frame and can be distinguished depending on whether a video frame arrival occurs ($\varphi' = 0$) or does not occur ($\varphi' = 0$) at the embedding point itself.

In the latter case the phase component related to the arrival process must remain unchanged ($c' = c$) and variations in the system state are only due to the number of packets transmitted (i.e., to the number of slots accessed) in the previous SIR++ frame.

On the other hand, if a frame arrival occurs the phase component related to the arrival process varies according to the Markov chain $\{C_n, n \in \mathbb{N}\}$ (with transition probability matrix $\|C_{ij}\|$) and the change in the system state is due both to the number of packet arrivals and to the number of packet transmissions in the previous SIR++ frame.

In any case the maximum number of slots utilized by the the tagged video terminal in the previous SIR++ frame is given by (6). Furthermore, the transition from one state to another state of the Markov chain $Y = \{[L,Z,\Phi,C,S,\Sigma]_{(n)}, n \in \mathbb{N}\}$ occurs with probability different from zero if and only if the variation in the queue length is equal to the difference between the number of packet arrivals and departures .

VII. STABILITY STUDY

From (8), after the following re-arrangement of states $[z,\varphi,c,s,\sigma] \rightarrow [\varphi,c,s,\sigma,z]$, the A_k $(k \ge 0)$ matrices have the following block periodic structure

$$A_k = \begin{bmatrix} 0 & A_k^* & 0 & \cdots & 0 \\ 0 & 0 & A_k^* & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & A_k^* \\ \tilde{A}_k & 0 & 0 & \cdots & 0 \end{bmatrix} \qquad (9)$$

where matrices A_k^* are null for $k > 1$. Specifically, it is

$$A_k^*[(c,s,\sigma,z),(c',s',\sigma',z')] =
\begin{cases}
S_{ss'} & \text{if} \quad \{c' = c\}, \\
& \{\sigma' = [(\sigma + \min[s', S_{\max}]) \bmod T_v]\}, \\
& \{z - u = (k-1)P_{\max} + z'\} \\
0 & \text{else}
\end{cases}$$

(9.a)

$$\tilde{A}_k[(c,s,\sigma,z),(c',s',\sigma',z')] =
\begin{cases}
S_{ss'}C_{ii'} & \text{if} \quad \{\sigma' = [(\sigma + \min[s', S_{\max}]) \bmod T_v]\}, \\
& \{z - u + arr(c') = (k-1)P_{\max} + z'\} \\
0 & \text{else}
\end{cases}$$

(9.b)

It can also be proven that $A = \sum_{h=0}^{\infty} A_h$ is an irreducible and stochastic matrix with the following block periodic structure

$$A = \begin{bmatrix} 0 & A^* & 0 & \cdots & 0 \\ 0 & 0 & A^* & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & A^* \\ \tilde{A} & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{10}$$

The invariant probability vector π of A can be structured as a vector of N_f block components each of which will be denoted by π_i, with $0 \le i \le N_f - 1$. By solving the (finite) system of linear equations

$$\begin{cases} \pi A & = & \pi \\ \pi \mathbf{e} & = & 1 \end{cases}$$

where $\mathbf{e} = [1,1,...,1]^T$, it follows that

$$\begin{cases} \pi_0 & = & \pi_{N_f-1}\tilde{A} & \\ \pi_i & = & \pi_0 \left(A^*\right)^i & 0 < i \le N_f - 1 \end{cases} \tag{11}$$

By using (11) the normalization condition $\sum_{i=0}^{N_f-1} \pi_i \mathbf{e} = 1$ can be written as follows

$$\pi_0 \left[\sum_{i=0}^{N_f-1} \left(A^*\right)^i \right] \mathbf{e} = 1 \tag{12}$$

Since A^* is stochastic

$$\pi_i \mathbf{e} = \frac{1}{N_f} \qquad i = 0,1,...,N_f - 1 \tag{13}$$

Because A is irreducible, from [13] (see page 153) it follows that the system is stable if and only if

$$\pi\beta < 1 \tag{14}$$

where the column vector β is defined as:

$$\beta = \sum_{h=1}^{\infty} h A_h \mathbf{e} \tag{15}$$

Vector β can be structured as a vector of N_f block components each of which will be denoted by β_i, with $0 \le i \le N_f - 1$. Hence, (14) can be rewritten

$$\sum_{i=0}^{N_f-1} \pi_i \beta_i < 1 \tag{16}$$

From (10) and keeping into consideration that for $k > 1$, matrices A_k^* are null, the following partitioning holds

$$\begin{cases} \beta_i & = & A_1^* \mathbf{e} & 0 \le i \le N_f - 2 \\ \beta_{N_f-1} & = & \sum_{h=1}^{\infty} h \tilde{A}_h \mathbf{e} & \end{cases} \tag{17}$$

