Mobile Networks and Applications 6, 427–434, 2001

Globally Constrained Power Control Across Multiple Channels in Wireless Data Networks

NICHOLAS BAMBOS
Department of Electrical Engineering and Department of Management Science Engineering, Stanford University, Stanford, CA 94305, USA

SUNIL KANDUKURI
Department of Electrical Engineering, Stanford University, Stanford, CA 94305, USA

Abstract. We investigate multi-channel transmission schemes for packetized wireless data networks. The transmitting unit transmits concurrently in several orthogonal channels (for example, distinct FDMA bands or CDMA codes) with randomly fluctuating interference and there is a global constraint on the total power transmitted across all channels at any time slot. Incoming packets to the transmitter are queued up in separate buffers, depending on the channel they are to be transmitted in. In each time slot, one packet can be transmitted in each channel from its corresponding queue. The issue is how much power to transmit in each channel, given the interference in it and the packet backlog, so as to optimize various power and delay costs associated with the system. We formulate the general problem taking a dynamic programming approach. Through structural decompositions of the problem, we design practical novel algorithms for allocating power to various channels under the global power constraint.

Keywords: wireless networks, multi-channel power control

1. Introduction

Power control in wireless networking mitigates interference, increasing the network capacity, and minimizes the power spent to achieve a target quality of service per communication link, prolonging the mobile battery life. Power control in voice oriented wireless networks has been studied extensively in the past [1–5,9,11,12,18–20], while in packet networks relatively more recently [6,8,15]. Previous studies of power control in wireless networks with packetized data traffic analyze the case of a single transmitter communicating with a single receiver in a channel with randomly fluctuating interference. During high interference periods packets may have to be queued up in the transmitter buffer and incur a delay cost instead of being transmitted at high power (due to high interference) and incur an unacceptable power cost.

In this paper, we consider a central transmitting unit communicating with multiple receivers, each in a separate channel which is orthogonal to all others and has randomly fluctuating interference. The total power transmitted in all channels has to be less than a certain power ceiling, which is determined by operational constraints associated with the transmitting unit. Arriving packets to the transmitter are queued up in separate buffers, depending on which user (receiver in channel) they are to be transmitted to. In each time slot one packet may be transmitted in each channel. Given the power ceiling, the dilemma is how much power to transmit in each channel, that is, how to distribute the constrained power to the various channels. Intuitively, one feels that channels with high backlog and low interference in a slot should be allocated more power than channels with low backlog and high interference. To capture this dilemma and understand the situation, in this paper we model the system within a dynamic programming framework [7,14]. Because packetized traffic may have to tolerate limited delay, we associate deadlines with packets. That is, if a packet is not transmitted successfully before its set deadline, then it expires and gets dropped from the buffer and a cost in incurred.

The need for combined power control across multiple channels with a global power ceiling can arise in cases where a mobile wireless data terminal transmits to multiple network access points concurrently (for example, multiple base stations in the case of cellular networks) over orthogonal radio channels, that is, FDMA bands or CDMA codes. In this situation of access and channel diversity the data terminal may be supporting various wireless computing applications on different channels. The electronic circuitry and the battery constraints determine the maximum power discharge across all channels, that is, the power ceiling mentioned above. A 'dual' scenario is that where a central transmitting unit, say a base station in a micro-cellular network architecture or a wireless LAN, transmits to multiple user receivers in orthogonal channels. We mostly refer to the latter scenario in this paper, however, the former is also clearly covered.

The paper is organized as follows. Section 2 introduces the stochastic model used in this study and the general formulation of the power control problem, based on a dynamic programming approach. That includes packet deadlines leading to high complexity and making necessary reduced formulations of the problem that can lead to tractable solutions. In section 3 the issue of allocating the power so as to maximize the aggregate system throughput is considered, as well as that of keeping the individual channel throughput constant in the presence of fluctuating interference. Novel practical algorithms are introduced for these two cases. In section 4 we consider a reduced formulation of the power control problem which still includes delay costs associated with the system buffers. However, the operational complexity still remains high. To address that, we consider in section 5 another formulation of the problem based on a decomposition to individual queues. This allows us to design a novel practical algorithm for multi-channel power control. We conclude with some final comments.

2. The multi-channel power control model – general formulation

Consider a central transmitting unit (for example, a base station) communicating with K users, each receiving in one of K distinct channels indexed by $k \in \{1, 2, 3, \ldots, K\}$. The system operates in *slotted time* and transmission is *packetized*. In each time slot, one packet may be transmitted in each channel. Time slots are indexed by $n \in \{1, 2, 3, \ldots\}$.

The transmitter has K buffers, one for each user. Packets arrive at the transmitter and are queued up in separate buffers, depending on which channel they are to be transmitted in. Each individual buffer is served in a FIFO (first-in-first-out) manner. At most one packet per user can arrive in each time slot and packet arrival streams are modeled as *independent Bernoulli processes*. That is, in each time slot, a k-user packet arrives with probability a_k and is queued up in the kth buffer, or none arrives with probability $1 - a_k$. All arrival events are independent of each other in the various time slots and amongst the various users/channels.

