INT. J. CONTROL, 2003, VOL. 76, NO. 6, 618–634

Sturdy control of discrete communication networks. Part I: Population control

JACOB HAMMER†

The problem of optimizing the efficiency of a digital communication network through the use of feedback control is considered. An important goal is to reduce the effects of traffic uncertainties on network efficiency, especially in cases where statistical models of the traffic uncertainties are not available. Optimization is based on the notion of 'asymptotic efficiency', which characterizes the efficiency of high capacity networks and is introduced in this paper. This part of the paper concentrates on the optimal selection of the data admitted into the network.

1. Introduction

A discrete (or digital) communication network is a medium for the transport of discrete elements called *cells*. Discrete communication networks are used for the transmission of computer data, digitized voice, digitized video or multimedia data. The set of all cells that are in transit through a discrete communication network form the *network traffic*. As our daily experience on the roads suggests, traffic needs to be controlled in order to improve the efficiency of a network. The control of traffic is accomplished by a *traffic control algorithm*. The traffic control algorithms developed in the present paper aim to maximize the amount of traffic that can pass through the network.

An important issue that affects the development of traffic control algorithms is the random and varied nature of network traffic. Of particular consequence is the fact that some voluminous components of network traffic lack reliable statistical models. The lack of statistical models diminishes the benefits of employing statistical filtering techniques for the design of traffic control algorithms. Discussions of the difficulties involved with the statistical modelling of some common categories of data can be found in Eckberg (1979), Daigle and Langford (1986), Hut (1988), Paxson and Floyd (1994) and Schwartz (1996).

Digitized video and multimedia data constitute examples of important classes of network traffic for which comprehensive statistical models are not available (e.g., Schwartz (1996, Chapter 3) and the references mentioned there). The statistics of digitized video and multimedia data vary widely, depending on content, scenery, style and compression method. Considering that digitized video and multimedia data are expected to form a large fraction of network traffic, there is a need to develop traffic control methodologies that do not depend on detailed statistical models of the traffic. One of the objectives of the present paper is to address this need, and to develop a traffic control theory that does not depend on detailed statistical models of network traffic.

In its principles, the present theory is similar to the theory of robust control. Robust control theory was developed to overcome the effects on control systems of unmodelled disturbances, uncertainties and noises. The uncertainties that affect communication networks relate to the composition and the transmission rate of the network traffic and not to disturbances, but this is only a semantic difference, of course. Robust control techniques rely on well-characterized structural properties of the uncertainties, like known amplitude bounds; they do not require statistical models. Likewise, the traffic control theory developed in this paper depends only on well characterized structural properties of the network traffic, and does not require statistical models.

There is an important dissimilarity between the nature of the uncertainties considered in the theory of robust control, and the nature of the uncertainties that affect network traffic. While robust control concentrates mostly on the effects of relatively 'small' uncertainties with fixed amplitude bounds, traffic control must deal with large uncertainties whose amplitude bounds may vary with time. To acknowledge this distinction, we replace the adjective 'robust' by the adjective 'sturdy' in the present context, and refer to our current topic as *sturdy traffic control*. In brief, sturdy traffic control deals with the development of traffic control algorithms which operate under the influence of large uncertainties characterized by (possibly time varying) amplitude bounds; no statistical models are used. The fact that sturdy traffic control techniques do not require statistical models of network traffic becomes an important advantage when such models are not readily available.

Another important aspect of sturdy traffic control relates to the fact that it prevents data loss: no data is lost in transit through a network whose traffic is controlled by sturdy traffic control algorithms. In contrast, statistical traffic control algorithms are often designed to

Received 1 July 2000. Revised 1 August 2002. Accepted 1 February 2003. Communicated by Professor H. Kimura.

† Department of Electrical and Computer Engineering, P.O. Box 116130, University of Florida, Gainesville, FL 32611, USA. e-mail: hammer@mst.ufl.edu

International Journal of Control ISSN 0020–7179 print/ISSN 1366–5820 online © 2003 Taylor & Francis Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/0020717031000102961

permit data loss during certain rare traffic events (see, e.g. Atkins 1980, Golestani 1991 and Chang 1994). We show that in many common situations, sturdy traffic control leads to full network utilization without incurring any data loss. In other words, data loss is not a 'necessary evil' in the process of achieving high network efficiency.

Our discussion in the present paper is concerned with networks that can transmit very large volumes of traffic. Accordingly, we concentrate on the efficiency of the network in the limit, when the volume of traffic tends to infinity. We refer to this efficiency as *asymptotic efficiency*. Asymptotic efficiency is relevant to the large capacity networks currently used for long distance digital communication. As we shall see, considerations of asymptotic efficiency lead to scalable traffic control algorithms, which are easy to adjust as the communication network expands.

The effects of traffic uncertainties on the asymptotic efficiency of a discrete communication network are discussed in more detail in the second part of this paper (Hammer 2003). As it turns out, sturdy traffic control algorithms often make it possible to achieve full asymptotic efficiency (i.e. asymptotic efficiency of 1), even in cases where there are significant uncertainties about the traffic flow. This is achieved with no data loss, as sturdy traffic control algorithms permit no data loss.

1.1. *Some terminology*

As mentioned earlier, the elements transmitted through a discrete network are called *cells*. A cell is simply a packet of data. In many modern digital communication networks, all cells contain the same volume of data (i.e. the same number of bits). Accordingly, we restrict our attention in this report to digital communication networks with cells of fixed data volume. When all cells contain the same number of bits, the volume of data passing through the network is directly determined by the number of cells.

A collection of cells that forms one contiguous and complete object (or data record) is named a *call*. Each cell carries information that identifies the call to which the cell belongs, as well as the relative position of the cell data within the call. This information makes it possible to rebuild a call from a mix of its cells. Therefore, it is not necessary to keep the cells in order as they pass through the network.

The calls entering the network are divided into a number of categories called *classes*. Examples of call classes include digitized phone calls, computer file transfers, digitized video signals, multimedia calls and network control signals. The latter are signals generated by network controllers to help manage the traffic. For most call classes, the cell transmission rate varies randomly during a call.

Different classes of calls usually impose different demands, or *service requirements*, regarding the fidelity of their transmission through the network (e.g. ATM Forum 1997, Handel *et al.* 1994). From the point of view of our present discussion, the most critical service requirement is the maximal delay (or jitter) a cell may experience while passing through the network. Note that some call classes, especially real-time call classes like digitized video and digitized phone calls, are very sensitive to transmission delays and jitter. Other call classes, like the class of computer data file transfers, are relatively insensitive to transmission delays.

A call originates from a *source*, and its destination is called a *target*. Sources include computers, digitized phone devices, digitized video devices, environmental sensors, control devices, security devices and others. A discrete communication network usually has a large number of independent sources and targets. There is, of course, a limit on the number of cells the network can transmit during each unit of time; this limit is called the *network capacity*.

Inasmuch as calls enter the network from independent sources and with random transmission rates, there is a possibility that the number of cells entering the network exceeds network capacity. This condition is called *network congestion*. An important role of traffic control algorithms is to prevent or resolve conditions of network congestion. Network congestion carries the risk of cell loss; however, the sturdy traffic control algorithms discussed later in this paper and in its sequel (Hammer 2003) resolve conditions of network congestion without incurring cell loss.

A contiguous segment of a network is called a *transmission link*; it is simply a segment of cable or a wireless link. A network also has a *backbone* (a very large large long-distance conduit), fed by many shorter transmission links. All transmission links are connected to each other and to the backbone by *routers*; a router is a computerized switch that can control network traffic. Routers contain memory devices to temporarily store passing cells; such devices are called *buffers*. Buffers are also present in the sources and the targets of the network.

Buffers can store cells during periods when the cell transmission rate exceeds network capacity, and release their stored cells during periods when the incoming cell flow is below network capacity. In this way, buffers can shape the traffic flow to improve network efficiency. The operation of the buffers is controlled by the traffic control algorithm. A cell stored in a buffer experiences, of course, a delay. This delay, combined with all other buffering and propagation delays encountered by the cell while passing through the network, must not exceed the maximal delay permitted by the cell's service requirements.

The backbone is the most costly component of the network, and our objective in this paper is to optimize its utilization. We assume that the combined capacity of the links feeding the backbone is larger than the backbone capacity, so that the capacities of the links do not constitute a limiting factor in the process of optimizing backbone utilization.

This part of the paper concentrates on finding conditions under which asymptotic backbone efficiency of 1 can be achieved without buffering. These conditions characterize the buffering output required for achieving asymptotic efficiency of 1; they are used in the second part of this paper (Hammer 2003) to develop buffering algorithms.

The possibility of achieving asymptotic efficiency of 1 depends, to a large extent, on the way in which the waveforms of the different calls passing through the network relate to each other. We say that the call waveforms form a *complete family* if there is a linear combination with non-negative coefficients of these waveforms which equals a non-zero constant. It is shown in § 4 that asymptotic efficiency of 1 can be achieved if and only if the waveforms of the calls passing through the network form a complete family of calls.

The paper is organized as follows. Section 2 introduces the basic setup and notation. The notion of asymptotic efficiency, which serves as our basic optimization criterion, is developed in § 3. Section 4 characterizes the conditions under which the network can achieve asymptotic efficiency of 1, while § 5 considers families of calls for which asymptotic efficiency of 1 cannot be achieved. Finally, § 6 examines the effects on network efficiency of limitations on the call supply.

2. Basics

The flow of cells through a network is represented by a sequence of integers. Each element of the sequence represents the number of cells that flow through a point of the network during a specified time interval of $\Delta > 0$. For an integer $k \geq 1$, the notation v_k indicates the number of cells that flow through the point v of the network during the time interval $((k-1)\Delta, k\Delta]$. We shall refer to the interval $((k-1)\Delta, k\Delta]$ as *step* k. Note that this interpretation implies that, for networks turned on at the time zero, the first significant step is $k = 1$. The length of the time interval Δ is selected based on the network flow dynamics; it should be short enough, so that the flow of cells does not vary significantly over a time interval of Δ, and so that network delays can be approximated by integer multiples of Δ. In this way, we can regard the entire network as a discrete time system acting on sequences of integers.

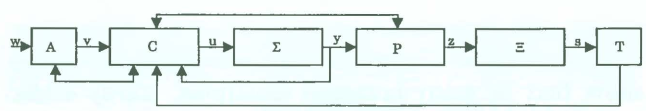

Figure 1.

A typical network can be described by figure 1.

