

# A demand-assignment algorithm based on a Markov modulated chain prediction model for satellite bandwidth allocation

Francesco Delli Priscoli · Dario Pompili

Published online: 8 February 2008  
© Springer Science+Business Media, LLC 2008

**Abstract** This article deals with the problem of the design of a control-based demand-assignment algorithm for a satellite access network using a Markov modulated chain traffic prediction model. The objective is to guarantee a target Quality of Service (QoS) to Internet traffic, while efficiently exploiting the air interface. The proposed algorithm is in charge of dynamically partitioning the uplink bandwidth capacity in a satellite spotbeam among the in-progress connections. Such partition is performed aiming at matching the QoS requirements of each connection and maximizing the satellite bandwidth exploitation. A closed-loop Control Theory approach is adopted to efficiently tackle the problem of the delay between bandwidth requests and bandwidth assignments, while minimizing the signaling overhead caused by control messages. The algorithm efficiently copes with both the satellite propagation delay and the delays inherent in the periodic nature of the bandwidth request mechanism. The proposed demand-assignment algorithm and Markov chain traffic prediction model are shown to improve the overall satellite network performance through extensive simulation experiments.

**Keywords** Satellite networks · Internet access · Control theory · Quality of service · IP traffic prediction

---

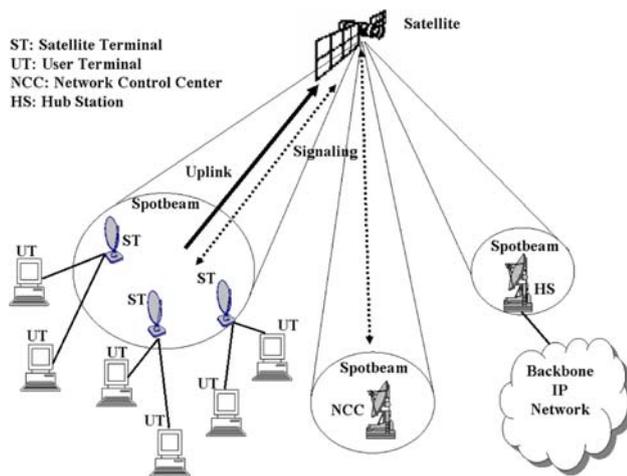
F. Delli Priscoli  
System Engineering Department, University of Rome  
“La Sapienza”, Rome, Italy

D. Pompili (✉)  
Electrical and Computer Engineering Department, Rutgers,  
The State University of New Jersey,  
Piscataway, NJ 08854, USA  
e-mail: pompili@ece.rutgers.edu

## 1 Introduction and background

High powered direct broadcast television satellites, using the European Digital Video Broadcast (DVB) standard, can be used to broadcast high volumes of data directly to home terminals. Currently, these are unidirectional transmission channels that do not allow an interaction between service providers and users. There are several ways to design a return channel for satellite broadcast/multicast services. Many believe terrestrial return channels to be the most cost-effective and practical solution. Commonly proposed terrestrial return channels are PSTN, ISDN, xDSL, and GSM/GPRS. However, there is a large world-wide interest for a DVB Return Channel via Satellite (DVB-RCS) [1], which could be particularly suitable to support a large amount of non real-time return connections provided that an appropriate bandwidth management mechanism be designed.

In this article, we propose a dynamic bandwidth management mechanism for an efficient and flexible partition of the uplink bandwidth capacity among the in-progress connections in a satellite spotbeam. This capacity is equal to the sum of the uplink carrier capacities assigned to such spotbeam. This partition is performed aiming at matching the Quality of Service (QoS) requirements of each connection and maximizing the satellite bandwidth exploitation. A Control Theory approach, which relies on a Markov Modulated Poisson Process (MMPP) model for IP traffic prediction, is adopted to efficiently tackle the problem of the delay between bandwidth requests and bandwidth assignments, while minimizing the signaling overhead caused by control messages. The proposed mechanism is fully compliant with the DVB-RCS standard [1]. In the following, we will refer to the terms *downstream* and *upstream* to indicate the traffic flowing from the



**Fig. 1** Satellite access network

Internet to a Satellite Terminal (ST) and from a ST to the Internet, respectively.

The considered Internet access scenario is outlined in Fig. 1, which shows a layer-2 switch satellite system including a Hub Station (HS), a certain number of Satellite Terminals (STs), and a Network Control Center (NCC), which is in charge of performing traffic control tasks. In the return direction, each ST can manage one or more upstream(s) coming from the User Terminals (UTs). Each upstream is relevant to a different connection involving a UT and entails the transmission of IP datagrams (hereon referred to as packets) from the considered UT, via the associated ST and the satellite, to the HS and the backbone IP network. In particular, in this scenario (see Fig. 1), an on-board packet switch is in charge of addressing packets towards a downlink carrier assigned to the destination spotbeam. However, the problems related to possible congestion of the downlink carriers, as well as those related to the Connection Admission Control (CAC), are outside the scope of this article. As described in [2], in fact, downlink and uplink management procedures can be decoupled, and both procedures have to provide appropriate inputs to the CAC. The interested reader is referred to [3] in order to have further details on such satellite system. Throughout this article, we will assume that each connection has its own specific set of requirements (e.g., minimum required bandwidth, maximum transfer delay, delay jitter, etc.) that are specified in the QoS contract established at connection setup [4, 5], as will be further described in Sect. 2.

In order to enhance the exploitation of the valuable satellite capacity, connections that have not very stringent delay requirements are not fixedly assigned uplink bandwidth portions. For such kind of connections, uplink bandwidth has to be managed according to a *demand-assignment* mechanism in which the STs periodically ask

the NCC for the *temporary* assignment of a certain portion of the bandwidth. Then, the NCC decides how the available uplink bandwidth should be optimally partitioned, while considering the QoS requirements of the connections asking for bandwidth assignment. Finally, the NCC communicates the relevant decisions to the STs.

Key problem of such a mechanism is that bandwidth assignments are received by STs fractions of second after bandwidth requests. This is caused by the high propagation delays of satellite networks. In particular, in Geostationary Earth Orbit (GEO) satellite systems [6] the one-way propagation delay is approximately equal to 250 ms. In addition, in order to keep the signaling overhead limited, a minimum time must elapse between two consecutive bandwidth requests sent by a ST. This issue can cause further delay in data transfer (thus increasing the unavoidable satellite propagation delay) as well as ST buffer overflows. A possible, although trivial, solution to such problem could be for a ST to request more bandwidth than that actually necessary. However, by so doing, there would be the risk of the so-called *over-assignment problem*, i.e., the assignment to a certain ST of capacity not actually necessary that could be, possibly, subtracted from other STs actually needing it.

This article copes with the above-mentioned problem by designing an innovative closed-loop demand-assignment mechanism that relies on an IP traffic prediction model based on a Markov Modulated Poisson Process (MMPP). The objective of this algorithm is to avoid further delays in data transfer and guarantee an efficient exploitation of the uplink satellite bandwidth. Although other papers have considered the Demand Assigned Multiple Access (DAMA) problem ([7–9]), up to the authors' knowledge, aside from works by the same authors (e.g., [10–12]), only few papers have addressed the demand-assignment problem with QoS guarantees in systems subject to delays (e.g., [13, 14]).

The remain of the article is organized as follows. Section 2 introduces the basic concepts utilized in the paper as well as the objectives of the capacity demand-assignment algorithm, while Sect. 3 presents the Satellite Terminal (ST) architecture. Section 4 describes the proposed demand-assignment procedure, and Sect. 5 details the Markov modulated Poisson process algorithm for IP traffic prediction. Finally, Sect. 6 shows numerical results while Sect. 7 concludes the article.

## 2 Basic definitions and QoS contract

In this section, we introduce some basic definitions that will be used throughout this article. For the sake of brevity, by 'uplink' we mean the 'return uplink', i.e., the link from

the ST to the satellite. Conversely, by ‘return’ traffic (or packets) we indicate the traffic (or packets) originated by UTs and directed to the HS via STs and the satellite. Let  $S$  denote the number of different STs in the considered spotbeam, where  $i = 1, 2, \dots, S$  identifies a generic ST with at least one in-progress connection. Let  $C(i)$  denote the number of different uplink connections simultaneously in progress involving the  $i$ th ST, and pair  $(i, j)$  denote a generic in-progress uplink connection involving the same ST, where  $j$  can assume values in the range  $[1, C(i)]$ .

Throughout this article, computation of the various variables will not be performed at any time  $t$ , but only at discrete-time instants  $t_h$  ( $h = 1, 2, \dots$ ) periodically occurring with a proper period  $T_{short}$  [s], i.e.,  $t_{h+1} = t_h + T_{short}$ . In the following, for the sake of notation simplicity, we will refer to these instants  $t_h$  as  $h$ . Moreover, by the  $h$ th time interval, we will refer to time interval  $[h, h + 1]$ <sup>1</sup>. Last but not least, let  $R_{ij}^{in}(h)$  [bps] denote the bit rate of the return traffic relevant to connection  $(i, j)$  that, at time  $h$ , is offered to the  $i$ th ST. Such bit rate is computed during a monitoring period as,

$$R_{ij}^{in}(h) = \frac{\sum_{k=h-M}^h L_{ij}^{in}(k)}{M \cdot T_{short}}, \tag{1}$$

where  $M$  is the duration of the monitoring period, expressed in number of discrete-time instants (i.e.,  $M \cdot T_{short}$  is the monitoring period duration, and  $L_{ij}^{in}(k)$  [bit] is the sum of the bit lengths of the return packets, relevant to connection  $(i, j)$ , that are incoming into the  $i$ th ST during the  $k$ th time interval.