Let p_k^n be the power transmitted in channel k during time slot n and $P_n = (p_n^1, p_n^2, \ldots, p_n^k, \ldots, p_n^K)$ the joint power vector. There is a power ceiling $P_{\max}$ that the sum of all transmitted powers cannot exceed, that is,

$$\sum_{k=1}^{K} p_n^k \leqslant P_{\max} \tag{1}$$

for every time slot n. Input signals to different channels are multiplexed and modulated before being transmitted [16].

Let i_n^k be the interference level in channel k over time slot n. It is assumed that the interference does not vary within each time slot, but fluctuates between consecutive ones. The joint interference vector over all channels during time slot n is given by

$$I_n = \left(i_n^1, i_n^2, \ldots, i_n^k, \ldots, i_n^K\right) \tag{2}$$

and is assumed to fluctuate according to a time-homogeneous irreducible Markov chain on a finite (but perhaps large) state space $\mathbf{I}$ with given transition probabilities

$$\mathrm{Prob}\left[I_{n+1} = I' \mid I_n = I\right] = \Phi\left(I, I'\right), \tag{3}$$

where $I, I' \in \mathbf{I}$. Later in the analysis, we consider the special case where the interference in each channel is statistically independent from that in others. Overall, it is assumed that the interference is unresponsive to the power transmitted in the channels, that is, it is modulated by some extraneous agent. In practical situations, the cochannel interference would be responsive to the power transmitted in the channel, because of other transmitters in it reacting to power variations of the one under consideration. For methodological purposes, however, we make the assumption that the interference is unresponsive, trying to separate the concern of reactive cochannel interference from that of power allocation across different channels, which is our focus here. Understanding the latter can allow one to develop efficient heuristics to address the case of responsive interference, as in [6], for example.

In each time slot, the packets that are positioned at the front of the user queues are concurrently transmitted – each in its corresponding channels – at certain chosen powers. The probability that a packet transmitted in channel k during time slot n is successfully received by the user receiver is $s^k(p_n^k, i_n^k)$, depending on the transmitted power p_n^k and the channel interference i_n^k. It is naturally assumed that each $s^k(p, i)$ function is increasing in the power argument p and decreasing in the interference one i. The particular form of $s^k(p, i)$ depends on the modulation scheme and the channel propagation characteristics. For example, under standard DPSK/F (differential binary phase-shift keying over a fading channel) and NC-FSK/F (non-coherent frequency-shift keying over a fading channel) [13,15] – and potentially several others – the success probability has the generic functional form

$$s(p, i) = \frac{p/i}{\alpha + \beta(p/i)} = \frac{p}{\alpha i + \beta p}, \tag{4}$$

where $\alpha \geqslant 0$, $\beta \geqslant 1$. Note that p/i is the SIR (signal to interference ratio). Another common functional form is

$$s(p, i) = 1 - \mathrm{e}^{-\gamma p/i}, \tag{5}$$

where $\gamma \geqslant 0$. Actually, the detailed formula of $s(p, i)$ turns out not to be critical for the purposes of our study. It is its functional form that matters, and this has to have the *structural properties* of being increasing in p and decreasing in i.

Upon successful reception of a transmitted packet in a channel, the transmitter attempts to transmit the next packet in the corresponding user queue. In the case of unsuccessful transmission, the transmitter continues to attempt transmission of the same packet in the following time slots, until it is successfully received (or until the packet deadline expires, as we shall see later). It is assumed that there exists reliable ACK/NACK mechanism for each user, which informs the transmitter whether the packet was successfully received or not, instantaneously at the end of each time slot. Furthermore, it is assumed that we can estimate the channel interference at the beginning of every time slot, and this stays constant throughout the time slot. It is assumed that all random: (1) packet arrival events, (2) successful or not transmission events, and (3) interference switching events, are statistically independent of each other.

The standard approach would be to transmit equal power $P_k = P_{\max}/K$ in each channel $k \in \{1, 2, \ldots, K\}$. In this paper, we consider power allocation schemes which transmit at different power level in each channel, according to its interference and the corresponding user packet backlog. The goal is to achieve throughput and power gains over the standard approach.

We formulate the problem within a dynamic programming framework. We consider that each packet of the kth user has a *deadline* of D^k time slots associated with it. That is, each k-user packet must be successfully transmitted within D^k time slots after its arrival, otherwise it expires and gets dropped from the kth buffer. We also consider that each user buffer at the transmitter can hold up to B^k packets. Observe that, since at most one packet may arrive to the kth FIFO buffer at any time slot, and every packet leaves the buffer within D^k time slots (or else it is simply dropped), the maximum number of packets that can ever be in the kth buffer in a time slot is simply

$$M^k = \min\{B^k, D^k\}. \tag{6}$$

Note that a kth buffer overflow can only occur when $B^k \leqslant D^k$. Otherwise, only packet drops may occur due to their deadlines expiring before being transmitted successfully. Indeed, when $B^k > D^k$, in the *worst case scenario* the front packet of the kth FIFO buffer has been in it for D^k time slots; hence, it must be either successfully transmitted in the current time slot or otherwise dropped from the buffer. Therefore, the number of packets in this buffer will never exceed D^k $(< B^k)$, and no buffer overflow will ever occur due to an arriving packet finding the buffer full.