If we denote $\beta^* = \beta_i$, for $0 \le i \le N_f - 2$, by substituting (17) in (16)

$$\left(\sum_{i=0}^{N_f-2} \pi_i \right) \beta^* + \pi_{N_f-1} \beta_{N_f-1} < 1 \tag{18}$$

Since

$$\beta^* = A_1^* \mathbf{e} = \left(\mathrm{I} - A_0^*\right) \mathbf{e} \tag{19}$$

the first addendum of (18) becomes

$$\left(\sum_{i=0}^{N_f-2} \pi_i \right) \beta^* = \left(\sum_{i=0}^{N_f-2} \pi_i \right) \left(\mathrm{I} - A_0^*\right) \mathbf{e} =$$

$$= \left(\sum_{i=0}^{N_f-2} \pi_i \mathbf{e} \right) - \left(\sum_{i=0}^{N_f-2} \pi_i \right) A_0^* \mathbf{e} = \frac{N_f - 1}{N_f} - \left(\sum_{i=0}^{N_f-2} \pi_i \right) A_0^* \mathbf{e} \tag{20}$$

where the last equality comes out from (13).

The second addendum in (18) can be computed by observing that

$$\beta_{N_f-1} = \sum_{h=1}^{\infty} h\tilde{A}_h \mathbf{e} = \left(\mathrm{I} - \tilde{A}_0\right)\mathbf{e} + \sum_{h=2}^{\infty}(h-1)\tilde{A}_h \mathbf{e} \tag{21}$$

In fact,

$$\pi_{N_f-1}\beta_{N_f-1} = \pi_{N_f-1}\left[\left(\mathrm{I} - \tilde{A}_0\right)\mathbf{e} + \sum_{h=2}^{\infty}(h-1)\tilde{A}_h \mathbf{e}\right] =$$

$$= \frac{1}{N_f} - \pi_{N_f-1}\left(\tilde{A}_0\mathbf{e}\right) + \pi_{N_f-1}\left[\sum_{h=2}^{\infty}(h-1)\tilde{A}_h \mathbf{e}\right] \tag{22}$$

By substituting (20) and (22) into (18) the following inequality holds

$$-\left(\sum_{i=0}^{N_f-2}\pi_i\right)A_0^*\mathbf{e} - \pi_{N_f-1}\left(\tilde{A}_0\mathbf{e}\right) + \pi_{N_f-1}\left[\sum_{h=2}^{\infty}(h-1)\tilde{A}_h \mathbf{e}\right] < 0 \tag{23}$$

$\left(\sum_{i=0}^{N_f-2}\pi_i\right)\left(A_0^*\mathbf{e}\right)$ represents the average number of guaranteed and high-priority-available-slots in a superframe, with the exception of the SIR++ frame in which a video-frame arrives.

The term

$$-\pi_{N_f-1}\left(\tilde{A}_0\mathbf{e}\right) + \pi_{N_f-1}\left[\sum_{h=2}^{\infty}(h-1)\tilde{A}_h \mathbf{e}\right] \tag{24}$$

is related to the SIR++ frame at the beginning of which a video-frame arrival occurs. By following the above line of reasoning it can be verified that this term gives the difference between the average number of packet arrivals and the number of guaranteed and high-priority-available-slots the tagged video terminal can access in this SIR++ frame.

Therefore, inequality (23) has a very intuitive interpretation as it states that the system is stable if and only if the mean number of packet arrivals in a superframe is less than the average number of guaranteed and high-priority-available-slots which can be accessed by the tagged video terminal in a superframe. This condition is assumed to hold throughout the paper.

