Telecommunication Systems 19:2, 147–186, 2002

Performance Analysis of Fair Channel Sharing Policies in an Integrated Cellular Voice/Data Network

REMCO LITJENS
Expertise Group QoS Control, KPN Research, The Netherlands

RICHARD J. BOUCHERIE *
Faculty of Mathematical Sciences, University of Twente, The Netherlands

Abstract. We study channel sharing in an integrated cellular voice/data network with a finite queue for data call requests that cannot be served immediately upon arrival. Using analytical techniques, a comparison of different fair channel sharing policies is made. As a main result, a closed-form expression is derived for the expected sojourn time (waiting time *plus* transfer time) of a data call, conditional on its size, indicating that the sojourn time is proportional to the call size. This attractive proportionality result establishes an additional fairness property for the channel sharing policies proposed in the paper. Additionally, as a valuable intermediate result, the conditional expected sojourn time of an admitted data call is obtained, given the system state at arrival, which may serve as an appreciated feedback information service to the data source. An extensive numerical study is included to compare the proposed policies and to obtain insight in the performance effects of the various system and policy parameters.

Keywords: integrated services networks, cellular networks, channel sharing policies, GSM, HSCSD, performance analysis, Markov chain analysis, processor sharing models, grade-of-service, quality-of-service, fairness

1. Introduction

It is generally expected that within the next decade, data transfer will rival voice communication as the dominant mobile service mode. This is reflected in the current evolution in mobile networks, which is primarily characterized by a transition from circuit-switched voice-oriented networks to integrated circuit- and packet-switched multi-service networks.

In much of the mobile world, this evolution consists of a transition from second-generation GSM (Global System for Mobile communications) (e.g., [Mouly and Pautet, 25]), which has been optimized for mobile voice telephony, to third-generation UMTS (Universal Mobile Telecommunications System) (e.g., [Dahlman et al., 12; Ojanperä and Prasad, 28]), which is designed to support a wider variety of services, e.g., video telephony and remote database access, with increased efficiency and flexibility.

* This research is carried out while Richard J. Boucherie was at the Universiteit van Amsterdam, supported by the Technology Foundation STW, Applied Science Division of NWO and the technology programme of the Ministry of Economic Affairs, The Netherlands.

An obvious intermediate phase in this transition is the upgrade of the GSM standard to support higher data bit rates.

1.1. Data communications in GSM

Currently, data communication capabilities in GSM networks are limited to SMSs (Short Message Services) and CSD (Circuit-Switched Data) transfers. An SMS message may contain up to 140 bytes, while a CSD session is carried over a full-rate traffic channel with a maximum information bit rate of 9.6 kbps.

A number of upgrades to GSM's data transfer capabilities are (being) specified by ETSI (European Telecommunications Standards Institute), in order to support the growing market for mobile data services. Just recently, a new data coding scheme with reduced overhead and hence also reduced protection, has been standardized for the circuit-switched data service, yielding a 14.4 kbps information bit rate. Secondly, the HSCSD (High-Speed Circuit-Switched Data) service is currently being implemented in GSM networks, enabling data calls to be allocated multiple circuit-switched traffic channels in parallel. HSCSD is discussed in more detail in the next subsection. As a third, and most significant upgrade of the GSM network, GPRS (General Packet Radio Service) (e.g., [Brasche and Walke, 4; Cai and Goodman, 6]) is expected to be deployed by network operators in 2000/2001. With GPRS, multiple traffic channels can be dynamically shared by multiple data calls in a packet-switched fashion. Next, a new higher-level modulation scheme will be introduced with EDGE (Enhanced Data rates for Global Evolution) (e.g., [Furuskär et al., 17]), which is designed to boost (HS)CSD and GPRS information bit rates. Finally, 3GPP (3rd-Generation Partnership Project) is standardizing UMTS, but this can hardly be recognized as a GSM upgrade, since it is based on an entirely new air interface.

1.2. High-speed circuit-switched data

HSCSD enables the assignment of a bundle of traffic channels to a single data call, thereby enhancing the potential information bit rate that can be offered. According to the specifications [ETSI 14,15], a single HSCSD data call can be assigned up to eight full rate traffic channels, i.e., an entire GSM carrier, while the assignment may be up- or downgraded during a call in order to optimize service quality and channel utilization or support newly arriving GSM or HSCSD calls, respectively. See figure 1 where four traffic channels are simultaneously used by an HSCSD call.

Bundling of traffic channels requires a new functionality (software) in both the mobile station and the base station controller (BSC). The *terminal adaptation function* in the mobile station is in charge of splitting and combining the n data substreams that are carried over n traffic channels, and thus forms the interface between the terminal equipment and the air interface. Across the air interface, the *inter-working function* in the base station controller performs these operations as an interface between the radio interface and the mobile switching center, which in turn forms the gateway to external networks.

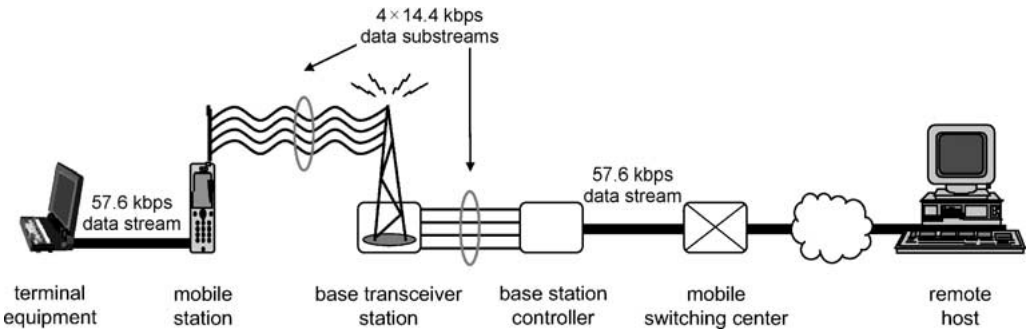

Figure 1. GSM/HSCSD network architecture.

As mentioned above, two distinct service modes can be implemented to enhance the data transfer capabilities of a GSM network. The primary advantage of GPRS over HSCSD is its enhanced flexibility and resource efficiency, inherently due to its packet-switched character. For this reason, GPRS is particularly suitable for bursty applications. The advantages of HSCSD with respect to GPRS are threefold:

(1) HSCSD can be commercially introduced at least a year before GPRS, potentially yielding a competitive advantage in the mobile data market, since it can give an operator the means to satisfy short-term demand, and already attract a small league of mobile data customers;

(2) since only software upgrades are required in the radio access network, HSCSD is much cheaper to deploy;

(3) due to its circuit-switched character, HSCSD data transfers will be more reliable and delay variations will be smaller; HSCSD is most appropriate for real-time high bit rate applications and for the transfer of large data files that require some minimum data transfer rate.

Note that the first release of GPRS does not incorporate any mechanisms for QOS provisioning or differentiation, hence at least initially HSCSD offers the *only* way to provide throughput or delay guarantees. Since HSCSD and GPRS are optimized for different types of applications with different QOS requirements, the data bearer services complement each other, and are likely to coexist in a matured GSM/(E)HSCSD/(E)GPRS network, whereby the 'E' indicates the use of EDGE.

A GSM network operator offering HSCSD services to its customers must implement new radio resource management algorithms to optimize resource efficiency and service quality. In this paper we concentrate on the performance of different admission control and channel assignment policies in a GSM/HSCSD network, in terms of voice and data call blocking probability, data call delays and channel utilization.

1.3. Literature review

Few papers have been published that present a performance analysis of an integrated GSM/HSCSD network. Calin and Zeghlache [8,9] present three different channel allocation policies which are evaluated either by simulation or brute force Markov chain analysis. With their *maximum capacity* policy, an HSCSD call requests a fixed number of channels and is blocked if the requested number of channels is not available. Under the *no rate adaptation* policy an admitted HSCSD call grabs as many channels as there are available, up to its technical maximum transfer capability, and the channel assignment remains fixed for the duration of a call. Under the *rate adaptation* policy a call can grab additional channels as they become available, up to its technical maximum. In the policies proposed by Calin and Zeghlache [8,9] data calls are never downgraded in their assignments, e.g., in order to support a newly originating voice (or data) call. As we will demonstrate, not only upgrading of channel assignments is desirable in order to optimize channel utilization and delay performance, but also downgrading is essential primarily to keep the voice call blocking probability low, which is expected to be of great concern to a mobile network operator freshly entering the data market. Jeng et al. [18] present an evaluation of three types of channel allocation policies in a GSM/HSCSD network. Besides the *maximum capacity* and *no rate adaptation* policies, the authors also propose a 'soft' policy where the *fixed* number of channels that is assigned to a new HSCSD data call, and hence also the admission control rule, is dynamically adjusted based on the blocking statistics. The policies are evaluated by a network simulation with 64 cells, including user mobility. The primary drawback of this study is that the traffic generated consists solely of HSCSD data calls and thus excludes GSM voice calls, while in a realistic network the scarce cell capacity is to be shared between both service types.

In the literature, related traffic management issues in GSM/GPRS networks have been studied mainly with a focus on (generally single cell) data-only networks and use system-level simulations in order to (i) investigate the impact of the system and environment parameters such as data traffic load, job size distribution and channel availability on performance measures such as call blocking, delay and throughput (see, e.g., [Brasche and Walke, 4; Cai and Goodman, 6; Calin et al., 7]); or (ii) compare different scheduling algorithms (e.g., First-Come First-Served, Round Robin, Earliest Deadline First, Static Priority Scheduling) to handle data traffic (see, e.g., [Ajib and Godlewski, 1; Johansson et al., 19; Pang et al., 29]). A multiple cell network is considered by Johansson et al. [19] in order to model frame errors due to interference. (Modified) versions of the Round Robin and Earliest Deadline First disciplines are generally considered most promising, although the latter is also expected to be rather complex from an implementation viewpoint [Pang et al., 29]. Integrated voice (GSM) / data (GPRS) networks with a dynamic radio resource allocation algorithm have been studied via simulation by Bianchi et al. [3], Chuang et al. [10] and Kennedy and Litjens [21] where the grade (voice) and quality of service (data) is determined as a function of the voice and data traffic loads, as well as the number of dedicated data traffic channels. Kennedy and Litjens [21] illustrate the dependency of the data call delays and the present number of (prioritised)

voice calls by means of a typical simulation trace. As an application of the model and analysis presented in the underlying paper, the performance of a fully segregated and a hybrid radio resource sharing scheme have been evaluated analytically by Litjens and Boucherie [22], in order to quantify the capacity gain that can be achieved when utilizing the idle periods between voice calls by filling these gaps with packet data.

UMTS studies that focus on related traffic management issues, include those by De Bernardi et al. [13], Ramakrishna and Holtzman [32] and Wu et al. [38]. De Bernardi et al. [13] compare different radio resource sharing schemes for a multiple cell CDMA network serving conversational (e.g., voice) and interactive (e.g., data) calls. The presented simulation results are in accordance with our results in section 5.1, obtained via analytical methods. An alternative to our fair channel sharing approach is considered by Ramakrishna and Holtzman [32], where it is analytically shown for a CDMA cell integrating voice and delay tolerant data calls, that the average data throughput is optimized if at any time the capacity that is unused by the prioritized voice calls is assigned in full to only a single (randomly or cyclically appointed) data call, even if the other (temporarily inactive) data calls still require a small idle rate to maintain proper power control and synchronisation. We note that the optimality of this scheme is restricted to CDMA-based networks, while it does not hold for, e.g., GSM/GPRS networks. Although an interesting and somewhat counterintuitive result, it is unlikely to be of use in live networks as it assumes that there is no limitation on an individual data call's peak transfer rate, while it disregards the undesired effects of the implied burstiness of the generated interference. Wu et al. [38] study a single CDMA cell serving premium (e.g., voice), assured (e.g., WWW) and best-effort (e.g., e-mail) services. The impact of different capacity sharing schemes (e.g., different radio resource reservation schemes) on the experienced grade and quality of service is evaluated by means of simulations.

A relatively large body of literature exists on performance modelling and analysis for *fixed* broadband multi-service networks (e.g., ATM, IP), a good overview of which is given in the final report of the COST 242 project (see [Roberts et al., 34]). The analysis in the underlying paper makes use of the approaches and results published by Avi-Itzhak and Halfin [2], who present a sojourn time (waiting time *plus* transfer time) analysis for a single server system serving a single type of jobs according to a processor sharing service discipline; Roberts [35], who suggests that the performance of (elastic) data traffic under TCP flow control can be modelled by a processor sharing queue; and by Núñez Queija et al. [27], who a.o. analyse the conditional expected transfer time of data calls in an integrated system with stream and elastic traffic.

1.4. Contribution and outline

We present an extensive performance analysis of a class of fair channel sharing policies in an integrated GSM/HSCSD network. Under any policy in the studied class, active data calls are guaranteed a minimum channel assignment, and may be granted the use of additional channels, until these are claimed by new voice or data calls. Data calls can be up- or downgraded in their channel assignment, but only at times of a voice or data

call arrival or termination. Admitted data calls that cannot start transfer immediately are queued. The primary contributions of the paper are threefold. First, a unified framework for channel sharing policies is presented. Second, the conditional expected sojourn time (waiting time *plus* transfer time) of data calls is explicitly analysed and a closed-form expression is derived for the sojourn time of a newly arriving data call, conditional on its size. Third, the paper presents an extensive numerical evaluation of four proposed channel sharing policies.

The outline of the paper is as follows. The mathematical framework is set in section 2. It includes the assumptions regarding call characteristics and call handling procedures, and thus defines the studied general class of channel sharing policies. At the end of the section, the assumptions are formulated in a Markov chain model. An extensive performance analysis of the policies in this class is provided in section 3, where besides some basic performance measures that can be directly derived from the Markov chain's equilibrium distribution, a conditional expected sojourn time analysis is presented for data calls. Subsequently, four different channel sharing policies that fit within the general class are described in section 4, while a numerical evaluation of these policies is given in section 5. Section 6 ends this paper with some concluding remarks.

2. Model

In this section we define the framework for our performance analysis. This framework consists of two distinct parts, the assumed call characteristics and the call handling procedures, which form the primary ingredients of the Markov chain model described at the end of this section.

2.1. Call characteristics

Consider a single cell in a GSM/HSCSD network, serving circuit-switched voice and data calls. Voice and data calls arrive according to two mutually independent Poisson processes, with arrival intensities λ_{voice} and λ_{data} calls per second, respectively. An admitted *voice call* is served with a single dedicated traffic channel for its entire duration, assumed to be exponentially distributed with mean $1/\mu_{\text{voice}}$. The voice traffic load is $\rho_{\text{voice}} \equiv \lambda_{\text{voice}}/\mu_{\text{voice}}$.

A *data call* is assumed to be the downlink transfer of a file with an exponentially distributed size. We assume an information bit rate of r kbits/s per traffic channel, and express the data call size in units of r kbits, so that both the data call size and the data call holding time given the exclusive use of a single traffic channel, are exponentially distributed with mean denoted $1/\mu_{\text{data}}$ (in r kbits or seconds, respectively). The data traffic load is given by $\rho_{\text{data}} \equiv \lambda_{\text{data}}/\mu_{\text{data}}$. The number of traffic channels that can be assigned to a data call must be between the requested minimum b and the technical maximum B, with $b \leqslant B$. Since in TDMA-based (Time-Division Multiple Access) GSM, a single carrier frequency is time-sliced into eight physical channels, $b, B \in \{1, 2, \ldots, 8\}$ must hold. When multiple traffic channels are assigned to a data call, we neglect the technical

requirement that these channels must be on the same carrier frequency, for reasons of analytical tractability. In our model it is assumed that for a given file there is at any time sufficient data available in the data buffer to be carried on the assigned radio traffic channels, thus supposing that end-to-end (TCP) flow control, deployed in real networks to limit the amount of data in transit, performs most efficiently. As the effectiveness of TCP over wireless access networks such as GSM/HSCSD is a serious matter of debate, this matter is elaborated below.

2.1.1. On the impact of TCP in a GSM/HSCSD network

Unless the data call is processed over an end-to-end circuit-switched connection, end-to-end error and congestion control are typically executed at the TCP (Transmission Control Protocol) transport layer operating on top of the IP (Internet Protocol) network layer. In several publications (e.g., [Parsa and Garcia-Luna-Aceves, 30; Rivadeneyra Sicilia and Miguel-Alonso, 33]), potential problems have been identified at the transport layer that arise in the case of a wireless access network. These are a consequence of the design premise that TCP was intended for relatively fast and reliable fixed networks, rather than for slow and unstable radio networks. Regarding *error control*, the relatively high frame error rates and corresponding link level delays that are common in the radio interface can easily induce TCP to generate unnecessary retransmissions. TCP may even confuse excessive delays with a connection loss, forcing the application to make an expensive reconnection. Furthermore, TCP's *congestion control* may falsely interpret such link level delays as a congestion symptom, whereas it may very well be caused by link level retransmissions of erroneous frames due to fading. In response, transmission windows and retransmission timeouts may be adjusted, leading to a potentially inappropriate flow reduction. As a consequence, data throughputs and resource efficiency are reduced, a highly undesirable effect in the radio interface, given its inherently scarce capacity.