Let us now introduce two important definitions relevant to a generic connection  $(i, j)$ .

**Definition 1** Let  $D_{ij}$  [s] denote the queuing delay that a packet relevant to connection  $(i, j)$  experiences from the instant at which it arrives at the ST (coming from a UT) to the instant at which it may be forwarded towards the uplink air interface.

**Definition 2** Let  $R_{ij}^{av}$  [bps] denote the average throughput of connection  $(i, j)$ , which is defined as the ratio of the number of bits transmitted during the connection lifetime and the connection duration.

The QoS requirements that have to be guaranteed to a connection are specified in a QoS contract [4, 5] that is established at connection setup. In particular, the QoS contract relevant to connection  $(i, j)$  may include the following requirements:

- (1) *Throughput QoS requirement* A first QoS requirement concerns the definition of the throughput to be

<sup>1</sup> For example, the generic expression  $A(h)$  will refer to the value of  $A$  in the  $h$ th time interval, i.e., in the interval  $[h, h + 1]$ , while  $B[\eta, k]$  will refer to the evolution of  $B$  in the interval  $[\eta, k]$ , where  $k$  may be greater than  $\eta + 1$ .

guaranteed to connection  $(i, j)$ , i.e., the so-called *Static Bit Rate*,  $R_{ij}^{static}$  [bps]. Note that  $R_{ij}^{static}$  can vary as time varies. An appropriate CAC procedure assures that the *Static Bit Rates* of in-progress connections satisfy the following constraint,

$$\sum_{i=1}^S \sum_{j=1}^{C(i)} R_{ij}^{static}[\eta, k] \leq R_{up}^{tot}, \quad t_0 \leq \eta < k \leq t_f, \tag{2}$$

where  $R_{up}^{tot}$  [bps] is the overall uplink capacity available in the considered spotbeam, and  $t_0$  [s] and  $t_f$  [s] are the starting and finishing times of the considered connections, respectively.

- (2) *Delay QoS requirement* If connection  $(i, j)$  is associated with a real-time application (voice, video-conference, etc.), a second fundamental QoS requirement concerns the maximum transfer delay, hereafter indicated as  $D_{ij}^{max}$  [s], that can be tolerated by connection  $(i, j)$ . This means that, in general, the queuing delay  $D_{ij}$  should not exceed  $D_{ij}^{max}$ . As a matter of fact, in case  $D_{ij}$  exceed  $D_{ij}^{max}$ , packets of connection  $(i, j)$  would be discarded by the ST. In this respect, for real-time connections, a small amount of packet loss can be tolerated, according to the specific real-time application, even though such a loss should be minimized.

In light of the above, it should be clear that the QoS contract relevant to a real-time connection  $(i, j)$  is characterized by both  $R_{ij}^{static}$  and  $D_{ij}^{max}$ , while the QoS contract relevant to a non real-time connection  $(i, j)$  is characterized by only  $R_{ij}^{static}$ .

The algorithm proposed in this article aims at minimizing the real-time traffic to be discarded because it has waited more than the maximum-tolerated delay (*Delay QoS Requirement*) and at maximizing the average throughput of non real-time traffic (*Throughput QoS Requirement*). We assume that packet expiration due to the overcome of  $D_{ij}^{max}$  be the only possible traffic loss, i.e., no queue overflows can occur. In other words, we make the assumption that each queue is dimensioned in such a way as to accept all possible coming packets. This way, even under heavy traffic conditions, *time-critical traffic* will not suffer from poor QoS due to queue overflows.

To this end, the  $i$ th ST is assigned with a semi-permanently *Static Bit Rate* that is equal to the sum of the static bit rates  $R_{ij}^{static}$  relevant to the connections in progress at such ST. In addition, it is granted a *Dynamic Bit Rate* that is temporarily assigned by the NCC on the ground of requests from the ST, according to an appropriate demand-assignment mechanism, which is detailed in Sect. 4.

Let  $R_{up}^{dyn}$  [bps] denote the available dynamic uplink capacity, which is defined as the uplink capacity relevant to

the considered spotbeam that is not statically assigned. Such a capacity, which can be dynamically assigned, is

$$R_{up}^{dyn}[\eta, k] = R_{up}^{tot} - \sum_{i=1}^S \sum_{j=1}^{C(i)} R_{ij}^{static}[\eta, k], \quad t_0 \leq \eta < k \leq t_f. \tag{3}$$

Specifically, let  $R_{ij}^{dyn}[\eta, k]$  denote the *Dynamic Bit Rate* assigned to connection  $(i, j)$  during time interval  $[\eta, k]$ . Clearly, for any time interval, the following uplink capacity constraint must be respected,

$$\sum_{i=1}^S \sum_{j=1}^{C(i)} R_{ij}^{dyn}[\eta, k] \leq R_{up}^{dyn}[\eta, k], \quad t_0 \leq \eta < k \leq t_f. \tag{4}$$

### 3 Satellite terminal architecture

Figure 2 shows the internal structure of a Satellite Terminal (ST). In particular, the  $i$ th ST is provided with a set of  $C(i)$  *FIFO buffers* (First In First Out). Each of these buffers stores the packets (waiting for being transmitted on the uplink channel) of one of the uplink connections the ST in question is involved in. A *Classifier*, which is fed with the traffic coming from the User Terminals (UTs) linked to the  $i$ th ST, is in charge of sorting the packets towards the  $C(i)$  FIFO buffers. In the following, for the sake of brevity, the FIFO buffer storing the packets relevant to connection  $(i, j)$  will be referred to as queue  $(i, j)$ . Let  $q_{ij}(h)$  [bit] denote the number of bits stored in queue  $(i, j)$  in the  $h$ th time interval. Let  $\delta_{ij}^{ass}[\eta, k] \in [0, 1]$  denote the fraction of the available dynamic uplink capacity  $R_{up}^{dyn}$  [bps] granted by the NCC to connection  $(i, j)$  during time interval  $[\eta, k]$ , i.e.,  $\delta_{ij}^{ass}(\eta, k) \cdot R_{up}^{dyn}$  represents the *Dynamic Bit Rate* at which, during time interval  $[\eta, k]$ , the  $i$ th ST is allowed to forward packets, relevant to connection  $(i, j)$ , towards the uplink air

interface. For any time interval  $[\eta, k] \in [t_0, t_f]$ , the following fundamental uplink capacity constraint, which is equivalent to (4), must be respected,

$$\sum_{i=1}^S \sum_{j=1}^{C(i)} \delta_{ij}^{ass}(\eta, k) \leq 1, \quad t_0 \leq \eta < k \leq t_f. \tag{5}$$

Packets stored in queue  $(i, j)$  can be either forwarded over the uplink air interface by the *Multiplexer* or, if they are relevant to real-time connections, discarded because they are expired, i.e., they have waited more than their maximum-tolerated delay  $D_{ij}^{max}$ .

### 4 Capacity demand-assignment procedure

In this section, we describe the proposed capacity demand-assignment procedure, whose objective is to efficiently and fairly share the available bandwidth of satellite uplink channels among STs. In particular, in Sect. 4.1 we introduce some definitions; in Sect. 4.2, we describe the centralized approach of the proposed procedure, and detail the key parameters sent by STs to the NCC to perform appropriate bandwidth assignments; in Sect. 4.3, we describe the details of the capacity demand-assignment procedure, while in Sect. 4.4 we synthetically summarize the main steps the procedure goes through.

#### 4.1 Definitions

Let us introduce the following definitions (see also Fig. 3):

- Let  $L$  denote the round-trip delay expressed in number of time intervals. Such a delay  $L$  is equal to  $2 \cdot \lceil (D_{prop} + T_{comput})/T_{short} \rceil$ , where  $D_{prop}$  [s] is the maximum propagation delay from any ST to the NCC, or in the opposite direction, and  $T_{comput}$  [s] is the ST (or NCC) demand-assignment computing time.
- Let  $\eta$  denote the generic discrete-time at which a ST sends a bandwidth request; these times will be referred to as *demand times*. For the sake of clarity, we assume that all STs synchronously carry out their bandwidth demands. Moreover, we assume that bandwidth demands are performed periodically. Nevertheless, the proposed algorithm can be straightforwardly extended to the case in which ST demands are asynchronous and bandwidth demands are not periodic.
- Let  $T_{inf}$  denote the period occurring between two consecutive bandwidth requests, expressed in number of time intervals, and  $N$  denote the ratio of  $L$  and  $T_{inf}$ , i.e.,  $N = L/T_{inf}$  (Fig. 3 is relevant to the case of  $N = 2$ ). The choice of parameter  $N$  has to be carried out by carefully trading off two contrasting requirements:

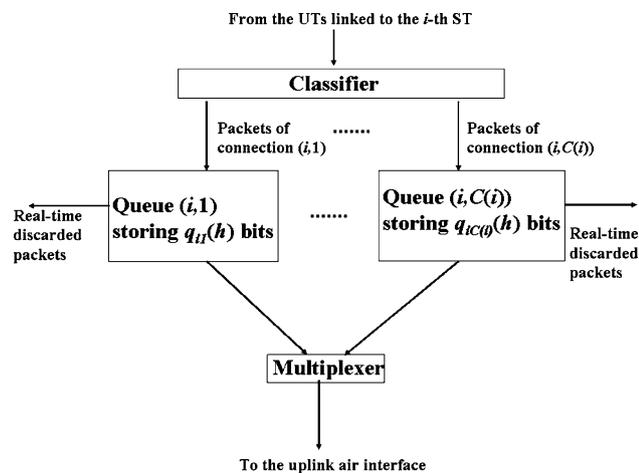
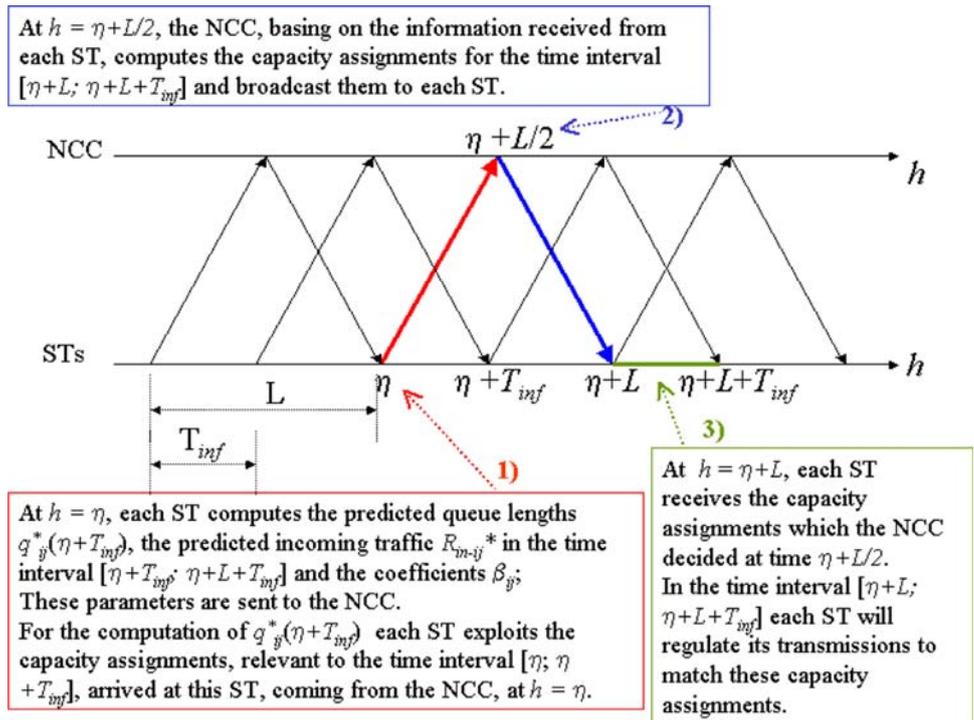


Fig. 2 Internal structure of a satellite terminal (ST)

**Fig. 3** Three-step example of the proposed capacity assignment procedure in the case of  $N = L/T_{inf} = 2$



- (1) Allowing a tight tracking of the traffic arrived at the STs from the UTs (by frequently sending bandwidth requests to the NCC);
- (2) Limiting the signaling overhead caused by such bandwidth requests.

Clearly, the former and the latter requirements lead to high and low values of  $N$ , respectively.

- Let  $R_{ij}^{in*}[\eta, k]^2$  denote the traffic prediction, performed by the  $i$ th ST at the  $\eta$ th time interval, of the bit rate that will enter queue  $(i, j)$  during time interval  $[\eta, k]$ . In Sect. 5, a Markov Modulated Poisson Process (MMPP) model is proposed to perform such predictions. This model outperforms other considered prediction models, as shown in the experiment scenarios (Sect. 6).

#### 4.2 Information sent to the NCC by STs

In the proposed demand-assignment mechanism, STs do not distributively calculate the bandwidth capacity they require. Conversely, they send to the NCC some key parameters that are used by the NCC itself to perform appropriate bandwidth assignments. This centralized approach, as apposed to the distributed approach in which STs directly compute the needed bandwidth, achieves a better *fairness* among STs and higher *bandwidth efficiency*

on the uplink satellite channel, as will be shown in the performance evaluation section.

Therefore, whenever at a time instant  $\eta$  a bandwidth request has to be performed, the  $i$ th ST sends to the NCC the following three pieces of information:

- (1)  $q_{ij}^*(\eta + T_{inf})$ : The  $C(i)$  predictions of the lengths of queues  $(i, j)$ ,  $j = 1, 2, \dots, C(i)$ , at time instant  $\eta + T_{inf}$ ; at time  $\eta$ , the  $i$ th ST computes these predictions as,

$$q_{ij}^*(\eta + T_{inf}) = q_{ij}(\eta) + \sum_{k=\eta}^{\eta+T_{inf}-1} R_{ij}^{in*}[\eta, k] \cdot T_{short} - \delta_{ij}^{ass}[\eta, \eta + T_{inf}] \cdot R_{up}^{dyn}[\eta, \eta + T_{inf}] \cdot T_{inf}. \tag{6}$$

- (2)  $\overline{R_{ij}^{in*}}[\eta + T_{inf}, \eta + T_{inf} + L]$ : The  $C(i)$  predictions of the average bit rates of the traffic that will enter queue  $(i, j)$  during time interval  $[\eta + T_{inf}, \eta + T_{inf} + L]$ ; at time  $\eta$ , the  $i$ th ST computes these predictions as,

$$\overline{R_{ij}^{in*}}[\eta + T_{inf}, \eta + T_{inf} + L] = \frac{\sum_{k=\eta+T_{inf}}^{\eta+T_{inf}+L-1} R_{ij}^{in*}[\eta, k]}{L}. \tag{7}$$

- (3)  $\beta_{ij}$ : The  $C(i)$  coefficients used to grant a higher weight to those queues  $(i, j)$  relevant to real-time connections that are losing bits because of packet expirations. These coefficients are computed as,

$$\beta_{ij} = 1 + K^{opt} \cdot \frac{B_{ij}^{loss}[\eta - T_{inf}, \eta]}{B_{ij}^{out}[\eta - T_{inf}, \eta]}, \tag{8}$$

<sup>2</sup> Throughout this article symbols with a \* apex represent prediction values.

where  $B_{ij}^{loss}[\eta - T_{inf}, \eta]$  [bit] represents the amount of bits discarded from queue  $(i, j)$  during time interval  $[\eta - T_{inf}, \eta]$  due to packet expirations;  $B_{ij}^{out}[\eta - T_{inf}, \eta]$  [bit] represents the whole amount of bits that, during the same time interval, is either retrieved from queue  $(i, j)$  and forwarded towards the air interface, or discarded due to packet expirations. In (8),  $K^{opt}$  is an appropriate feedback gain whose value determines the aggressiveness of the feedback control law (the optimal value  $K^{opt} = 0.97$  has been used in the simulation experiments reported in Sect. 6).

### 4.3 Demand-assignment procedure

As shown in Fig. 3 (which refers to the case of  $N = L/T_{inf} = 2$ ), basing on the pieces of information received from the STs, at time  $\eta + L/2$ , the NCC has to determine the capacity assignments  $\delta_{ij}^{ass}[\eta + L, \eta + L + T_{inf}]$  for any  $(i, j)$  pair. The *conservative approach* proposed in this article is to select these assignments aiming at *fully emptying* the ST queues at time  $\eta + L + T_{inf}$  (this is the last time at which the assignment decided by the NCC at time  $\eta + L/2$  will be effective).

The expected length of queue  $(i, j)$  at time  $\eta + L + T_{inf}$  is,

$$q_{ij}^*(\eta + L + T_{inf}) = q_{ij}^*(\eta + T_{inf}) + \overline{R_{ij}^{in}}^*[\eta + T_{inf}, \eta + T_{inf} + L] \times L - \{ \delta_{ij}^{ass}[\eta + T_{inf}, \eta + L] + \delta_{ij}^{ass}[\eta + L, \eta + L + T_{inf}] \} \times R_{up}^{dyn}[\eta + T_{inf}, \eta + L + T_{inf}] \cdot L. \tag{9}$$

Note that the term  $\delta_{ij}^{ass}[\eta + T_{inf}, \eta + L]$  appearing in (9) is relevant to capacity assignments already performed by the NCC at time instants earlier than  $\eta + L/2$ . Equation 9 assumes  $L > T_{inf}$ , i.e.,  $N > 1$ , which is the most common case. Nevertheless, the extension to the opposite case is straightforward. Then, the *target capacity assignments*, indicated as  $\delta_{ij}^{ass*}[\eta + L, \eta + L + T_{inf}]$ , which the NCC needs to assign at time  $\eta + L/2$  in order to empty the ST queues at time  $\eta + L + T_{inf}$ , can be obtained by imposing that the right hand side of (9) be equal to zero, i.e.,  $q_{ij}^*(\eta + L + T_{inf}) = 0$ . This yields to

$$\delta_{ij}^{ass*}[\eta + L, \eta + L + T_{inf}] = \frac{q_{ij}^*(\eta + T_{inf})}{R_{up}^{dyn}[\eta + T_{inf}, \eta + L + T_{inf}] \cdot L} + \frac{\overline{R_{ij}^{in}}^*[\eta + T_{inf}, \eta + T_{inf} + L]}{R_{up}^{dyn}[\eta + T_{inf}, \eta + L + T_{inf}]} - \delta_{ij}^{ass}[\eta + T_{inf}, \eta + L]. \tag{10}$$

The target capacity assignment in (10), however, can be actually granted only if the uplink capacity constraint in (5)

is satisfied. In order to force the respect of this constraint and take into account parameters  $\beta_{ij}$  in (8) (thus giving higher weight to those real-time connections suffering from poor QoS), the actual capacity assignments  $\delta_{ij}^{ass}[\eta + L, \eta + L + T_{inf}]$  is computed according to the following *normalization*,

$$\delta_{ij}^{ass}[\eta + L, \eta + L + T_{inf}] = \frac{\beta_{ij} \cdot \delta_{ij}^{ass*}[\eta + L, \eta + L + T_{inf}]}{\sum_{i=1}^S \sum_{j=1}^{C(i)} \beta_{ij} \cdot \delta_{ij}^{ass*}[\eta + L, \eta + L + T_{inf}]}. \tag{11}$$

Note that, by using the above-mentioned expression, the uplink capacity constraint is satisfied with the sign of equality meaning that all the available dynamic uplink capacity is actually assigned.