VIII. SOLUTION OF THE MARKOV CHAIN

Following the standard methodology for the solution of an M/G/1-type Markov chain, our first step was to calculate the G matrix. A general technique to compute the G matrix relies on the following recursive algorithm [13]::

$$G = \lim_{n\to\infty} G_n$$

where

$$G_n = \begin{cases} A_0 & n = 0 \\ \sum_{j=0}^{\infty} A_j G_{n-1}^j & n > 0 \end{cases} \tag{25}$$

Because of the size of the A_k $(k \geq 0)$, this algorithm is computationally demanding. Thus, in order to solve (25) we used the algorithm described in [20] based on the recursive state reduction which provides a quadratic convergence, has a low computational cost and is numerically stable. For more details see ([20], [21]).

After computing the G matrix, we calculate the stationary probability vector $\mathbf{x}$ of the transition probability matrix P. By partitioning $\mathbf{x}$ in a manner congruent to the partition of matrix P we have $\mathbf{x} = [\mathbf{x_0}, \mathbf{x_1}, \mathbf{x_2}\ldots]$. To derive the subvector $\mathbf{x_0}$ we applied the Latouche algorithm ([22]) which requires the solution of the following linear system

$$\begin{cases} \mathbf{x_0} = \mathbf{x_0}K \\ \mathbf{x_0}\kappa = 1 \end{cases} \tag{26}$$

where

$$K = \sum_{h=0}^{\infty} B_h G^h \tag{27}$$

$$\kappa = \mathbf{e} + \sum_{i\geq 1} B_i \sum_{0\leq k\leq i-1} G^k \mu$$

$$\mu = \left(I - \sum_{i\geq 1} A_i \sum_{0\leq k\leq i-1} G^k\right)^{-1}\mathbf{e} \tag{28}$$

All the other components of vector $\mathbf{x}$ are obtained by applying the Ramaswami algorithm ([13], [23])

$$\mathbf{x_i} = \left[\mathbf{x_0} \overline{B}_i + \sum_{j=1}^{i-1} \mathbf{x_j} \overline{A}_{i+1-j} \right] \left(I - \overline{A}_1 \right)^{-1} \qquad i \geq 1 \tag{29}$$

where

$$\overline{A}_n = \sum_{i=n}^{\infty} A_i G^{i-n} \ , \ \overline{B}_n = \sum_{i=n}^{\infty} B_i G^{i-n} \qquad n \geq 0 \tag{30}$$

It must be observed that $\mathbf{x}_i = P\{L = i, Z = z, \Phi = \varphi, C = c, S = s, \Sigma = \sigma\}$ provides the steady state probability of being at level i at an embedding point, with

$$i \in N, \ 0 \leq z < P_{\max}, \ 0 \leq \varphi < N_f,$$

$$0 \leq c \leq V_{\max}, \ 0 \leq s < S_{\dim}, \ 0 \leq \sigma < N_v$$

The distribution of the buffer occupancy after a video-frame arrival can be easily computed through the following relation

$$\pi(i,z) = \frac{\sum_c \sum_s \sum_\sigma P\{L = i, Z = z, \Phi = 0, C = c, S = s, \Sigma = \sigma\}}{\sum_l \sum_z \sum_c \sum_s \sum_\sigma P\{L = l, Z = z, \Phi = 0, C = c, S = s, \Sigma = \sigma\}}$$

(31)

and by observing that $Q = L \times P_{\max} + Z$.

IX. PERFORMANCE ANALYSIS

The system parameters for a microcellular environment are inspired from [9], hence the transmission rate is 1.8 Mbit/s and each SIR++ frame is 5 ms long and each frame has 72 slots. Further, it is assumed that cells are provided with one carrier.

In our analysis the system parameters for the speech traffic have been chosen in order to operate in the stability region, [11]. Specifically, we selected 4 R-slots with $Class = 0$ per SIR++ frame used by speech sessions to notify their bandwidth requests and a permission probability $p = 0.1$. The maximum number of speech sessions $(T_{s\_\max})$ which guarantees stability and $P_{drop} = 0.01$ for several N_s values has been evaluated via simulative analysis and results are summarized in Table 3 .

Table 3: Maximum number of speech sessions guaranteeing stability and $P_{drop} = 0.01$ for several N_s values.

N_s	T_{s_max}
60	128
62	133
63	135
64	137
65	140

For each N_s value in **Table 3** via a simulative analysis and by using the approach shown in Section V we estimated the transition probability matrix of the discrete-time, discrete-state Markov chain $\{S_n, n \in \mathrm{N}\}$. From the results obtained it was easy to verify that conditions for lumpability are met for $S_{\dim} = 3$. Furthermore, the number of states of the discrete-time, discrete-state Markov chain $\{C_n, n \in \mathrm{N}\}$, which represents the behaviour of each video source, was chosen equal to four, i.e., $V_{\max} = 3$.