Although little practical experience exists regarding these matters, due to the fact that mobile data communications is still in its infancy, a great variety of theoretical performance studies and proposals for TCP improvement can be found in the literature. Aside from physical layer solutions of increased forward error correcting coding to lower the frame error rate at the cost of a reduced throughput, higher-layer solutions for the anticipated problems attempt to fool TCP by hiding the lossiness of the wireless link (e.g., [Parsa and Garcia-Luna-Aceves, 30; Rivadeneyra Sicilia and Miguel-Alonso, 33]). In contrast to the concerns that triggered these studies, the performance analysis of standard TCP for GSM/GPRS networks presented by Meyer [24] demonstrates that TCP and GPRS's ARQ mechanism are well harmonised, as the ARQ scheme is appropriately designed to ensure that TCP observes just packet delays rather than packet losses. Although no HSCSD-specific TCP performance study is available in the literature, it is expected that appropriate measures are taken, if necessary, to avoid any performance degradation caused by TCP's misinterpretations of radio interface events.

Our model ignores frame errors on the data traffic channels and hence any experienced (queueing) delay is indeed caused by radio interface *congestion* only (not due to link level retransmissions). Such delays therefore suitably induce TCP to slow down the

source rate. Under an ideal TCP feedback mechanism, and assuming that the wireless segment is the primary bottleneck in the end-to-end connection, we argue that the variability at which TCP feeds the data buffer in the GSM/HSCSD network, is induced by and hence in direct correspondence with the variability of the data transfer rate over the air interface, so that it is indeed plausible to assume that the buffers are never empty as long as the file is not fully transferred (see also [Roberts and Massoulie, 35]).

2.2. *Call handling procedures*

Denote the cell capacity, i.e., the number of traffic channels available in the considered cell, with C_{total}. A *channel sharing policy* prescribes how the channel pool is shared by voice and data calls, and specifies both a call admission control policy and a channel assignment policy.

In this paper we will study the performance of a specific class of channel sharing policies. A common characteristic of all policies in this class is that at a given time all active data calls share the available channel capacity *fairly*, in our model defined such that at any given time the available resources are distributed evenly over the present data calls. It is readily verified that within the considered setting, i.e., a single-link system whose varying capacity is to be distributed over a single service class (data calls), an even resource distribution over the present data calls is indeed both *max–min fair* and *proportionally fair* (see [Kelly, 20]). Under each such fair channel sharing policy, the evolution of the system can be modelled as a 2-dimensional continuous-time Markov chain $(V(t), D(t))_{t\geqslant 0}$, where $V(t)$ and $D(t)$ are defined as the number of voice and data calls, respectively, that is, present at time t. The system states are denoted (v, d). The data calls present in the system are either *active*, i.e., in transfer on one or more traffic channels, or *queued*, in case no channels can be assigned immediately upon admission. A queued call becomes active once sufficient resources are freed and it remains active until it terminates.

A channel sharing policy in the given class is characterized by three basic functions. Firstly, $\beta(v, d)$ is the expected number of channels that is assigned to each active data call in state (v, d), i.e., the total number of traffic channels assigned to data calls divided by the number of active data calls. The circuit-switched character of the HSCSD service allows only an integer number of channels to be assigned to an active data call. The proposed *fair* channel sharing policies (re)assign, at each call arrival or termination event, the traffic channels available for data transfer as follows. Each active data call receives a basic assignment of $\underline{b}(v, d) \equiv \lfloor \beta(v, d) \rfloor$ channels which is the maximum assignment that can be uniformly awarded, while the remaining channels are randomly distributed over the active data calls, so that some fortunate data calls obtain an additional channel and are assigned $\overline{b}(v, d) \equiv \lceil \beta(v, d) \rceil = \underline{b}(v, d) + 1$ channels. Hence an active data call is assigned either $\underline{b}(v, d)$ or $\overline{b}(v, d)$ channels with respective probabilities $\underline{p}(v, d) \equiv \lceil \beta(v, d) \rceil - \beta(v, d)$ and $\overline{p}(v, d) \equiv 1 - \underline{p}(v, d) = \beta(v, d) - \lfloor \beta(v, d) \rfloor$. It is intuitively clear and easily verified that

$$\underline{p}(v, d)\, \underline{b}(v, d) + \overline{p}(v, d)\, \overline{b}(v, d) = \beta(v, d). \tag{1}$$

Secondly, the maximum number of data calls that can be allowed in the cell when there are v voice calls present, is denoted $d_{\max}(v)$. Incorporated in $d_{\max}(v)$ is the possibility of queueing data calls.

An admitted data call that cannot start service immediately is held in a fixed-size first-come first-served queue that can store up to Q_{data} requests, until the system has evolved to a state where sufficient capacity is freed to serve the call. We stress that freed capacity should indeed be immediately assigned to a queued data call, if possible, rather than allowing the active calls to be served faster. Although not all manufacturers support this feature, the GSM/HSCSD standards allow queueing of circuit-switched calls (see [ETSI, 16]). A fresh (or handover) call request that cannot be assigned a traffic channel immediately can be put in a BSC queue in the hope that sufficient capacity is freed before a timeout occurs. Effectively, this means that the admit/reject decision is postponed for a few seconds. In our model the queueing feature is implemented for HSCSD data calls only, as a higher delay tolerance is expected from the corresponding users. Still, the value of Q_{data} must be small in order to appropriately mimic the typically small timeout value. As will be demonstrated in section 5, the optimal size of Q_{data} depends on multiple factors, e.g., the cell capacity C_{total}, the data traffic load ρ_{data}, and even on the average file size $1/\mu_{\text{data}}$, since for a given data traffic load, a smaller average file size implies that more data calls can be queued and still be served within the allowed call setup time.

In order to define an equivalent function for the maximum number of admissable voice calls, we note that the admission of a newly arriving voice call depends on the actual voice/data call configuration, i.e., on the system state (v, d), rather than on d alone. The rationale behind this is that the number of active and queued data calls is determined by both v and d, and hence the number of available traffic channels as well. Note that it is possible for a data call to be queued even when some traffic channels are idle, either because these idle channels are reserved for voice calls, or because the number of channels that is (or can be made) available is less than minimum requirement b. As a consequence, (v, d) determines whether or not a new voice call can be admitted, and thus the third function to be specified is denoted $v_{\max}(v, d)$, defined such that $v_{\max}(v, d) - v$ is the maximum number of voice calls that can still be admitted, starting from state (v, d).

Consider a simple illustrative example with $C_{\text{total}} = 2$, $b = B = 2$, $Q_{\text{data}} = 1$, and full sharing of traffic channels between voice and data calls. The corresponding state space is depicted in figure 2, where for each admissable state (v, d), $v_{\max}(v, d)$ and $d_{\max}(v)$ are given. The block arrows indicate the possible state transitions. Note that $v_{\max}(0, 1) = 0$ since there is one active data call, occupying both channels, so that a newly arriving voice call must be blocked. Compare this with state $(1, 1)$ with one active voice call and one queued data call, where $v_{\max}(1, 1) = 2$, since there is one idle channel that may be assigned to a new voice call. Apparently, we need to know the number of voice calls v in the system, to know how many of the d data calls are active or queued, which in turn determines how many more voice calls can be admitted.

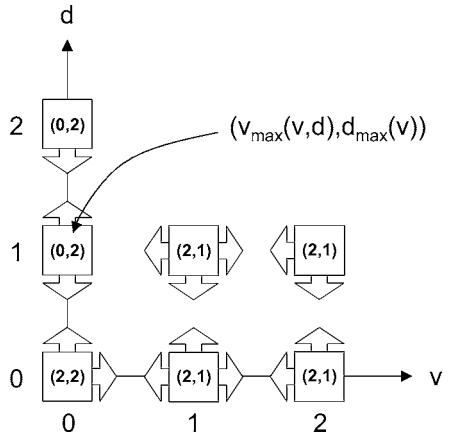

Figure 2. Illustration of state space, $v_{\max}(v, d)$ and $d_{\max}(v)$.

Two additional parameters that will be useful in the analysis can be derived from the basic functions. Denote with $v_{\max} \equiv v_{\max}(0, 0)$ and $d_{\max} \equiv d_{\max}(0)$ the absolute maximum number of voice and data calls the system can support, respectively.

These three basic functions are sufficient to specify an *admission control* policy. If at time t a newly originating voice call sees the system in state (v, d), it is admitted if and only if $v < v_{\max}(v, d)$, i.e., if and only if there is still a spare channel to serve the voice call (there is no queue for voice calls). Similarly, a new data call is admitted if $d < d_{\max}(v)$. If $Q_{\text{data}} > 0$ then $d = d_{\max}(v)$ if and only if the queue is full. All blocked calls are cleared from the system. Furthermore, these functions fully specify a *channel assignment* policy as well. In state (v, d), each of the v voice calls is active and assigned a single channel. Of the d data calls present, $d_a(v, d) \equiv \min\{d, d_{\max}(v) - Q_{\text{data}}\}$ are active, fairly sharing a total of $d_a(v, d)\,\beta(v, d)$ channels, while the remaining $d_q(v, d) \equiv d - d_a(v, d)$ data calls are queued. An active data call can never be pushed back into the queue. Rather, it is assigned at least b channels at any time, until its transfer is completed.

2.3. Markov chain

Under any channel sharing policy in the considered class, the evolution of the system can be described by an irreducible 2-dimensional continuous-time Markov chain $(V(t), D(t))_{t \geqslant 0}$, with states denoted (v, d). The state space of the Markov chain is given by

$$\mathbb{S} \equiv \big\{(v, d) \in \mathbb{N}_0 \times \mathbb{N}_0 \colon\ v \leqslant v_{\max}(v, d) \text{ and } d \leqslant d_{\max}(v)\big\}.$$

Ordering the state space lexicographically in (v, d), the infinitesimal generator is given by

$$\mathcal{Q} \equiv \begin{pmatrix} \mathcal{C}_0 & \mathcal{A}_0 & \mathcal{O} & \dots & \dots & \mathcal{O} \\ \mathcal{B}_1 & \mathcal{C}_1 & \mathcal{A}_1 & \mathcal{O} & \ddots & \vdots \\ \mathcal{O} & \mathcal{B}_2 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \mathcal{A}_{v_{\max}-2} & \mathcal{O} \\ \vdots & \ddots & \mathcal{O} & \mathcal{B}_{v_{\max}-1} & \mathcal{C}_{v_{\max}-1} & \mathcal{A}_{v_{\max}-1} \\ \mathcal{O} & \dots & \dots & \mathcal{O} & \mathcal{B}_{v_{\max}} & \mathcal{C}_{v_{\max}} \end{pmatrix} \in \mathbb{R}^{|\mathbb{S}|} \times \mathbb{R}^{|\mathbb{S}|},$$

with

$$\begin{aligned} |\mathbb{S}| &= \sum_{v=0}^{v_{\max}} \big(d_{\max}(v) + 1\big) \leqslant (v_{\max} + 1) \times (d_{\max} + 1) \\ &\leqslant (C_{\text{total}} + 1) \times \left(\left\lfloor \frac{C_{\text{total}}}{b} \right\rfloor + Q_{\text{data}} + 1 \right). \end{aligned}$$

The *super-diagonal* blocks $\mathcal{A}_v$ generate voice call arrival events, for $v = 0, \dots, v_{\max} - 1$. $\mathcal{A}_v$ is of dimension $(d_{\max}(v)+1) \times (d_{\max}(v+1)+1)$ and has entries $\mathcal{A}_v(d, d) \equiv \lambda_{\text{voice}} \mathbb{1}\{v < v_{\max}(v, d)\}$ for $d = 0, \dots, d_{\max}(v+1)$, and $\mathcal{A}_v(d, d') \equiv 0$ for $d \neq d'$, where the indicator function $\mathbb{1}\{\cdot\}$ returns 1 if the argument event is true, and 0 otherwise. Note that the inclusion of the factor $\mathbb{1}\{v < v_{\max}(v, d)\}$ is necessary to cope with a situation as described in section 2.2, where a state $(v + 1, d)$ is in principle admissable, yet it cannot be reached from state (v, d). The *subdiagonal* blocks $\mathcal{B}_v$ generate voice call termination events, for $v = 1, \dots, v_{\max}$. $\mathcal{B}_v$ is of dimension $(d_{\max}(v) + 1) \times (d_{\max}(v - 1) + 1)$ and has entries $\mathcal{B}_v(d, d) \equiv v\mu_{\text{voice}}$ for $d = 0, \dots, d_{\max}(v)$, and $\mathcal{B}_v(d, d') \equiv 0$ for $d \neq d'$. Finally, the square blocks $\mathcal{C}_v$ on the *diagonal* generate data call arrival and termination events, for $v = 0, \dots, v_{\max}$. $\mathcal{C}_v$ is of dimension $(d_{\max}(v) + 1) \times (d_{\max}(v) + 1)$ and has entries $\mathcal{C}_v(d - 1, d) \equiv \lambda_{\text{data}}$ and $\mathcal{C}_v(d, d - 1) \equiv \beta(v, d)\, d_a(v, d)\, \mu_{\text{data}}$ for $d = 1, \dots, d_{\max}(v)$. Furthermore, the diagonal entries of $\mathcal{C}_v$ are such that the entries of each row of $\mathcal{Q}$ sum up to 0. All other entries of $\mathcal{C}_v$ are equal to zero.

Since the finite state space Markov chain is irreducible, there is a unique probability vector π that satisfies the system of linear equations (e.g., [Tijms, 36])

$$\boldsymbol{\pi}\, \mathcal{Q} = \mathbf{0},$$

with $\mathbf{0}$ the vector with all entries zero, and $\boldsymbol{\pi}$ lexicographically ordered in the system states $(v, d) \in \mathbb{S}$.

3. Performance analysis

Now that the model is formulated, we are able to specify the relevant performance measures. First, some basic performance measures are given, that can be derived from the equilibrium distribution of the Markov chain, while the remainder of this section focuses on the conditional expected sojourn time of a data call of size x.

3.1. Basic performance measures

The following performance measures can be calculated directly from the stationary state probabilities.

From a system perspective, the resource efficiency yielded by each channel sharing policy can be measured by the *expected channel utilization,*

$$\mathbf{U} \equiv C_{\text{total}}^{-1} \sum_{(v,d)\in\mathbb{S}} \big(v + \beta(v,d)\, d_a(v,d)\big)\, \pi(v,d).$$

The performance of the channel sharing policies with respect to the speech and data services is primarily measured by the *blocking probabilities,*

$$\mathbf{P}_{\text{voice}} \equiv \sum_{(v,d)\in\mathbb{S}} \mathbb{1}\big\{v = v_{\max}(v,d)\big\}\, \pi(v,d),$$

and

$$\mathbf{P}_{\text{data}} \equiv \sum_{0\leqslant v\leqslant v_{\max}} \pi\big(v, d_{\max}(v)\big),$$

using the PASTA (Poisson Arrivals See Time Averages) property (e.g., [Wolff, 37]). Another performance measure that is easily determined is the *expected number of calls* of each type in the system,

$$\mathbf{N}_{\text{voice}} \equiv \sum_{(v,d)\in\mathbb{S}} v\, \pi(v,d) = \rho_{\text{voice}}(1 - \mathbf{P}_{\text{voice}}),$$

and

$$\mathbf{N}_{\text{data}} \equiv \sum_{(v,d)\in\mathbb{S}} d\, \pi(v,d),$$

where the equality follows from Little's formula (e.g., [Wolff, 37]). $\mathbf{N}_{\text{data}}$ can be split up into the *expected number of active* and *queued data calls* in the system, given by $\mathbf{N}_{\text{data}}^{\text{active}} \equiv \sum_{(v,d)\in\mathbb{S}} d_a(v,d)\, \pi(v,d)$ and $\mathbf{N}_{\text{data}}^{\text{queue}} \equiv \sum_{(v,d)\in\mathbb{S}} d_q(v,d)\, \pi(v,d)$, respectively.