Here, w represents the pool of calls requesting admission to the network and A represents the gate to the network. The gate A implements the *call admission process*, which selects the calls that are allowed to enter the network. With the gate is associated a buffer and controller unit C. The symbol Σ represents the short network links that connect the gate to the backbone Ξ; it is connected to the backbone by the router P, which includes a buffer and controller. Finally T, represents the target with its controller and buffer. The operation of the buffers and controllers A, C, and P is controlled by the traffic control algorithm. The diagram is simplified by representing a single link between any two points, while in reality many links may exist.

The signals v, u, y, z and s of figure 1 are sequences of non-negative integers. We denote by v_k, u_k, y_k, z_k and s_k their respective values at (the end of) step k. Each such value represents the number of cells passing through the respective point of the network during step k. Assuming the network is turned on at the time zero, we take $k \geq 1$.

The network link Σ induces a delay of κ steps, so we have

$$y_k = u_{k-\kappa}$$

The sequence $z := \{z_k\}_{k \geq 1}$ denotes the input signal to the backbone Ξ. Let $\phi > 0$ be the maximal number of cells that can pass through the backbone during a time interval of length Δ. Somewhat abusing the standard terminology, we call ϕ the *capacity* of the backbone. Then, the backbone input sequence z must satisfy the requirement

$$0 \leq z_k \leq \phi, \qquad k = 1, 2, \ldots$$

For the sake of simplicity, we ignore in this discussion any capacity limitations of the network segment Σ. In practice, Σ represents a large number of links feeding the backbone, and the combined capacity of these links usually exceeds backbone capacity. We concentrate on the optimization of backbone flow, to maximize the utilization of the most costly part of the network.

2.1. *The calls*

We turn now to the description of the calls passing through the network. In general terms, we consider calls of finite duration T whose flow rates are bounded by piecewise constant sequences. Let $q, T \geq 1$ be two integers, and partition the interval $[1, T]$ into q disjoint subintervals called *segments*. When $q \geq 2$, the segments are of the form $I_1 := [1, t_1], I_2 := [t_1 + 1, t_2], \ldots, I_q := [t_{q-1} + 1, T]$; when $q = 1$, there is the single segment $I_1 := [1, T]$. We denote by $\lambda_i := t_i - t_{i-1}$ the number of steps included in the segment $I_i, i = 2, \ldots, q$, with $\lambda_1 := t_1$.

Given a list of q integers $\varphi(1), \varphi(2), \ldots, \varphi(q)$, we define the piecewise constant sequence

$$\varphi_k := \begin{cases} 0 & \text{for } k \leq 0 \\ \varphi(1) & \text{for } 1 \leq k \leq t_1, \\ \varphi(2) & \text{for } t_1 + 1 \leq k \leq t_2, \\ \vdots & \\ \varphi(q) & \text{for } t_{q-1} + 1 \leq k \leq T \\ 0 & \text{for } T + 1 \leq k \end{cases} \tag{1}$$

Here, $\varphi(j)$ is the constant value of the sequence φ over the segment I_j. The integers $t_1, t_2, \ldots, T$ are called the *switching times* of the sequence. Note that by using segments of length 1, every sequence with finite support can be represented in the form (1). We refer to the interval $[1, T]$ as the *call cycle*. The integer T usually represents a common multiple of the durations of the calls of interest, so all calls of interest become compatible with the call cycle. In the case of very long calls, T may indicate a convenient breakpoint of a call. We refer to the interval $[T + 1, 2T]$ as the *second call cycle*, and so on.

A call c is represented as a sum of two piecewise constant sequences, one of which represents the nominal flow of the call, while the other represents the uncertainty. Specifically

$$c = \chi + v$$

where χ and v are piecewise constant sequences over the partition $\{I_1, \ldots, I_q\}$. The nominal flow is represented by χ, which is given and fixed; the uncertainty is represented by v, which may vary from one sample of the call c to another. The only *a priori* information available about the uncertain part v is an amplitude bound $\rho \geq 0$, so that

$$0 \leq v(j) \leq \rho, \qquad j = 1, 2, \ldots, q$$

no statistical model of the uncertainty is presumed.

The total delay τ_t incurred by a cell while passing through the network is determined by two factors: the physical propagation delay through the network τ_p and the delay τ_b incurred while the cell is stored in the buffer system, so that

$$\tau_t = \tau_p + \tau_b$$

Letting τ_c denote the maximal delay that a cell of the given call class may incur, we obtain the requirement

$$\tau_p + \tau_b \leq \tau_c \tag{2}$$

In many cases, the physical propagation delay τ_p can be regarded as approximately fixed. Then, equation (2) can be transformed into a bound on the maximal time a cell may linger within the buffer system. Define the quantity

$$\tau := \tau_c - \tau_p$$

which is determined by the call class and the length of the various links. Then, we obtain the bound

$$\tau_b \leq \tau$$

on the maximal time a cell of the considered call class may linger in the buffer system. We call τ the *buffering delay bound*. The buffering delay bound may vary with the class and the distance between the source and the target.

As mentioned earlier, the calls attempting entry into the network are classified into *call classes*, depending on their characteristics and on their service requirements. We assume that there are m different call classes $C^1, \ldots, C^m$. The calls of each call class share the same nominal waveform, the same uncertainty characteristics, and the same buffering delay bound. A call of the class C^i will be denoted by

$$c^i = \chi^i + v^i \tag{3}$$

where χ^i represents the nominal part and v^i represents the uncertain part. The value of the call c^i at the step k is written as $c^i_k = \chi^i_k + v^i_k$, while the value of the call c^i on the segment I_j is written as $c^i(j) = \chi^i(j) + v^i(j)$. The maximal permissible buffering delay for a call of the class C^i is denoted by $\tau(i)$. An *empty call class* is a call class that is zero over the entire call cycle $[1, T]$.

The term *call pool* refers to the population of calls awaiting admission into the network, i.e. to the available supply of calls. Of course, one needs to assume that the peak flow capacity required to transmit the entire call pool exceeds backbone capacity; otherwise, all waiting calls can be transmitted directly, and there is no place for optimization. The optimization process depends, among other factors, on the composition of the call pool. In the following sections we consider the optimization of network performance under various scenarios regarding the call pool composition.

For the sake of simplicity, we shall assume that the admission process into the backbone is performed at the compensator P, and that admitted calls start entering the backbone at the time step $k = 1$. The first assumption means that the call pool is at the entrance of P, while the second assumption means that there is an immediate response to the admission decision. In many practical applications, the propagation delay from C to P is relatively short, in which case the difference between admission at C and admission at P is minor.

2.2. *Buffering*

We turn now to a preliminary discussion of the buffering mechanism of the compensator P of figure 1. Being located at the gate of the backbone, the compensator P serves as the main flow control device of the backbone. Recall that one of the basic requirements imposed on a sturdy discrete communication network is that there be no loss of cells; consequently, no cells can be lost within the compensator P. This requirement leads to a recursive representation of the buffering mechanism of P, as follows.

Let the initial condition be that the buffers of P are empty at the initial step $k = 1$ and consider a step $k \geq 1$. Referring to figure 1, the number of cells that have entered P by the end of step k is given by $\sum_{i=1}^{k} y_i$, while the number of cells that have exited P by the end of step k is given by $\sum_{i=1}^{k} z_i$. Considering that there is no loss of cells within P, the cell population ζ_{k+1} stored in the buffers of P at the start of step $k+1$ is given by

$$\zeta_{k+1} = \sum_{i=1}^{k} y_i - \sum_{i=1}^{k} z_i, \qquad \zeta_1 = 0 \tag{4}$$

This implies, in particular, that we must have

$$\sum_{i=1}^{k} y_i - \sum_{i=1}^{k} z_i \geq 0 \qquad \text{for all } k = 1, 2, \ldots$$

To obtain a recursive model of P, define the variable

$$b_k := z_k - y_k$$

which is the number of new cells stored in P during step k. Then, equation (4) can be rewritten in the recursive form

$$\zeta_{k+1} = \zeta_k + b_k, \qquad k = 1, 2, \ldots$$

so that b_k serves as the control variable of P. In order to control the backbone flow, the traffic control algorithm assigns the values of $b_k, k = 1, 2, \ldots$.

In the sequel, we assume that the buffer of P is a *random access buffer*, i.e. that stored cells can be retrieved from the buffer of P in any desired order, irrespective of the order in which they were stored. We also assume that the buffer is large enough, so that buffer capacity is not a limiting factor. These assumptions are valid in many modern day routers.

3. Asymptotic efficiency: preliminary considerations

The present section introduces one of the main notions of our discussion: asymptotic efficiency. This notion relates to the flow of traffic through high capacity backbones. In qualitative terms, asymptotic efficiency indicates the efficiency of backbone utilization in the limit, when backbone capacity tends to infinity; 'efficiency' here refers to the fraction of backbone capacity filled by the flow. Thus, asymptotic efficiency of 1 means that the fraction of unused backbone capacity tends to zero, as backbone capacity grows to infinity. The notion of asymptotic efficiency will lead us to traffic control algorithms that are *scalable* in the sense that their basic mode of operation does not change when backbone capacity is increased. This facilitates effortless upgrades of system capacity.

The process of optimizing the utilization of the backbone is accomplished by two means: (i) by selecting the calls that are admitted into the network, and (ii) by reshaping the waveforms of the admitted calls. The reshaping of call waveforms is achieved through buffering, by storing some cells of a call and releasing them later. Of course, reshaping must be performed without violating any service requirements. In our discussion, the optimization process is performed mainly by the compensator P; the compensator C usually serves as an auxiliary storage facility for buffered cells.

We start our discussion by examining network optimization under a number of simplifying assumptions. This will help us gain insight into the basic optimization phenomena, with a minimum of technical complications. The first simplifying assumption we make is that all calls are uncertainty free. Referring to (3), each call class $C^i, i = 1, \ldots, m$, consists of calls of the form

$$c^i = \chi^i \quad \text{and} \quad v^i = 0$$

We also assume in this preliminary examination that there are no limitations on the call supply. In practical terms this means that for each call class $C^i, i = 1, \ldots, m$, the number of calls contained in the call pool is larger than the number of calls that can be simultaneously transmitted through the backbone.

Another assumption we make in our preliminary investigation is that no buffering action is performed on the calls. In such case, backbone efficiency can be controlled only through the call admission process, by selecting the calls that best fill the backbone. The analysis of performance without buffering is, in fact, an analysis of the output of P, after all buffering has been completed. This preliminary analysis leads us to a characterization of the optimal output of the compensator P. The issues of limited call supply, buffering, and call uncertainties are considered later in this part and in the second part (Hammer 2003) of the paper.