The backlog state of the kth user's buffer at the transmitter during time slot n is given by the M^k-dimensional vector

$$R_n^k = \left(r_n^k(1), r_n^k(2), \ldots, r_n^k(m), \ldots, r_n^k\left(M^k\right)\right), \tag{7}$$

$m \in \{1, 2, \ldots, M^k\}$, where the integer $r_n^k(m)$ is the *residual lifetime* of the packet at the mth place of the FIFO buffer of the kth user at the nth time slot. The residual lifetime is the remaining number of time slots till the packet's transmission deadline expires; the latter is initiated when the packet arrives in the buffer. If there is no packet at the mth place of the kth buffer at the nth time slot, then we artificially assign $r_n^k(m) = \infty$. Therefore, if there are only q packets in the kth buffer during the nth time slot, then $R_n^k = (r_n^k(1), r_n^k(2), \ldots, r_n^k(q), \infty, \infty, \ldots, \infty)$. The collective backlog state of all K buffers at the central transmitter unit during time slot n is denoted by

$$R_n = \left\{R_n^1, R_n^2, \ldots, R_n^k, \ldots, R_n^K\right\}. \tag{8}$$

Let $\mathbf{R}$ be the set of all possible states that the collective backlog state R_n may attain during its evolution. The complete *system state* is $\{R_n, I_n\}$, which tracks both the backlog and the the interference states.

It is easy to see that under the assumptions introduced before, $\{R_n, I_n\}$ is a discrete-time Markov chain on the state space $\mathbf{R} \times \mathbf{I}$. The transition probabilities can be calculated easily, from those of the interference state (3) and the individual backlog states of each buffer. For example, it is easy to see that $R_n^k = (r_n^k(1), r_n^k(2), \ldots, r_n^k(q), \infty, \infty, \ldots, \infty)$ can switch to $R_{n+1}^k = (r_n^k(1) - 1, r_n^k(2) - 1, \ldots, r_n^k(q) - 1, D^k, \infty, \ldots, \infty)$ with probability $a^k(1 - s^k(p_n^k, i_n^k))$, corresponding to (1) a packet arriving, (2) the transmitted packet not being received successfully, and (3) no packet being dropped during time slot n. Similarly, $R_n^k = (r_n^k(1), r_n^k(2), \ldots, r_n^k(q), \infty, \infty, \ldots, \infty)$ can switch to $R_{n+1}^k = (r_n^k(2) - 1, r_n^k(3) - 1, \ldots, r_n^k(q) - 1, D^k, \infty, \ldots, \infty)$ with probability $a^k s^k(p_n^k, i_n^k)$, corresponding to (1) a packet arriving, (2) the transmitted packet being received successfully, and (3) no packet being dropped during time slot n. Given the independence of all random events (arrivals, transmissions, and interference transitions) that we have assumed, we can similarly calculate the transition probabilities easily

$$\begin{aligned}\mathrm{Prob}\big[(R_{n+1}, I_{n+1}) = (R', I') \mid (R_n, I_n) = (R, I), P_n = P\big]\\ = \Pi_P\{(R, I), (R', I')\}\end{aligned} \tag{9}$$

for all $(R, I), (R', I') \in \mathbf{R} \times \mathbf{I}$ and a given power vector P, by considering the possible state transitions and calculating the probabilities through independent coin flips.

During the nth time slot the system incurs a power cost $\sum_{k=1}^K p_n^k$, reflecting the total power spent transmitting in the various channels, plus a backlog cost $\mathcal{B}(R_n)$. The function $\mathcal{B}()$ from $\mathbf{R}$ to the positive real numbers reflects the delay or holding cost of the packets in the various buffers of the system, as well as potential deadline expiration and overflow costs. Indeed, note that when the backlog state of the kth buffer becomes $R_n^k = (0, r_n^k(2), \ldots, r_n^k(q), \infty, \ldots, \infty)$, the deadline of the front packet in this buffer has expired (its residual lifetime has reached 0) and the packet has to be dropped from the buffer. This packet dropping cost can easily be incorporated into the backlog cost $\mathcal{B}(R_n)$ for the corresponding backlog state.

The issue is how to optimally control the power $\{P_n\}$ in the various time slots, so as to minimize the expected cost incurred by the system over its evolution. We can cast the problem into a dynamic programming framework [7,14] as follows. Suppose the system is run for N time slots total. Let $V_n(R, I)$ be the minimal cost-to-go at time $n \leqslant N$ from state (R, I) – that is, the minimal expected cost to be incurred under optimal power control till time N, given that the system starts at time slot n from state (R, I). The cost-to-go will satisfy the dynamic programming equation

$$\begin{aligned}V_n(R, I) = \min_{P \in \mathbf{P}} \Bigg\{ &\sum_{k=1}^K p^k + \mathcal{B}(R)\\ &+ \sum_{(R', I') \in \mathbf{R} \times \mathbf{I}} \Pi_P\{(R, I), (R', I')\}\\ &\times V_{n+1}(R', I') \Bigg\},\end{aligned} \tag{10}$$

where $\mathbf{P} = \{(p^1, \ldots, p^k, \ldots, p^K)\colon \sum_{k=1}^K p^k \leqslant P_{\max}\}$ is the set of power vectors that do not exceed the power ceiling.