The methodology adopted to analyze the performance of SIR++ in handling video traffic is as follows. We assess SIR++ starting from a basic scenario Sc_0 (**Fig. 4**). Variations are then introduced to estimate the sensitivity of SIR++ to different loads and operating conditions. **Fig. 4** shows the relationship between the assessment scenarios. For each scenario the buffer occupancy distribution of the tagged video terminal is derived.

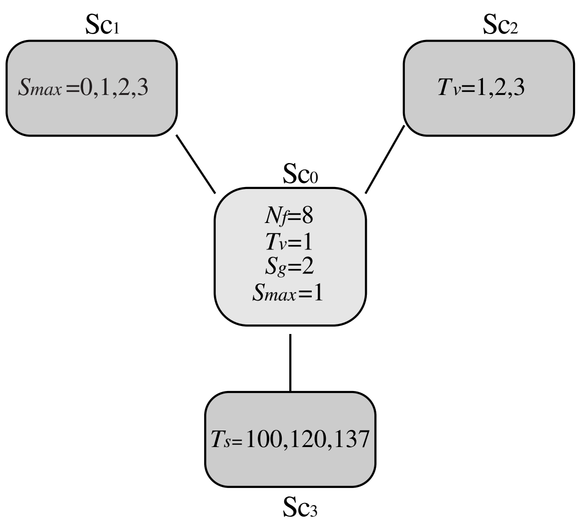

Figure 4: Assessment scenarios.

Since the analytic model proposed in the paper is approximate (the number of slots unused by speech is modeled by a Markov chain whose transition probability matrix $\|S_{hk}\|$ is assumed *exactly lumpable*), simulation is used to check how accurate it is.

Fig. 5 shows the percentiles of the buffer occupancy distribution after video frame arrivals for the basic scenario (Sc_0) as obtained with the analytic and simulative approach. From the figure it can be appreciated that the approximation introduced by the lumpability assumption is negligible. Furthermore, the influence of the Markov chain assumption is such that simulative and analytic results start diverging only for a queue length of approximately 30 packets, corresponding to a probability value in the order of $1-10^{-3}$, thus justifying the assumption. Similar results (not shown) have been obtained for all the scenarios considered.

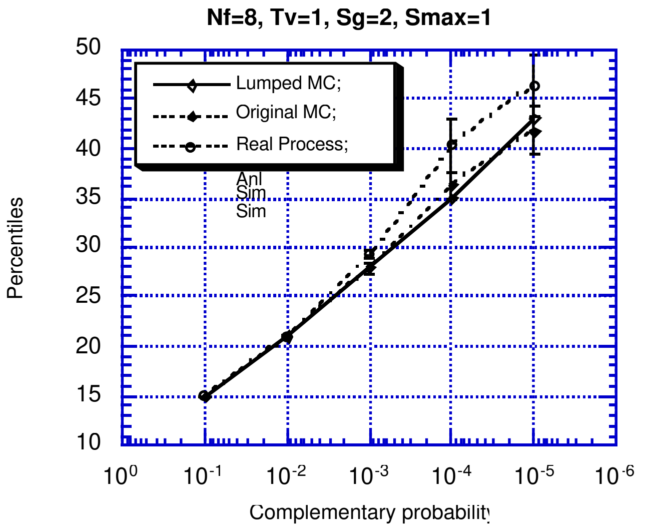

Figure 5: Accuracy assessment for the basic scenario.

Fig. 6 shows, for scenario Sc_1, the buffer occupancy distribution after video frame arrivals, for $S_{\max}=0,1,2,3$. Clearly, $S_{\max}=0$ corresponds to the case in which the tagged video terminal transmits packets only in the slot guaranteed to it in a SIR++ frame (i.e., the system operates as a pure TDM system). **Fig. 6** highlights the remarkable improvement of the system when the number of high-priority-available-slots changes from 0 to 1. A further increase in the number of high-priority-available-slots beyond 1, does not produce any noticeable improvement. This can be easily justified by observing that in this scenario, a bandwidth allocation to the tagged video terminal with $S_g=2$ and $S_{\max}=1$ is very close to its peak bit rate.