The *expected number of channels assigned to an active data call* is given by

$$\mathbf{B}_{\text{data}} \equiv \frac{\sum_{(v,d)\in\mathbb{S}} \beta(v,d)\, \pi(v,d)}{\sum_{(v,d)\in\mathbb{S}} \mathbb{1}\{d_a(v,d) > 0\}\, \pi(v,d)},$$

which must obviously lie somewhere within the range of technical limitations, i.e., $\mathbf{B}_{\text{data}} \in [b, B]$. The relevance of this measure lies therein that the time-average throughput of an active data call is equal to $r\, \mathbf{B}_{\text{data}}$. It is stressed that the access delay is not incorporated in $\mathbf{B}_{\text{data}}$.

For the data service, the delivered quality of service, expressed as the expected sojourn time of a data call, is important from the users' point of view. Again using Little's formula, the *expected waiting time* and the *expected transfer time of a data call* are given by

$$\mathbf{T}_{\text{data}}^{\text{queue}} \equiv \frac{\mathbf{N}_{\text{data}}^{\text{queue}}}{\lambda_{\text{data}}(1 - \mathbf{P}_{\text{data}})},$$

and

$$\mathbf{T}_{\text{data}}^{\text{active}} \equiv \frac{\mathbf{N}_{\text{data}}^{\text{active}}}{\lambda_{\text{data}}(1 - \mathbf{P}_{\text{data}})},$$

respectively, so that the *expected sojourn time of a data call* is given by $\mathbf{T}_{\text{data}}^{\text{sojourn}} \equiv \mathbf{T}_{\text{data}}^{\text{queue}} + \mathbf{T}_{\text{data}}^{\text{active}}$.

3.2. *Analysis of the conditional expected sojourn time*

The expected sojourn time $\mathbf{T}_{\text{data}}^{\text{sojourn}}$ of a data call can be easily computed, once the steady state probabilities have been determined, as demonstrated above. However, for data calls we are also interested in determining $\mathbf{T}_{\text{data}}^{\text{sojourn}}(x)$, the expected sojourn time of an admitted data call of size x, in order to be able to judge to what extent the different channel sharing policies are able to establish fairness w.r.t. data calls of various sizes. As a valuable intermediate result in the analysis, the conditional expected sojourn time of an admitted data call of size x is obtained, *given the system state at arrival*, which may be fed back to the data source as an informative indication of the expected call handling time.

It is convenient to partition the state space $\mathbb{S}$ and introduce some additional notation. Denote with $\mathbb{S}_0 \equiv \{(v, d) \in \mathbb{S}\colon\ d = 0\}$ the set of states with no data calls, and with $\mathbb{S}_+ \equiv \mathbb{S}\backslash\mathbb{S}_0$ its complement. Further, denote with $\mathbb{S}_+^{a_0} \equiv \{(v, d) \in \mathbb{S}_+\colon\ d_a(v, d) = 0\}$ the subset of $\mathbb{S}_+$ with no active data calls, while $\mathbb{S}_+^{a_+} \equiv \mathbb{S}_+\backslash\mathbb{S}_+^{a_0}$ is its complement within $\mathbb{S}_+$. Lastly, let $\mathbb{S}_+^{q_0} \equiv \{(v, d) \in \mathbb{S}_+\colon\ d_q(v, d) = 0\}$ be the subset of $\mathbb{S}_+$ with no queued data calls, and let $\mathbb{S}_+^{q_+} \equiv \mathbb{S}_+\backslash\mathbb{S}_+^{q_0}$ be its complement in $\mathbb{S}_+$. This partitioning is illustrated in figure 3.

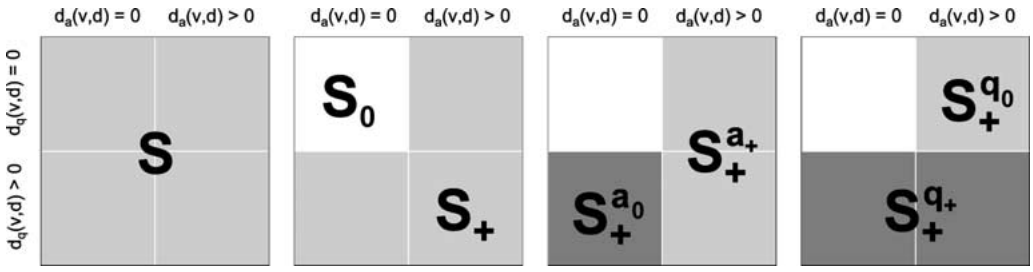

Figure 3. Illustration of state space partitioning.

For each $(v, d) \in \mathbb{S}_+$ define $\sigma_{v,d}(x)$ as the random sojourn time of an admitted data call of size x, arriving at *given* system state (v, d) with v voice and d data calls (d *includes* the new call), and let $\widehat{\sigma}_{v,d}(x) \equiv \mathbf{E}\{\sigma_{v,d}(x)\}$ denote its expectation. Then the expected sojourn time of an admitted data call of size x is given by

$$\mathbf{T}_{\text{data}}^{\text{sojourn}}(x) = \frac{1}{1 - \mathbf{P}_{\text{data}}} \sum_{(v,d)\in\mathbb{S}_+} \widehat{\sigma}_{v,d}(x)\, \pi(v, d-1), \tag{2}$$

where $\pi(v, d)/(1 - \mathbf{P}_{\text{data}})$ is the equilibrium probability that an *accepted* data call finds v voice calls and d *other* data calls in the system, and thus

$$\mathbf{T}_{\text{data}}^{\text{sojourn}} = \int_{x=0}^{\infty} \mathbf{T}_{\text{data}}^{\text{sojourn}}(x)\, \mu_{\text{data}} \exp\{-x\, \mu_{\text{data}}\}\, \mathrm{d}x.$$

In order to determine the functions $\widehat{\sigma}_{v,d}(x)$, $(v, d) \in \mathbb{S}_+$, $x \in \mathbb{R}^+$, we may split the expected sojourn time into a waiting time and a transfer time component, along the lines followed by [Avi-Itzhak and Halfin, 2], where a sojourn time analysis is presented for a single server system serving a single type of jobs with a processor sharing service discipline, and a limited number of service positions.

In analogy with the sojourn time variables, denote with $\omega_{v,d}$ the random waiting time of a newly admitted data call of size x, entering the system in state (v, d), which is obviously independent of the call size, due to the first-come first-served queueing discipline, and let $\widehat{\omega}_{v,d} \equiv \mathbf{E}\{\omega_{v,d}\}$ be its expectation. Similarly, denote with $\tau_{v_o,d_o}(x)$ the random transfer time of an admitted data call of size x, that starts its transfer in system state $(v_o, d_o) \in \mathbb{S}_+^{a+}$, and let $\widehat{\tau}_{v_o,d_o}(x) \equiv \mathbf{E}\{\tau_{v_o,d_o}(x)\}$ be its expectation. The final ingredient required to formulate theorem 1 below is $\mathcal{J}((v, d); (v_o, d_o))$, defined as the probability that a data call entering the system in state $(v, d) \in \mathbb{S}_+$ starts its transfer in state $(v_o, d_o) \in \mathbb{S}_+^{a+}$.

Theorem 1. The conditional expected sojourn time $\widehat{\sigma}_{v,d}(x)$, $(v, d) \in \mathbb{S}_+$, $x \in \mathbb{R}^+$, can be expressed as the sum of the expected waiting time and the expected transfer time, as follows:

$$\widehat{\sigma}_{v,d}(x) = \widehat{\omega}_{v,d} + \sum_{(v_o,d_o)\in\mathbb{S}_+^{a+}} \widehat{\tau}_{v_o,d_o}(x)\, \mathcal{J}\big((v, d); (v_o, d_o)\big). \tag{3}$$

Proof. The proof of theorem 1 is given in the appendix. □

The $\widehat{\omega}_{v,d}$, $(v, d) \in \mathbb{S}_+$, are not required to obtain $\mathbf{T}_{\text{data}}^{\text{sojourn}}(x)$, which becomes clear when substituting (3) in (2). Still, the values of $\widehat{\omega}_{v,d}$ are of interest to get some insight into the variability of the waiting times in light of the practical restriction on how long a data call request can be queued. Furthermore, a mobile station may be informed of the expected delay until it can start its data transfer.

Now that we have justified the above split-up of the expected sojourn time, we will derive closed-form expressions to calculate the expected waiting and transfer times,

as well as the required transition probabilities. Once these expressions are derived, all ingredients are available to compute the expected conditional sojourn time of a data call, using (2) and (3).

3.2.1. Waiting time analysis

Consider the reducible continuous-time Markov chain that results when the data call arrival process in the original Markov chain is turned off upon arrival of a tagged data call. With this adjustment, the system state explicitly indicates the tagged call's position in the queue, which enables us to determine its expected waiting time, given by the time until the queue is emptied. The adjusted Markov chain has infinitesimal generator $\widetilde{\mathcal{Q}}$, which is similar to $\mathcal{Q}$, except that all data call arrival rates have been set to zero, i.e., $\lambda_{\text{data}} = 0$, while the diagonal elements are adjusted accordingly. Hence only the submatrices $\widetilde{\mathcal{C}}_v$ differ from $\mathcal{C}_v$, $v = 0, \ldots, v_{\max}$.

The state space of the adjusted Markov chain remains $\mathbb{S}$. For our purposes, it is convenient to make use of the partitioning of $\mathbb{S}_+$ into an absorbing set $\mathbb{S}_+^{q0}$ of states where the queue is empty and its transient complement, $\mathbb{S}_+^{q+}$.

As was argued above, the waiting time of a queued data call in the original Markov chain is independent of the future data call arrival process. Hence $\widehat{\omega}_{v,d}$ is equal to the expected time it takes for the adjusted chain to evolve from state (v, d) into any state in the absorbing set $\mathbb{S}_+^{q0}$, i.e., $\widehat{\omega}_{v,d}$ is the expected *absorption time* of $\mathbb{S}_+^{q0}$, starting from state (v, d). Trivially, $\widehat{\omega}_{v,d} = 0$ for all $(v, d) \in \mathbb{S}_+^{q0}$ since the tagged call can start its transfer immediately upon arrival. Let the vector $\widehat{\boldsymbol{\omega}}_+^n$, $n \in \mathbb{N}_0$, contain the cumulative expected waiting time $\widehat{\omega}_{v,d}^n$ after n state transitions from initial state $(v, d) \in \mathbb{S}_+^{q+}$, lexicographically ordered in the (v, d). Then this vector of cumulative expected waiting times evolves according to the following recursive relation:

$$\widehat{\boldsymbol{\omega}}_+^{n+1} = \widehat{\boldsymbol{\omega}}_+^* + \widetilde{\mathcal{P}}_{++}\, \widehat{\boldsymbol{\omega}}_+^n, \quad n \in \mathbb{N}_0, \text{ with initial condition } \widehat{\boldsymbol{\omega}}_+^0 = \mathbf{0}, \tag{4}$$

where the vector $\widehat{\boldsymbol{\omega}}_+^*$ contains the expected waiting times ($\widehat{\omega}_{v,d}^*$, $(v, d) \in \mathbb{S}_+^{q+}$) until the next state transition, obtained by conditioning on the possible events:

$$\widehat{\omega}_{v,d}^* = \left[\lambda_{\text{voice}}\, \mathbb{1}\{v < v_{\max}\} + v\, \mu_{\text{voice}} + \beta(v, d)\, d_a(v, d)\, \mu_{\text{data}}\right]^{-1}.$$

$\widetilde{\mathcal{P}}_{++}$ denotes the one-step transition probability matrix of the embedded discrete-time Markov chain that follows the transitions within $\mathbb{S}_+^{q+}$ of the adjusted continuous-time Markov chain with generator $\widetilde{\mathcal{Q}}$, i.e.,

$$\widetilde{\mathcal{P}}_{++}\big((v, d); \big(v', d'\big)\big) = \begin{cases} \dfrac{\widetilde{\mathcal{Q}}((v, d); (v', d'))}{-\widetilde{\mathcal{Q}}((v, d); (v, d))} & \text{if } (v, d) \neq \big(v', d'\big), \\ 0 & \text{otherwise,} \end{cases}$$

for all (v, d), $(v', d') \in \mathbb{S}_+^{q+}$. Note that probability matrix $\widetilde{\mathcal{P}}_{++}$ is substochastic since it excludes all possible state transitions from $\mathbb{S}_+^{q+}$ to $\mathbb{S}_+^{q0}$, and has dimensions $|\mathbb{S}_+^{q+}| \times |\mathbb{S}_+^{q+}|$.

Finally, the initial condition of the recursive relation simply reflects that the cumulative expected waiting time of the tagged data call is initialized to zero when it enters the system.

We can now formulate the following theorem.

Theorem 2. The expected waiting time of a data call which enters the system in state (v, d) in $\mathbb{S}_+^{q+}$ is contained in the vector $\widehat{\boldsymbol{\omega}}_+$, given by

$$\widehat{\boldsymbol{\omega}}_+ = \left(\mathcal{I} - \widetilde{\mathcal{P}}_{++}\right)^{-1} \widehat{\boldsymbol{\omega}}_+^*.$$

Proof. The proof of theorem 2 is given in the appendix. □

3.2.2. Transition probabilities

We now compute the probability matrix $\mathcal{J}$ containing the probabilities $\mathcal{J}((v, d); (v_o, d_o))$ that a call entering the system in state $(v, d) \in \mathbb{S}_+$ starts its transfer in state $(v_o, d_o) \in \mathbb{S}_+^{a+}$. Hence matrix $\mathcal{J}$ is of dimension $|\mathbb{S}_+| \times |\mathbb{S}_+^{a+}|$, some of whose entries are readily determined. For instance,

$$\mathcal{J}\big((v, d); (v, d)\big) = 1 \quad \text{for } (v, d) \in \mathbb{S}_+^{q0},$$

since the arriving data call can immediately start its transfer.

In order to compute $\mathcal{J}$, we augment the state space, $\mathbb{S}$, of the original Markov chain, i.e., *including* data call arrivals and generated by $\mathcal{Q}$, with an extra dimension. Denote with $N(t)$ the location in the queue of a tagged data call at time t, and let $N(t) = 0$ if the data call is no longer queued, i.e., it is either in transfer or already fully processed. The augmented Markov chain is denoted $(N(t), V(t), D(t))_{t \geqslant 0}$, with states (n, v, d).

For a given channel sharing policy, the state space of the augmented Markov chain is given by

$$\mathbb{S}^* \equiv \big\{(n, v, d) \in \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0 \colon\ n \leqslant d_q(v, d) \text{ and } (v, d) \in \mathbb{S}\big\}.$$

The state space is partitioned into an absorbing subset, $\mathbb{S}_0^* \equiv \{(n, v, d) \in \mathbb{S}^* \colon n = 0\}$, in which the tagged call is no longer queued, and its transient complement, $\mathbb{S}_+^* \equiv \mathbb{S}^* \backslash \mathbb{S}_0^*$. We further adjust the augmented Markov chain, by reducing all rates out of any state in $\mathbb{S}_0^*$ to zero, thereby enforcing that each such state becomes absorbent. The adjusted chain is a reducible continuous-time Markov chain with state space $\mathbb{S}^*$, that consists of $|\mathbb{S}_0^*| = |\mathbb{S}|$ absorbing states and 1 transient class $\mathbb{S}_+^*$.

Denote with $\mathcal{P}^*$ the one-step transition probability matrix of the embedded discrete-time Markov chain that follows the transitions of the augmented continuous-time Markov chain, with entries given by

$$\mathcal{P}^*\big((n, v, d); \big(n', v', d'\big)\big)$$
$$= \begin{cases} 1 & \text{if } n = 0 \text{ and } (n', v', d') = (n, v, d),\ \big(n', v', d'\big) \in \\ & \quad \big\{(n, v+1, d), (n, v, d+1), (n-1, v, d-1)\big\},\ \text{or} \\ \dfrac{\mathcal{Q}((v, d); (v', d'))}{-\mathcal{Q}((v, d); (v, d))} & \text{if } n > 0 \text{ and } \big(n', v', d'\big) = (n, v-1, d) \text{ and} \\ & \quad d_a(v-1, d) = d_a(v, d),\ \text{or } \big(n', v', d'\big) = \\ & \quad (n-1, v-1, d) \text{ and } d_a(v-1, d) = d_a(v, d) + 1, \\ 0 & \text{otherwise,} \end{cases}$$

for all (n, v, d), $(n', v', d') \in \mathbb{S}^*$. Clearly, all transitions that have nonzero probability of occurrence correspond to voice call arrivals, voice call terminations, data call arrivals, and data call terminations, respectively. Note that n changes at each data call termination event, as well as at the termination of a voice call whose released channels enable a queued data call to become active. $\mathcal{P}^*$ can be written in the form

$$\mathcal{P}^* \equiv \begin{pmatrix} \mathcal{I} & \mathcal{O} \\ \mathcal{P}^*_{+0} & \mathcal{P}^*_{++} \end{pmatrix},$$

where $\mathcal{I}$ is the identity matrix, $\mathcal{O}$ is the null-matrix, and $\mathcal{P}^*_{+0}$ and $\mathcal{P}^*_{++}$ are substochastic submatrices of $\mathcal{P}^*$ corresponding to the transitions from $\mathbb{S}^*_+$ to $\mathbb{S}^*_0$ and $\mathbb{S}^*_+$, respectively.