The dynamic programming equation can be solved recursively backwards (starting from N with a given terminal cost) to compute the optimal power control $P_n^*(R, I), n \in \{1, 2, \ldots, n, \ldots, N\}$, for each $(R, I) \in \mathbf{R} \times \mathbf{I}$. In principle, the above dynamic program could be solved numerically for a long time horizon N (to wash out the transient and boundary effects) to obtain the optimal power to transmit at any time instant $P^*(R, I)$ given the backlog and interference states R and I, correspondingly. However, because of the complexity of the backlog state R, solving the dynamic programming recursion numerically is a highly nontrivial matter. To overcome this problem we consider below other radically simplified reformulations of the problem. There may be 'suboptimal' but lead to simple ways for deciding how much power to transmit in each channel.

3. Throughput oriented power allocation

As mentioned before, the simplest power control strategy is to statically allocate power $P_{\max}/K$ to each channel, independently of its backlog and interference states. We call that *static power allocation*. Obviously, this strategy does not allow shifting power to channels/users where the interference is high from those where it is not and vice versa. In its most restricted form it would make $P_{\max}/K$ available even to channels with empty buffers at particular time slots where it cannot be used, instead of allocating it to those that really need it. That would be the case if the transmission processes within each channel were totally decoupled and each channel were operated independently with a fixed power assigned to it. This is basically the situation in current technology, where each channel would have its own transmitter decoupled from all others and not allowing shifting of power from channel to channel. A slight modification – and improvement – over the previous approach is the situation where the total available power $P_{\max}$ is equally distributed to the channels that have non-empty buffers in the current time slot.

3.1. Maximizing the instantaneous aggregate throughput

An interesting approach is to try to allocate the power to the different channels in such a way that we maximize the aggregate packet throughput per time slot. Let us first assume that none of the K buffers is empty. When the interference vector in a time slot is $I = (i^1, i^2, \ldots, i^k, \ldots, i^K)$ and a power vector $P = (p^1, p^2, \ldots, p^k, \ldots, p^K)$ is used, it is easy to see that the expected total number of packets that are successfully transmitted in this time slot across all channels is simply

$$T(P, I) = \sum_{k=1}^{K} s^k\big(p^k, i^k\big), \tag{11}$$

since the transmission events are independent of each other and success occurs with probability $s^k(p^k, i^k)$ in the kth channel. For each fixed I, we choose the power vector $P_*(I) = (p_*^1(I), p_*^2(I), \ldots, p_*^k(I), \ldots, p_*^K(I))$ that maximizes the instantaneous throughput $T(P, I)$ or

$$T(P_*, I) = \sum_{k=1}^{K} s^k\big(p_*^k(I), i^k\big) = \max_{P \in \mathbf{P}} \left\{ \sum_{k=1}^{K} s^k\big(p^k, i^k\big) \right\}, \tag{12}$$

where $\mathbf{P} = \{(p^1, \ldots, p^k, \ldots, p^K)\colon \sum_{k=1}^{K} p^k \leqslant P_{\max}\}$ is the space of power vectors that do not exceed the power ceiling $P_{\max}$. Since all $s^k(p^k, i^k)$ are increasing in their power argument p^k, it easy to see that the maximum throughput will be attained when the power vector is on the boundary of the power space $\mathbf{P}$, that is,

$$\sum_{k=1}^{K} p_*^k(I) = P_{\max}. \tag{13}$$

Moreover, for any fixed I, there is a unique optimal power vector $P_*(I)$, which maximizes the aggregate throughput, under the given interference pattern I. Assuming the interference Markov process $\{I_n\}$ is in equilibrium at state I with probability $\phi(I)$ (stationary distribution), the average packet throughput in equilibrium will be

$$\overline{T} = \sum_{I \in \mathbf{I}} T(P, I)\phi(I), \tag{14}$$

for any given power vector. It should be noted that by choosing the optimal power vector $P_*(I)$ for each interference state I, we maximize the overall average throughput in equilibrium. The calculation of the optimal power vector $P_*(I)$ can be obtained analytically for simple cases of the functions $s(p, i)$, but, in general, it would be identified numerically for each interference pattern.