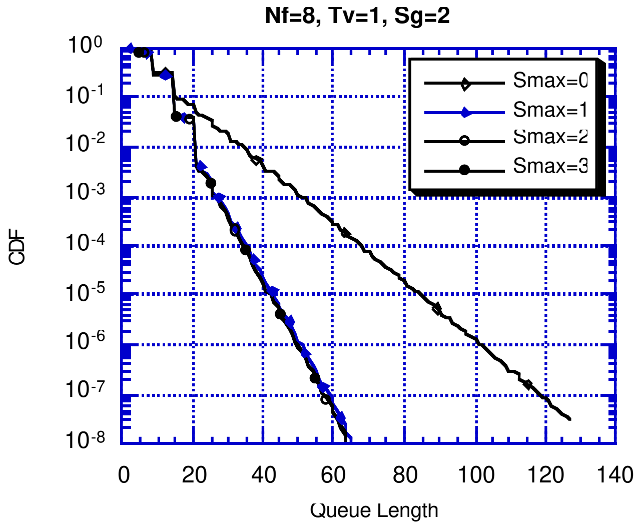

Figure 6: Influence of the $S_{\max}$ parameter.

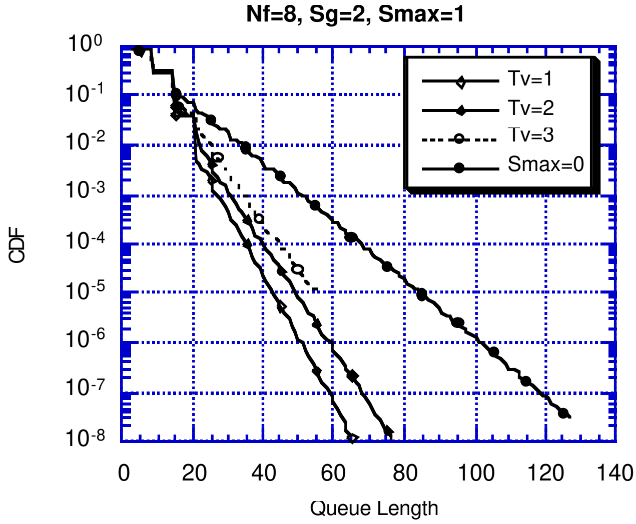

Figure 7: Influence of the number of video sessions.

Fig. 7 shows, for scenario Sc_2, the buffer occupancy distribution after video frame arrivals, for $T_v=1,2,3$. Since there is only one high-priority-available-slot (i.e., $S_{\max}=1$), by increasing the number of video sessions, the total bandwidth achieved by each session decreases. Hence, the buffer length of the tagged video terminal grows. For comparison's sake, **Fig. 7** also contrasts this case with the case $S_{\max}=0$ already discussed in the previous scenario.

To further deepen our analysis when $T_v=2$ we also consider the case in which $S_{\max}=2$, i.e., each video session can utilize in a SIR++ frame, at maximum, one high-priority-available-slot. Results are shown in **Fig. 8** from which it can be deduced that the setting $S_{\max}=2$, performs better than $S_{\max}=1$.

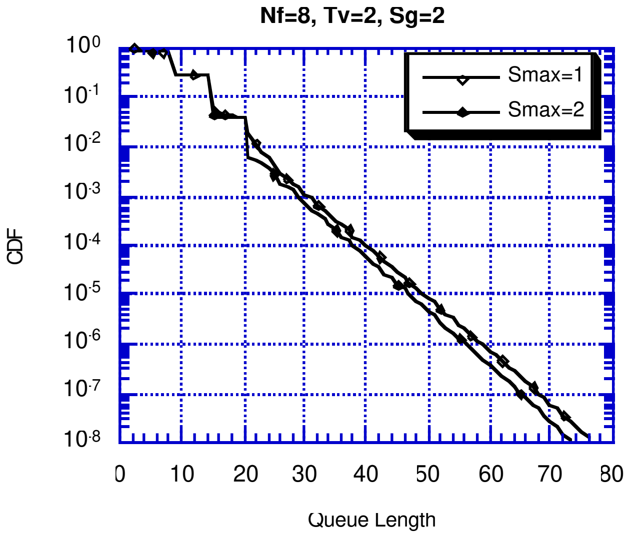

Figure 8: Influence of $S_{\max}$ when $T_v = 2$.