### 4.4 Recapitulation

In conclusion, the proposed capacity demand-assignment procedure follows the steps hereafter summarized (see also Fig. 3 for the case of  $N = 2$ ):

- (1) *Time  $\eta$* : The STs compute the forecast queue lengths  $q_{ij}^*(\eta + T_{inf})$  according to (6), the forecast bit rates  $\overline{R_{ij}^{in}}^*[\eta + T_{inf}, \eta + T_{inf} + L]$  according to (7), and coefficients  $\beta_{ij}$  according to (8). All these parameters are sent to the NCC.
- (2) *Time  $\eta + L/2$* : The NCC receives these three pieces of information, detailed in Sect. 4.2, from each ST, i.e.,  $q_{ij}^*(\eta + T_{inf})$ ,  $\overline{R_{ij}^{in}}^*[\eta + T_{inf}, \eta + T_{inf} + L]$ , and  $\beta_{ij}$ , and computes the capacity assignments  $\delta_{ij}^{ass}[\eta + L, \eta + L + T_{inf}]$  to be granted to all the connections during time interval  $[\eta + L, \eta + L + T_{inf}]$ , according to (10) and (11). Such assignments are then broadcast to the STs.
- (3) *Time  $\eta + L$* : The  $i$ th ST receives the capacity assignments  $\delta_{ij}^{ass}[\eta + L, \eta + L + T_{inf}]$  from the NCC. These assignments determine the traffic, i.e. number of packets, that the  $i$ th ST is authorized to forward from queues  $(i, j)$  towards the uplink air interface during time interval  $[\eta + L, \eta + L + T_{inf}]$ . Moreover, the  $i$ th ST utilizes these capacity assignments at next demand time(s) for the computation of the forecast queue lengths, according to (6).

As a final remark, note that a ST can rearrange the capacity granted by the NCC among the connections it is involved in. In this rearrangement, the ST can take into account updated information concerning the present lengths of queues (this information was not available to the NCC when it computed the capacity assignments), according to appropriate criteria (e.g., see [15, 16]). So, at time  $\eta + L$ , the  $i$ th ST can compute the overall fraction of the uplink

capacity, indicated as  $\alpha_i[\eta + L, \eta + L + T_{inf}]$ , assigned to it during time interval  $[\eta + L, \eta + L + T_{inf}]$ , as

$$\alpha_i[\eta + L, \eta + L + T_{inf}] = \sum_{j=1}^{C(i)} \delta_{ij}^{ass}[\eta + L, \eta + L + T_{inf}]. \tag{12}$$

Therefore, the capacity fraction  $\delta_{ij}[\eta + L, \eta + L + T_{inf}]$  actually granted to connection  $(i, j)$  during time interval  $[\eta + L, \eta + L + T_{inf}]$  can differ from the fraction assigned by the NCC (i.e.,  $\delta_{ij}^{ass}[\eta + L, \eta + L + T_{inf}]$ ). Nevertheless, the following constraint must be respected for each ST,

$$\sum_{j=1}^{C(i)} \delta_{ij}[\eta + L, \eta + L + T_{inf}] \leq \alpha_i[\eta + L, \eta + L + T_{inf}], \quad \forall i \in [1, S]. \tag{13}$$

### 5 Markov modulated Poisson process model for IP traffic prediction

In this section, we propose a Markov Modulated Poisson Process (MMPP) model for IP traffic prediction. As described in Sect. 4, it is crucial for satellite terminals to predict the incoming IP traffic in order to send accurate bandwidth requests to the NCC. Specifically, each ST needs to forecast the traffic associated with each connection in order to assess the evolution of the relevant queue.

The MMPP algorithm for IP traffic prediction is constituted by two phases: 1) *Identification/Tuning* phase and 2) *Traffic Prediction* phase. The Identification/Tuning phase finds the *order* of the MMPP chain model (i.e., number of states) and the associated parameters (i.e., Poisson rates) according to aggregated measurements on the behavior of the IP data stream. In the design of the prediction algorithm, we assume *ergodicity* of the traffic stream and exchange *time averages*, which are measured on the traffic stream, for *ensemble averages*, which are used to configure the MMPP chain model. The Traffic Prediction phase relies on the tuned MMPP model in order to perform traffic prediction.

The remain of the section is organized as follows. In Sect. 5.1, the main features of a MMPP chain model are recalled, and the general problem of data generation is presented. In Sect. 5.2, the Identification/Tuning phase is described and the MMPP matching problem is detailed. Finally, in Sect. 5.3 the Traffic Prediction phase is presented.

#### 5.1 Basics of the Markov modulated Poisson process

A Markov Modulated Poisson Process (MMPP) [17–19] of order  $N_M$  consists of a Markov chain model in which each

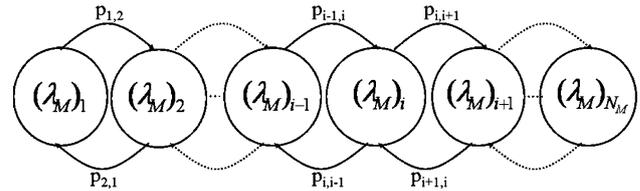


Fig. 4 Markov Modulated Poisson Process (MMPP) chain model

state  $i = 1, \dots, N_M$  is associated with a Poisson process that is characterized by rate<sup>3</sup>  $(\lambda_M)_i \in \mathbb{R}^{0,+}$ . A MMPP is a time varying Poisson process, whose rate is changed (*modulated*) according to the Markov chain.

Let us consider in the following a discrete-time Markov chain, as shown in Fig. 4, that is sampled every  $\Delta t$  [s], starting from an initial time  $t_0$  [s]. For the sake of simplicity, we will use the index  $h$  to represent the discrete-time instant  $t_h = t_0 + h \cdot \Delta t$ . If  $\mathcal{S}_h, h \in \mathbb{N}^0$ , is a state moving at discrete-time instants between a finite number of states  $i = 1, \dots, N_M$  with transition probabilities only depending on the previous state, then  $\mathcal{S}_h$  is defined as *Markov Process*.

Let matrix  $\mathcal{P} = [p_{ij}] \in \mathbb{R}^{0,+N_M \times N_M}$  be the one-step state transition matrix whose elements are the probabilities to transit from one state into another in one time-step, as shown in Fig. 4. These are formally defined as,

$$p_{ij} = \Pr\{\mathcal{S}_{h+1} = j | \mathcal{S}_h = i\}, \quad \forall i, j = 1, \dots, N_M. \tag{14}$$

This matrix  $\mathcal{P}$  can be shown to be a *stochastic matrix*, since it has the property that: i) all its elements are nonnegative ( $p_{ij}, \forall i, j = 1, \dots, N_M$ , are probabilities) and ii) the sum of the elements in each row equals 1 (i.e., each state has at least one possible transition into another state). These properties can be mathematically expressed as,

$$p_{ij} \in \mathbb{R}^{0,+}; \sum_{s=1}^{N_M} p_{is} = 1, \quad \forall i, j = 1, \dots, N_M. \tag{15}$$

Note that  $p_{ij}$  do not depend on time, i.e., on  $h$ . Properties in (15) define the *chain* structure of the proposed Markov process. Intuitively, this structure imposes that the evolution of a MMPP occurs between adjacent states. This choice is motivated by the need for a simple and low-complexity predicting structure that is able to capture the temporal correlation and self-similarity properties of incoming IP traffic belonging to a given connection.

Let  $\Theta_M(h)$ , at time  $t_h$ , be a column vector with  $(\Theta_M)_i(h) = \Pr\{\mathcal{S}_h = i\}$  as  $i$ th entry, representing the *state distribution* of the chain at step  $h$ . It holds that,

<sup>3</sup> Throughout this article we will use the following notation:  $\mathbb{R}^{0,+} \equiv \mathbb{R}^+ \cup \{0\}$ , and  $\mathbb{N}^0 \equiv \mathbb{N} \cup \{0\}$ .

$$(\Theta_M)_i(h) \in \mathbb{R}^{0,+}; \sum_{s=1}^{N_M} (\Theta_M)_s(h) = 1, \quad (16)$$

$$\forall h \in \mathbb{N}^0, \forall i = 1, \dots, N_M.$$

Under this assumption, the evolution of a MMPP is completely described by the following recursive equation,

$$\Theta_M(h+1)^T = \Theta_M(h)^T \cdot \mathcal{P}, \quad (17)$$

where  $T$  is the transpose operator.