To see how the above ideas apply in practice, consider the simplest possible example of two channels ($K = 2$) and that the success probability per packet transmission in each channel is given by $s(p, i) = 1 - \mathrm{e}^{-\gamma(p/i)}$. Therefore,

$$\begin{aligned} T(P, I) &= s\big(p^1, i^1\big) + s\big(p^2, i^2\big) \\ &= 1 - \mathrm{e}^{-\gamma p^1/i^1} + 1 - \mathrm{e}^{-\gamma p^2/i^2}. \end{aligned} \tag{15}$$

Given fixed (i^1, i^2), the maximum throughput is attained for $p^1 + p^2 = P_{\max}$, so setting $p^2 = P_{\max} - p^1$ in the expression for $T(P, I)$ above and differentiating over p^1, we get that $T(P, I)$ is maximized for

$$p_*^1\big(i^1, i^2\big) = \left(\frac{i^1}{i^1 + i^2}\right) P_{\max} - \frac{1}{\gamma}\left(\frac{i^1 i^2}{i^1 + i^2}\right) \log\left(\frac{i^1}{i^2}\right) \tag{16}$$

and

$$p_*^2\big(i^1, i^2\big) = \left(\frac{i^2}{i^1 + i^2}\right) P_{\max} + \frac{1}{\gamma}\left(\frac{i^1 i^2}{i^1 + i^2}\right) \log\left(\frac{i^1}{i^2}\right). \tag{17}$$

It should be noted that the optimization should really occur over the user buffers that are nonempty, since to channels for which their buffer is empty we should allocate zero power. We call the above process for allocating power to the channels *throughput maximizing power control*.

3.2. SIR-oriented power allocation

Another approach to formulate the power control problem is to attempt to maintain a required signal-to-interference ratio (SIR) in each channel [9]. For standard $s(p, i)$ functions that actually have the form $s(p/i)$, this implies that the average throughput per channel would be kept constant. Let us first assume that none of the user buffers is empty. Given the interference $I = (i^1, i^2, \dots, i^k, \dots, i^K)$ in a time-slot, we want to assign the powers so that the SIR in the kth channel is at a required level $\xi^k = p^k/i^k$, which typically reflects the quality of service level required in the channel. Therefore, $p^k = \xi^k i^k$, assuming that such powers can be supported, that is, $\sum_{k=1}^{K} p^k \leqslant P_{\max}$. If the latter is not possible the powers have to be scaled back so that they add up to $P_{\max}$. There are many ways to scale down the powers, but a reasonable one is to proportionally scale down the SIRs so that they can be supported at total power $P_{\max}$, distributing the burden proportionally across all channels. The power allocated to the kth channel when the interference is I, is simply

$$p_*^k(I) = \begin{cases} \xi^k i^k, & \text{if } \sum_{k=1}^{K} \xi^k i^k \leqslant P_{\max}, \\ \left(\dfrac{P_{\max}}{\sum_{k=1}^{K} \xi^k i^k}\right)\xi^k i^k, & \text{if } \sum_{k=1}^{K} \xi^k i^k > P_{\max}. \end{cases} \tag{18}$$

Note that in any case we have $\sum_{k=1}^{K} p_*^k(I) \leqslant P_{\max}$. An obvious modification – and improvement – of the algorithm is to allocate power only to the channels/users where the packet backlog is non-zero, so that no power is wasted in channels where there is no information packet to transmit.

4. Incorporating backlog costs at reduced model complexity

The previous power allocation algorithms take into account the interference levels in the various channels but largely ignore the packet backlog state of each user (except, for not allocating any power to empty buffers). Therefore, they are interference-sensitive but largely backlog-insensitive. As a results the system analyst and designer cannot incorporate backlog costs that may be important for high performance engineering of such systems. Indeed, in order to model and control performance tradeoffs like power versus delay and packet drop – which are critical for quality of service support and QoS-oriented system design – we need to be able to model and optimize cost structures that are both backlog and interference sensitive.

In this section we consider special 'reduced' formulations of the power allocation problem, which are both backlog and interference aware, and do capture the important performance tradeoffs for efficient system design. They are basically 'relaxed' versions of the general formulation presented previously, which trade reasonable loss of model accuracy and generality for substantial reduction of model complexity. Being fairly tractable at an appropriate level of abstraction, they provide substantial intuition and insight regarding the design and operation of such systems.

Recall the general formulation introduced in section 2, where each packet of the kth queue has a 'hard' deadline D^k (time-to-live) associated with it. Relax this modeling attribute by stretching $D^k \to \infty$ for all queues $k \in \{1, 2, 3, \dots, K\}$, hence, 'washing out' the deadline effect on the cost structure of the formulation, since no packet will ever expire.

Under this model relaxation, the backlog state (of the reduced model) becomes

$$Q_n = \{q_n^1, q_n^2, \dots, q_n^k, \dots, q_n^K\}, \tag{19}$$

where q_n^k is the number of packets in the kth buffer during the nth time slot. It is easy to see that (Q_n, I_n) is a Markov chain controlled by $P \in \mathbf{P}$. The transition probabilities can be easily computed given the control P. Indeed, note that q^k switches to $q^k + 1$ with probability $a^k(1 - s^k(p^k, i^k))$ in general, corresponding to a packet arrival and an unsuccessful packet transmission. There are two special cases: when the kth buffer is full and when it is empty. When the buffer is full ($q^k = B^k$), the transition $q^k \to q^k + 1$ occurs with probability 0. When the buffer is empty, we must differentiate between the system operating in a store-and-forward mode (i.e., a packet must first be fully received in a slot before in can be transmitted in the next one) or cut-through mode (a packet can be received and transmitted in the same slot). When the buffer is empty ($q^k = 0$), the transition $q^k \to q^k + 1$ occurs with probability a^k in the store-and-forward mode, and with probability $a^k(1 - s^k(p^k, i^k))$ in the cut-through mode. Similarly, the transition $q^k \to q^k - 1$ occurs with probability $(1 - a^k)s^k(p^k, i^k)$ in general, corresponding to no arrival of a new packet in the current time slot and departure of one due to successful transmission. When the buffer is empty ($q^k = 0$), the previous transition occurs with probability 0. Since all random events occur independently of each other, say, by flipping independent coins, we can analogously compute the probabilities of all the possible system state transition and denote them by