Theorem 3. The probability $\mathcal{J}((v, d); (v_o, d_o))$ that a call entering the system in state $(v, d) \in \mathbb{S}_+$ (d includes the new call) starts its transfer in state $(v_o, d_o) \in \mathbb{S}^{a+}_+$, is equal to the element $\mathcal{J}^*((d_q(v, d), v, d); (0, v_o, d_o))$ of the probability matrix $\mathcal{J}^*$ given by

$$\mathcal{J}^* = \big(\mathcal{I} - \mathcal{P}^*_{++}\big)^{-1}\mathcal{P}^*_{+0}.$$

The matrix $\mathcal{J}^*$ is of dimension $|\mathbb{S}^*_+| \times |\mathbb{S}^*_0|$ and contains the probabilities that the augmented chain, starting in any transient state in $\mathbb{S}^*_+$ is eventually absorbed in each of the recurrent states in $\mathbb{S}^*_0$.

Proof. The proof of theorem 3 is given in the appendix. □

3.2.3. Transfer time analysis

We now focus on the conditional expected transfer time $\widehat{\tau}_{v,d}(x)$ of a tagged active data call of length $x \geqslant 0$, starting its transfer in the presence of v voice and d data calls, $(v, d) \in \mathbb{S}^{a+}_+$, including itself and all queued data calls. Data call length x is expressed in units of r kbits so that it takes x seconds to transfer a file of length x on a single dedicated traffic channel. In the following an explicit expression for vector $\widehat{\boldsymbol{\tau}}(x) = (\widehat{\tau}_{v,d}(x),\ (v, d) \in \mathbb{S}^{a+}_+)$ is derived. The presented transfer time analysis is based on similar results reported by Núñez Queija [26] and Núñez Queija et al. [27].

For this, we need to make another modification to the original Markov chain, and hence introduce another generator, which is denoted $\mathcal{Q}^\bullet$. The modified chain is characterized by the presence of *one permanently active data call*, i.e., there is one active data

call that never leaves the system, but shares in the available traffic channels as if it were a regular active data call. The behavior of this permanent data call, i.e., the tagged call whose transfer time is to be determined, is identical to that of a regular data call, except for the fact that it cannot terminate within a given short time Δ which is considered in the proof of lemma 4 below. Generator $\mathcal{Q}^\bullet$ is similar to $\mathcal{Q}$, but of smaller dimensions, since the rows and columns corresponding to all states $(v, d) \notin \mathbb{S}_+^{a+}$ are crossed out. For all $(v, d) \in \mathbb{S}_+^{a+}$, the data call departure rates are modified as follows:

$$\mathcal{Q}^\bullet\big((v, d); (v, d-1)\big) = \beta(v, d)\,\big(d_a(v, d) - 1\big)\,\mu_{\text{data}},$$

and the diagonal elements of $\mathcal{Q}^\bullet$ are such that the entries of each row of $\mathcal{Q}^\bullet$ sum up to 0.

Furthermore, $\mathcal{B} \equiv \text{diag}(\beta(v, d),\ (v, d) \in \mathbb{S}_+^{a+})$ is the diagonal matrix of average data transfer rates, lexicographically ordered in (v, d). Note that, since $\beta(v, d) > 0$ for all $(v, d) \in \mathbb{S}_+^{a+}$, the diagonal matrix $\mathcal{B}$ is nonsingular and thus $\mathcal{B}^{-1}$ exists.

We may now formulate the following lemma.

Lemma 4. For $x \geqslant 0$, the vector of conditional expected transfer times $\widehat{\boldsymbol{\tau}}(x)$ satisfies the following differential equation and initial condition:

$$\frac{\partial}{\partial x}\widehat{\boldsymbol{\tau}}(x) = \mathcal{B}^{-1}\mathbf{1} + \mathcal{B}^{-1}\mathcal{Q}^\bullet\widehat{\tau}(x), \tag{5}$$

$$\widehat{\boldsymbol{\tau}}(0) = \mathbf{0}. \tag{6}$$

Proof. The proof of lemma 4 is given in the appendix. □

Theorem 5 below presents the explicit expression of the conditional expected transfer time.

Theorem 5. Let $\boldsymbol{\pi}^\bullet \equiv (\pi^\bullet_{v,d},\ (v, d) \in \mathbb{S}_+^{a+})$ be the stationary probability distribution vector corresponding to the Markov chain with one permanently active data call, i.e., $\boldsymbol{\pi}^\bullet\mathcal{Q}^\bullet = \mathbf{0}$. Further, let $\boldsymbol{\gamma} = (\gamma_{v,d},\ (v, d) \in \mathbb{S}_+^{a+})$ be the unique solution to

$$-\mathcal{B}^{-1}\mathcal{Q}^\bullet\boldsymbol{\gamma} = \mathcal{B}^{-1}\mathbf{1} - \frac{1}{\pi^\bullet\mathcal{B}\,\mathbf{1}}\mathbf{1}, \tag{7}$$

$$\boldsymbol{\pi}^\bullet\mathcal{B}\,\boldsymbol{\gamma} = 0. \tag{8}$$

Then the unique solution to the system of differential equations (5) is given by

$$\widehat{\boldsymbol{\tau}}(x) = \frac{x}{\boldsymbol{\pi}^\bullet\mathcal{B}\,\mathbf{1}}\mathbf{1} + \big[\mathcal{I} - \exp\{x\mathcal{B}^{-1}\mathcal{Q}^\bullet\}\big]\boldsymbol{\gamma}. \tag{9}$$

Proof. The proof of theorem 5 is given in the appendix. □

Remark 1. The constant $\boldsymbol{\pi}^\bullet \mathcal{B}\,\mathbf{1} = \sum_{(v,d)\in\mathbb{S}_+^{a+}} \beta(v,d)\,\pi^\bullet(v,d)$ can be interpreted as the expected number of channels that is assigned to the permanently active data call, in the modified Markov chain, generated by $\mathcal{Q}^\bullet$. An equivalent expression follows from

$$\begin{aligned}
\boldsymbol{\pi}^\bullet \mathcal{B}\,\mathbf{1} = & \sum_{(v,d)\in\mathbb{S}_+^{a+}} d_a(v,d)\,\beta(v,d)\,\pi^\bullet(v,d) \\
& - \mu_{\text{data}}^{-1} \sum_{(v,d)\in\mathbb{S}_+^{a+}} \big(d_a(v,d)-1\big)\,\beta(v,d)\,\mu_{\text{data}}\,\pi^\bullet(v,d) \\
= & \sum_{(v,d)\in\mathbb{S}_+^{a+}} d_a(v,d)\,\beta(v,d)\,\pi^\bullet(v,d) \\
& - \mu_{\text{data}}^{-1} \sum_{(v,d)\in\mathbb{S}_+^{a+}} \lambda_{\text{data}}\,\mathbb{1}\big\{(v,d+1)\in\mathbb{S}_+^{a+}\big\}\,\pi^\bullet(v,d) \\
\equiv & \mathbf{C}_{\text{data}}^\bullet - \rho_{\text{data}}(1-\mathbf{P}_{\text{data}}^\bullet),
\end{aligned}$$

where $\mathbf{C}_{\text{data}}^\bullet$ is the average number of channels used for data transfer, and $\mathbf{P}_{\text{data}}^\bullet$ is the blocking probability of a newly arriving data call, both in the modified Markov chain. In the derivation above, the second equality sign is due to the fact that in steady state the average number of data calls leaving the system per time unit must equal the average number of data calls entering the system per time unit. Note that since $\rho_{\text{data}}(1-\mathbf{P}_{\text{data}}^\bullet)$ is the average number of channels used by nonpermanent data calls, $\mathbf{C}_{\text{data}}^\bullet - \rho_{\text{data}}(1-\mathbf{P}_{\text{data}}^\bullet)$ is indeed equal to the expected number of channels assigned to the permanent data call.

Remark 2. Although the proof of theorem 5 may seem to indicate that instead of the constant $\boldsymbol{\pi}^\bullet \mathcal{B}\,\mathbf{1}$ basically any value could have been used, we stress that $\boldsymbol{\pi}^\bullet \mathcal{B}\mathbf{1}$ is indeed the only constant that allows (7) to be solved for $\boldsymbol{\gamma}$, as is easily demonstrated by premultiplication of (7) by $\boldsymbol{\pi}^\bullet \mathcal{B}$.

Corollary 6 below derives an asymptotic result, establishing an additional fairness property of the investigated class of channel sharing policies in that the transfer time of a data call is approximately linear in the data call size.

Corollary 6. The following asymptotic result immediately follows from (9):

$$\lim_{x\to\infty} \left\{ \widehat{\boldsymbol{\tau}}(x) - \frac{x}{\boldsymbol{\pi}^\bullet \mathcal{B}\,\mathbf{1}} \mathbf{1} \right\} = \boldsymbol{\gamma}.$$

Proof. The proof of corollary 6 is given in the appendix. □

Remark 3. The asymptotic result presented in corollary 6 is readily supported by the following intuitive argument. Consider a file of size x, expressed as the transfer time in seconds given the exclusive use of one traffic channel. As $x \to \infty$ the average number of assigned channels over the file's lifetime becomes more and more independent of the

system state at the file's arrival and the precise evolution trace of all other (voice or data) calls. In fact, in the limit the average number of assigned channels is precisely equal to $\boldsymbol{\pi}^{\bullet}\mathcal{B}\,\mathbf{1}$, and hence the expected transfer time is equal to the deterministic file size divided by the average number of assigned traffic channels. We note hereby that the significance of the constant $\boldsymbol{\gamma}$ becomes negligible.

4. Channel sharing policies

In this section we propose four distinct channel sharing policies for voice and data calls that fit within the considered class. The first policy, called *FullSeg*, is a *static* policy, in that voice and data calls are served with two completely separate channel pools. The other policies, *FixCap, FullShare* and *FullShareRes* allow *dynamic* sharing of the traffic channels, to varying extents. For each policy the basic functions $v_{\max}(v, d)$, $d_{\max}(v)$ and $\beta(v, d)$ are specified as well as $v_{\max}$ and $d_{\max}$, while we recall that $d_a(v, d)$ and $d_q(v, d)$ are implicitly defined by these basic functions.

4.1. FullSeg policy

Under the *FullSeg* policy, the voice and data services are completely segregated, in that the cell capacity of C_{total} traffic channels is split into two disjoint pools with $C_{\text{voice}} \geqslant 1$ and $C_{\text{data}} \equiv C_{\text{total}} - C_{\text{voice}} \geqslant b$ channels for voice and data calls, respectively. Since there is no interaction between the two service types, the performance analysis can be done separately. Still, in order to demonstrate that the *FullSeg* policy falls within the studied class of channel sharing policies, the three characteristic functions will be specified.

For *voice* calls, the resulting model is simply an $M/M/C_{\text{voice}}/C_{\text{voice}}$ Erlang loss model with voice traffic load ρ_{voice}. Note that

$$v_{\max} \equiv v_{\max}(0, 0) = v_{\max}(v, d) = C_{\text{voice}}.$$

The voice call blocking probability is given by the well-known Erlang loss formula, e.g., [Tijms, 36]. *Data* calls request a transfer capacity of B channels, but will settle for any capacity between b and B. During a data call, the channel assignment is dynamically adapted to either utilize freed capacity or to support newly admitted data calls. Hence the average number of channels assigned to an active data call in system state (v, d), is given by

$$\beta(v, d) \equiv \min\left\{B, \frac{C_{\text{data}}}{d_a(v, d)}\right\} \quad \text{if } d_a(v, d) > 0.$$

The maximum number of data calls in the segregated system is given by

$$d_{\max} \equiv d_{\max}(0) = d_{\max}(v) = \left\lfloor \frac{C_{\text{data}}}{b} \right\rfloor + Q_{\text{data}},$$

which functions as the admission control threshold. Note that all these functions do not depend on v, as expected.

In this fully segregated model, the voice and data services can be evaluated separately, but all performance measures obtained are identical to those that would be found if the segregated models were evaluated simultaneously as one model.

4.2. *FixCap policy*

Under the *FixCap* policy, data calls request a fixed capacity of $\beta_{FixCap} \in \{b, \ldots, B\}$ traffic channels. Voice service is protected from the potentially demanding data calls, by reserving C_{voice} channels for voice calls only. The remaining $C_{\text{total}} - C_{\text{voice}}$ channels are shared between voice and data calls, without any priorities or service preemption.

Voice calls are admitted if at least 1 channel is available, i.e., if

$$v < v_{\max}(v, d) \equiv C_{\text{total}} - \beta_{FixCap}\, d_a(v, d).$$

The maximum number of voice calls in the system is given by $v_{\max} \equiv v_{\max}(0, 0) = C_{\text{total}}$. A *data* call is admitted if it can either start service immediately, i.e., β_{FixCap} free channels can be found among the $C_{\text{total}} - C_{\text{voice}}$ shared channels, or if the queue is not full. Mathematically, this condition for data call admission is formulated as follows: a data call is admitted if

$$d < d_{\max}(v) \equiv \left\lfloor \frac{C_{\text{total}} - \max\{C_{\text{voice}}, v\}}{\beta_{FixCap}} \right\rfloor + Q_{\text{data}}.$$

Once activated, data calls hold on to the assigned $\beta(v, d) \equiv \beta_{FixCap}$ channels until call termination, continuously transmitting at a fixed bit rate. The maximum number of data calls in the system is given by $d_{\max} \equiv d_{\max}(0)$. Note that since

$$\beta(v, d) \equiv \beta_{FixCap}$$

indicates a fixed transfer rate in each state $(v, d) \in \mathbb{S}_+^{a+}$, $\boldsymbol{\gamma} = \mathbf{0}$ immediately follows from (7) and (8), so that (9) yields $\widehat{\boldsymbol{\tau}}(x) = \beta_{FixCap}^{-1}\, x\, \mathbf{1}$, as expected.

The *FixCap* channel sharing policy is different from the three other proposed policies in the sense that the data calls are not elastic, i.e., the policy does not allow the data calls to dynamically capture or release traffic channels, in order to enhance service quality and channel utilization, or support newly arriving (voice or data) calls. The policy is included in the performance comparison for reference purposes.

4.3. *FullShare policy*

Under the *FullShare* policy, data calls request a transfer capacity of B channels, but will settle for any capacity between b and B. Data calls maximally utilize all available channels, with channel assignments that are dynamically adapted to either utilize freed capacity or to support newly admitted (voice or data) calls. The cell capacity C_{total} is fully shared between voice and data calls, whereby data calls are always forced to give up excess capacity, i.e., capacity above b, when needed. An important distinction between the *FullShare* policy and the other proposed policies, is that here no parameters need to be set by the network operator.

Voice calls are admitted if at least 1 channel is, or can be made, available, i.e., if

$$v < v_{\max}(v, d) \equiv C_{\text{total}} - b\, d_a(v, d).$$

Note that under this policy, if $b = 1$, voice calls can be admitted only if no data calls are queued, while for $b > 1$, it is possible for a voice call to be admitted, even if one or more data calls are queued. The maximum number of voice calls in the system is given by $v_{\max} \equiv v_{\max}(0, 0) = C_{\text{total}}$. A *data* call is admitted if at least b channels are, or can be made available, or if the queue is not full. This condition can be mathematically formulated as follows:

$$d < d_{\max}(v) \equiv \left\lfloor \frac{C_{\text{total}} - v}{b} \right\rfloor + Q_{\text{data}}.$$

Once activated, each active data call will receive $\min\{B, \lfloor (C_{\text{total}} - v)/d_a(v, d) \rfloor\} \geqslant b$ channels, while the remaining channels are randomly distributed, not exceeding technical constraint B. Thus on average each active data call is given

$$\beta(v, d) \equiv \min\bigl\{B, (C_{\text{total}} - v)/d_a(v, d)\bigr\}$$

channels if $d_a(v, d) > 0$. The maximum number of data calls in the system is given by $d_{\max} \equiv d_{\max}(0)$.