Throughout this article, we will neglect the *transitory state*, and we will consider the Markov chain in its *steady state*, where the state distribution  $\Theta_M$  becomes time-independent, i.e., does not depend anymore on step  $h$ . In the steady state, (17) simplifies to

$$\Theta_M^T = \Theta_M^T \cdot \mathcal{P}. \quad (18)$$

Note that vector  $\Theta_M$  is the left eigenvector of stochastic matrix  $\mathcal{P}$  corresponding to the eigenvalue 1. This is a nonnegative vector, and the sum of its elements equals 1, exactly as in (16), but now without the dependency on  $h$ , i.e.,

$$(\Theta_M)_i \in \mathbb{R}^{0,+}; \sum_{s=1}^{N_M} (\Theta_M)_s = 1, \quad \forall i = 1, \dots, N_M. \quad (19)$$

Let us now introduce the *observation or measurement process*  $\psi_h, h \in \mathbb{N}^0$ , that accounts for the incoming IP traffic that we want to predict. In the following, we will not represent traffic according to the fluid approximation, although fluid models<sup>4</sup> are widely used in addressing a variety of network control problems such as congestion control [20, 21], routing [22], and pricing [23, 24]. Rather, we will consider a more realistic model where traffic is actually made up of discrete-packets of variable length. In our approach, in order to measure traffic, we logically segment a traffic data stream into *fixed-length blocks*, whose size  $B$  [bit] is a constant predefined number of bits. The block size  $B$  is the finest granularity in the measurement of traffic and can be as small as needed. This approach allows us to deal with traffic composed of variable length IP packets, as is the Internet traffic case, while keeping the complexity of the predicting algorithm low. This approximation should not be seen as a limiting factor, since we are interested in predicting neither the packet length nor the inter-arrival time between consecutive packets. In fact, as explained in Sect. 4, we need to compute  $R_{ij}^{in*}[\eta, k]$ , which is the prediction, performed by the  $i$ th ST at time  $\eta$ , of the bit rate that will enter queue  $(i, j)$  during time interval  $[\eta, k]$ . Thus, we are interested in predicting the average bit rate during a time interval.

<sup>4</sup> Fluid models replace discrete packets with continuous flows.

If we divide the monitoring interval  $T_{mon}$  [s] into  $H$  sub-intervals each with time length  $\Delta t$  [s], i.e.,  $T_{mon} = H \cdot \Delta t$ , if the maximum number of expected traffic blocks is  $Q$ , then the following relation among  $B, Q$ , and  $\Delta t$  must hold,

$$\frac{B \cdot Q}{\Delta t} \geq R_{max}^{in}, \quad (20)$$

where  $R_{max}^{in}$  [bps] is the maximum expected bit rate. There is a trade-off between the *granularity* of the traffic measurement, which depends on  $B$ , and the *complexity* of the proposed predicting algorithm, which increases as  $Q$  increases, as will be clear later on. In particular, the smaller  $B$ , the larger  $Q$  must be in order to satisfy (20), given a maximum expected bit rate  $R_{max}^{in}$ . As far as the choice of  $T_{mon}$ , i.e.,  $H$ , is concerned, the longer the monitoring time, the more statistically relevant the measures of the traffic are, but the longer it takes to have updated measures. An appropriate value for  $T_{mon}$  should be chosen by considering this trade-off, as well as the expected statistical variability of the traffic.

The *observation or measurement process*  $\psi_h, h \in \mathbb{N}^0$ , accounts for the number of measured blocks arriving in time interval  $[t_h, t_{h+1}]$ . It has values in the set  $\{0, 1, \dots, Q\}$ , and probabilistically depends on  $\mathcal{S}_h$  via matrix  $\mathcal{C} = [c_{q+1j}] \in \mathbb{R}^{0,+^{(Q+1) \times N_M}}$ ,  $q = 0, 1, \dots, Q, j = 1, 2, \dots, N_M$ . Generic element  $c_{q+1j}$  represents a Poisson distribution, i.e.,

$$c_{q+1j} = \Pr\{\psi_h = q | \mathcal{S}_h = j\} = \frac{(\lambda_M)_j^q \cdot e^{-(\lambda_M)_j}}{q!}, \quad (21)$$

where  $(\lambda_M)_j \in \mathbb{R}^{0,+}$  is the arrival bit rate of the Poisson process associated with the state  $j$ . If the state parameter  $(\lambda_M)_j$  is 0, then its associated emission bit rate will be zero as well. This deterministic process is called *zero process*. The row of  $\mathcal{C}$  that represents the zero process is equal to  $[1 \ 0 \ \dots \ 0]$ . Let  $\Xi$  be the vector with  $\Pr\{\psi_h = q\}$  as  $(q + 1)$ th entry, then,

$$\Xi = \mathcal{C} \cdot \Theta_M. \quad (22)$$

For the sake of compactness, the Poisson parameters  $(\lambda_M)_i$  are arranged in a column vector  $\Lambda_M = [(\lambda_M)_i] \in \mathbb{R}^{0,+^{N_M}}$ . At time step  $h$ , the MMPP, which can be described by the tuple  $\langle \mathcal{P}; \Lambda_M; N_M \rangle$ , will generate  $\psi_h$  bit emissions according to the Poisson process with arrival rate  $(\lambda_M)_i$ , if the state of the Markov chain is  $i$ . This implies that if  $(\lambda_M)_i = 0$ , then no bit emissions will be generated.

In our prediction model we leverage the *first order statistics* of the bit emission  $\psi_h$ . The first order statistic is described by the cumulative distribution function, which is defined as  $\mathcal{F}(v) = \Pr\{\psi_h \leq v\}$ , with  $v = 0, 1, \dots, Q$ . The MMPP cumulative distribution function  $\mathcal{F}_M$  is a weighted average of the cumulative distributions of each Poisson process associated with a state of the chain. Thus,

$$\mathcal{F}_M(v) = \sum_{i=1}^{N_M} (\Theta_M)_i \cdot \mathcal{F}_{(\lambda_M)_i}(v), \tag{23}$$

where

$$\mathcal{F}_{(\lambda_M)_i}(v) = e^{-(\lambda_M)_i} \cdot \sum_{q=0}^v \frac{(\lambda_M)_i^q}{q!}. \tag{24}$$

### 5.2 Identification/tuning phase: the MMPP matching problem

After computing the cumulative distribution function of the arrival process  $\psi_h$ , the vectors of the first order parameters of the MMPP chain,  $\Lambda_M$ ,  $\Theta_M$ , and the model order  $N_M$  are determined by solving a *nonnegative least square* problem [25]. When all these first order parameters are determined, the identification problem is solved, and traffic prediction can be performed using the data generation of the tuned MMPP chain, as described in Sect. 5.1.

The model identification problem can be cast as follows: *Find*  $N_M$ ,  $\Theta_M$ , and  $\Lambda_M, \forall i = 1, 2, \dots, N_M$ , *Given* the observations on the arrivals  $\psi_h$  for each time step  $h = 0, 1, 2, \dots, H - 1$  within the monitoring interval  $T_{mon}$ , so that they form a MMPP with *first order statistics*  $\mathcal{F}_M$  matching those of the measured traffic  $\psi_h$ , i.e.,  $\mathcal{F}_{data}$ , as accurately as possible. The state parameter vectors  $\Theta_M = (\Theta_M)_i$  and  $\Lambda_M = (\Lambda_M)_i$ , with  $i = 1, 2, \dots, N_M$ , can then be used to predict future traffic according to (23) and (24). Note that if the order  $N_M$  of the MMPP chain is too high, not only the complexity of the prediction algorithm skyrockets, but also the model will tend to reproduce measurement errors in the observation process. Conversely, if the order is too low, the prediction model will lack accuracy, since too few Poisson states will be modulated to reproduce the traffic characteristics. For this reason, we decided not to set an a priori value for  $N_M$ ; rather, we included the model order as key variable in the proposed identification problem.

The cumulative distribution function of the data sequence is a staircase function. It is computed as follows,

$$\mathcal{F}_{data}(v) = \frac{1}{H} \cdot \sum_{j=0}^v \sum_{h=0}^{H-1} \delta(\psi_h, j), \quad v = 0, 1, 2, \dots, Q, \tag{25}$$

where  $\delta(\psi_h, j)$  is the *Kronecker delta*, i.e.,  $\delta(\psi_h, j)$  equals 1 iff  $\psi_h = j$  while it is 0 otherwise, and  $H$  is the total number of subintervals which the monitoring interval  $T_{mon}$  is divided in.