Under our simplifying assumptions, consider the case of a network whose call pool consists of m classes $C^1, \ldots, C^m$ of piecewise constant calls over the partition $\{I_1, \ldots, I_q\}$ of the integer interval $[1, T]$. All calls of the class C^i have the waveform c^i, and c^i_k is the value of c^i at the step k. Assume that none of the calls is completely empty, i.e. that for each $i \in \{1, \ldots, m\}$ there is an integer $k \in \{1, \ldots, T\}$ such that

$$c_k^i > 0 \tag{5}$$

Now, let $\alpha_i \geq 0$ be the number of calls of the class c^i admitted into the backbone. Then, the total number of cells injected into the backbone at the step k is given by

$$z_k = \sum_{i=1}^{m} \alpha_i c_k^i$$

The integers $\alpha_1, \ldots, \alpha_m$ are determined by the traffic control algorithm through the admission process, and are called the *call populations*. The assumption of unlimited call pool means that there is no restriction on $\alpha_1, \ldots, \alpha_m$. The *amplitude* $A(z)$ of the cell stream z is defined by

$$A(z) := \max\{z_k : k = 1, \ldots, T\}$$

We are now ready to introduce our main optimization criterion. Let ϕ be the backbone capacity. Then, the maximal number of cells the backbone can carry during the call cycle $[1, T]$ is given by $T\phi$. Now, for the backbone capacity ϕ, let $\alpha_1(\phi), \ldots, \alpha_m(\phi)$ be the call populations admitted into the backbone. This notation brings into the forefront the obvious fact that the call populations depend on the backbone capacity. The total number of cells entering the backbone at the step k is then given by

$$z_k(\phi) = \sum_{i=1}^{m} \alpha_i(\phi) c_k^i \tag{6}$$

It will be convenient to refer to the waveform $z(\phi)$ as the 'traffic control algorithm', and regard it as a rule that assigns populations $\alpha_1(\phi), \ldots, \alpha_m(\phi)$ for each backbone capacity $\phi \geq 1$. Then, the *efficiency* $\eta(z(\phi))$ of the traffic control algorithm $z(\phi)$ is defined by

$$\eta(z(\phi)) := \frac{\sum_{k=1}^{T} z_k(\phi)}{T\phi} \tag{7}$$

It indicates the fraction of the available backbone throughput that is being utilized by the traffic control algorithm $z(\phi)$. Clearly

$$0 \leq \eta(z(\phi)) \leq 1$$

The *asymptotic efficiency* $\eta(z)$ of the traffic control algorithm $z(\cdot)$ is defined by

$$\eta_\infty(z) := \lim_{\phi \to \infty} \eta(z(\phi))$$

The asymptotic efficiency approximates the efficiency of the traffic control algorithm $z(\phi)$ when executed on backbones with large capacity ϕ. A traffic control algorithm with asymptotic efficiency of 1 utilizes almost the entire throughput potential of the backbone, when used on large backbones. Maximization of the asymptotic efficiency forms the basic optimization criterion in our discussion. The following statement characterizes the conditions for achieving asymptotic efficiency of 1. (Recall that $c^i(j)$ is the value of the call c^i on the segment I_j.)

Theorem 1: *Let $F := \{c^1, \ldots, c^m\}$ be a family of piecewise constant call classes over the partition $\{I_1, \ldots, I_q\}$. Then, the following two statements are equivalent:*

(i) *There is a traffic control algorithm with asymptotic efficiency of 1 for the family F.*

(ii) *There is a list of non-negative integers $\gamma_1, \ldots, \gamma_m$ such that the linear combination $c(j) := \sum_{i=1}^{m} \gamma_i c^i(j)$ is a non-zero constant function of $j, j = 1, \ldots, q$.*

Proof: Assume first that there is a list of non-negative integers $\gamma_1, \ldots, \gamma_m$ for which the linear combination

$$c(j) = \sum_{i=1}^{m} \gamma_i c^i(j)$$

is a non-zero constant function of j, and denote by $c := c(j)$ the corresponding constant value. Using the integer division algorithm, we can write

$$\phi = mc + r$$

where m and r are non-negative integers and $0 \leq r < c$. Consider now the traffic control algorithm $z(\phi)$ defined by setting $\alpha_i(\phi) := m\gamma_i, i = 1, \ldots, q$, in (6). Then

$$z_k(\phi) = mc \qquad \text{for all } k = 1, \ldots, T$$

so that $\sum_{k=1}^{T} z_k(\phi) = Tmc$. Substituting into (7), we get

$$\eta(z(\phi)) = \frac{Tmc}{T\phi} = \frac{T\phi - Tr}{T\phi} = 1 - \frac{r}{\phi}$$

Letting $\phi \to \infty$, and taking into account the fact that $0 \leq r < c$ for all ϕ, it follows that

$$\lim_{\phi \to \infty} \eta(z(\phi)) = 1,$$

and the proposed traffic control algorithm yields asymptotic efficiency of 1.

Conversely, assume that there is a traffic control algorithm $z(\phi)$ that yields asymptotic efficiency of 1, so that $\eta_\infty(z) = \lim_{\phi \to \infty} \eta(z(\phi)) = 1$. Given a sequence $\phi(1), \phi(2), \ldots \to \infty$ of backbone capacities, the algorithm assigns a sequence of integer lists $\{\alpha_1(n), \ldots, \alpha_m(n)\}$, $n = 1, 2, \ldots$, where $\alpha_i(n) \geq 0$ is the number of calls of the class C^i admitted into the backbone when the backbone capacity is $\phi(n)$. Then, when the backbone capacity is $\phi(n)$, the number of cells injected by the traffic control algorithm into the backbone during step k is given by

$$z_k(n) := \sum_{i=1}^{m} \alpha_i(n) c_k^i$$

The total number of cells injected during the entire interval $[1, T]$ is

$$\varphi(n) := \sum_{k=1}^{T} z_k(n) \tag{8}$$

Let

$$A(n) := \max_{k=1,\dots,T} z_k(n) = \max_{k=1,\dots,T} \sum_{i=1}^{m} \alpha_i(n) c_k^i \tag{9}$$

be the largest number of cells injected into the backbone during one step (i.e. the flow amplitude). Clearly, the backbone capacity must satisfy

$$\phi(n) \geq A(n), \qquad n = 1, 2, \dots \tag{10}$$

The efficiency achieved by this traffic control algorithm for a backbone capacity $\phi(n)$ is

$$\eta(\phi(n)) = \frac{\varphi(n)}{T\phi(n)} \tag{11}$$

By assumption, this traffic control algorithm achieves an asymptotic efficiency of 1; consequently

$$\lim_{n\to\infty} \eta(\phi(n)) = 1 \tag{12}$$

Forming the rational number

$$\eta_n := \frac{A(n)}{\phi(n)}$$

and using (10) and (11), we obtain

$$\eta(\phi(n)) \leq \eta_n \leq 1$$

Combining this with (12) implies that

$$\lim_{n\to\infty} \eta_n = 1$$

Next, (10) and (9) imply that

$$0 \leq \frac{\sum_{i=1}^{m} \alpha_i(n) c_k^i}{\phi(n)} \leq 1, \quad k = 1, \dots, T, \quad n = 1, 2, \dots \tag{13}$$

Since $\{[\sum_{i=1}^{m} \alpha_i(n) c_k^i]/\phi(n)\}_{n=1}^{\infty}$ is a bounded set of real numbers by (13), the completeness property implies that it must contain a convergent subsequence. Consequently, there is a subsequence of lists

$$S = \{\alpha_1(n_e), \dots, \alpha_m(n_e) : e = 1, 2, \dots\} \tag{14}$$

for which the limit

$$a_k := \lim_{e\to\infty} \frac{\sum_{i=1}^{m} \alpha_i(n_e) c_k^i}{\phi(n_e)}$$

exists for all $k = 1, \dots, T$. In view of (13), one has

$$0 \leq a_k \leq 1, \quad k = 1, \dots, T \tag{15}$$

Further, by (8), (11) and (12), we have

$$\lim_{e\to\infty} \eta(\phi(n_e)) = \frac{a_1 + \cdots + a_T}{T} = 1$$

By (15), this yields that

$$a_k = 1, \qquad k = 1, \dots, T$$

so that a_k is constant over the entire interval $[1, T]$. Denoting by $a(j)$ the (constant) value of a_k over the segment I_j, we can rewrite the last equation in the form

$$a(j) = 1, \qquad j = 1, \dots, q \tag{16}$$

Let us restrict ourselves now to the sequence S, and let us abbreviate the notation of (14) by removing the subscript e, to write

$$S = \{\alpha_1(n), \dots, \alpha_m(n) : n = 1, 2, \dots\}$$

Then, we can rewrite (16) in the form

$$\lim_{n\to\infty} \frac{\sum_{i=1}^{m} \alpha_i(n) c^i(j)}{\phi(n)} = 1, \qquad j = 1, \dots, q \tag{17}$$

Recalling that all calls are constant on segments of the partition $\{I_1, \dots, I_q\}$, we can rewrite (13) in the form

$$0 \leq \frac{\sum_{i=1}^{m} \alpha_i(n) c^i(j)}{\phi(n)} \leq 1, \qquad j = 1, \dots, q, \quad n = 1, 2, \dots \tag{18}$$

Now, since all quantities are positive, equation (18) implies that

$$0 \leq \frac{\alpha_i(n) c^i(j)}{\phi(n)} \leq 1$$

for all $i = 1, \dots, m, j = 1, \dots, q$, and $n = 1, 2, \dots$, so that $\{\alpha_i(n) c^i(j)/\phi(n)\}_{n=1}^{\infty}$ is a bounded set of real numbers. The completeness property of the real numbers implies again that this set must contain a convergent subsequence. Consequently, there is a subsequence $S' = \{\alpha_1(n_h), \dots, \alpha_m(n_h) : h = 1, 2, \dots\}$ of S such that the limits

$$\lim_{h\to\infty} \frac{\alpha_i(n_h) c^i(j)}{\phi(n_h)}$$

exist for all $i = 1, \dots, m$, and all $j = 1, \dots, q$. Combining this with the fact that for each $i = 1, \dots m$ there is at least one value of j for which $c^i(j) \neq 0$ (see (5)), we conclude that the limits

$$M_i := \lim_{h\to\infty} \frac{\alpha_i(n_h)}{\phi(n_h)} \tag{19}$$

exist for all $i = 1, \dots, m$, where $M_1, \dots, M_m$ are real non-negative numbers.