$$\begin{aligned} \operatorname{Prob}\big[(Q_{n+1}, I_{n+1}) = (Q', I') \mid (Q_n, I_n) = (Q, I); \\ P_n = P\big] = \Gamma_P\{(Q, I), (Q', I')\} \end{aligned} \tag{20}$$

for fixed power $P \in \mathbf{P}$ and any $(Q, I), (Q', I') \in \mathbf{Q} \times \mathbf{I}$, where $\mathbf{Q}$ is the set of all possible backlog states in this case.

The cost incurred in each time slot is the total power transmitted and a holding/delay cost for the packets in the buffer. Let $C_H(Q)$ be the holding cost (positive real number) incurred in a time slot when the backlog state is Q. For example, we could take $C_H(Q) = \sum_{k=1}^{K} c_H^k q^k$ to be a linear cost, weighted with positive factors c_H^k; if the latter are all equal to one, the cost is simply the total backlog of packets in the system. Assume that we run the system for N steps (time slots). Let $J_n(Q, I)$ be the cost-to-go at time n, that is the minimum expected total cost to be incurred from time n to N, given that the system starts from state (Q, I) at time

n and is operated under the optimal power control strategy. The dynamic programming equation then becomes

$$J_n(Q, I) = \min_{P \in \mathbf{P}} \Bigg\{ \sum_{k=1}^{K} p^k + C_H(Q) + \sum_{(Q', I') \in \mathbf{Q} \times \mathbf{I}} \Gamma_P \big\{(Q, I), \big(Q', I'\big)\big\} \times J_{n+1}\big(Q', I'\big) \Bigg\} \tag{21}$$

for $n \in \{1, 2, \ldots, n, \ldots, N\}$. The minimization over $\mathbf{P} = \{(p^1, \ldots, p^k, \ldots, p^K)\colon \sum_{k=1}^{K} p^k \leqslant P_{\max}\}$ guarantees that global power ceiling is not exceeded. At the boundary of $\mathbf{Q}$, where at least one buffer is full, there is the possibility of a packet overflow. For such a boundary state the above general dynamic programming equation should be amended to include an overflow cost incurred when a packet arrives to a full queue and has to be dropped. We could also include some terminal cost associated with the packets that remain in the buffer at the end of the N step horizon. Solving the dynamic programming equation recursively we can get the optimal power control $P_n^*(Q, I)$, $n \in \{1, 2, \ldots, n, \ldots, N\}$, for $(Q, I) \in \mathbf{Q} \times \mathbf{I}$.

The complexity of the backlog state Q (e.g., number of states in $\mathbf{Q}$) is far smaller than that of the more refined backlog state R (e.g., number of states in $\mathbf{R}$) of the model presented in the previous section. In that respect, under this formulation the problem is substantially more tractable than before. However, despite the reduction achieved by 'relaxing' the general formulation to the current one, the complexity is still very high. This is due to two reasons. First, we need to compute the dynamic programming recursion over an long horizon N (undetermined) to obtain the optimal power control in equilibrium. More importantly, however, the state space structure is inherently very complex. To dimension the complexity, consider the simple example where there are 10 queues (channels), each having 10 buffer places. Then $\mathbf{Q}$ has 10^{10} states! Considering that they exist 10 interference levels per channel, I has 10^{10} states across all 10 channels! Finally, if there are 10 power levels (discretized) per channel, the number of states in $\mathbf{P}$ is again 10^{10}. This is a huge space, indeed, over which we need to minimize! One can explore various decomposition and approximation methods or heuristics to handle the state space explosion. To suppress the complexity we consider below a further 'relaxed' formulation which now brings the problem into the domain of realistic implementation.

5. Decomposition to individual channels and power balancing

The main source of model complexity above is the 'entanglement' of the different channels, which occurs through the global power ceiling that has to be observed jointly across all channels. This 'entanglement' leads to a multiplicative (geometric) explosion of the number of states in the complete multi-channel system, as the number of channels increases.

In order to reduce the complexity down to practical implementation levels, we follow two methodological steps. In the *first step*, we artificially decompose the system into its K individual channels, by removing the global power constraint (relaxing $P_{\max} \to \infty$) and decoupling the interference states of the various channels. This results in the channels being disentangled, so that each one can be studied independently of the others. Hence, the problem is drastically reduced to K similar ones. We then study the power control problem for each channel obtaining the optimal power to transmit in each time slot, given the individual backlog and interference states of the specific channel under consideration. Under this analysis, the sum of the individually optimal powers obtained may exceed $P_{\max}$. In the *second step*, if the sum of computed powers exceeds $P_{\max}$, we scale down the powers appropriately by load balancing them across the channels. That way the channels get again entangled through the power constraint (at a more superficial level now of course). The key benefit is that, although the state complexity in the original formulation grows geometrically with the number of channels K, it grows linearly in the reduced one based on decomposition to the individual queues.