The distribution function of the MMPP is a linear combination of Poisson distributions, as can be inferred from (23). The cumulative distribution function of the data,  $\mathcal{F}_{data}$ , must be approximated by the MMPP cumulative distribution function  $\mathcal{F}_M$ . This implies that  $\mathcal{F}_{data}$  must be

approximated by a *nonnegative linear combination* of cumulative distributions of Poisson processes, i.e.,

$$\mathcal{F}_d \simeq \mathcal{D}\Theta_M, \tag{26}$$

where  $\mathcal{F}_d$  is the vector of size  $Q + 1$  with elements  $(\mathcal{F}_d)_v = \mathcal{F}_{data}(v)$ , with  $v = 0, 1, \dots, Q$ , and  $Q$  is the maximum number of expected traffic blocks. The columns of matrix  $\mathcal{D} \in \mathbb{R}^{0,+,(Q+1) \times N_M}$  represent the cumulative distribution functions  $\mathcal{F}_{(\lambda_M)_i}$  in (24). If the states of the MMPP modeling a given data set were given, i.e., the  $(\lambda_M)_i$  parameters were known, matrix  $\mathcal{D}$  would be mathematically determined, and  $\Theta_M$  could be easily computed by solving the linear equation system in (26). Without the knowledge of the states or the number of states, which is a more realistic scenario, the same approach yields  $N_M$ ,  $\Theta_M$ , and  $\Lambda_M$  if  $\mathcal{D}$  is replaced by an enlarged version  $\mathcal{D}' \in \mathbb{R}^{0,+,(Q+1) \times N'_M}$ . This matrix  $\mathcal{D}'$  has more columns than  $\mathcal{D}$  ( $N'_M \geq N_M$ ), but the same number of rows. Each column  $(\mathcal{D}')_j, j = 1, 2, \dots, N'_M$ , of this enlarged matrix represents a possible state, i.e., a possible Poisson cumulative distribution. In fact, in each subinterval of the monitoring period the domain of possible block arrivals  $[0, Q]$  is discretized in order to get a broad choice of candidate states. The discretization step is chosen to increase linearly because the variance of a Poisson process is equal to its rate. The computed cumulative distribution vector  $\mathcal{F}_d$  must be reconstructed as a nonnegative linear combination of the columns of  $\mathcal{D}'$ . We can now formulate the optimization problem for the MMPP model identification

#### **P: MMPP identification problem**

*Given:*  $\mathcal{D}', Q, N'_M, \psi_h, h = 0, 1, \dots, H - 1$

*Find:*  $x = (x)_j, j = 1, 2, \dots, N'_M$  (27)

*Minimize:*  $\|\mathcal{F}_d - \mathcal{D}'x\|_2$  (28)

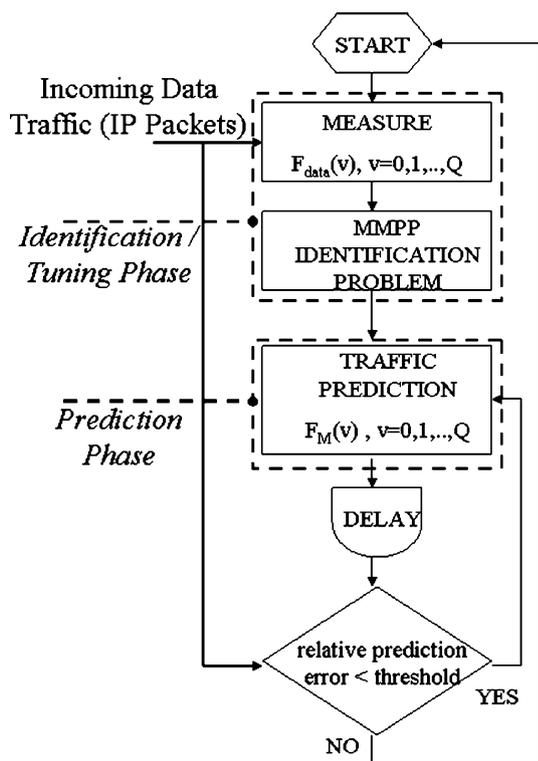
*Subject to:*  $(x)_j \geq 0, j = 1, 2, \dots, N'_M$ , (29)

$(\mathcal{F}_d)_v = \mathcal{F}_{data}(v), v = 0, 1, \dots, Q.$  (30)

We solved problem **P** using the *Khun Tucker's algorithm* [26]. The MMPP parameters are completely defined by the solution of this optimization problem, as follows. Let  $x^*$  be the optimal value of  $x$ . The number of non-zero components of  $x^*$  gives automatically the MMPP model order  $N_M \leq N'_M$ . In addition, the indices of the nonzero components of  $x^*$  represent  $(\lambda_M)_j$ , whereas the values of the nonzero  $x^*$  components give  $(\Theta_M)_j$ . This way, the model order  $N_M$  is automatically determined.

### 5.3 Traffic prediction phase

Once in the Identification/Tuning phase the MMPP identification problem **P** has been solved, and all the parameters



**Fig. 5** Flow chart of the MMPP prediction algorithm

have been tuned, the Traffic Prediction phase is in charge of predicting future traffic, according to (23) and (24). In Fig. 5, a flow chart of the entire MMPP algorithm is depicted. Each traffic prediction value is periodically compared to the real incoming data traffic, so that it is possible to know when it is necessary to re-enter the Identification/Tuning phase in order to tune the model parameters. To this end, the relative prediction error is constantly monitored to prevent error drift. In fact, if the relative error is higher than a predetermined threshold, a new identification problem  $\mathbf{P}$  is solved, with as input the latest available traffic measurements. This mechanism allows following the dynamics of the input traffic, while saving precious computational resources.

## 6 Performance evaluation

In this section, we present the simulation results of the capacity demand-assignment algorithm for satellite bandwidth allocation (described in Sect. 4), which relies on the Markov Modulated chain model for IP traffic prediction (presented in Sect. 5). The simulation tool OPNET [27] has been adopted for developing a satellite simulator and testing the performance of the proposed algorithms. OPNET simulator has three logical levels: *Network Level* (a GEO satellite system has been considered for its

challenging high propagation delay, together with Satellite Terminals, User Terminals, Hub Stations, and a Network Control Center), *Node Level* (consisting of all the demons and algorithms of the protocol stack), and *Process Level* (Finite State Machines (FSM) developed in C++ that implement the proposed algorithms and the associated protocols).

The statistical characteristics of the considered application traffic sources are reported in Table 1. In particular, we considered five types of applications: two real-time applications, using the User Datagram Protocol (UDP), namely Voice over IP and Video conference, and three non real-time applications, using the Transmission Control Protocol (TCP), namely FTP, E-mail, and Web browsing [28]. The simulated scenario is shown in Fig. 1, which considers a GEO satellite system (satellite round-trip time is approximately 500 ms). The considered spotbeam includes 3 STs. Each of these STs is connected (e.g., via Ethernet LAN) to 5 User Terminals (UTs). Each of these UTs has five in-progress connections relevant to the five applications listed in Table 1, in such a way that every UT is involved in 1 Voice-over-IP, 1 Video-conference, 1 FTP, 1 E-mail, and 1 Web-browsing connection. Thus, in this setting,  $S = 3$  and  $C(i) = 5$  ( $i = 1, 2, 3$ ), according to the notation introduced in Sect. 2. Parameters  $R_{ij}^{static}$  are all set to zero for any  $i \in [1, 3]$  and any  $j \in [1, 5]$ . For real-time traffic, we have considered the maximum queue-delay  $D_{ij}^{max}$  equal to 50 ms and 100 ms for Voice-over-IP and Video-conference connections, respectively, according to the QoS parameters proposed by the ETSI TIPPHON project

**Table 1** Application source parameters

Source parameter	Distribution
Email inter. time [s]	exp (300)
Email group size <sup>a</sup>	constant (3)
Email size [KB]	unif_int [1,99]
File transf. inter-req. time [s]	exp (180)
File transf. size [KB]	const (500)
Webpage inter. time [s]	exp (60)
Video frame frequency [frame/s]	constant (15)
Video frame size [KB]	unif_int [2,3]
Voice spurt length [s] <sup>b</sup>	exp (0.352)
Voice silence length [s]	exp (0.65)
Voice encoder scheme	G.711 (Silence)
Voice frames per packet <sup>c</sup>	constant (1)

<sup>a</sup> Indicates the number of ‘queued emails’ to be sent

<sup>b</sup> Specifies the time spent by the calling party in speech mode in a speech-silent cycle

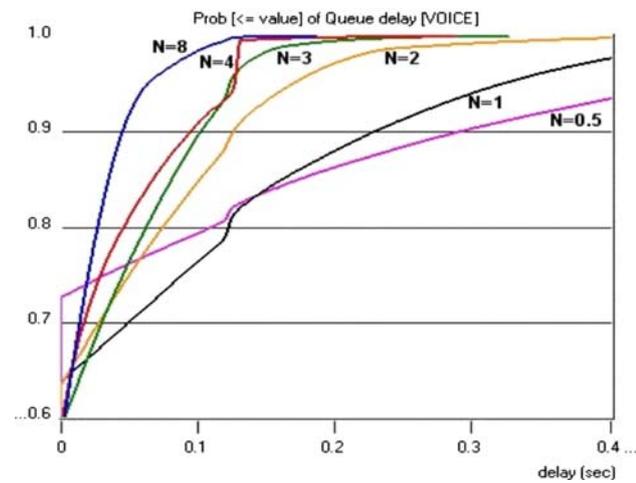
<sup>c</sup> This attribute determines the number of encoded voice frames grouped into a voice packet, before being sent by the application to the lower layers

[29]. Finally, the available uplink capacity in the considered spotbeam  $R_{up}^{tot}$  is set to 10 Mbps.