Now specializing (17) to the subsequence S', and using (19), we obtain

$$\sum_{i=1}^{m} M_i c^i(j) = 1 \qquad \text{for all } j = 1, \dots, q$$

But then, invoking Lemma 1 below, it follows that there are rational numbers $R_1, \ldots, R_m \geq 0$ such that

$$\sum_{i=1}^{m} R_i c^i(j) = 1 \qquad \text{for all } j = 1, \ldots, q \tag{20}$$

Finally, since $R_1, \ldots, R_m$ are non-negative rational numbers, there are integers $b_i \geq 0$ and $d_i > 0$ such that $R_i = b_i/d_i$, $i = 1, \ldots, m$. Defining the integers $d = d_1 d_2 \ldots d_m$ and $\gamma_i := R_i d \geq 0$, $i = 1, \ldots, m$, and multiplying (20) by d, we obtain

$$\sum_{i=1}^{m} \gamma_i c^i(j) = d \qquad \text{for all } j = 1, \ldots, q$$

This shows that (i) implies (ii), and our proof concludes □

The following is a technical result used in the proof of Theorem 1.

Lemma 1: *Let $c^1, \ldots, c^m$, f be non-negative and rational valued piecewise constant functions over the partition $\{I_1, \ldots, I_q\}$ of the integer interval $[1, T]$. Assume there are real numbers $M_1, \ldots, M_m \geq 0$ such that $\sum_{i=1}^{m} M_i c^i = f$. Then there also are rational numbers $R_1, \ldots, R_m \geq 0$ such that $\sum_{i=1}^{m} R_i c^i = f$.*

Proof: By assumption, there are numbers $M_1, \ldots, M_m \geq 0$ satisfying

$$\sum_{i=1}^{m} M_i c^i(j) = f(j), \qquad j = 1, \ldots, q \tag{21}$$

where $c^i(j)$ and $f(j)$ are rational numbers for all $i = 1, \ldots, m, j = 1, \ldots, q$. Now, if $f = 0$ for all $j = 1, \ldots, q$, then we can take $R_i = 0$ for all $i = 1, \ldots, m$, and Lemma 1 is valid. Otherwise, there must be at least one non-zero member of the set $M := \{M_1, \ldots, M_m\}$. Omit the functions c^i that correspond to zero members of the set M; reduce the value of m to the number of remaining functions c^i, and renumber the functions accordingly. Then, we can assume without loss of generality that $M_i > 0$ for all $i = 1, \ldots, m$.

Now, consider the following system of q linear equations with the indeterminate $x_1, \ldots, x_m$

$$\left.\begin{aligned} c^1(1)x_1 + \cdots + c^m(1)x_m &= f(1) \\ c^1(2)x_1 + \cdots + c^m(2)x_m &= f(2) \\ &\vdots \\ c^1(q)x_1 + \cdots + c^m(q)x_m &= f(q) \end{aligned}\right\} \tag{22}$$

This reduces to (21) upon setting $x_i = M_i$, $i = 1, \ldots, q$. Define the quantities

$$\begin{gathered} M := \begin{pmatrix} M_1 \\ \vdots \\ M_m \end{pmatrix} \quad x := \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} \quad f := \begin{pmatrix} f(1) \\ \vdots \\ f(q) \end{pmatrix} \\ C := \begin{pmatrix} c^1(1) & \cdots & c^m(1) \\ \cdot & \cdots & \cdot \\ c^1(q) & \cdots & c^m(q) \end{pmatrix} \quad R := \begin{pmatrix} R_1 \\ \vdots \\ R_m \end{pmatrix} \end{gathered} \tag{23}$$

Then, equation (22) can be rewritten in the form

$$Cx = f \tag{24}$$

The fact that there are numbers $M_1, \ldots, M_m$ satisfying (21) means that $x = M$ is a solution of (24). This implies that the vector f is contained in the image (or column span) of the matrix C, i.e. $f \in \operatorname{Im} C$.

Next, since all coefficients of the matrices C and f are rational numbers, we can consider (24) as a system of linear equations over the field of rational numbers. Since $f \in \operatorname{Im} C$, there is a rational vector r satisfying the equation $Cr = f$, so that a rational solution exists. We show next that r can be selected with non-negative entries.

Let $n := \dim \operatorname{Ker} C$ be the dimension of the kernel of the matrix C. When $n = 0$, equation (24) has a unique solution, which implies that $M = r$. Thus, M is itself a rational vector in this case, and Lemma 1 is satisfied with $R = M$. Consider now the case $n \geq 1$. Regarding (24) as an equation over the field of rational numbers, let $b^1, \ldots, b^n$ be rational vectors that form a basis of $\operatorname{Ker} C$. Then, every solution x of the equation $Cx = f$ over the field of rational numbers is of the form $x = r + \sum_{i=1}^{n} \varepsilon_i b^i$, where $\varepsilon_1, \ldots, \varepsilon_n$ are rational numbers. But (24) can, of course, also be regarded as a linear equation over the field of real numbers. Then, the same facts imply that every real vector solution ξ of the equation $C\xi = f$ is of the form $\xi = r + \sum_{i=1}^{n} \delta_i b^i$, where $\delta_1, \ldots, \delta_n$ are real numbers. In particular, since $x = M$ is a solution of (24), there are real numbers $d_1, \ldots, d_n$ satisfying $M = r + \sum_{i=1}^{n} d_i b^i$.

Let

$$\mu := \min\{M_1, \ldots, M_m\}$$

be the smallest entry of the vector M, let $b^i_1, \ldots, b^i_m$ be the entries of the vector $b^i, i = 1, \ldots, n$ and let

$$b := \max\{|b^i_j| : i = 1, \ldots, n, j = 1, \ldots, m\}$$

be the largest absolute value of an entry of the vectors $b^1, \ldots, b^n$. Recalling that all entries of M are strictly positive, we have $\mu > 0$. Considering that the field of real numbers is the closure of the field of rational numbers, there is, for each $i = 1, \ldots, n$, a real number β_i satisfying the following conditions: (i) the difference

$(d_i - \beta_i)$ is a rational number; and (ii) $|\beta_i| < \mu/(nb)$. Define the vector

$$R := r + \sum_{i=1}^{n}(d_i - \beta_i)b^i$$

This vector is clearly a solution of (24), and it is a rational vector since $r, b^1, \ldots, b^n$ and $(d_1 - \beta_1), \ldots, (d_n - \beta_n)$ are all rational. Furthermore, since

$$R = M - \left(\sum_{i=1}^{n}\beta_i b^i\right)$$

it follows by the definition of $\beta_1, \ldots, \beta_n$ that all entries of R are positive. Thus, R satisfies the requirements of our Lemma, and our proof concludes. □

Definition 1: A family of call classes $F = \{c^1, \ldots, c^m\}$ is *complete* if there are integers $\alpha_1, \ldots, \alpha_m \geq 0$ such that the linear combination $\sum_{i=1}^{m}\alpha_i c^i = c$ is a non-zero constant function over the interval $[1, T]$.

In view of Theorem 1, asymptotic efficiency of 1 is possible if and only if the calls entering the backbone constitute a complete family of call classes. As a result, complete families are the only families of call classes capable of utilizing the entire capacity of large backbones, and whence they are of basic significance in our present context.

4. Complete families of calls

When the calls entering the backbone form a complete family, asymptotic efficiency of 1 can be obtained simply by a proper selection of the call populations. Indeed, let $F = \{c^1, \ldots, c^m\}$ be a complete family of call classes. Let $V^m(F)$ be the set of all lists of m non-negative integers $(\beta_1, \ldots, \beta_m)$ for which the linear combination $\sum_{i=1}^{m}\beta_i c^i$ is constant and non-zero over the entire interval $[1, T]$. We refer to $V^m(F)$ as the *set of constancy coefficients* of the family F. The definition of a complete family implies that $V^m(F) \neq \emptyset$. Consider the following traffic control algorithm.

Population control for complete families: Let $F = \{c^1, \ldots, c^m\}$ be a complete family of call classes, let $V^m(F)$ be its set of constancy coefficients, and let ϕ be the backbone capacity. Choose a list $(\beta_1, \ldots, \beta_m) \in V^m(F)$, and let $c := \sum_{i=1}^{m}\beta_i c^i$ be the corresponding non-zero constant value. Using the integer division algorithm, write

$$\phi = m(\phi)c + \rho$$

where $m(\phi) \geq 0$ is an integer and $0 \leq \rho < \phi$. Define the call populations

$$\alpha_i(\phi) := m(\phi)\beta_i, \qquad i = 1, \ldots, m$$

The traffic control algorithm is then given by

$$z(\phi) := \sum_{i=1}^{m}\alpha_i(\phi)c_i \tag{25}$$

The following statement, which is a direct consequence of the proof of Theorem 1, shows that, for complete families of calls, asymptotic efficiency of 1 can be achieved simply through admission control by using (25).

Corollary 1: *Let $F = \{c^1, \ldots, c^m\}$ be a complete family of calls entering the backbone, let $V^m(F)$ be its set of constancy coefficients, and let ϕ be the backbone capacity. Then, the traffic control algorithm* (25) *achieves asymptotic efficiency of 1.*

In view of Corollary 1, it is important to develop tools to determine whether a given family of call classes is complete or not. One way of making such a determination is by examining the solutions of a system of linear equations, as described in the following statement. (For a vector $a = (a_1, \ldots, a_m)$, denote by a^{T} the transposed vector, and write $a \geq 0$ whenever $a_i \geq 0$ for all $i = 1, \ldots, m$.)

Proposition 1: *Let $F = \{c^1, \ldots, c^m\}$ be a family of piecewise constant call classes over the partition $\{I_1, \ldots, I_q\}$. Let C be the $q \times m$ matrix associated with the family F as in* (23). *Then, the family F is complete if an only if the linear equation $Cx = (1, \ldots, 1)^{\mathrm{T}}$ has a solution $x \geq 0$.*

Proof: Assume first that F is complete. Then, there are integers $\alpha_1, \ldots, \alpha_m \geq 0$ such that $\sum_{i=1}^{m}\alpha_i c^i = c$, where $c > 0$ is a constant. Assembling the rational numbers $a_i := \alpha_i/c$ into the vector $a := (a_i, \ldots, a_m)^{\mathrm{T}}$, we obtain a solution $x = a$ of the equation $Cx = (1, \ldots, 1)^{\mathrm{T}}$, with $x \geq 0$.