Let us now consider the analysis of one queue, decoupled from the others. Dropping the superscript k used previously to index the queues and channels, the notation is reduced as follows. At time slot n, the number of packets in the queue is q_n, the channel interference level is i_n, the power transmitted is p_n, the probability of a packet arrival is a, the probability of successful packet transmission is $s(p_n, i_n)$, and the buffer size is B. Let the interference switch between different states according to the Markov chain with transition probabilities

$$\mathrm{Prob}\big[i_{n+1} = i' \mid i_n = i\big] = F\big(i, i'\big) \tag{22}$$

for all n's and $i, i' \in \mathcal{I}$, where $\mathcal{I}$ is the set of all states that the interference can attain in the channel.

Each packet has a deadline D, and let r_n ($< D$) be the residual lifetime of the front (first) packet in the first-come-first-served queue. This is the time-to-live of the packet under transmission, until its deadline expires and is dropped from the buffer incurring some drop cost discussed below. For simplicity assume that B is larger than D, therefore there are no buffer overflows and packets can only be dropped because of deadline expiration. We study the problem as follows, including the possibility of having packet deadlines. Our main criterion is simplicity that will allow practical implementation. Suppose that at time some time n, the backlog is $q_n = q$, the interference is $i_n = i$.

We formulate the following dynamic program over the horizon 0 to D, which captures the evolution of the system until the terminal event of the head packet in the queue being successfully transmitted before D or otherwise expiring at $D+1$ and being dropped from the buffer.

$$U_m(q,i) = \min_p \Big\{ p + C_h(q) + \big[1 - s(p,i)\big] \times \Big(a \sum_{i' \in \mathcal{I}} F\big(i,i'\big) U_{m+1}\big(q+1, i'\big) + (1-a) \sum_{i' \in \mathcal{I}} F\big(i,i'\big) U_{m+1}\big(q, i'\big) \Big) \Big\}. \quad (23)$$

The variable $U_m(q,i)$ is the cost-to-go in this context, that is, the minimal average cost incurred until the head packet gets successfully transmitted or dropped from the buffer because its deadline expires. In every time slot, there is a power cost p incurred (that is, the power used to transmit the packet), as well as a holding (delay) cost $C_h(q)$, which is an increasing function of the backlog size q – for example, it could be a linear cost $c_h q$ or a quadratic one $c_h q^2$. If the head packet has not been transmitted successfully by time D, then it is dropped from the buffer at time $D+1$ and a drop cost $C_d(q')$ is incurred, which is an increasing function of the number of packets q' left in the buffer when the head one is dropped.

The formulation of the drop cost is important, as it has to capture the 'deadline pressure' and 'expiration risk' of the other packets in the buffer. To elaborate on that, consider that the system incurs a cost c_d every time that a packet is dropped due to its deadline expiring. Then, structure the terminal drop cost of the dynamic program as

$$C_d\big(q'\big) = c_d + c_d f\big(q'\big) = c_d\big[1 + f\big(q'\big)\big]. \quad (24)$$

The first term reflects the cost of the head packet that was lost and the second one the cost associated with potential losses of other packets in the buffer because of their deadlines expiring, since the head one had been blocking them for two long. Thus, $f(q')$ reflects the 'pressure' that the other packets in the buffer put on the head one. Of course, $f(q')$ should be an increasing function of q' (the backlog of packets in the buffer when the head one is dropped). Indeed, the risk that more packets are dropped later increases as more packets accumulate in the buffer and are delayed, 'spending out' their time-to-live horizons.

It turns out that the minimization in equation (23) can be obtained analytically. Indeed, we can rewrite the equation as:

$$U_m(q,i) = \min_p \big\{ p - s(p,i) X_m(q,i) + Y_m(q,i) \big\}, \quad (25)$$

where

$$X_m(q,i) = a \sum_{i' \in \mathcal{I}} F\big(i,i'\big) U_{m+1}\big(q+1, i'\big) + (1-a) \sum_{i' \in \mathcal{I}} F\big(i,i'\big) U_{m+1}\big(q, i'\big) \quad (26)$$

and

$$Y_m(q,i) = X_m(q,i) + C_h(q). \quad (27)$$

It is easy to see that for fixed i, the expression in (25) is minimized (recalling that $s(p,i)$ is increasing in p) for $p_*(i,q,m)$ such that

$$\left.\frac{\partial s(p,i)}{\partial p}\right|_{p_*} = \frac{1}{X_m(q,i)}. \quad (28)$$