Note that as the system becomes overloaded with data traffic, i.e., $\lambda_{\text{data}} \to \infty$, the voice call blocking probability approaches 100% due to the fact that freed channels will always be claimed immediately by a queued data call (provided that $Q_{\text{data}} > 0$). This suggests the need to protect the voice service, which is precisely the aim of the *FullShareRes* policy.

4.4. FullShareRes policy

The *FullShareRes* policy is very similar to the *FullShare* policy, except that now voice calls are strictly prioritized, with service preemption, over data calls. In order to prevent voice calls from crowding out data calls, C_{data} channels are reserved for data service only, while the remaining $C_{\text{total}} - C_{\text{data}}$ channels are shared.

Voice calls are admitted if at least 1 channel is, or can be made, available, i.e., if

$$v < v_{\max}(v, d) \equiv C_{\text{total}} - C_{\text{data}}.$$

The maximum number of voice calls in the system is given by $v_{\max} \equiv v_{\max}(0, 0) = C_{\text{total}} - C_{\text{data}}$. A *data* call is admitted if a minimum capacity of b channels can be guaranteed to it, or if the queue is not full. Mathematically formulated, the condition is as follows:

$$d < d_{\max}(v) \equiv \lfloor C_{\text{data}}/b \rfloor + Q_{\text{data}}.$$

Once activated, data calls share the available channels fairly, precisely as described for the *FullShare* policy. Hence

$$\beta(v,d) \equiv \min\bigl\{B, (C_{\text{total}} - v)/d_a(v,d)\bigr\},$$

with $\beta(v,d) \geqslant b$ if $d_a(v,d) > 0$. Note that no more than $\lfloor C_{\text{data}}/b \rfloor$ data calls can be active, even if there are no voice calls in the system. The reason for this is that no more data calls can be *guaranteed* a minimum assignment of b channels, if a large number of voice calls were to arrive and claim all $C_{\text{total}} - C_{\text{data}}$ shared channels (recall that an active data call cannot be pushed back into the queue). As a consequence, the maximum number of data calls in the system is given by $d_{\max} \equiv d_{\max}(0) = \lfloor C_{\text{data}}/b \rfloor + Q_{\text{data}}$.

4.5. *Overview of channel sharing policies*

We conclude this section with an overview of the four presented channel sharing policies. Figure 4 graphically summarizes how the cell capacity C_{total} can be assigned to the different services, while table 1 lists all functions required for the performance analysis. The functions $\beta(v,d)$ in this table are defined only if $d_a(v,d) > 0$.

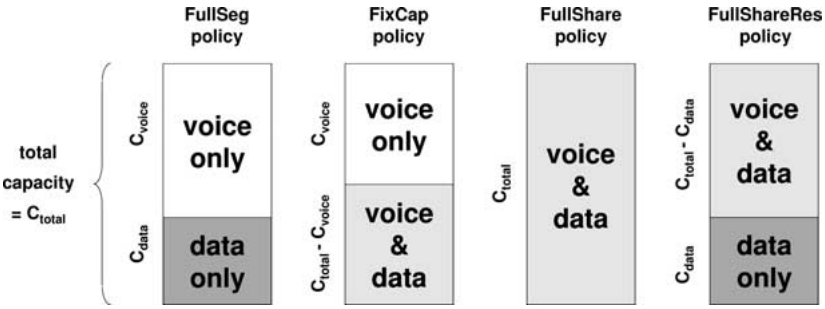

Figure 4. Overview of channel sharing policies.

Table 1
Overview of channel sharing policies.

	$\beta(v,d)$	$v_{\max}(v,d)$	$d_{\max}(v) - Q_{\text{data}}$
FullSeg	$\min\left\{B, \frac{C_{\text{data}}}{d_a(v,d)}\right\}$	C_{voice}	$\left\lfloor \frac{C_{\text{data}}}{b} \right\rfloor$
FixCap	β_{FixCap}	$C_{\text{total}} - \beta_{FixCap}\, d_a(v,d)$	$\left\lfloor \frac{C_{\text{total}} - \max\{C_{\text{voice}}, v\}}{\beta_{FixCap}} \right\rfloor$
FullShare	$\min\left\{B, \frac{C_{\text{total}} - v}{d_a(v,d)}\right\}$	$C_{\text{total}} - b\, d_a(v,d)$	$\left\lfloor \frac{C_{\text{total}} - v}{b} \right\rfloor$
FullShareRes	$\min\left\{B, \frac{C_{\text{total}} - v}{d_a(v,d)}\right\}$	$C_{\text{total}} - C_{\text{data}}$	$\left\lfloor \frac{C_{\text{data}}}{b} \right\rfloor$

5. Numerical results

This section presents an extensive numerical study. As it would take up too much space to study the effect of all model parameters, some parameters are prefixed at a realistic level while the remaining parameters are varied within a realistic range around their default value and their impact on the relevant performance measures is investigated. Table 2 below gives an overview of all model parameters and indicates either their prefixed value or their default values and the range of considered values around this default value.

Regarding the prefixed parameters, r is based on the latest channel coding scheme for a full rate data traffic channel, b and B correspond to the expected multichannel capabilities of an HSCSD terminal, and μ_{data} and μ_{voice} are set to correspond with an average e-mail size (320 kbits: $\mu_{\text{data}}^{-1} = 320/14.4$ s) and an average voice call holding time ($\mu_{\text{voice}}^{-1} = 50$ s). Note that parameters C_{data} and C_{voice} are not required for all channel sharing policies (see table 1). The voice call arrival rate λ_{voice} is chosen such that for a cell with 3 frequencies ($C_{\text{total}} = 21$) the voice call blocking probability is 1% provided that all C_{total} channels are available for voice transfer (for 3 frequencies: $\rho_{\text{voice}} = 12.837$ Erlang). For those cases with fewer or more frequencies, the voice traffic load is linearly adjusted as indicated in table 2. Finally, the data traffic load ρ_{data} is varied between 0 Erlang and ρ_{voice}. Since μ_{data} is fixed, λ_{data} is adjusted to obtain the desired data traffic load.

In the remainder of this section, a number of numerical experiments is executed in order to obtain insight in the effect of the variable parameters on the performance measures. First, we present a comparison of the proposed channel sharing policies.

5.1. Comparison of channel sharing policies

The proposed channel sharing policies are compared with default settings for all model parameters except for the data traffic load ρ_{data} which is varied between 0 Erlang and $\rho_{\text{voice}} = 12.837$ Erlang. As figure 5 (left) shows, under low data traffic loads the channel utilization is optimal under the *FixCap* and *FullShare* policies, since they do not reserve any capacity strictly for data transfers. As the data traffic load grows, however, only the most work-conserving policies (*FullShare* and *FullShareRes*) are able to establish

Table 2
Numerical results: parameter settings.

Parameter	*Prefixed value*	*Parameter*	*Default value*	*Range*
r	14.4 kbps	C_{total}	21 channels	$\{7, 14, 21, 28\}$
b	1 channel	C_{data}	6 channels	$\{2, 4, 6, 8\}$
B	4 channels	C_{voice}	$C_{\text{total}} - C_{\text{data}}$ channels	–
β_{FixCap}	2 channels	Q_{data}	5 calls	$\{0, \ldots, 20\}$
		ρ_{data}	$0.5 \cdot \rho_{\text{voice}}$ Erlang	$(0, 1] \cdot \rho_{\text{voice}}$
μ_{data}	0.0450 calls/s	λ_{data}	$\mu_{\text{data}} \cdot \rho_{\text{data}}$ calls/s	–
		ρ_{voice}	$0.6113 \cdot C_{\text{total}}$ Erlang	–
μ_{voice}	0.0200 calls/s	λ_{voice}	$\mu_{\text{voice}} \cdot \rho_{\text{voice}}$ calls/s	–

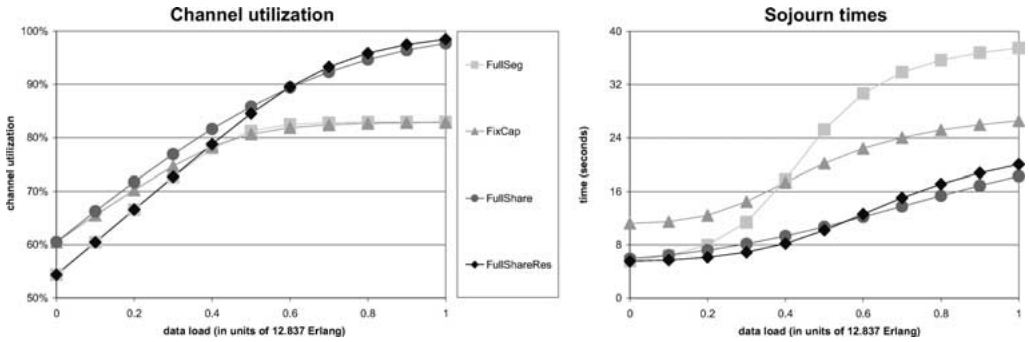

Figure 5. Comparison of channel sharing policies: expected channel utilization versus data traffic load (left) and expected sojourn times versus data traffic load (right).

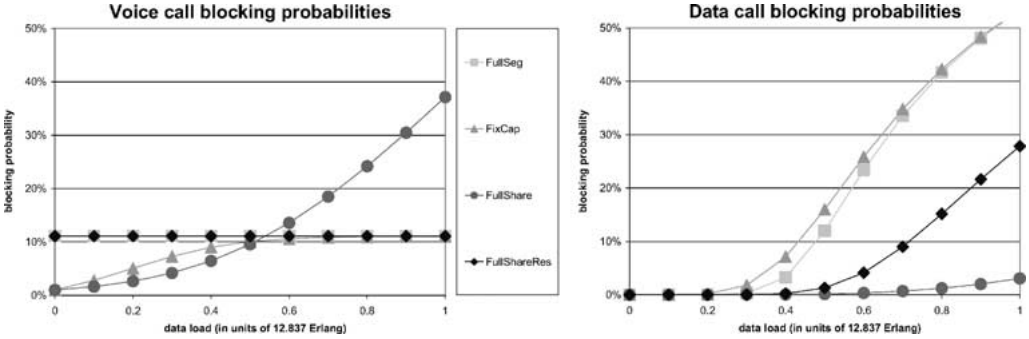

Figure 6. Comparison of channel sharing policies: voice (left) and data (right) call blocking probability versus data traffic load.

a significant channel utilization, since under these policies data calls can occupy up to four otherwise idle (reserved for voice calls) traffic channels.

Under the same parameter settings, figure 6 presents the voice (left) and data (right) call blocking probability as a function of the data traffic load for all four policies. This figure reveals the primary disadvantage of the *FullShare* policy in the sense that it cannot protect the voice service from being crowded out by the data traffic. A low data call blocking probability along with a rapidly increasing voice call blocking probability clearly indicates this. In contrast, under low data traffic loads the *FullShare* policy is optimal. Note from the channel utilization and blocking probabilities that the performance of the *FixCap* policy converges to that of the *FullSeg* policy for increasing data traffic loads, because the data calls will fully occupy all channels available for data transfer. Since the *FullSeg* policy assigns only a single channel to each data call under high data traffic loads, the number of active data calls is twice as high while the transfer is twice as slow compared to the fixed assignment of two channels under the *FixCap* policy. The convergence of the channel utilization and blocking probabilities proves that the two opposite effects cancel out.

Regarding the corresponding expected sojourn times, observe from figure 5 (right) that for low data traffic loads, the *FixCap* policy is strictly outperformed by the other

Table 3
Comparison of channel sharing policies.

	$\mathbf{U}$		$\mathbf{P}_{\text{voice}}$		$\mathbf{P}_{\text{data}}$		$\mathbf{T}_{\text{data}}^{\text{sojourn}}$	
ρ_{data}	≈ 0	$\gg 0$	≈ 0	$\gg 0$	≈ 0	$\gg 0$	≈ 0	$\gg 0$
FullSeg	o	−	−	+	+	−	+	−
FixCap	+	−	+	+	+	−	−	o
FullShare	+	+	+	−	+	+	+	+
FullShareRes	o	+	−	+	+	o	+	+

policies that allow assignments of more than β_{FixCap} traffic channels. As ρ_{data} grows, the expected sojourn times grow under each policy, including the *FixCap* policy due to an increasing waiting time component. The *FullSeg* policy, not allowing data calls to utilize idle voice channels, suffers from this restriction most notably under high data traffic loads, as indicated by the long expected sojourn times. The *FullShare* and *FullShareRes* policies establish very similar quality of service curves.

Table 3 provides an overview of the performance of the investigated channel sharing policies regarding the principal performance measures. Separately for low ($\rho_{\text{data}} \approx 0$) and high ($\rho_{\text{data}} \gg 0$) data traffic loads, a policy scores a '−', 'o' or '+' reflecting the relative performance with respect to the other policies. Evidently, none of the policies strictly outperforms the alternatives with respect to all performance measures, which prohibits a trivial policy selection. We argue that both the work-conserving *FullShare* and *FullShareRes* policies are prefered over the *FullSeg* and the *FixCap* policies, primarily because they allow statistical multiplexing of voice and data traffic and thus generally establish a high channel utilization and, correspondingly, low sojourn times and blocking probabilities. In the initial phase of a light data traffic load, it is recommended to implement the *FullShare* policy, as any channel reservation would only raise the voice call blocking probability. However, when the data traffic load grows from light to moderate or heavy, it appears best to deploy the *FullShareRes* policy, as a mobile network operator is likely to be very hesitant about affecting its voice client base when operating in the data market. The rationale for this is that the policy best utilizes the elasticity and relative delay tolerance of the data calls, while protecting the voice users by posing an acceptable upper bound on the voice call blocking probability, which is independent from the data traffic load. Since it is most robust against a data traffic load increase, we select the *FullShareRes* policy for further study in the remainder of our numerical investigation. In practice, we suggest that a desired trade-off between the grade and quality of service measures is established by making the reservation level adaptive to the traffic load.

5.2. *Performance effects of* C_{data}

This section focuses on the trade-off between the different performance measures as we reserve fewer or more traffic channels for data transfer. The data traffic load ρ_{data} is varied between 0 Erlang and $\rho_{\text{voice}} = 12.837$ Erlang, C_{data} is taken from $\{2, 4, 6\}$

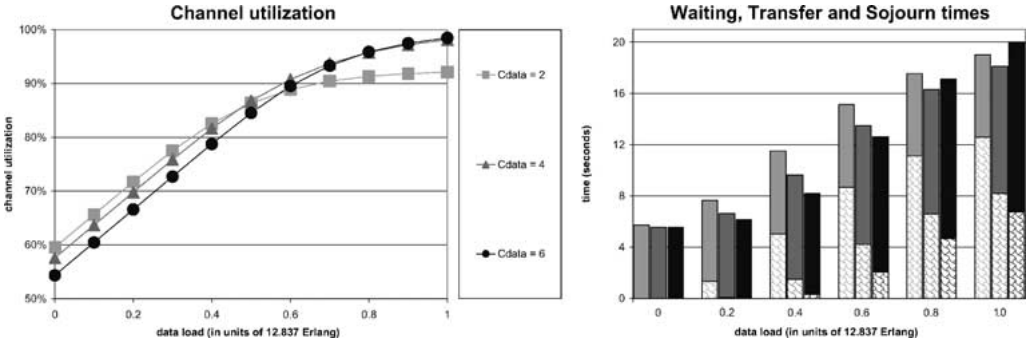

Figure 7. Performance effects of C_{data}: channel utilization (left), expected waiting, transfer and sojourn times (right) versus data traffic load.

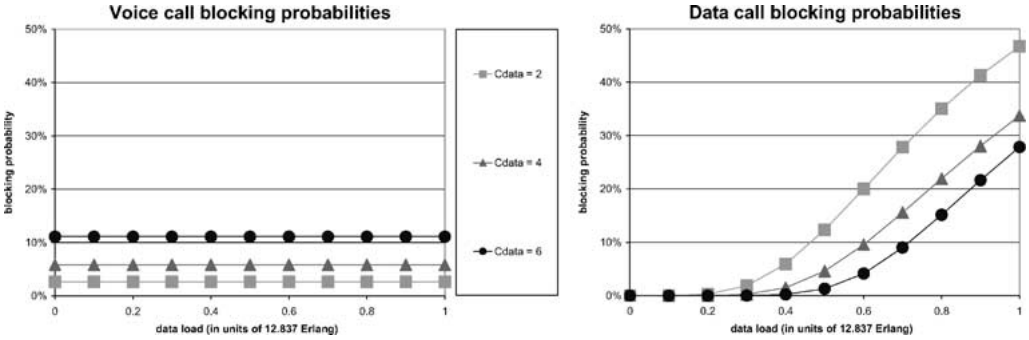

Figure 8. Performance effects of C_{data}: voice (left) and data (right) call blocking probability versus data traffic load.

while all other model parameters are set to their default values. As figure 7 (left) shows, C_{data} must be adapted to the data traffic load if an operator wishes to maximize channel utilization, in accordance with the upper envelope of the utilization curves. As the data traffic load increases, C_{data} should be regularly incremented in order to keep the channel utilization maximal. The trade-off is that aiming for optimal channel utilization may imply unacceptable voice call blocking probabilities under high data traffic loads (see figure 8 (left)).