Conversely, assume that the equation $Cx = (1, \ldots, 1)^{\mathrm{T}}$ has a solution $x = (x_1, \ldots, x_m)^{\mathrm{T}} \geq 0$. In view of Lemma 1, x can be selected as a vector of rational numbers. There are then integers $b_i \geq 0$ and $d_i > 0$ such that $x_i = b_i/d_i$, $i = 1, \ldots, m$. Defining the integers $d := d_1 d_2 \ldots d_m > 0$ and $\beta_i := x_i d \geq 0$, $i = 1, \ldots, m$, it follows that $\sum_{i=1}^{m}\beta_i c^i = d$, a constant. This shows that F is a complete family of call classes. □

Proposition 1 provides a simple test for completeness of a family of calls. More intuitive insight into the notion of completeness can be obtained by inspecting call values on segments, as follows. Consider, for example, the case of a family of call classes $F = \{c^1, \ldots, c^m\}$ over a two segment partition $\{I_1, I_2\}$. We say that two calls c^i, $c^j \in F$ form a *reversed pair* if

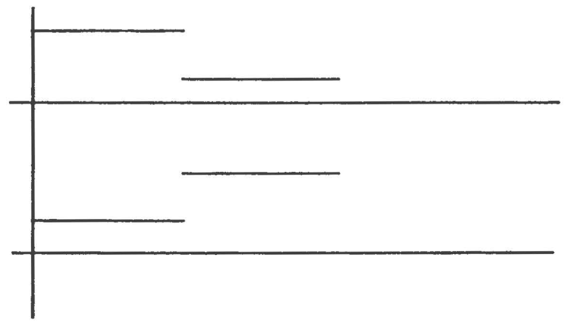

Figure 2.

$c^i(1) \leq c^i(2)$ while $c^j(1) \geq c^j(2)$; or if $c^i(1) \geq c^i(2)$ while $c^j(1) \leq c^j(2)$. Figure 2 demonstrates a reversed pair.

It follows from Proposition 2 below that a family of call classes over a two segment partition is complete if and only if it contains a reversed pair. This observation provides insight into the relationship between individual call values and the property of completeness.

Now, assume that $q \geq 2$. Let c'^i be the restriction of the call $c^i \in F$ to the segment set $\{I_1, \ldots, I_{q-1}\}$. Define the family $F' = \{c'^1, \ldots, c'^m\}$ over the $q-1$ segments $\{I_1, \ldots, I_{q-1}\}$, i.e. excluding the last segment. Clearly, if F is a complete family of call classes, so is F'; in such case one also has $V^m(F) \subset V^m(F')$.

Conversely, assume that the family F' is complete, and let $\alpha_1, \ldots, \alpha_m \in V^m(F')$ be a list of integers in the constancy set of F'. Then, the linear combination $\alpha_1 c'^1 + \cdots + \alpha_m c'^m$ is constant over the set of segments $\{I_1, \ldots, I_{q-1}\}$. Returning to our original partition $\{I_1, \ldots, I_q\}$, if follows that the call $c''(j) := \sum_{i=1}^m \alpha_i c^i$ is constant over the segments $\{I_1, \ldots, I_{q-1}\}$, but may have a different value on segment I_q. Thus, the call c'' can be regarded as a two segment call, where the first segment is the union $(I_1 \cup \cdots \cup I_{q-1})$ and the second segment is I_q. The following statement characterizes completeness in recursive terms. It suggests a recursive process for extending a complete family of call classes onto a larger time interval.

Proposition 2: *Let $F := \{c^1, \ldots, c^m\}$ be a family of call classes over the partition $\{I_1, \ldots, I_q\}$ of the interval $[1, T]$, where $q \geq 2$. Let c'^i be the restriction of a call c^i to the segments $\{I_1, \ldots, I_{q-1}\}$. Assume that the family $F' := \{c'^1, \ldots, c'^m\}$ is complete, and let $V^m(F')$ be its set of constancy coefficients. Then, the following two statements are equivalent:*

(i) *F is a complete family of call classes.*

(ii) *There are two lists of integers $\{\alpha_1, \ldots, \alpha_m\}$, $\{\beta_1, \ldots, \beta_m\} \in V^m(F')$ such that the two calls $c''^1 := \sum_{i=1}^m \alpha_i c^i$ and $c''^2 := \sum_{i=1}^m \beta_i c^i$ form a reversed pair of calls over the two segments $(I_1 \cup \cdots \cup I_{q-1}), I_q$.*

Proof: Assume first that $F = \{c^1, \ldots, c^m\}$ is a complete family of call classes over the segments $\{I_1, \ldots, I_q\}$, and let $\gamma_1, \ldots, \gamma_m \in V^m(F)$ be a list of integers in the constancy set of F. Then, the linear combination $d := \sum_{i=1}^m \gamma_i c^i$ is constant over the entire interval $[1, T]$. Referring to part (ii) of Proposition 2, set $\alpha_i := \gamma_i, \beta_i := \gamma_i, i = 1, \ldots, m$; the resulting calls c''^1 and c''^2 are then constant over the two segments $(I_1 \cup \cdots \cup I_{q-1}), I_q$, forming a (trivial) reversed pair over these two segments. This shows that (i) implies (ii).

Conversely, let the family F' be complete, and assume that part (ii) of Proposition 2 is valid. Then, there are lists $\{\alpha_1, \ldots, \alpha_m\}$, $\{\beta_1, \ldots, \beta_m\} \in V^m(F')$ for which the two calls

$$\left.\begin{aligned} c''^1 &:= \textstyle\sum_{i=1}^m \alpha_i c^i \\ c''^2 &:= \textstyle\sum_{i=1}^m \beta_i c^i \end{aligned}\right\} \tag{26}$$

form a reversed pair over the two segments $I' := (I_1 \cup \cdots \cup I_{q-1})$ and I_q. Let $c''^1(j)$, $c''^2(j)$ designate the values of the respective calls over the segment $I_j, j = 1, \ldots, q$. Then, by the definition of a reversed pair, $c''^1(j) \leq c''^1(q)$ and $c''^2(j) \geq c''^2(q)$ (or $c''^1(j) \geq c''^1(q)$ and $c''^2(j) \leq c''^2(q)$) for all $j = 1, \ldots, q-1$. Lemma 3 below implies then that there are integers $\alpha, \beta \geq 0$ such that the linear combination $c'' := \alpha c''^1 + \beta c''^2$ is constant and non-zero over the union $I' \cup I_q$, i.e. over the entire interval $[1, T]$. Substituting (26) into the last expression, we obtain the combination

$$c'' := \sum_{i=1}^{m} (\alpha\alpha_i + \beta\beta_i) c^i$$

which is constant and non-zero over the entire interval $[1, T]$. This demonstrates that the family F is complete, and our proof concludes. □

Proposition 2 can be used as a tool for generating in a recursive manner complete families of call classes by extending the time interval segment by segment. The following algorithm is an example of such a process.

Algorithm 1: Generating a complete family of call classes. Let $\{I_1, I_2, \ldots\}$ be a partition of the interval $[0, T]$.

Start: Choose an integer $c^1(1) > 0$. Note that this forms a one-member family $\{c^1(1)\}$, which is complete over the segment I_1.

Step 1. Assume that a complete family $\{c^1, \ldots, c^m\}$ over the segments $\{I_1, \ldots, I_m\}$ has been constructed for some integer $m \geq 1$. Define an additional call class $c^{m+1}(j) := c^1(j), j = 1, \ldots m$.

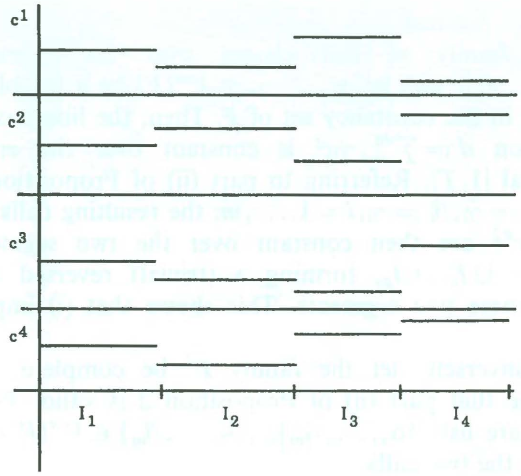

Figure 3.

Step 2. Extend the calls $c^1, \ldots, c^{m+1}$ to the interval I_{m+1} as follows:

(i) To extend the calls c^1 and c^{m+1}: let $a, b \geq 0$ be two integers satisfying the requirements $a \geq \max\{c^1(j), j = 1, \ldots, m\}$ and $b \leq \min\{c^1(j), j = 1, \ldots, m\}$. If m is even, set $c^1(m+1) := a$ and $c^{m+1}(m+1) := b$; otherwise, set $c^1(m+1) := b$ and $c^{m+1}(m+1) := a$. This completes step 2 for the case $m = 1$.

(ii) To extend the calls $c^2, \ldots, c^m$ for $m \geq 2$: choose any $(m-1)$ non-negative integers $e_2, \ldots, e_m$, and set $c^i(m+1) := e_i$, $i = 2, \ldots, m$.

We comment that Algorithm 1 cannot generate every possible complete family of call classes over a given set of segments. Nevertheless, Algorithm 1 provides a convenient method of producing complete families. An interesting aspect of Algorithm 1 is the fact that it can be used to build patterns of call classes that form a complete family, irrespective of the specific values of the calls. These families are complete for any call values that are compatible with the pattern. The following example demonstrates such a pattern.

Example 1: Figure 3 shows call patterns generated by Algorithm 1 for $m = 4$.

To conclude, we have seen that full utilization of large backbones (without buffering) is possible only with complete families of call classes. In the next section we examine backbone efficiency in cases where a complete family of call classes is not available.

5. Incomplete families of calls

In view of Theorem 1, the case of complete families of calls is the only case in which asymptotic efficiency of 1 can be achieved by admission control alone. In the present section we examine the question of how to determine the optimal call populations for general families of calls. Critical to the discussion is the notion of 'relative efficiency', defined next.

Let $F = \{c^1, \ldots, c^m\}$ be a family of calls, let α_i be the number of calls of the class c^i entering the backbone, $i = 1, \ldots, m$, and let ϕ be the backbone capacity. The total number of cells entering the backbone at the step k is

$$z_k := \sum_{i=1}^{m} \alpha_i c_k^i, \qquad k = 1, \ldots, T \tag{27}$$

Since cell loss is not allowed, the amplitude must satisfy $A(z) \leq \phi$. The *relative efficiency* $\eta_r(z)$ of the cell stream z is defined by

$$\eta_r(z) := \frac{\sum_{k=1}^{T} z_k}{TA(z)} \tag{28}$$

when $A(z) > 0$ and $\eta_r := 0$ when $z = 0$. Note that

$$0 \leq \eta_r(z) \leq 1 \tag{29}$$

In view of equation (7) and the fact that $A(z) \leq \phi$, we have

$$\eta(z) = \eta_r(z) \frac{A(z)}{\phi} \leq \eta_r(z) \leq 1 \tag{30}$$

Thus, the efficiency cannot exceed the relative efficiency, and $\eta(z) = \eta_r(z)$ when $A(z) = \phi$. We sometimes use the notation

$$\eta_r(\alpha_1, \ldots, \alpha_m) := \eta_r(z)$$

to emphasize the dependence on the integers $\alpha_1, \ldots, \alpha_m$. The maximal relative efficiency is then given by

$$\eta_r^*(F) := \sup\{\eta_r(\alpha_1, \ldots, \alpha_m) : \alpha_1, \ldots, \alpha_m \in Z^+\}$$

where Z^+ is the set of all non-negative integers. The following is then true.