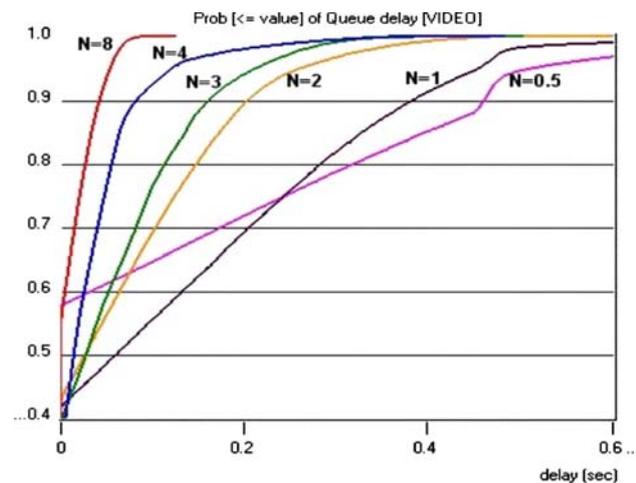
Figures 6 and 7 show the cumulative distribution of the queuing delay  $D_{ij}$  experienced by the packets relevant to real-time applications, as the parameter  $N = L/T_{inf}$  ranges in the interval [0.5, 8], i.e., given the fixed GEO satellite round-trip time  $L$ , as the period between consecutive bandwidth requests  $T_{inf}$  varies (see Sect. 4.1). Specifically, Fig. 6 depicts the queuing delay cumulative distribution for Voice-IP datagrams, while Fig. 7 depicts the same metric for Video-IP datagrams. Note that by increasing the value of  $N$  queuing delays decrease, and the probability that the queuing delay exceed the maximum-tolerated delay threshold decreases. Nevertheless, the signaling overhead

increases when  $N$  increases. So, as already stressed in Sect. 4,  $N$  must be selected by trading off these contrasting requirements. With this respect, Figs. 6 and 7 show that  $N = 4$  is the lowest value for which it is possible to statistically respect the QoS delay requirement for real-time applications. Thus, this value has been considered in the simulations.

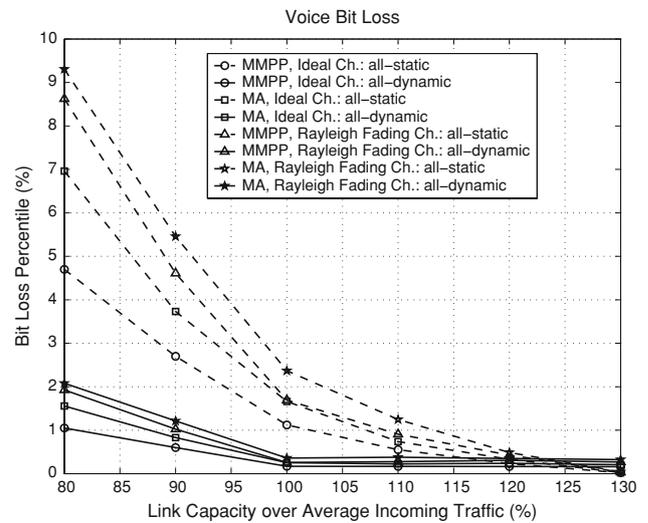
In Figs. 8–10, we compared through simulation experiments the performance achieved using the proposed capacity-assignment algorithm, i.e., the procedure described in Sect. 4 (this case will be hereinafter referred to as *all-dynamic*), with the one achieved by fixedly partitioning the available bandwidth among the three STs (this case will be hereinafter referred to as *all-static*). Note that in the



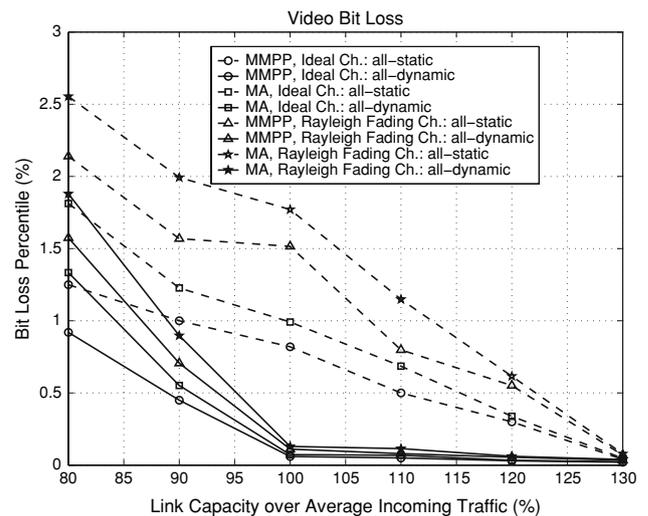
**Fig. 6** Queuing delay cumulative distribution functions (CDFs) of Voice IP datagrams for different values of  $N$  in the range [0.5, 8]



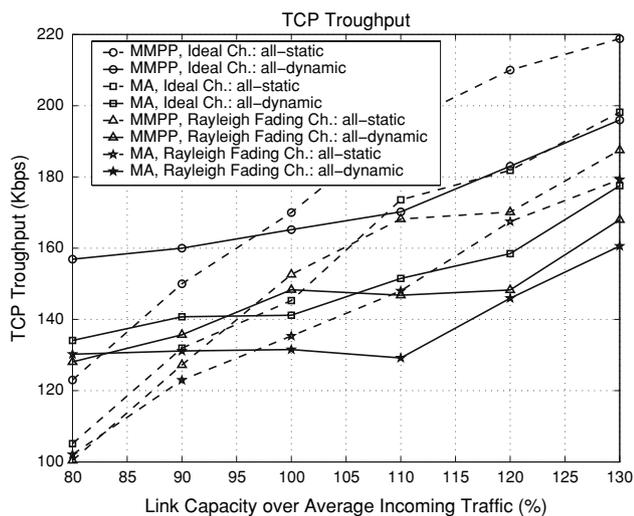
**Fig. 7** Queuing delay cumulative distribution functions (CDFs) of Video IP datagrams for different values of  $N$  in the range [0.5, 8]



**Fig. 8** Voice bit loss percentile comparisons expressed for significant capacity values



**Fig. 9** Video bit loss percentile comparisons expressed for significant capacity values



**Fig. 10** TCP average bit rate comparisons expressed for significant capacity values

*all-static* case each ST is fixedly assigned a third of the total available uplink capacity and no demand-assignment mechanism is required.

In the performed experiments, the following key performance parameters are monitored, which represent the QoS targets stated in Sect. 2:

- *Bit Loss Percentile* (for real-time traffic), which is defined as the ratio of the bits discarded because they have waited more than the maximum tolerated delay, and the sum of the transmitted and discarded bits.
- *Average Throughput* (for non real-time traffic), which is defined as the average end-to-end TCP data rate, discounted by datagram retransmissions due to losses on the satellite channel.

The simulation parameters are the same as in the previous simulation experiments, with the exception of the available uplink capacity  $R_{up}^{tot}$ . As a matter of fact, we now consider the satellite system behavior when the available uplink capacity  $R_{up}^{tot}$  is varied with respect to an *average offered uplink capacity* of 5.5 Mbps, which is defined as the ratio of the total traffic (expressed in bits) offered to the three STs by the UTs during the simulation time and the simulation time duration itself. In particular, we assumed the available uplink capacity to range in the interval [80, 130%] of the average offered uplink capacity, 80 and 130% representing heavily overloaded and underloaded traffic scenarios, respectively. This normalization makes performance results to be independent on the uplink capacity and on the amount of offered traffic; rather, results will only depend on their ratio.

Specifically, we compared the *all-dynamic* and the *all-static* bandwidth assignment algorithms (*solid* and *dotted* curves in Figs. 8–10, respectively) when two competing

traffic prediction schemes are considered, i.e., the *Markov Modulated Poisson Process (MMPP) chain model* scheme (Sect. 5) and the *Moving Average (MA)* scheme, which uses as traffic prediction the projection of incoming traffic averaged during a fixed-length moving time window. In addition, we considered two different channel models for the satellite air interface, an ideal model (*Ideal Ch.*), where the transmission loss is only affected by a deterministic component depending on distance and frequency, and a Rayleigh fading model (*Rayleigh Fading Ch.*), which captures signal dips due to transmission anomalies and multipaths. While in the ideal channel bit errors have been modeled to occur randomly, in the Rayleigh fading channel bit errors occur in bursts.

In Figs. 8–10, the 95% confidence intervals are not shown for the sake of clarity. At any rate, performance results are averaged on several experiments in such a way that the 95% relative confidence intervals be smaller than 5%.

Figure 8 depicts the *Bit Loss Percentile* for the Voice over IP application as a function of the above-mentioned capacity values. The dashed curves refer to the *all-dynamic* case, while the solid ones to the *all-static* one. The figure highlights that, if the satellite system is overloaded (link capacity over average incoming traffic lower than 100%), the *all-dynamic* approach is remarkably more efficient than the *all-static* one, since the proposed demand-assignment procedure succeeds in exploiting the advantages of statistical multiplexing. Clearly, as the capacity availability grows, these advantages reduce and for an high capacity availability (link capacity over average incoming traffic higher than 125%), the *all-static* has a slight advantage over the *all-dynamic* case. This is due to the fact that the former does not require the demand-assignment procedure (which entails delay and overhead savings). These considerations hold for both considered channel models, i.e., ideal and Rayleigh fading, and both prediction schemes, i.e., MMPP and MA. In particular, for a given channel model, the MMPP prediction model outperforms the MA model both in the all-static and all-dynamic cases, i.e., lower bit losses are experienced when the traffic prediction is performed with the MMPP model, since this achieves a more accurate traffic prediction than the MA scheme.