Figure 7 (right) illustrates the effect that C_{data} has on the waiting, transfer and sojourn times. The height of each vertical bar reflects the expected sojourn time, consisting of a waiting time (bottom segment) and a transfer time (top segment) component. For very light data traffic loads, in particular for $\rho_{\text{data}} \downarrow 0$, delay values are plotted at the '$\rho_{\text{data}} = 0$' mark on the horizontal axis. Aside from the unsurprising results that the expected waiting, transfer and hence also the sojourn times increase with ρ_{data}, while the waiting time becomes more dominant with an increase in ρ_{data}, the figure also illustrates that an increase in C_{data} does *not* necessarily imply an enhancement of the delivered QOS for admitted data calls. The reason for this effect is that the data call blocking probability decreases with an increase in C_{data} (see figure 8 (right)), so that the *admitted* data calls have more competition for radio resources.

Suppose the maximum voice and data call blocking probabiblities an operator allows in its network are 5% and 10%. Then for the given setting, the presented performance results enable us to conclude that the optimal number of dedicated data traffic channels is $C_{\text{data}} = 2$ for $\rho_{\text{data}} \leqslant 0.45 \cdot 12.837$ Erlang and $C_{\text{data}} = 4$ for $0.45 \times 12.837 < \rho_{\text{data}} \leqslant 0.6 \cdot 12.837$ Erlang, while for greater data traffic loads, there is insufficient cell capacity to meet the blocking requirements. Note that under the proposed channel reservation strategy, the channel utilization is maximal, while the expected data call sojourn times are still below 15 s. Viewing the problem from a slightly different angle, one can determine the maximum value of C_{data} such that the voice call blocking probability remains below a prespecified value, e.g., 5%, and subsequently determine the data call blocking probability and the expected sojourn times as demonstrated. If these performance measures for the data services are not satisfactory, the operator must increase the cell capacity, e.g., by assigning an additional frequency to it.

5.3. *Performance effects of* Q_{data}

We now investigate the performance effects of increasing the size of the data call request queue (Q_{data}) for the *FullShareRes* policy. Default settings are used for all model parameters except for the data traffic load ρ_{data} which is varied between 0 Erlang and $\rho_{\text{voice}} = 12.837$ Erlang and Q_{data} which is taken from $\{0, 5, 10\}$. First, note that under the considered policy the voice call blocking probability is obviously independent of ρ_{data} and Q_{data}, as illustrated by figure 10 (left). In contrast, the data call blocking probability (see figure 10 (right)) increases with ρ_{data} and decreases with Q_{data}. For the given range of ρ_{data} the carried data traffic load $\rho_{\text{data}}(1 - \mathbf{P}_{\text{data}})$ still increases with the offered data traffic load ρ_{data}, which explains the channel utilization, increasing to 100% (see figure 9 (left)) as the data call blocking probability approaches 1 to stabilize the carried data load, independently of any further increase in ρ_{data}. Lastly, the lower data call blocking probability induced by a larger data call request queue marginally improves the channel utilization.

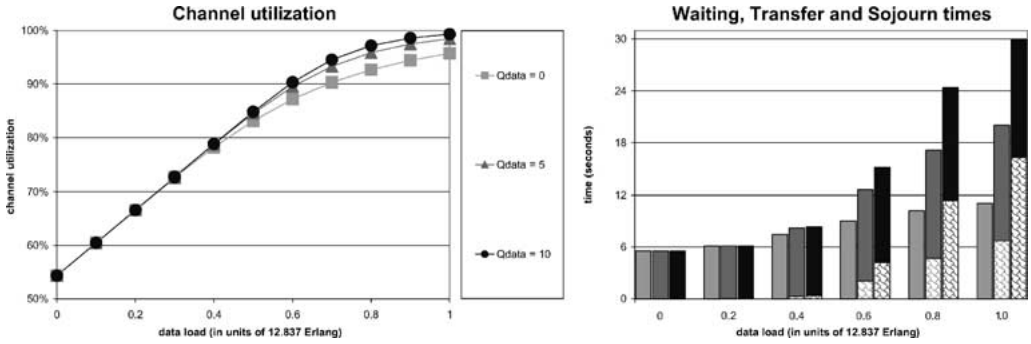

Figure 9. Performance effects of Q_{data}: channel utilization (left), expected waiting, transfer and sojourn times (right) versus data traffic load.

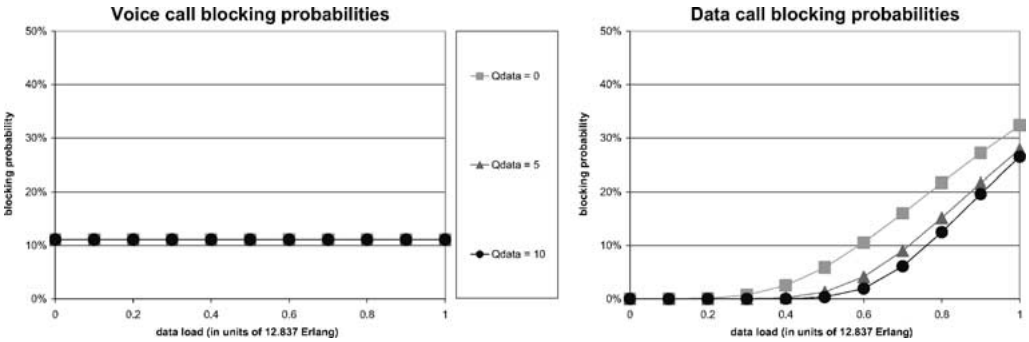

Figure 10. Performance effects of Q_{data}: voice (left) and data (right) call blocking probability versus data traffic load.

For $\rho_{\text{data}} \in \{0.0, 0.2, 0.4, 0.8, 1.0\} \cdot \rho_{\text{voice}}$ figure 9 (right) presents the expected sojourn times and the corresponding split-up in waiting (bottom segment) and transfer (top segment) times. Obviously, there is no waiting time if there is no queue ($Q_{\text{data}} = 0$). Once again, the delay values plotted at the '$\rho_{\text{data}} = 0$' mark on the horizontal axis must be interpreted as the limit values as $\rho_{\text{data}} \downarrow 0$. Under low data traffic loads, the sojourn time is dominated by the transfer time which is bounded from below by 5.56 s, corresponding with a continuous assignment of B traffic channels to each data call. As ρ_{data} increases, both the expected waiting and the transfer times go up, until the data traffic load becomes so high that each data call is served with no more than $b = 1$ dedicated traffic channel *plus* a fair share of those shared channels that do not carry voice calls. For data traffic loads beyond this value, the expected transfer time remains constant at $C_{\text{data}} \cdot (320/(14.4(C_{\text{total}} - \mathbf{N}_{\text{voice}}))) \approx 13.91$ s with $\mathbf{N}_{\text{voice}} \approx 11.41$, $C_{\text{total}} - \mathbf{N}_{\text{voice}}$ the expected number of channels available for data transfer, and C_{data} the number of active data calls (recall that $b = 1$). The expected waiting times continue to increase, converging to $Q_{\text{data}} \cdot (320/(14.4(C_{\text{total}} - \mathbf{N}_{\text{voice}})))$, since in a cell overloaded with data calls, an admitted data call always takes the last position in the queue and hence must wait until Q_{data} data calls finish their transfer. For $Q_{\text{data}} = 5$ and 10, the corresponding upper bounds on the expected waiting times are 11.59 and 23.18 s. Recalling that the expected transfer time was calculated to converge to 13.91 s, this illustrates that it depends on the queue size whether the expected sojourn time will ever be dominated by the expected waiting time. For $Q_{\text{data}} = 10$ the waiting time is already dominant at $\rho_{\text{data}} = 12.837$ (see figure 9 (right)), while for $Q_{\text{data}} = 5$ the waiting time will never dominate.

5.4. Waiting times and optimizing Q_{data}

In the previous section it was illustrated that the expected waiting time was increasing in both the data traffic load and the queue size. In this section we focus on the effect of the queue size on the expected waiting time and the data call blocking probability, and indicate how a network operator can choose the optimal value of its queue size. A cell capacity C_{total} of 7, 14, 21 or 28 traffic channels is considered with $C_{\text{data}} = 2, 4, 6$ or 8, respectively. The queue size is varied from 0 to 20. All other model parameters are set

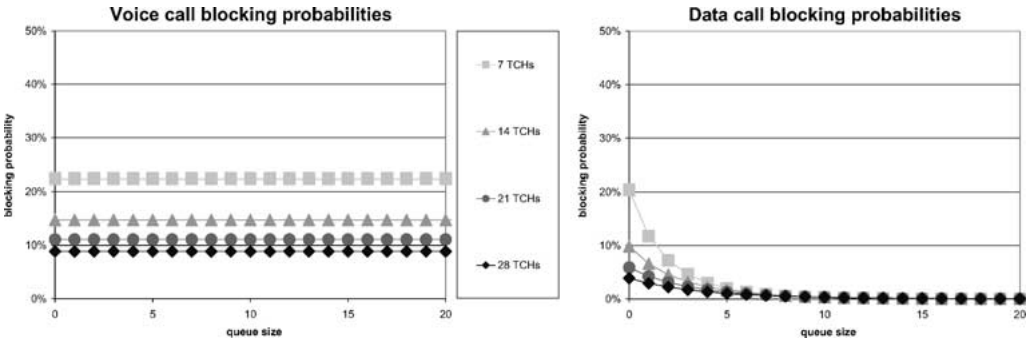

Figure 11. Waiting times and optimizing Q_{data}: voice (left) and data (right) call blocking probability versus data traffic load.

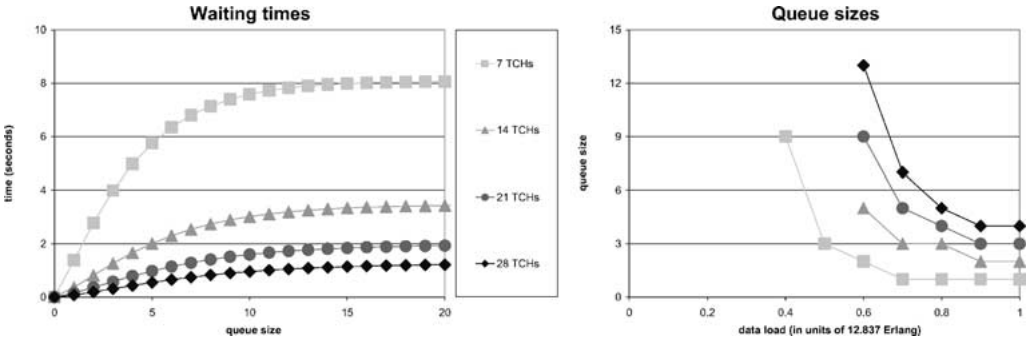

Figure 12. Waiting times and optimizing Q_{data}: expected waiting times versus queue size (left) and maximum allowable queue sizes versus data traffic load (right).

to the default values. Recall that this implies proportionality of the voice and data traffic loads with respect to the cell capacity (see table 2).

Figure 11 shows the voice (left) and data (right) call blocking probabilities. Naturally, the voice call blocking probability is unaffected by the variation in queue size, while its dependency on the cell capacity is readily determined using the Erlang loss formula. The data call blocking probability decreases with the queue size, while the expected waiting time (see figure 12 (left)) converges to a constant as the queue becomes so large that its last positions are virtually never taken (0% data call blocking probability). Regarding the effect of the cell capacity, the presented numerical results support the well-known result that the benefits of statistical multiplexing become greater as the capacity increases. This can be seen from the fact that although the offered load is assumed proportional to the cell capacity, cells with higher capacity are strictly better off with respect to both performance measures displayed.

Recall from section 2 that the mere purpose of implementing a queue for data call requests is to postpone call blocking momentarily in the hope that resources are freed to serve the call. The amount of additional set-up delay that can be allowed is limited. Results as presented in figure 12 (left) can be used to derive the maximum queue size

that can be implemented such that the expected additional call set-up time (waiting time) is less than an operator-specified service requirement of, say, $\omega_{\max}$ seconds. Although in practice a 90% percentile of the waiting time would be a more appropriate measure to determine the optimal queue size, the expected waiting time given by our model provides a useful and analytically obtainable first-order indication. As illustrated in figure 12 (right) for $\omega_{\max} = 4$ s, the maximal queue size strongly depends on the data traffic load. Under very low data traffic loads, the expected waiting time may be sufficiently low even with an infinite queue size, which is the case in figure 12 (left) for the cells with 14, 21 or 28 traffic channels, where the waiting time converges to a maximum below $\omega_{\max}$. As the data traffic load becomes heavier, the queue must be shortened in order to meet the $\omega_{\max}$ requirement, which causes an additional indirect increase in the data call blocking probability, aside from the direct and obvious effect of the heavier load. The observation that a larger queue size is allowed in cells with higher capacity can be explained with the statistical multiplexing effect described above. Note that in figure 12 (right) the curve for $C_{\text{total}} = 14$ does not decrease for $\rho_{\text{data}} = 0.8 \cdot 12.837$ Erlang due to the discretization effect: for each Q_{data} there is a *range* of data traffic load values for which this queue size is optimal.

5.5. *Conditional expected sojourn times*

In corollary 6 it was stated that the conditional expected transfer times $\widehat{\boldsymbol{\tau}}(x)$ are asymptotically linear in the data call size x. Since both the expected waiting time $\mathbf{T}_{\text{data}}^{\text{queue}}$ and the transition probability distribution $\mathcal{J}$ are independent of the call size, the conditional expected *sojourn* time $\mathbf{T}_{\text{data}}^{\text{sojourn}}(x)$ must be asymptotically linear in x as well (see (2) and (3)). For default values of all parameters but ρ_{data}, which is taken from $\{0.1, 0.5, 1.0\} \cdot \rho_{\text{voice}}$, figure 13 demonstrates the convergence of the exact expected sojourn times to the derived asymptotes. For each value of ρ_{data} in the left figure, the asymptote is the dashed line with the small open markers that almost coincides with the exact curve for $\rho_{\text{data}} \in \{0.1, 1.0\} \cdot \rho_{\text{voice}}$, while it more visibly deviates from the exact curve for $\rho_{\text{data}} = 0.5 \cdot \rho_{\text{voice}}$. The range of x values considered is from 0 to $100\,r$ kbits,

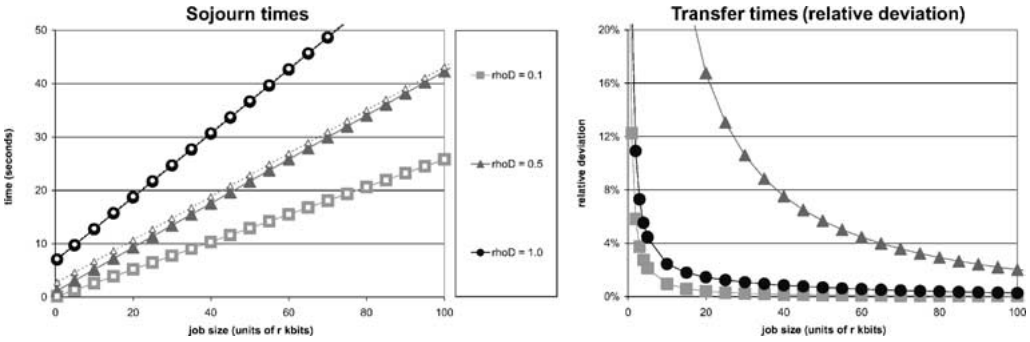

Figure 13. Conditional expected sojourn times: convergence of conditional expected sojourn times.

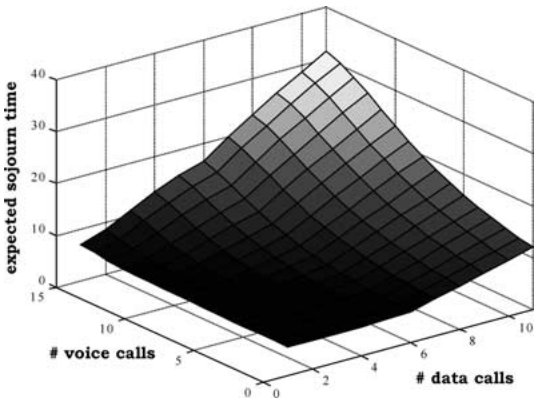

Figure 14. Conditional expected sojourn times: conditional expected sojourn time versus system state at arrival.

the latter value corresponding to the 99% percentile of the data call size distribution. Although in this example the exact expected sojourn time values converge to the asymptote from below, this observation does not hold in general, as we learned from other experiments with different parameter settings.