Theorem 2: *Let $F = \{c^1, \ldots, c^m\}$ be a family of call classes over the partition $\{I_1, \ldots, I_q\}$, where none of the call classs $c^1, \ldots, c^m$ is identically zero. Then, the following are true:*

(i) *There are integers $\alpha_1^*, \ldots, \alpha_m^* \geq 0$ such that $\eta_r^*(F) = \eta_r(\alpha_1^*, \ldots, \alpha_m^*)$, where $\eta_r^*(F)$ is the maximal relative efficiency of the family F, and $\eta_r^*(F) > 0$.*

(ii) *The maximal relative efficiency $\eta_r^*(F)$, as well as the call populations $\alpha_1^*, \ldots, \alpha_m^*$ yielding it, can be determined through the solution of a linear programming problem.*

Proof: Substituting (27) into (28), and recalling that λ_j is the number of steps of the segment I_j and that

$c^i(j)$ is the value of the call c^i on the segment I_j, we have

$$\eta_r(\alpha_1,\ldots,\alpha_m)=\sum_{k=1}^{T}\sum_{i=1}^{m}\frac{\alpha_i}{TA(z)}c_k^i=\sum_{j=1}^{q}\sum_{i=1}^{m}\lambda_j\frac{\alpha_i}{TA(z)}c^i(j)$$

Then, inequality (29) implies

$$0\leq\sum_{j=1}^{q}\sum_{i=1}^{m}\lambda_j\frac{\alpha_i}{TA(z)}c^i(j)\leq 1$$

Now, define the quantities

$$\beta_i:=\frac{\alpha_i}{A(z)}\quad\text{and}\quad\Psi_i:=\frac{1}{T}\sum_{j=1}^{q}\lambda_j c^i(j),\quad i=1,\ldots,m \tag{31}$$

Note that $\Psi_1,\ldots,\Psi_m$ are determined by the calls $c^1,\ldots,c^m$ and the intervals $I_1,\ldots,I_q$, and are therefore fixed and specified. Since none of the calls is identically zero, we have $\Psi_i>0$ for all $i=1,\ldots,m$. Note also that $\beta_i\geq 0, i=1,\ldots,m$. Letting $z(j)$ be the value of the flow z on the segment I_j, we can write

$$\eta_r(\alpha_1,\ldots,\alpha_m)=\sum_{i=1}^{m}\beta_i\Psi_i$$

and

$$\sum_{i=1}^{m}\beta_i c^i(j)=\frac{z(j)}{A(z)}\leq 1,\qquad j=1,\ldots,q$$

Let us regard for a moment $\beta_1,\ldots,\beta_m$ as indeterminate non-negative real numbers, while regarding $\Psi_1,\ldots,\Psi_m$, $c^1(1),\ldots,c^1(q)$, $c^2(1),\ldots,c^2(q),\ldots,c^m(1),\ldots,c^m(q)$ as specified non-negative real numbers. Then, consider the linear programming problem of finding a maximum of the function

$$L(\beta_1,\ldots,\beta_m):=\sum_{i=1}^{m}\beta_i\Psi_i \tag{32}$$

subject to the constraints

$$\sum_{i=1}^{m}\beta_i c^i(j)\leq 1,\quad j=1,\ldots,q \tag{33}$$

and

$$\beta_i\geq 0,\qquad i=1,\ldots,m \tag{34}$$

Let S be the set of all real numbers $\beta_1,\ldots,\beta_m$ that satisfy the inequalities (33) and (34), and denote

$$\mu(c^i):=\max\{c^i(j):j=1,\ldots,q\}$$

Since none of the calls $c^1,\ldots,c^m$ is identically zero, it follows that $\mu(c^i)>0$ for all $i=1,\ldots,m$; that S is a closed set; and that

$$0\leq\beta_i\leq 1/\mu(c^i),\qquad i=1,\ldots,m.$$

These facts imply that S is closed and bounded, and whence is a compact set. Consequently the continuous function L has a maximum over S, and our linear programming problem has a solution.

As in every linear programming problem, the maximal value of L is attained on a boundary point of the set S. In other words, L has a maximum point at which at least m of the $(q+m)$ constraints (33) and (34) are satisfied with equality. These equality constraints induce a system of at least m simultaneous linear equations. Any solution $\beta_1^*,\ldots,\beta_m^*$ of this system of linear equations yields the maximal value of L, i.e.

$$L(\beta_1^*,\ldots,\beta_m^*)\geq L(\beta_1,\ldots,\beta_m)\quad\text{for all }\beta_1,\ldots,\beta_m\in S$$

Since $\{c^i(j)\}$ are all integers, we are dealing here with a system of linear equations over the field of rational numbers. Consequently, we can select the solution $\beta_1^*,\ldots,\beta_m^*$ to consists of rational numbers.

Next, since the rational numbers $\Psi_1,\ldots,\Psi_m$ are all positive, the nature of the constraints (33) and (34) implies that $L(\beta_1^*,\ldots,\beta_m^*)>0$, so that at least one of $\beta_1^*,\ldots,\beta_m^*$ is not zero. Consequently, not all the m constraints of (34) are satisfied with equality; since at least m of the constraints (33) and (34) must hold with equality, it follows that at least one of the constraints (33) holds with equality, i.e. that there is an integer $p\in\{1,\ldots,q\}$ such that

$$\sum_{i=1}^{m}\beta_i^* c^i(p)=1 \tag{35}$$

Now, let $\alpha>0$ be the least common integer denominator of the rational numbers $\beta_1^*,\ldots,\beta_m^*\geq 0$, and define the integers

$$\alpha_i:=\alpha\beta_i^*\geq 0,\qquad i=1,\ldots,m \tag{36}$$

Consider the linear combination $z:=\sum_{i=1}^{m}\alpha_i c^i=\alpha[\sum_{i=1}^{m}\beta_i^* c^i(j)]$. In view of (33) and (34), this yields

$$z(p)=\alpha\quad\text{and}\,z(j)\leq\alpha\quad\text{for all}\,j=1,\ldots,q$$

Whence, the amplitude $A(z)=\alpha$, which implies that

$$L(\beta_1^*,\ldots,\beta_m^*)=\eta_r(\alpha_1,\ldots,\alpha_m)$$

In other words, the maximal relative efficiency is achieved by the populations $\alpha_1,\ldots,\alpha_m$ of (36) and part (i) of Theorem 2 is valid with $\alpha_i^*:=\alpha_i, i=1,\ldots,m$. □

There is an intimate connection between relative efficiency and asymptotic efficiency. In fact, the maximal asymptotic efficiency is equal to the maximal relative efficiency and is achieved similarly, as indicated next.

Optimal population control for incomplete families: Let $F=\{c^1,\ldots,c^m\}$ be a family of non-empty call classes transmitted over a backbone of capacity ϕ. Let

$z^* := \sum_{i=1}^{m} \alpha_i^* c^i$ be a flow that achieves the maximal relative efficiency $\eta_r^*(F)$ for the family F, and let $A(z^*)$ be the amplitude of z^*. Using the integer division algorithm, write $\phi = \beta A(z^*) + \rho$, where $\beta \geq 0$ is an integer and $0 \leq \rho < A(z^*)$. Define the backbone flow

$$z(\phi) := \beta z^*. \tag{37}$$

Theorem 3: *In the above notation, let $\eta(\phi)$ be the efficiency of the flow $z(\phi)$ of* (37). *Then, the following hold:*

(i) *For any flow $z := \sum_{i=1}^{m} \gamma_i c^i$ of the family F, the efficiency satisfies $\eta(z) \leq \eta_r^*(F)$.*

(ii) *The maximal asymptotic efficiency is given by $\eta^* := \lim_{\phi\to\infty} \eta(\phi)$.*

(iii) *$\eta^* = \eta_r^*(F)$, where $\eta_r^*(F)$ is the maximal relative efficiency of the family F.*

Proof: (i) Let $\eta_r(z)$ be the relative efficiency of the flow z, and recall that the maximal relative efficiency $\eta_r^*(F)$ exists by Theorem 2. Then, we clearly have that $\eta_r(z) \leq \eta_r^*(F)$, and, since $\eta(z) \leq \eta_r(z)$ by (30), we obtain that $\eta(z) \leq \eta_r^*(F)$. This proves part (i) of Theorem 3.

Regarding part (ii) of Theorem 3, let $z^* := \sum_{i=1}^{m} \alpha_i^* c^i$ be a flow with relative efficiency $\eta_r(z^*) = \eta_r^*(F)$, as given by Theorem 2 (i). Let $A(z^*)$ be the amplitude of z^*, and note that $A(z^*) > 0$. For the flow (37), the amplitude satisfies $A(z(\phi)) = \beta A(z^*)$. Direct substitution into (28) shows that the relative efficiency $\eta_r(z(\phi))$ of the flow $z(\phi)$ satisfies

$$\eta_r(z(\phi)) = \eta_r(z^*) = \eta_r^*(F) \tag{38}$$

independently of the value of the integer $\beta > 0$. Using (30), (37), and (38), we get

$$\eta(z(\phi)) = \eta_r(z(\phi))\frac{A(z(\phi))}{\phi} = \eta_r^*(F)\frac{\beta A(z^*)}{\phi}$$

$$= \eta_r^*(F)\left[\frac{\phi-\rho}{\phi}\right] = \eta_r^*(F)\left[1-\frac{\rho}{\phi}\right]$$

Taking into account that $0 \leq \rho < A(z^*)$ and that $A(z^*)$ is a fixed number, it follows that

$$\lim_{\phi\to\infty} \eta(z(\phi)) = \lim_{\phi\to\infty} \eta_r^*(F)\left[1-\frac{\rho}{\phi}\right] = \eta_r^*(F)$$

In view of part (i) of this proof, this implies that $\eta(z) \leq \lim_{\phi\to\infty} \eta(z(\phi))$ for any flow z of the family F. Consequently, the maximal asymptotic efficiency satisfies $\eta^* = \lim_{\phi\to\infty} \eta(z(\phi))$, and our proof concludes. □

In view of Theorem 3 and (37), the flow that achieves maximal asymptotic efficiency consists of fixed proportions of the call classes, as characterized by the integers $\alpha_1^*, \ldots, \alpha_m^*$ of Theorem 2. In other words, the flow that achieves the maximal asymptotic efficiency is obtained through a scalable process, by using integer multiples of the basic flow package $z^* = \alpha_1^* c^1 + \cdots + \alpha_m^* c^m$. Thus, when backbone capacity is increased, one only needs to scale the flow upwards, leaving the consistency unchanged.