With the worst case approach we evaluate the performance of the tagged video terminal when the number of speech sessions is at its maximum. However, in order to asses the influence of the speech load, we also evaluate the performance of the system when the number of speech sessions is lower than the allowed maximum.

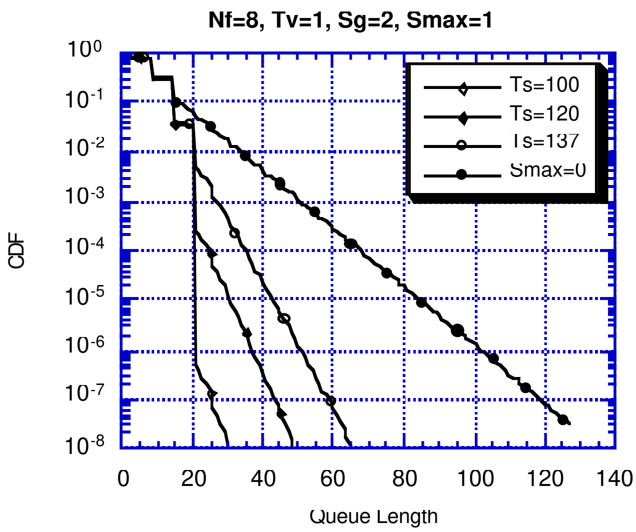

Figure 9: Influence of the number of speech sessions.

Fig. 9 shows, for scenario S_3, the buffer occupancy distribution after video frame arrivals, for $T_s = 100,\ 120,\ 137$. Clearly, the buffer length of the tagged video terminal decreases when the number of speech sessions decreases. This can be easily justified by considering that the probability of having one high-priority-available-slot in a SIR++ frame grows when the number of speech sessions decreases.

We further deepened the analysis by exploiting the H.263 feature of varying the number of SIR++ frames between two video frame arrivals, i.e., the value of the N_f parameter. Specifically, instead of a frame every 40 msec ($N_f = 8$) we assume a coding mode resulting in a frame every 80 msec ($N_f = 16$). Of course, this results in a degradation of picture quality but, on the other hand, has the advantage of less required bandwidth. As an example, a single slot per SIR++ frame is sufficient to guarantee a bandwidth higher than the average bit rate for the sequence "grandma", [16], when $N_f = 16$.

Fig. 10 shows the buffer occupancy distribution after video frame arrivals for $N_f = 16$. This figure relates to the case in which the number of reserved slots per SIR++ frame to the tagged video terminal is equal to one. This setting considers that, for $N_f = 16$, the average video bit rate is less than one slot per SIR++ frame, whereas a rate of two slots per SIR++ frame is higher than the peak bit rate. From Fig. 10 it can be realized the remarkable improvement introduced by the utilization of one high-priority-available-slot. Note that in this case the tagged video terminal can rely on a bandwidth which is close to the peak rate. Hence, the usage of two high-priority-available-slots does not introduce any significant improvement.

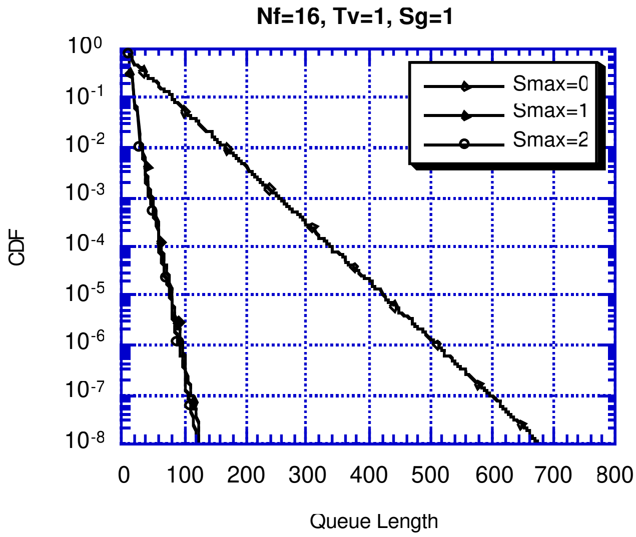

Figure 10: Influence of the N_f parameter.