Furthermore, the right figure indicates that the speed of convergence, which is expressed as the relative deviation between the exact and the approximate expected *transfer* time, appears to be lowest for moderate data loads, which we intuitively expect to hold in general. The reason for using the *transfer* time rather than the *sojourn* time to determine the speed of convergence is that we do not want the different values of the expected waiting time to distort the comparison. In general, the speed of convergence is predominantly determined by the second-largest eigenvalue of the generator $\mathcal{B}^{-1}\mathcal{Q}^{\bullet}$, as can be seen from the proof of corollary 6. It is extremely difficult to obtain analytical insight in the relation between the eigenvalues of $\mathcal{B}^{-1}\mathcal{Q}^{\bullet}$ and the model parameters.

An intermediate result in the determination of the conditional expected sojourn times $\mathbf{T}_{\text{data}}^{\text{sojourn}}(x)$ is given by the $\widehat{\sigma}_{v,d}(x)$, the conditional expected sojourn time of an admitted data call of size x arriving in system state $(v, d) \in \mathbb{S}_+$ (recall that d includes the new call). This result may be very useful as a feedback information service to the caller. Figure 14 presents an illustrative example of $\widehat{\sigma}_{v,d}(x)$ versus $(v, d) \in \mathbb{S}_+$ for an admitted data call of size 320 kbits, given the default parameter settings. The figure supports the intuition that $\widehat{\sigma}_{v,d}(x)$ is increasing in both v and d, i.e., that the expected conditional sojourn time is longer as the data call finds the cell to be more congested upon its arrival. The numerical example further illustrates that the expected sojourn time of a data call is most sensitive to a change in the number of *data* calls, since an additional active data call claims at least as many traffic channels as a voice call. Note the abrupt change in slope at $d = d_{\max}(v) - Q_{\text{data}} = 6$, indicating an increased sensitivity of the expected sojourn time of a queued data call with respect to the number of data calls queued ahead of it.

6. Concluding remarks and discussion

We have presented a extensive analytical performance evaluation of a class of fair channel sharing policies in an integrated GSM/HSCSD network with a finite queue for data call requests that cannot be served immediately upon arrival. Markov chain analysis has been applied to obtain simple performance measures such as channel utilization, voice and data call blocking probabilities and the average data call waiting, transfer and sojourn times. Furthermore, using differential equations, a closed-form expression has been derived for the expected sojourn time of a data call, conditional on its size, indicating that the sojourn time is asymptotically proportional to the call size and hence the proposed policies provide fairness with respect to various data call sizes. As a valuable intermediate result in the analysis, the conditional expected sojourn time of an admitted data call is obtained, given the system state at arrival, which may serve as an appreciated feedback information service to the data source.

Four typical channel sharing policies within the given class have been specified for numerical evaluation. Among these the *FullShareRes* policy has been argued to be most promising. Under this policy, an operator-specified number of traffic channels is reserved for data transfer only, while the remaining channels are shared by voice and data calls. On the shared channels voice calls are strictly prioritized, with service preemption over data calls. At any time, the active data calls fairly share all available channels up to the terminals' multichannel capabilities, with channel assignments that are dynamically adapted to either utilize freed capacity or to support newly admitted data calls.

Expecting that a mobile network operator is likely to be rather hesitant to degrade its voice service when entering the data market, the *FullShareRes* policy performs best when the data traffic load grows from light to moderate or heavy, given its work-conserving property, the implied high channel utilization, and the protection it offers to the voice users, independent of an increase in the data traffic load. The desired trade-off between the grade and quality of service measures can be achieved by adapting the reservation level to the data traffic load. Initially, only for very light data traffic loads, it seems a waste to reserve any channels for data transfers, so an operator is better off with the *FullShare* policy, sharing all channels and reducing the preferential treatment of voice calls such that data calls can only be downgraded to b traffic channels, e.g., $b = 1$. Since it is most robust against a data traffic load increase, the *FullShareRes* policy has been selected for a further numerical investigation, presented to obtain insight in the performance effects of the various system and policy parameters, and to illustrate the sojourn time expectations that may be fed back to a data caller.

Whereas the presented analysis requires the assumption of exponentially distributed data call sizes to obtain explicit expressions for, e.g., (conditional) expected sojourn times, it has been observed that, e.g., the size of World Wide Web pages typically has a heavy tailed distribution (e.g., [Crovella and Bestavros, 11; Paxson and Floyd, 31]). It is therefore of interest to determine the impact of the data call size distribution on the performance of the investigated model. In this light, we note that the data service model is basically a combination of a finite First-Come First-Served (FCFS) access queue and

(effectively) a Processor Sharing (PS) transfer queue served at a varying rate due to the voice call arrival and termination process. The performance of an FCFS queue is known to degrade under a more heavily tailed job size distribution, as a rare large job in service causes huge queueing delays and hence significant blocking (in correspondence with the Pollaczek–Khintchine formula (e.g., [Tijms, 36])). In contrast, the performance of a PS queue, known to be insensitive to the job size distribution under a fixed service capacity (e.g., [Tijms, 36]), *improves* under a more heavily tailed job size distribution under a *varying* service capacity as in the model of our paper, a remarkable phenomenon that is observed and analytically supported by Litjens and Boucherie [23]. The net performance effect of the data call size distribution is determined by the dominant queue (access or transfer queue), and is thus strongly dependent on the maximum number of active data calls. As this number is generally proportional to the cell capacity, a heavy tailed data call size distribution tends to worsen the net performance in low-capacity cells, but enhance the net performance in high-capacity cells.

In an adjusted form, the application of the model and mathematical results of the present paper to a performance evaluation of an integrated services GSM/GPRS network is under investigation. Initial results are published by Litjens and Boucherie [22], while future work aims at extending this research to a GSM/GPRS model with two priority classes for the data service, i.e., prioritized data and best effort. Within this framework, we aim to develop admission control rules that can provide probabilistic QOS guarantees for the high-priority data service, while serving the best-effort calls with the varying excess capacity. Furthermore, an extension to a GSM/GPRS model including video and data services is studied. There the QOS experienced by video services is defined as the time- or call-average throughput, while the call duration is unaffected by the amount of attention given. Finally, the impact of the data call size distribution tail on the experienced QOS in an integrated services model is an interesting topic for further research.

Acknowledgements

The authors would like to thank Rudesindo Núñez Queija of the Center for Mathematics and Computer Science, The Netherlands, and Hans van den Berg of KPN Research, The Netherlands, for helpful discussions and comments.

Appendix A

This appendix contains the proofs of theorems 1–3, lemma 4, theorem 5 and corollary 6.

A.1. Proof of theorem 1

Proof. Denote with $\mathbf{s} \in \mathbb{S}_+^{a+}$ the random system state where the tagged data call starts its transfer, and define the auxiliary random variable $\tau_{v,d}^*(x)$ as the transfer time of the

tagged call which enters the system in (v, d). Conditioning on $\mathbf{s}$ gives

$$\begin{aligned}\widehat{\sigma}_{v,d}(x) &= \mathbf{E}\{\sigma_{v,d}(x)\} = \mathbf{E}\{\omega_{v,d} + \tau^*_{v,d}(x)\} \\ &= \mathbf{E}\{\omega_{v,d}\} + \sum_{(v_o,d_o)\in\mathbb{S}^{a+}_+} \mathbf{E}\{\tau_{v_o,d_o}(x) \mid \mathbf{s} = (v_o, d_o)\}\, \mathcal{J}\big((v, d); (v_o, d_o)\big) \\ &= \widehat{\omega}_{v,d} + \sum_{(v_o,d_o)\in\mathbb{S}^{a+}_+} \widehat{\tau}_{v_o,d_o}(x)\, \mathcal{J}\big((v, d); (v_o, d_o)\big),\end{aligned}$$

where the third equality uses the Markov property in that $\tau_{v_o,d_o}(x)$ is independent of (v, d). □

A.2. Proof of theorem 2

Proof. The expected waiting time of a data call which enters the system in state (v, d) is given in the limit value of the cumulative expected waiting time vector of the adjusted Markov chain, governed by recursive relation (4), and is given by

$$\widehat{\boldsymbol{\omega}}_+ = \lim_{n\longrightarrow\infty} \widehat{\boldsymbol{\omega}}^n_+.$$

The limit value follows from solving the linear system of balance equations provided by (4):

$$\widehat{\boldsymbol{\omega}}_+ = \widehat{\boldsymbol{\omega}}^*_+ + \widetilde{\mathcal{P}}_{++}\, \widehat{\boldsymbol{\omega}}_+.$$

The convergence of the cumulative expected waiting times is due to the transiency of $\mathbb{S}^{q+}_+$ and the fact that no further costs are incurred in the absorbing set $\mathbb{S}^{q0}_+$. Indeed, since $\widetilde{\mathcal{P}}_{++}$ is a substochastic matrix representing transient states, $\widetilde{\mathcal{P}}^n_{++} \to 0$, which implies that all of the eigenvalues of $\widetilde{\mathcal{P}}_{++}$ have absolute values strictly less than 1. Hence the eigenvalues of $\mathcal{I} - \widetilde{\mathcal{P}}_{++}$ are all nonzero, and thus the matrix is indeed nonsingular and its inverse $(\mathcal{I} - \widetilde{\mathcal{P}}_{++})^{-1}$ exists, which concludes the proof. □

A.3. Proof of theorem 3

Proof. It is obvious that the probability $\mathcal{J}((v, d); (v_o, d_o))$ that a call entering the original system in state $(v, d) \in \mathbb{S}_+$ starts its transfer in state $(v_o, d_o) \in \mathbb{S}^{a+}_+$ is equal to the probability $\mathcal{J}^*((d_q(v, d), v, d); (0, v_o, d_o))$ that the augmented process, starting in state $(d_q(v, d), v, d) \in \mathbb{S}^*_+$ is eventually absorbed in state$(0, v_o, d_o) \in \mathbb{S}^*_0$. Hence we only need to show that the probability matrix $\mathcal{J}^*$ can indeed be calculated as stated in the theorem.

Consider the augmented chain in transient state $(n, v, d) \in \mathbb{S}^*_+$. Conditioning on the first transition out of (n, v, d) yields, for any $(0, v_o, d_o) \in \mathbb{S}^*_0$,

$$\begin{aligned}\mathcal{J}^*\big((n, v, d); (0, v_o, d_o)\big) &= \mathcal{P}\big((n, v, d); (0, v_o, d_o)\big) \\ &+ \sum_{(n',v',d')\in\mathbb{S}^*_+} \mathcal{P}\big((n, v, d); (n', v', d')\big)\, \mathcal{J}^*\big((n', v', d'); (0, v_o, d_o)\big).\end{aligned}$$

In matrix form, this can be formulated as

$$\mathcal{J}^* = \mathcal{P}^*_{+0} + \mathcal{P}^*_{++}\mathcal{J}^*.$$

Note that $\mathcal{P}^*_{++}$ is a substochastic matrix representing transient states, so that $(\mathcal{P}^*_{++})^n \to 0$, which implies that all of the eigenvalues of $\mathcal{P}^*_{++}$ have absolute values strictly less than 1. Hence the eigenvalues of $\mathcal{I} - \mathcal{P}^*_{++}$ are all nonzero, and thus the matrix is nonsingular and its inverse $(\mathcal{I} - \mathcal{P}^*_{++})^{-1}$ exists, which concludes the proof. □

A.4. Proof of lemma 4

Proof. The lemma is proven by marginal analysis. Consider a time interval of length $\Delta > 0$, with Δ sufficiently small such that the tagged call cannot terminate within this time, hence $\Delta < x/\overline{b}(v,d)$ in state $(v,d) \in \mathbb{S}^{a+}_+$. Recall the definitions and properties of $\underline{b}(v,d)$ and $\overline{b}(v,d)$ from section 2.2. Condition on all the possible events occurring in this interval, starting out in state $(v,d) \in \mathbb{S}^{a+}_+$. For notational convenience and readability, the boundary constraints are not explicitly considered. Equations for the boundary can be derived by analogy with the results below:

$$\begin{aligned}
\widehat{\tau}_{v,d}(x) = \Delta & \\
&+ \lambda_{\text{voice}}\,\Delta\,\widehat{\tau}_{v+1,d}\big(x - \mathrm{O}(\Delta)\big) \\
&+ v\,\mu_{\text{voice}}\,\Delta\,\widehat{\tau}_{v-1,d}\big(x - \mathrm{O}(\Delta)\big) \\
&+ \lambda_{\text{data}}\,\Delta\,\widehat{\tau}_{v,d+1}\big(x - \mathrm{O}(\Delta)\big) \\
&+ \big(\beta(v,d)\,d_a(v,d) - \underline{b}(v,d)\big)\,\underline{p}(v,d)\,\mu_{\text{data}}\,\Delta\,\widehat{\tau}_{v,d-1}\big(x - \mathrm{O}(\Delta)\big) \\
&+ \big(\beta(v,d)\,d_a(v,d) - \overline{b}(v,d)\big)\,\overline{p}(v,d)\,\mu_{\text{data}}\,\Delta\,\widehat{\tau}_{v,d-1}\big(x - \mathrm{O}(\Delta)\big) \\
&+ \underline{p}(v,d)\,\big(1 - \big(\lambda_{\text{voice}} + v\,\mu_{\text{voice}} + \lambda_{\text{data}} \\
&\quad + \big(\beta(v,d)\,d_a(v,d) - \underline{b}(v,d)\big)\mu_{\text{data}}\big)\Delta\big)\,\widehat{\tau}_{v,d}\big(x - \underline{b}(v,d)\Delta\big) \\
&+ \overline{p}(v,d)\,\big(1 - \big(\lambda_{\text{voice}} + v\,\mu_{\text{voice}} + \lambda_{\text{data}} \\
&\quad + \big(\beta(v,d)\,d_a(v,d) - \overline{b}(v,d)\big)\,\mu_{\text{data}}\big)\Delta\big)\,\widehat{\tau}_{v,d}\big(x - \overline{b}(v,d)\Delta\big) \\
&+ \mathrm{o}(\Delta),
\end{aligned}$$

where $\mathrm{O}(\Delta)$ ($\mathrm{o}(\Delta)$) is standard notation for some unspecified function $F(\Delta)$ ($f(\Delta)$) having the property that $\lim_{\Delta\to 0} F(\Delta)/\Delta = c$ for some $c \in \mathbb{R}$ ($\lim_{\Delta\to 0} f(\Delta)/\Delta = 0$), i.e., $F(\Delta)$ ($f(\Delta)$) becomes negligibly small (compared to Δ) as $\Delta \to 0$. The fifth and seventh line on the right hand side correspond with the case that the tagged data call is assigned $\underline{b}(v,d)$ channels, while in the sixth and the eighth line the tagged call is assigned one extra channel, i.e., $\overline{b}(v,d)$ channels.