Of course, in order to achieve the maximal asymptotic efficiency with the family F, the pool of calls waiting for admission into the backbone must contain a sufficient number of calls of each class. Specifically, the call pool must contain at least $\beta\alpha_i^*$ calls of the class c^i for each $i = 1, \ldots, m$. The next section addresses situations where this requirement is not met.

6. Incomplete call families with limited call supply

Consider a family of call classes $F = \{c^1, \ldots, c^m\}$ to be transmitted over a backbone of capacity ϕ. For each integer $i = 1, \ldots, m$, let p_i be the number of calls of the class c^i present in the call pool at the initial time. We refer to $p_1, \ldots, p_m$ as the *call pool populations*. It is convenient to define the ratios

$$\rho_i : \frac{p_i}{\phi}, \quad i = 1, \ldots, m$$

called the *call pool parameters*. As $\phi \to \infty$, we assume that the call pool populations increase so as to maintain constant values of the call pool parameters $\rho_1, \ldots, \rho_m$. This maintains constant proportions among the different call populations as the backbone capacity grows to infinity. The call pool parameters are specified system parameters and are not all equal zero.

If one would attempt to transmit all calls of the call pool simultaneously through a backbone of capacity ϕ, one would obtain the flow $z = \sum_{i=1}^{m} \rho_i \phi c^i$. The amplitude of this flow is $A(z) = \phi A(\sum_{i=1}^{m} \rho_i c^i)$. As mentioned earlier, in order for the backbone optimization problem to be meaningful, one must have $A(z) > \phi$; otherwise, all waiting calls can be transmitted simultaneously, and there is no place for optimization. This yields the requirement $\phi A(\sum_{i=1}^{m} \rho_i c^i) > \phi$, or

$$A\left(\sum_{i=1}^{m} \rho_i c^i\right) > 1 \tag{39}$$

Our discussion in the present section is subject to the condition (39).

Consider now a flow $z := \sum_{i=1}^{m} \alpha_i c^i$, where $\alpha_1, \ldots, \alpha_m \geq 0$ are integers, not all of which are zero. The limitation on the call pool populations imposes the conditions

$$\alpha_i \leq p_i = \rho_i \phi, \qquad i = 1, \ldots, m$$

Let $\eta_r(\alpha_1, \ldots, \alpha_m)$ be the relative efficiency of the flow z, as given by (28). Then, the maximal relative efficiency η_r^*

that can be achieved under the present circumstances is given by

$$\eta_r^* := \sup\{\eta_r(\alpha_1,\ldots,\alpha_m) : 0 \le \alpha_i \le p_i, A(z) \le \phi, i = 1,\ldots,m\}$$

Defining the variables $\beta_i := \alpha_i/A(z)$ as in (31), we can write the restriction $\alpha_i \le p_i$ in the equivalent form

$$\beta_i \le \frac{p_i}{A(z)}, \qquad i = 1,\ldots,m \tag{40}$$

Let us revisit now the function $L = \sum_{i=1}^m \beta_i \Psi_i$ of (32) which, as shown in the proof of Theorem 2, represents the relative efficiency. To accommodate our present additional restrictions, we have to add the m constraints (40) to the constraints (33) and (34) considered earlier. However, the constraints (40) are not linear constraints, due to the fact that the amplitude $A(z)$ depends on $\beta_1,\ldots,\beta_m$. Thus, the new optimization problem is no longer a linear programming problem—a potential complication.

In order to circumvent this complication, consider the constraints

$$\beta_i \le \rho_i, \qquad i = 1,\ldots,m \tag{41}$$

Recall that the call pool parameters $\rho_1,\ldots,\rho_m$ are fixed and specified system parameters. Consequently, the maximization of the function L subject to the constraints (33), (34) and (41) is a linear programming problem. As we show below, for the maximization of asymptotic efficiency, (40) and (41) are equivalent constraints. This fact allows us to represent the optimization of large backbones as a linear programming problem, and indicates another important advantage of the notion of asymptotic efficiency. The equivalency of (40) and (41) for asymptotic efficiency originates from the fact that, for an optimal flow $z^*(\phi)$, one has $\lim_{\phi\to\infty} A(z^*(\phi))/\phi = 1$; see the proof of Theorem 4 below for details.

We start our technical discussion by showing that the problem of maximizing the function L under the constraints (33), (34) and (41) has a rational solution. This an important fact, since the optimal call populations are derived later from this solution.

Proposition 3: *The linear programming problem of maximizing the function L of* (32) *subject to the constraints* (33), (34) and (41) *has a solution that consists of a list of rational numbers $\beta_1^*,\ldots,\beta_m^*$.*

The proof of Proposition 3 is in the Appendix.

We can now characterize the general features of the problem of optimizing backbone utilization with call pool constraints. In particular, we show that the maximal asymptotic efficiency of the backbone is given by the solution of the linear programming problem of Proposition 3.

Theorem 4: *Let $F = \{c^1,\ldots,c^m\}$ be a family of call classes over the partition $\{I_1,\ldots,I_q\}$, and let $\rho_1,\ldots,\rho_m$ be the call pool parameters. For a backbone capacity ϕ, let $\eta(\phi)$ be the maximal backbone efficiency possible for F with the call pool parameters $\rho_1,\ldots,\rho_m$. Denote by L^* the maximal value of the function L of* (32), *subject to the constraints* (33), (34) *and* (41). *Then,* $\lim_{\phi\to\infty}\eta(\phi) = L^*$.

Proof: Consider a flow $z = \sum_{i=1}^m \alpha_i c^i$ with amplitude $0 < A(z) \le \phi$, let $\beta_i := \alpha_i/A(z), i = 1,\ldots,m$, as in (31), and let L be given by (32). Using (30) and (32), the efficiency $\eta(z)$ of z satisfies

$$\eta(z) := \frac{A(z)}{\phi} L(\beta_1,\ldots,\beta_m) =: \Lambda(\beta_1,\ldots,\beta_m, A(z))$$

Using the constraint $A(z) \le \phi$ (lossless transmission)

$$\begin{aligned} 0 \le L - \Lambda &= \frac{\phi - A(z)}{\phi} L \\ &\le \left(\sup\frac{\phi - A(z)}{\phi}\right) \sup L \le \sup\frac{\phi - A(z)}{\phi} \end{aligned} \tag{42}$$

since all quantities are non-negative and $0 \le L \le 1$.

For a flow of amplitude $A(z)$ and a backbone capacity $\phi \ge A(z)$, use the integer division algorithm to set the integers $\beta, \varepsilon \ge 0$ according to

$$\phi = \beta A(z) + \varepsilon, \quad \text{where } 0 \le \varepsilon < A(z) \tag{43}$$

Assuming that ϕ is large enough, we have $\beta > 0$. Define $z_\beta := \beta z$, which has the amplitude $A(z_\beta) = \beta A(z)$. Substituting z_β for z in (42), we get

$$|L - \Lambda| \le \sup\left[\frac{\varepsilon}{\phi}\right] \to 0 \qquad \text{as } \phi \to \infty$$

for every z, as long as the largest possible number β of packages z is transmitted. Consequently, under these circumstances, the maximum of Λ approaches the maximum L^* of L as $\phi \to \infty$. But then, since the maximum of Λ is the maximal efficiency $\eta(\phi)$ by definition, we conclude that $\lim_{\phi\to\infty}\eta(\phi) = L^*$. □

The following technical property of the linear programming problem of Proposition 3 is needed for our discussion (a proof of Lemma 2 is provided in the Appendix).

Lemma 2: *Let $\beta_1^*,\ldots,\beta_m^*$ be as in Proposition 3. Then, the amplitude $A(\sum_{j=1}^m \beta_j^* c^j) = 1$.* □

We can now characterize the optimal call populations.

Optimal population control for incomplete families with call pool restrictions: Let $F = \{c^1, \dots, c^m\}$ be a family of calls with call pool parameters $\rho_1, \dots, \rho_m$, flowing into a backbone of capacity ϕ. Let $\beta_1^*, \dots, \beta_m^*$ be a rational solution of the linear programming problem of Proposition 3, and let $\alpha > 0$ be an integer common denominator of $\beta_1^*, \dots, \beta_m^*$. Denote

$$\alpha_i := \alpha\beta_i^*, \qquad i = 1, \dots, m \tag{44}$$

Define the flows

$$z := \sum_{i=1}^{m} \alpha_i c^i \quad \text{and} \quad z_\beta := \beta z \tag{45}$$

where $\beta \geq 0$ is the integer of (43).

The fact that the flow z_β of (45) achieves maximal asymptotic efficiency under the stated restrictions is validated by the following.

Proposition 4: (i) *The Population Control Algorithm above achieves maximal asymptotic efficiency under its stated restrictions, and* (ii) *the flow z of* (45) *has the amplitude $A(z) = \alpha$, where α is from* (44).

Proof: Part (i) follows directly from Theorem 4 and Proposition 3. Part (ii) is a consequence of Lemma 2, since

$$A(z) = A\left(\alpha \sum_{i=1}^{m} \beta_i^* c^i\right) = \alpha A\left(\sum_{i=1}^{m} \beta_i^* c^i\right) = \alpha$$

□

As we can see, the flow z of (45) forms a basic 'packing unit' for calls travelling through the backbone. To achieve the maximal asymptotic efficiency, the population control algorithm admits calls from the call pool in combinations that form integer multiples of the packing unit z. In this way, the population control algorithm provides a scalable solution of the optimization problem for large backbones. Note that, in general, the combination z is not unique—there can be different basic packing units that achieve (the same) maximal asymptotic efficiency.

Our discussion in this paper dealt with the development of an optimal policy of admitting calls into the network. We have taken a functional approach to this issue, viewing it as a global optimization problem. In comparison, the classical approach to admission control tilts more toward an instant-by-instant evaluation of the network load (compare to Decina and Toniatti 1990, Rathgeb 1991, CCITT 1992). In the second part of this paper (Hammer 2003), we expand our framework to include reshaping of call waveforms. This will allow us to further improve network performance for incomplete families of calls and for calls with random components.

Appendix

This appendix contains some technical results and proofs.

Lemma 3: Let (δ_1, δ_2), (δ_3, δ_4) be two pairs of non-negative integers, where $\delta_1 \leq \delta_2$ and $\delta_3 \geq \delta_4$. Then, there are two integers $\alpha, \beta \geq 0$ not both zero, such that $\alpha\delta_1 + \beta\delta_3 = \alpha\delta_2 + \beta\delta_4$.