This is confirmed in Fig. 9, which shows the same metric of Fig. 8, i.e., *Bit Loss Percentile*, but refers to the video-conference application. The same considerations as in Fig. 8 apply. In particular, the lower bit loss percentile values obtained for video conference depend on the higher maximum tolerated delay of this application with respect to the voice one (100 vs. 50 ms, respectively).

Finally, Fig. 10 depicts the average TCP throughput for non real-time applications. When the system is lightly or heavily overloaded (link capacity over average incoming

traffic lower than 95%) the all-dynamic bandwidth assignment approach outperforms the all-static one, i.e., achieves 10–30% higher throughput values. In addition, Fig. 10 shows that, also for non real-time traffic, MMPP prediction algorithm achieves better performance than its competing MA scheme. This can be roughly quantified in 20–25% higher throughput values for any given link capacity over average traffic ratio. Moreover, as pointed out for Figs. 8 and 9, these results hold for both considered satellite channel models.

## 7 Conclusions

This article dealt with the problem of the design of a control-based demand-assignment algorithm for a satellite access network guaranteeing a target quality of service to Internet traffic, while efficiently exploiting the air interface. The proposed algorithm dynamically partitions the uplink bandwidth capacity in a satellite spotbeam among the in-progress connections. A Control Theory approach, which relies on a Markov modulated chain model for IP traffic prediction, is adopted to efficiently tackle the problem of the delay between bandwidth requests and bandwidth assignments, while minimizing the signaling overhead caused by control messages.

Through extensive simulation experiments, the lowest bandwidth request frequency to statistically respect the QoS delay requirement for real-time applications was computed, and the proposed demand-assignment algorithm and the Markov modulated chain traffic prediction model were shown to improve the overall satellite network performance under different satellite channel models.

## References

- Digital Video Broadcasting (DVB): Interaction Channel for Satellite Distribution Systems. ETSI EN 301 790 v1.2.2, Tech. Rep., Dec 2000.
- Delli Priscoli, F., & Pietrabissa, A. (2002). Resource management for ATM based geostationary satellite networks with on-board processing. *Computer Networks (Elsevier)*, 39(1), 43–60.
- Pompili, D., Delli Priscoli, F., & Santoro, G. (2004). Enhanced IP-based satellite architecture compatible with many satellite platforms. In *Proc. of Conference of Information Society Technologies (IST)*, The Hague, Nov.
- Almquist, P. (1992). Type of service in the internet protocol suite. IETF RFC 1349, Tech. Rep., July.
- Braden, R., Clark, D., & Shenker, S. (1994). Integrated services in the internet architecture: an overview. IETF RFC 1633, Tech. Rep., July.
- Rosenberg, C. (2000). Broadband satellite systems: a network perspective. In *Proc. of IEEE Global Telecommunications Conference (GLOBECOM)*, San Francisco, Nov.
- Mitchell, P. D. (2003). *Effective medium access control for geostationary satellite systems*. Ph.D. Thesis. UK: University of York.
- Jiang, Z., Li, Y., & Leung, V. C. M. (2002). A predictive demand assignment multiple access protocol for broadband satellite networks supporting internet applications. In *Proc. of IEEE International Conference on Communications (ICC)*, vol. 5 (pp. 2973–2977). New York, May.
- Gerakoulis, D. (2000). Demand assignment control in a satellite switched CDMA network. *International Journal of Network Management (Wiley InterScience)*, 10(4), 175–189.
- Delli Priscoli, F., Pompili, D., & Santoro, G. (2004). A QoS-aware bandwidth on demand assignment mechanism in a GEO satellite system. In *Proc. of Conference of Information Society Technologies (IST)*, Lyon, France, June.
- Delli Priscoli, F., & Pietrabissa, A. (2002). Control-theoretic band-on-demand protocol for satellite networks. In *Proc. of Conference on Control Applications (CCA)*, Glasgow, UK, Sep.
- Delli Priscoli, F. (1999). Design and implementation of a simple and efficient medium access control for high speed wireless local area networks. *IEEE Journal on Selected Areas in Communications*, 17(11), 2052–2064.
- Acar, G., & Rosenberg, C. (1999). Algorithms for bandwidth on demand. In *Proc. of Ka Band Utilization Conference*, Taormina, Italy, Oct.
- Acar, G., & Rosenberg, C. (2002). Weighted Fair Bandwidth-On-Demand (WFBoD) for geostationary satellite networks with on-board processing. *Computer Networks (Elsevier)*, 39(1), 5–20.
- Becchetti, L., Delli Priscoli, F., Mahonen, P., Munoz, L., & Inzerilli, T. (2001). Enhancing IP service provision over heterogeneous wireless networks: A path towards 4G. *IEEE Communications Magazine*, 39(8), 74–81.
- Delli Priscoli, F., & Isidori, A. (2002). A control-engineering approach to traffic control in wireless networks. In *Proc. of Conference on Decision and Control (DCD)*, Las Vegas, Dec.
- Cock, K. D., Gestel, T. V., & Moor, B. D. (1997). Stochastic system identification for ATM network traffic models: A time domain approach. Internal Report K. U. (Leven-Belgium), Tech. Rep, July.
- Cock, K. D., & Gestel, T. V. (1998). Identification of the circulant modulated poisson. In *Proc. of the Mathematics Theory of Networks and Systems (NTNS)* (pp. 742–779), Dec.
- Ryd, T. (1996). An EM algorithm for estimation in Markov-modulated Poisson processes. *Computational Statistics and Data Analysis (Elsevier)*, 21(4), 431–447.
- Mo, J., & Walrand, J. (2000). Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5), 556–567.
- Alpcan, T., & Basar, T. (2000). A variable rate model with QoS guarantees for real-time internet traffic. In *Proc. of SPIE Symposium on Information Technologies*, vol. 2411, Boston, MA, Nov.
- Altman, E., & Basar, T. (1998). Multi-user rate-based flow control. *IEEE/ACM Transactions on Communications*, 47(7), 940–949.
- Yaiche, H., Mazumdar, R. R., & Rosenberg, C. (2000). A game theoretic framework for bandwidth allocation and pricing in broadband networks. *IEEE/ACM Transactions on Networking*, 8(5), 667–678.
- Basar, T., & Srikant, R. (2002). Revenue-maximizing pricing and capacity expansion in a many-users regime. In *Proc. of IEEE Conference on Computer Communications (INFOCOM)* (pp. 1556–1563). New York, June.
- Lawson, C., & Hanson, R. (1995). Solving least square problem. *Classic Applied Mathematics Series*, 15.
- Qi, H. D., Qi, L., & Sun, D. (2003). Solving Karush-Kuhn-Tucker systems via the trust region and the conjugate gradient methods. *SIAM Journal on Optimization*, 14(2), 439–463.
- OPNET, <http://www.opnet.com/>.

28. Calvert, K., Doar, M., & Zegura, E. (1997). Modeling internet topology. *IEEE Communications Magazine*, 35(6), 160–163.
29. TIPHON: Definition of QoS Classes. ETSI TS 101 329-2, Tech. Rep., July 2000.

### Author Biographies



**Francesco Delli Priscoli** graduated in Electronic Engineering “summa cum laude” from the University of Rome “La Sapienza” in 1986. He received the Ph.D. in System Engineering from the same university in 1991. From 1986 to 1991 he worked in the “Studies and Experimentation” Department of Telespazio (Rome). Since 1991, he is working for the University of Rome “La Sapienza” (from 1991 to 2001 as a Researcher and since 2001 as

an Associate Professor) where, in October 2004, he obtained the qualification of “Full Professor”. At present, it holds the courses “Automatic Controls”, “System Theory”, and “Network Control and Management”. In the framework of his activity, he has researched in the nonlinear control theory and in the area of control-based resource management procedures for the third and fourth generation of mobile systems. He is the author of about 150 technical papers on the above topics appeared on major international reviews (about 50) and conferences (about 100). In 2000 he has been scientific consultant for the Italian Council of Ministers in the framework of the auction for the assignment of the Italian Universal Mobile Telecommunication System (UMTS) licenses. In 2002 he has been the Guest Editor (together with Prof. J. P. Thomesse) of the Special Issue on “Control

Methods for Telecommunication Networks” of Control Engineering Practice. He has been scientific responsible, for the University of Rome “La Sapienza”, of 14 projects financed by the European Union (fourth, fifth, and sixth framework programmes) and the European Space Agency (ESA), dealing with resource management for UMTS, and broadband terrestrial and satellite wireless systems. He is also a project evaluator for the European Commission.



**Dario Pompili** received his Ph.D. in Electrical and Computer Engineering from the Georgia Institute of Technology in June 2007 after working at the Broadband Wireless Networking Laboratory (BWN-Lab) under the direction of Prof. I. F. Akyildiz. In 2005, he was awarded Georgia Institute of Technology BWN-Lab Researcher of the Year for “outstanding contributions and professional achievements”. He had previously received his

“Laurea” (integrated B.S. and M.S.) and Doctorate Degrees in Telecommunications Engineering and System Engineering from the University of Rome “La Sapienza,” Italy, in 2001 and 2004, respectively, after working with the DIS Department, advised by Prof. F. Delli Priscoli. In September 2007, he joined the Electrical and Computer Engineering Department at Rutgers, The State University of New Jersey, as Assistant Professor. His research interests include ad hoc and sensor networks, underwater acoustic communications, wireless sensor and actor networks, and network optimization and control.