X. CONCLUSIONS

The access protocol SIR++ is designed to address aspects which are considered of great importance in advanced mobile systems, i. e. accommodating the characteristics (requirements) of a variety of traffic classes (services); and, guaranteeing efficient use of bandwidth. The key concepts on which SIR++ is based are the provision of a minimum guaranteed bandwidth complemented with "best effort" bandwidth, and priority for servicing traffic classes with different throughput/delay constraints. The latter feature is enabled by exploiting bandwidth not used by intermittent traffic sources, typically speech, and employing collision-free, Round-Robin, mechanisms for allocating "best effort" bandwidth.

The potential of SIR++ has been investigated by considering a mix of speech, low bitrate (H.263) video, and data, where video is provided extra ("best effort") bandwidth in addition to a guaranteed minimum. The results obtained show that under such conditions there is a remarkable improvement in the performance for video applications even when the number of guaranteed slots is very low.

The paper has addressed operation of the protocol under ideal propagation conditions and synchronization/timing. Further study is ongoing to assess the impact of realistic transmission environment and the influence of variations in the use of in-band signaling.

REFERENCES

[1] M. Callendar, "Future Public Land Mobile Telecommunication Systems", IEEE Personal Communications, Vol. 1, No. 4, pp. 18-22, Fourth Quarter 1994.

[2] J. Rapeli, "UMTS - Targets, System Concept, and Standardization Activity in a Global Framework", IEEE Personal Communications Magazine, Vol. 2, No. 1, pp. 20-28, February 1995.

[3] ITU-R Recommendation M.1034, “Requirements for the radio interface(s) for Future Public Land Mobile Telecommunication Systems (FPLMTS)”.

[4] ITU-R Recommendation M.1035, “Framework for the radio interface(s) and radio sub-system functionality for Future Public Land Mobile Telecommunication Systems (FPLMTS)".

[5] D. J. Goodman, R. A. Valenzuela, K. T. Gaylard, and B. Ramamurthi, "Packet Reservation Multiple Access for Local Wireless Communications", *IEEE Trans. Commun.*, vol. COM-37, pp. 885-890, August 1989.

[6] S. Nanda, D. J. Goodman, and U. Timor "Performance of PRMA: A Packet Voice Protocol for Cellular Systems", *IEEE Trans. Veh. Tech.*, vol. VT-40,, pp. 584-598, August 1991.

[7] J. De Vile, "A Reservation Multiple Access Scheme for an Adaptive TDMA Air Interface", in *Proc. of Fourth WINLAB Workshop on Third Generation Wireless Information Networks*, New Jersey, USA, Oct. 1993.

[8] A. Urie, M. L. Streeton, and C. Mourot, "An Advanced TDMA Mobile Access System for UMTS", *IEEE Personal Communications*, vol. 2, pp. 38-47, February 1995.

[9] J. Dunlop, J. Irvine, D. Robertson, and P. Cosimini, "Performance of Statistically Multiplexed Access Mechanisms for a TDMA Radio Interface", *IEEE Personal Communications*, vol. 2, pp. 56-64, June 1995.

[10] G. Anastasi, D. Grillo, L. Lenzini, and E. Mingozzi, "A Bandwidth Reservation Protocol for Speech/data Integration in TDMA-Based Advanced Mobile Systems, in Proc. *INFOCOM '96*, San Francisco, 1996.

[11] G. Anastasi, D. Grillo, L. Lenzini, E. Mingozzi, "A Contention/Reservation Access Protocol for Speech and Data Integration in TDMA-Based Advanced Mobile Systems ", to appear in ACM Wireless Networks, Special Issue on Channel Access in Wireless Networks.

[12] ITU-T, “Video Coding for Low Bitrate Communication”, Draft ITU-T Recommendation H.263, December 5, 1995.

[13] M. F. Neuts, "Structured Stochastic Matrices of M/G/1 Type and Their Applications", Marcel Dekker, Inc., 1989.

[14] L. Lenzini, B. Meini, and E. Mingozzi, “An Approximate PRMA++ Performance and Stability Analysis”, submitted for publication.

[15] D.S. Kim, R.L. Smith, “ An Exact Aggregation-Disaggregation Algorithm for Mandatory Set Decomposable Markov Chain”, in Numerical Solution of Markov Chains, Marcel Dekker, Inc., 1991, pp. 89-103.

[16] ftp://bonde.nta.no/pub/tmn/qcif_source/.