Rearranging terms, and substituting (1), yields

$$\underline{p}(v,d)\,\underline{b}(v,d)\,\frac{\widehat{\tau}_{v,d}(x) - \widehat{\tau}_{v,d}(x - \underline{b}(v,d)\,\Delta)}{\underline{b}(v,d)\,\Delta}$$

$$
\begin{aligned}
&+ \overline{p}(v,d)\,\overline{b}(v,d)\,\frac{\widehat{\tau}_{v,d}(x) - \widehat{\tau}_{v,d}(x - \overline{b}(v,d)\,\Delta)}{\overline{b}(v,d)\,\Delta} \\
&= 1 \\
&\quad + \lambda_{\text{voice}}\,\widehat{\tau}_{v+1,d}\big(x - \mathrm{O}(\Delta)\big) \\
&\quad + v\,\mu_{\text{voice}}\,\widehat{\tau}_{v-1,d}\big(x - \mathrm{O}(\Delta)\big) \\
&\quad + \lambda_{\text{data}}\,\widehat{\tau}_{v,d+1}\big(x - \mathrm{O}(\Delta)\big) \\
&\quad + \beta(v,d)\,\big(d_a(v,d) - 1\big)\,\mu_{\text{data}}\,\widehat{\tau}_{v,d-1}\big(x - \mathrm{O}(\Delta)\big) \\
&\quad + \underline{p}(v,d)\,\big(-\lambda_{\text{voice}} - v\,\mu_{\text{voice}} - \lambda_{\text{data}} \\
&\qquad - \big(\beta(v,d)\,d_a(v,d) - \underline{b}(v,d)\big)\,\mu_{\text{data}}\big)\widehat{\tau}_{v,d}\big(x - \underline{b}(v,d)\,\Delta\big) \\
&\quad + \overline{p}(v,d)\,\big(-\lambda_{\text{voice}} - v\,\mu_{\text{voice}} - \lambda_{\text{data}} \\
&\qquad - \big(\beta(v,d)\,d_a(v,d) - \overline{b}(v,d)\big)\,\mu_{\text{data}}\big)\widehat{\tau}_{v,d}\big(x - \overline{b}(v,d)\,\Delta\big) \\
&\quad + \frac{\mathrm{o}(\Delta)}{\Delta}.
\end{aligned}
$$

Letting $\Delta \downarrow 0$, and substituting (1) once again, gives

$$
\begin{aligned}
\beta(v,d)\,\frac{\partial \widehat{\tau}_{v,d}(x)}{\partial x} &= \underline{p}(v,d)\,\underline{b}(v,d)\,\lim_{\Delta\downarrow 0}\left(\frac{\widehat{\tau}_{v,d}(x) - \widehat{\tau}_{v,d}(x - \underline{b}(v,d)\,\Delta)}{\underline{b}(v,d)\,\Delta}\right) \\
&\quad - \overline{p}(v,d)\,\overline{b}(v,d)\,\lim_{\Delta\downarrow 0}\left(\frac{\widehat{\tau}_{v,d}(x) - \widehat{\tau}_{v,d}(x - \overline{b}(v,d)\,\Delta)}{\overline{b}(v,d)\,\Delta}\right) \\
&= 1 \\
&\quad + \lambda_{\text{voice}}\,\widehat{\tau}_{v+1,d}(x) \\
&\quad + v\,\mu_{\text{voice}}\,\widehat{\tau}_{v-1,d}(x) \\
&\quad + \lambda_{\text{data}}\,\widehat{\tau}_{v,d+1}(x) \\
&\quad + \beta(v,d)\,\big(d_a(v,d) - 1\big)\,\mu_{\text{data}}\,\widehat{\tau}_{v,d-1}(x) \\
&\quad + \big(-\lambda_{\text{voice}} - v\,\mu_{\text{voice}} - \lambda_{\text{data}} - \beta(v,d)\,\big(d_a(v,d) - 1\big)\,\mu_{\text{data}}\big)\,\widehat{\tau}_{v,d}(x).
\end{aligned}
$$

Note that since $\underline{b}(v,d), \overline{b}(v,d) > 0$ both limits are well-defined. This system of differential equations may equivalently be written in matrix notation,

$$
\mathcal{B}\,\frac{\partial}{\partial x}\widehat{\boldsymbol{\tau}}(x) = \mathbf{1} + \mathcal{Q}^{\bullet}\widehat{\boldsymbol{\tau}}(x) \quad \Longleftrightarrow \quad \frac{\partial}{\partial x}\widehat{\boldsymbol{\tau}}(x) = \mathcal{B}^{-1}\mathbf{1} + \mathcal{B}^{-1}\mathcal{Q}^{\bullet}\widehat{\boldsymbol{\tau}}(x).
$$

To conclude the proof, the initial condition simply reflects the fact that the transfer time $\tau_{v,d}(0)$ of an 'empty' data call is zero, almost surely. □

A.5. Proof of theorem 5

Proof. In order to prove that the system of differential equations (5) with initial condition (6) has a *unique* solution, note that it is a system of the form $(\partial/\partial x)\widehat{\boldsymbol{\tau}}(x) = \mathbf{a}_{\mathrm{o}} + \mathcal{A}\,\widehat{\boldsymbol{\tau}}(x) \equiv f(\widehat{\boldsymbol{\tau}}(x))$, where f is a linear function with continuous partial derivatives

with respect to the entries of its argument vector. The existence and uniqueness of a solution $\widehat{\boldsymbol{\tau}}(x)$ for every initial vector, immediately follows from, e.g., [Braun, 5, theorem 3, p. 412].

The existence of a vector $\boldsymbol{\gamma}$ that satisfies (7) and its uniqueness up to a translation along the vector $\mathbf{1}$, are guaranteed by results in Markov decision theory. Interpreting $\boldsymbol{\gamma}$ as the vector of relative values in a Markov reward chain governed by the generator $\mathcal{Q}^{\bullet}$ and with immediate cost vector $(1/\eta)(\mathbf{1} - \mathcal{B}\mathbf{1}/(\boldsymbol{\pi}^{\bullet}\mathcal{B}\,\mathbf{1}))$, where η is the maximum rate of change in the Markov chain, and understanding that the long-term average costs are zero,

$$\boldsymbol{\pi}^{\bullet}\left(\frac{\mathcal{B}\mathbf{1}}{\boldsymbol{\pi}^{\bullet}\mathcal{B}\,\mathbf{1}} - \mathbf{1}\right) = 1 - 1 = 0,$$

e.g., [Tijms, 36, theorem 3.1, p. 167] can be directly applied after uniformization of the continuous-time Markov chain. Note that indeed a translation of $\boldsymbol{\gamma}$ along the vector $\mathbf{1}$ does not alter the solution, since $\exp\{x\mathcal{B}^{-1}\mathcal{Q}^{\bullet}\}\mathbf{1} = \mathbf{1}$, which is readily verified using the Taylor expansion of $\exp\{x\mathcal{B}^{-1}\mathcal{Q}^{\bullet}\}$. Hence in (7) a single degree of freedom exists in choosing $\boldsymbol{\gamma}$, which is used to normalize $\boldsymbol{\gamma}$ as in (8).

The theorem is then proven by substituting the claimed unique solution into the system of differential equations and verifying whether it indeed holds. With $\widehat{\boldsymbol{\tau}}(x)$ as given in (9),

$$\begin{aligned}
\frac{\partial}{\partial x}\widehat{\boldsymbol{\tau}}(x) &= \frac{1}{\boldsymbol{\pi}^{\bullet}\mathcal{B}\,\mathbf{1}}\mathbf{1} - \exp\{x\mathcal{B}^{-1}\mathcal{Q}^{\bullet}\}\mathcal{B}^{-1}\mathcal{Q}^{\bullet}\boldsymbol{\gamma} \\
&= \mathcal{B}^{-1}\mathbf{1} + \mathcal{B}^{-1}\mathcal{Q}^{\bullet}\big[\mathcal{I} - \exp\{x\mathcal{B}^{-1}\mathcal{Q}^{\bullet}\}\big]\boldsymbol{\gamma} \\
&= \mathcal{B}^{-1}\mathbf{1} + \mathcal{B}^{-1}\mathcal{Q}^{\bullet}\frac{x}{\boldsymbol{\pi}^{\bullet}\mathcal{B}\,\mathbf{1}}\mathbf{1} + \mathcal{B}^{-1}\mathcal{Q}^{\bullet}\big[\mathcal{I} - \exp\{x\mathcal{B}^{-1}\mathcal{Q}^{\bullet}\}\big]\boldsymbol{\gamma} \\
&= \mathcal{B}^{-1}\mathbf{1} + \mathcal{B}^{-1}\mathcal{Q}^{\bullet}\left[\frac{x}{\boldsymbol{\pi}^{\bullet}\mathcal{B}\,\mathbf{1}}\mathbf{1} + \big[\mathcal{I} - \exp\{x\mathcal{B}^{-1}\mathcal{Q}^{\bullet}\}\big]\boldsymbol{\gamma}\right] \\
&= \mathcal{B}^{-1}\mathbf{1} + \mathcal{B}^{-1}\mathcal{Q}^{\bullet}\widehat{\boldsymbol{\tau}}(x),
\end{aligned}$$

where (7) is substituted to obtain the second equality, and the third equality follows from adding $\mathcal{B}^{-1}\mathcal{Q}^{\bullet}(x/(\boldsymbol{\pi}^{\bullet}\mathcal{B}\,\mathbf{1}))\mathbf{1}$, which is equal to zero, since $\mathcal{Q}^{\bullet}$ is a generator of a Markov chain, and hence $\mathcal{Q}^{\bullet}\mathbf{1} = \mathbf{0}$. To conclude the proof, observe that (9) satisfies the initial condition (6). □

A.6. Proof of corollary 6

Proof. To see this, note that $\mathcal{B}^{-1}\mathcal{Q}^{\bullet}$ is the generator of an irreducible finite state space Markov chain, with equilibrium distribution vector $\boldsymbol{\pi}^{\bullet}\mathcal{B}/(\boldsymbol{\pi}^{\bullet}\mathcal{B}\,\mathbf{1})$. Hence

$$\lim_{x\to\infty}\exp\{x\mathcal{B}^{-1}\mathcal{Q}^{\bullet}\} = \mathbf{1}\frac{\boldsymbol{\pi}^{\bullet}\mathcal{B}}{\boldsymbol{\pi}^{\bullet}\mathcal{B}\,\mathbf{1}},$$

the matrix with each row equal to the equilibrium distribution vector, and

$$\lim_{x\to\infty}\left(\mathcal{I}-\exp\left\{x\mathcal{B}^{-1}\mathcal{Q}^{\bullet}\right\}\right)\boldsymbol{\gamma}=\boldsymbol{\gamma}-\mathbf{1}\frac{\boldsymbol{\pi}^{\bullet}\mathcal{B}}{\boldsymbol{\pi}^{\bullet}\mathcal{B}\,\mathbf{1}}\boldsymbol{\gamma}=\boldsymbol{\gamma}$$

using (8). Note that this simplest form of the asymptotic result follows from normalizing $\boldsymbol{\gamma}$ as is done in (8). □

References

[1] W. Ajib and P. Godlewski, Service disciplines performance for WWW traffic in GPRS system, in: *Proceedings of 3G Mobile Telecommunication Technologies '00*, 2000, pp. 431–435.

[2] B. Avi-Itzhak and S. Halfin, Expected response times in a non-symmetric time sharing queue with a limited number of service positions, Proceedings of ITC 12 (1988) 5.4B.2.1–7.

[3] G. Bianchi, A. Capone, L. Fratta and L. Musumeci, Packet data service over GSM networks with dynamic stealing of voice channels, in: *Proceedings of GLOBECOM '95*, 1995, pp. 1152–1156.

[4] G. Brasche and B. Walke, Concepts, services, and protocols of the new GSM phase 2+ general packet radio service, IEEE Communications Magazine 35(8) (1997) 94–104.

[5] M. Braun, *Differential Equations and Their Applications* (Springer, New York, 1983).

[6] J. Cai and D.J. Goodman, General packet radio service in GSM, IEEE Communications Magazine 35(10) (1997) 122–131.

[7] D. Calin, S. Malik and D. Zeghlache, Traffic scheduling and fairness for GPRS air interface, in: *Proceedings of IEEE VTC '99*, 1999, pp. 834–838.

[8] D. Calin and D. Zeghlache, Performance analysis of high speed circuit switched data (HSCSD) over GSM, in: *Proceedings of IEEE ICC '98*, 1998, pp. 1586–1590.

[9] Calin, D. and D. Zeghlache, High speed circuit switched data over GSM: Potential traffic policies, in: *Proceedings of IEEE VTC '98*, 1998, pp. 1274–1278.

[10] Y.M. Chuang, T.Y. Lee and Y.B. Lin, Trading CDPD availability and voice blocking probability in cellular networks, IEEE Network (March/April 1998) 48–54.

[11] M. Crovella and A. Bestavros, Self-similarity in World Wide Web traffic: evidence and possible causes, IEEE/ACM Transactions on Networking 5(6) (1997) 835–846.

[12] E. Dahlman, B. Gudmundson, M. Nilsson and J. Sköld, UMTS/IMT-2000 based on wideband CDMA, IEEE Communications Magazine 36(9) (1998) 70–80.

[13] R. De Bernardi, D. Imbeni, L. Vignali and M. Karlsson, Load control strategies for mixed services in WCDMA, in: *Proceedings of IEEE VTC '00*, 2000, pp. 825–829.

[14] ETSI, GSM 02.34; Digital cellular telecommunications system (phase 2+): high speed circuit switched data (HSCSD) stage 1, ETSI, France (1997).

[15] ETSI, GSM 03.34; Digital cellular telecommunications system (phase 2+): high speed circuit switched data (HSCSD) stage 2, ETSI, France (1998).

[16] ETSI, GSM 08.08; Digital cellular telecommunications system (phase 2): Mobile-services switching centre – base station system (MSC-BSS) interface; layer 3 specification , ETSI, France (1998).

[17] A. Furuskär, M. Frodigh, H. Olofsson and J. Sköld, System performance of EDGE, a proposal for enhanced data rates in existing digital cellular systems, in: *Proceedings of IEEE VTC '98*, 1998, pp. 1284–1289.

[18] J.-Y. Jeng, C.-W. Lin and Y.-B. Lin, Dynamic scheduling for GSM data services, IEICE Transactions on Communications E80 B(2) (1997) 296–300.

[19] C. Johansson, L. de Verdier and F. Khan, Performance of different scheduling strategies in a packet radio system, in: *Proceedings of ICUPC '98*, 1998, pp. 267–271.

[20] F.P. Kelly, Charging and rate control for elastic traffic, European Transactions on Telecommunications 8(1) (1997) 33–37.

[21] K.D. Kennedy and R. Litjens, Performance evaluation of a hybrid radio resource allocation algorithm in a GSM/GPRS network, in: *Proceedings of PIMRC '99*, 1999, pp. 131-136.
[22] R. Litjens and R.J. Boucherie, Radio resource sharing in a GSM/GPRS network, in: *Proceedings of the ITC Specialist Seminar on Mobile Systems and Mobility*, 2000, pp. 261–274.
[23] R. Litjens and R.J. Boucherie, Elastic calls in an integrated services network: the heavier the tail the better the quality-of-service, Report AE 1/00, Institute of Actuarial Sciences & Econometrics, Universiteit van Amsterdam (2000).
[24] M. Meyer, TCP performance over GPRS, in: *Proceedings of IEEE WCNC '99*, 1999, pp. 1248–1252.
[25] M. Mouly and M.-B. Pautet, *The GSM System for Mobile Communications* (Palaiseau, France, published by the authors, 1992).
[26] R. Núñez Queija, Processor-sharing models for integrated-services networks, Ph.D. thesis, Technische Universiteit Eindhoven (1999).
[27] R. Núñez Queija, J.L. van den Berg and M.R.H. Mandjes, Performance evaluation of strategies for integration of elastic and stream traffic, Proceedings of ITC 16 (1999) 1039–1050.
[28] T. Ojanperä and R. Prasad, An overview of air interface multiple access for IMT-2000/UMTS, IEEE Communications Magazine 36(9) (1998) 82–95.
[29] Q. Pang, A. Bigloo, V.C.M. Leung and C. Scholefield, Service scheduling for general packet radio service classes, in: *Proceedings of WCNC '99*, 1999, pp. 1229–1233.
[30] C. Parsa and J.J. Garcia-Luna-Aceves, Improving TCP performance over wireless networks at the link layer, Mobile Networks and Applications 5 (2000) 57–71.
[31] A. Paxson and S. Floyd, Wide area traffic: The failure of Poisson modelling, IEEE/ACM Transactions on Networking 3(3) (1995) 226–244.
[32] S. Ramakrishna and J.M. Holtzman, A scheme for throughput maximization in a dual-class CDMA system, in: *Proceedings of ICUPC '97*, 1997, pp. 623–627.
[33] J.M. Rivadeneyra Sicilia and J. Miguel-Alonso, A communication architecture to access data services through GSM, in: *Proceedings of INDC '98*, 1998, pp. 1–11.
[34] J.W. Roberts et al., eds., *Broadband Network Teletraffic (COST 242)* (Springer, Berlin/Heidelberg, 1996).
[35] J.W. Roberts and L. Massoulié, Bandwidth sharing and admission control for elastic traffic, in: *Proceedings of the ITC Specialist Seminar on Teletraffic Issues Related to Multimedia and Nomadic Communications*, 1998.
[36] H.C. Tijms, *Stochastic Modelling and Analysis: A Computational Approach* (Wiley, Chichester, 1986).
[37] R.W. Wolff, *Stochastic Modeling and the Theory of Queues* (Prentice-Hall, Englewood Cliffs, NJ, 1989).
[38] W. Wu, W. Seah, A. Lo and C.C. Ko, Differentiated service provisioning in third-generation CDMA cellular networks, in: *Proceedings of the ITC Specialist Seminar on Mobile Systems and Mobility*, 2000, pp. 27–38.