Proof: It is clearly enough to show that there are two integers $\alpha, \beta \geq 0$ not both zero, such that

$$\alpha(\delta_2 - \delta_1) = \beta(\delta_3 - \delta_4).$$

By assumption, the two integers $a_1 := (\delta_2 - \delta_1)$ and $a_2 := (\delta_3 - \delta_4)$ are both non-negative. Now, if $a_1 = 0$, we can choose $\alpha > 0$ and $\beta = 0$. If $a_1 \neq 0$ but $a_2 = 0$, we can choose $\beta > 0$ and $\alpha = 0$. Finally, if $a_1 \neq 0$ and $a_2 \neq 0$, then the existence of α, β is a consequence of the fact that two positive integers have a positive common multiple. □

Proof (of Algorithm 1): We show that, for every integer $m \geq 1$, Algorithm 1 yields a complete family of call classes $c^1, \dots, c^m$ over the segments $\{I_1, \dots, I_m\}$. We use induction on m.

For the case $m = 1$, the start step of Algorithm 1 generates a complete family over the single-segment partition I_1, since any constant non-zero call class over a single segment partition forms a complete family.

Next, let $m = p \geq 1$ be an integer, and assume that every family $F(p) := \{c^1, \dots, c^p\}$ of p calls over the p segments $\{I_1, \dots, I_p\}$ created by Algorithm 1 is complete. Let $\{a_1, \dots, a_p\} \in V^p(F(p))$ be a list of integers, where $V^p(F(p))$ is the set of constancy coefficients of $F(p)$; the linear combination

$$d := a_1 c^1 + \cdots + a_p c^p$$

is then constant over the union $(I_1 \cup \cdots \cup I_p)$.

Now, consider the family $F(p+1) := \{c^1, \dots, c^{p+1}\}$ over the segment set $I_1, \dots, I_{p+1}$, built from the family $F(p)$ by one more iteration of Algorithm 1. Define two lists of integers $\alpha_1, \dots, \alpha_{p+1}$ and $\beta_1, \dots, \beta_{p+1}$ by setting $\alpha_1 := a_i$, $i = 1, \dots, p$, $\alpha_{p+1} := 0$; and $\beta_1 := 0$, $\beta_i := a_i$, $i = 2, \dots, p, \beta_{p+1} = a_1$, and consider the two calls

$$c''^1 := \sum_{i=1}^{p+1} \alpha_i c^i$$

$$c''^2 := \sum_{i=1}^{p+1} \beta_i c^i$$

over the segment set $\{I_1, \ldots, I_{p+1}\}$. Recalling that Step 1 of Algorithm 1 assigns $c^{p+1}(j) = c^1(j)$ for all $j = 1, \ldots, p$, it follows by the selection of $a_1, \ldots, a_p$ that

$$c''^1(j) = d \quad \text{and} \quad c''^2(j) = d \quad \text{for all } j = 1, \ldots, p \tag{46}$$

Further define

$$r^1(j) := \alpha_2 c^2(j) + \cdots + \alpha_{p+1} c^{p+1}(j)$$

$$r^2(j) := \beta_1 c^1(j) + \cdots + \beta_p c^p(j), \quad j = 1, \ldots, p+1$$

The structure of the lists $\{\alpha_i\}$ and $\{\beta_i\}$ implies that $r^1(j) = r^2(j)$ for all $j = 1, \ldots, p+1$. Setting $r(j) := r^1(j) = r^2(j)$, we have

$$c''^1(j) = r(j) + a_1 c^1(j),$$

$$c''^2(j) = r(j) + a_1 c^{p+1}(j), \quad j = 1, \ldots, p+1$$

In view of Step 2(i) of Algorithm 1, this implies that $c''^1(p+1) \geq c''^2(p+1)$ when p is even, while $c'''[1(p+1) \leq c''^2(p+1)$ when p is odd, since $a_1 \geq 0$. Combining this with (46), we obtain that c''^1 and c''^2 form a reversed pair over the two segments $(I_1 \cup I_2 \cup \cdots \cup I_p)$ and I_{p+1}. But then Proposition 2 shows that $F(p+1)$ is a complete family of call classes over the segments $\{I_1, \ldots, I_{p+1}\}$, and the validity of Algorithm 1 follows by induction. □

Proof (of Proposition 3): We can decompose our present situation into two subcases: (i) At the maximum point $\beta_1^*, \ldots, \beta_m^*$, none of the constraints (41) hold with equality; and (ii) At the maximum point $\beta_1^*, \ldots, \beta_m^*$, one or more of the constraints (41) hold with equality. In case (i), the constraints (41) can be ignored, and the existence of the rational solution $\beta_1^*, \ldots, \beta_m^*$ follows by Theorem 2.

Regarding case (ii), let $n \geq 1$ be the number of constraints of (41) that hold with equality, i.e. $\beta_{i(1)}^* = \rho_{i(1)}$, $\beta_{i(2)}^* = \rho_{i(2)}, \ldots, \beta_{i(n)}^* = \rho_{i(n)}$, and denote the set $\sigma' := \{i(1), \ldots, i(n)\}$. Recall that the pool parameters $\rho_1, \ldots, \rho_m$ are rational numbers. Now, if $n = m$, then $\beta_i^* = \rho_i$ for all $i = 1, \ldots, m$, and Lemma 3 is valid by the rationality of $\rho_1, \ldots, \rho_m$. Otherwise, and let σ be the set of all integers $i \in \{1, \ldots, m\}$ that are not in the set σ'. Define the functions

$$L' := \sum_{i \in \sigma} \beta_i \Psi_i \quad \text{and} \quad L'' = L' + \sum_{j \in \sigma'} \rho_j \Psi_j$$

In view of the fact that $\beta_i^* := \rho_i$ for all $i \in \sigma'$, it follows that the maximum of the function L'' occurs at the same values of $\beta_i, i \in \sigma$, as the maximum of the function L. Furthermore, the corresponding values of $\beta_i, i \in \sigma$, induce a maximum of the function L' as well.

Thus, our problem reduces to finding a maximum point of the function L', subject to the constraints (33) and (34), where we set $\beta_i := \rho_i$ for all $i \in \sigma'$. We can then rewrite (33) and (34) in the form

$$\sum_{i \in \sigma} \beta_i c^i(j) \leq 1 - \sum_{i \in \sigma'} \rho_i c^1(j), \quad j = 1, \ldots, q \tag{47}$$

$$\beta_i \geq 0, \quad i \in \sigma \tag{48}$$

This yields an optimization problem similar to the one considered in Theorem 2, since the right side of (47) is rational for all $j = 1, \ldots, q$. The arguments used in the proof of Theorem 2 show then that there are rational numbers $\{\beta_i^* : i \in \sigma\}$ at which L' attains its maximum, subject to the constraints. This concludes our proof. □

Proof (of Lemma 2): In view of the constraints (33), we must have

$$A\left(\sum_{i=1}^{m} \beta_i^* c^i(j)\right) \leq 1 \tag{49}$$

We consider now each possible scenario separately.

Case 1: All the constraints (41) hold with equality at the maximum point $\beta_1^*, \ldots, \beta_m^*$. Then, $\beta_i^* = \rho_i$ for all $i = 1, \ldots, m$, and whence $A(\sum_{i=1}^m \beta_i^* c^i(j)) = A(\sum_{i=1}^m \rho_i c^i(j))$. But in view of (39), this equality violates (49), which implies that Case 1 cannot occur.

Case 2: One or more of the constraints (33) holds with equality at the maximum point $\beta_1^*, \ldots, \beta_m^*$. Then, $\sum_{i=1}^m \beta_i^* c^i(j) = 1$ for an integer $j \in \{1, \ldots, q\}$, which togather with (49) directly implies that $A(\sum_{i=1}^m \beta_i^* c^i) = 1$.

Case 3: All the constraints (33) hold with strict inequality at the maximum point $\beta_1^*, \ldots, \beta_m^*$. Now, since Case 1 cannot occur, there must be an integer $s \in \{1, \ldots, m\}$ such that $\beta_s^* < \rho_s$. Also, since Ψ_s is strictly positive (see the paragraph following (31)), the function L of (32) is a monotone strictly increasing function of $\beta_s \geq 0$. Thus, the value of L can be increased by increasing the value of β_s beyond β_s^*; indeed, the value of β_s can be increased until one of the constraints (33) becomes valid with equality, or until β_s becomes equal to ρ_s (whichever occurs first). But then $\beta_1^*, \ldots, \beta_m^*$ do not constitute a maximum point of L, contrary to their definition. This shows that Case 3 cannot occur at a maximum point of L. Consequently, only Case 2 is possible, and our proof concludes. □

References

Atkins, J. D., 1980, Path control: the transport network of SNA. *IEEE Transactions on Communications*, **COM 28**, 527–538.

ATM Forum, 1997, Technical specifications: approved ATM forum specifications.

CCITT, 1992, Traffic and congestion control in B-ISDN, I, CCIT Study Group XVIII, Geneva.

Chang, C. S., 1994, Stability, queue length, and delay of deterministic and stochastic queueing networks. *IEEE Transactions on Automatic Control*, **AC 39**, 913–931.

Daigle, J. N., and Langford, J. D., 1986, Models for analysis of packet voice communication systems. *IEEE JSAC*, **SAC-4**, 847–855.

Decina, M., and Toniatti, T., 1990, On bandwidth allocation to bursty virtual connections in ATM networks. *Proceedings of the IEEE ICC*, April 1990, Atlanta, GA, USA.

Eckberg, A. E., Jr., 1979, The single server queue with periodic arrival process and deterministic service times. *IEEE Transactions on Communications*, **COM 27**, 556–562.

Golestani, S. J., 1991, A framing strategy for congestion management. *IEEE JSAC*, **SAC9**, 1064–1077.

Hammer, J., 2003, Sturdy control of discrete communication networks. Part II: Call reshaping. *International Journal of Control*, **76**, 643–656.

Handel, R., Huber, M. N., and Schroder, S., 1994, *ATM Networks, Concepts, Protocols, Applications*, 2nd edn (Wokingham, England: Addison-Wesley).

Hui, J. Y, 1988, Resource allocation for broadband networks. *IEEE JSAC*, **SAC6**, 1598–1608.

Paxson, V., and Floyd, S., 1994, Wide area traffic: the failure of poisson modeling. *Proceedings of SIGCOM*, London, UK, August, pp. 257–268.

Rathgeb, E. P., 1991, Modeling and performance comparison of policing mechanisms for ATM networks. *IEEE JSAC*, **SAC9**, 325–334.

Schwartz, M., 1996, *Broadband Integrated Networks* (New Jersey: Prentice Hall).