[17] `http://www.nta.no/brukere/DVC/.`

[18] D. P. Heyman, A. Tabatabai, T. V. Lakshman, “Statistical Analysis and Simulation Study of Video Teleconference Traffic in ATM Networks”, IEEE Transaction on Circuit and Systems for Video Technology, Vol. 2, No. 1, March 1992.

[19] U. Narayan Bhat, “Elements of Applied Stochastic Processes”, John Wiley & Sons, 1984, Second Edition.

[20] D. Bini, B. Meini, "On the Solution of a Non-linear Matrix Equation Arising in Queuing Problems", SIAM J. Matrix Analysis and Applications, 17,4 pp. 906--926, 1996

[21] D. Bini, B. Meini, "On Cyclic Reduction Applied to a Class of Toeplitz-like Matrices arising in queuing Problems", in Proceedings of the Second International Workshop on Numerical Solution of Markov Chains, Raleigh, North Carolina, 1995, pp. 21-38.

[22] G. Latouche, "Newton's Iteration for Non-linear Equations in Markov Chains", IMA J. of Numerical Analysis, 1994 (14), pp 583-598.

[23] V. Ramaswami "A Stable Recursion for the Steady State Vector in Markov Chains of M/G/l type", Stochastic Models, 4, pp. 183-188, 1988.

Giuseppe Anastasi received the Laurea degree in Electronic Engineering and the PhD degree in Computer Engineering, both from the University of Pisa, Italy, in 1990 and 1995, respectively. In 1991 he joined the Department of Information Engineering of the University of Pisa where he is currently assistant professor of Computer Engineering. His main fields of interest include wireless and mobile networking, high speed networking, service integration, modeling and performance evaluation of computer networks. He is a member of the IEEE Computer Society.
Email: anastasi@iet.unipi.it

Davide Grillo received a degree in Statistics from the University of Rome in 1965. In the same year he joined the Fondazione Ugo Bordoni, Rome, Italy, where he is currently manager, Personal Communications. He was a Visiting Scientist at the Siemens AG Central Laboratory, Munich, and the University of Dortmund (Germany), and the IBM Zurich Research Laboratory (Switzerland). His areas of research have included: telephone network operation and control; switching exchange architecture; packet switching networks; LAN/MAN architecture, interconnection, and control; and resource allocation strategies in the radio sub-system of mobile networks. He has been on the program committees of numerous conferences and workshops, and served as chairman for the 8th ITC Seminar on Universal Personal Telecommunication in 1992. He was a Guest Editor of an issue on LAN interconnection of the IEEE JSAC, a special issue of PERFORMANCE EVALUATION dedicated to high-speed telecommunication systems, and a special issue of IEEE Personal Communications on the European research and standardization activities for advanced mobile systems. He has been an active member of SG2 of the ITU-T for questions on traffic engineering of telephone networks and ISDN, where, since 1989, he has been leading a group concerned with traffic engineering of networks supporting mobile and UPT services. He is involved in TG 8/1 (FPLMTS) of the ITU-R, where he acts as Liaison Rapporteur to ITU-T SG2. He has been involved in the RACE projects on mobile communications (RACE I 1043, and RACE II MONET and ATDMA) with responsibility in the co-ordination of traffic performance modelling activities.
He is currently on the Editorial Board of *IEEE Personal Communications* .
He is a member of the IEEE Communications Society.
E-mail: grillo@fub.it

Luciano Lenzini holds a degree in Physics from the University of Pisa, Italy. He joined CNUCE, an institute of the Italian National Research Council (CNR) in 1970. Starting in 1973, he spent a year and a half at the IBM Scientific Centre in Cambridge, Massachusetts, working on computer networks. He has since directed several national and international projects including: RPCNET, the first Italian packet switching network; STELLA, the first European broadcasting satellite network; and OSIRIDE, the first Italian OSI network. His current research interests include integrated service networks, the design and performance evaluation of both Metropolitan Area Network MAC protocols and packet-based radio access mechanisms for third generation mobile systems. He is author or co-author of numerous publications in these fields. He is currently on the Editorial Board of *Computer Networks and ISDN Systems.* He has been on the program committees of numerous conferences and workshops, and served as chairman for the 1992 IEEE Workshop on Metropolitan Area Networks. At present he is a Full Professor at the Department of Information Engineering of the University of Pisa.
E-mail: lenzini@iet.unipi.it