For example, for

$$s(p,i) = \frac{p/i}{\alpha + \beta p/i} = \frac{p}{\alpha i + \beta p},$$

we get

$$p_*(i,q,m) = \begin{cases} \dfrac{1}{\beta}\big(\sqrt{\alpha i X_m(q,i)} - \alpha i\big), & i < \dfrac{X_m(q,i)}{\alpha}, \\ 0, & i \geqslant \dfrac{X_m(q,i)}{\alpha}. \end{cases} \quad (29)$$

Similarly, for $s(p,i) = 1 - \mathrm{e}^{-\gamma(p/i)}$, we get

$$p_*(i,q,m) = \begin{cases} -\dfrac{i}{\gamma} \log \dfrac{i}{\gamma X_m(q,i)}, & i < \gamma X_m(q,i), \\ 0, & i \geqslant \gamma X_m(q,i). \end{cases} \quad (30)$$

It is easy to see that the dynamic programming recursion (23), (25) can be solved backwards explicitly from $m = D$ to $m = 1$, by recursive substitution. Actually, note that given the terminal cost $C_d(q)$ to be incurred if the front packet is dropped, we have

$$U_D(q,i) = \min_p \big\{ p + C_h(q) + \big[1 - s(p,i)\big] \times \big(aC_d(q) + (1-a)C_d(q-1)\big)\big\}. \quad (31)$$

Therefore, $X_D(q,i) = aC_d(q) + (1-a)C_d(q-1)$ and $Y_D(q,i) = X_D(q,i) + C_h(q)$. Solving for all $m = 1, 2, 3, \ldots, D$, we get the optimal power $p_*(i,q,m)$ to transmit at step m, given that the interference is i and the backlog q. Note that the complexity now is manageable, since for a system with 10 interference levels and a deadline of 10 steps per packet (hence, we would never have more than 10 packets in the buffer), the number of $p_*(i,q,m)$ values is $10 \times 10 \times 10$ or 1000.

Given the previous analysis, the power to be transmitted in a channel (decoupled from the others) during time slot n is simply $p_*(i_n, q_n, D - r_n)$, where i_n is the interference in that slot, q_n the backlog, and $D - r_n$ is the index m used above to mark how far advanced the process of transmission of the head packet is towards its deadline horizon.

Consider now the system of K channels with the power ceiling $P_{\max}$. If the power ceiling is not violated, we can transmit power $p_*^k(i_n^k, q_n^k, D^k - r_n^k)$ in the kth channel, where i_n^k is the interference in it, q_n^k the backlog, and r_n^k the residual lifetime of the head packet in the queue. The power ceiling not being violated means that $\sum_{k=1}^K p_*^k(i_n^k, q_n^k, D^k - r_n^k) \leqslant P_{\max}$. In that case, the channels operate as if they were totally decoupled. However, if $\sum_{k=1}^K p_*^k(i_n^k, q_n^k, D - r_n^k) > P_{\max}$, then the powers need to be scaled down and load balanced to observe the power ceiling collectively. One way is to scale them down proportionally, that is:

$$p_*^k(i_n^k, q_n^k, D^k - r_n^k) = \begin{cases} p_*^k\big(i_n^k, q_n^k, D^k - r_n^k\big), & \text{if } pk \leqslant P_{\max} \\ \left(\dfrac{p_*^k(i_n^k, q_n^k, D^k - r_n^k)}{\sum_{k=1}^K p_*^k(i_n^k, q_n^k, D^k - r_n^k)}\right) P_{\max}, & \text{if } pk > P_{\max} \end{cases} \tag{32}$$

where $pk = \sum_{k=1}^K p_*^k(i_n^k, q_n^k, D^k - r_n^k)$. There are other ways to scale down the powers to observe the power ceiling, for example, using weights that are increasing functions of the backlog.

It should be noted that the new 'entanglement' of the channels, brought about by rebalancing the individual powers to observe the power ceiling, couples again the channels which had been decoupled initially. However, the approach taken here has substantially reduced complexity and the previous algorithm is now easily implementable.

6. Conclusions

We have studied the issue of power control across several channels and under a maximum total power constraint. The problem has been formulated within a very general dynamic programming framework. Despite the insight the general formulation provides, it is too general for practical implementation. Therefore, through selective model reductions we have designed novel power control algorithms that can be practically implemented.

References

[1] M. Andersin, Z. Rosberg and J. Zander, Gradual removals in cellular PCS with constrained power control and noise, Wireless Networks 2(1) (1996) 27–43.

[2] M. Andersin, Z. Rosberg and J. Zander, Soft and safe admission control in cellular networks, IEEM/ACM Transactions on Networking 5(2) (1997) 255–265.

[3] N. Bambos, Toward power-sensitive network architectures in wireless communications: concepts, issues and design aspects, IEEE Personal Communications Magazine 5(3) (1998) 50–59.

[4] N. Bambos, S.C. Chen and G.J. Pottie, Radio link admission algorithms for wireless networks with power control and active link quality protection, in: *Proc. of the IEEE Infocom* (1995).

[5] N. Bambos, S.C. Chen and G.J. Pottie, Radio link admission algorithms for wireless networks with power control and active link protection, IEEE/ACM Transactions in Networking (2000) to appear.

[6] N. Bambos and S. Kandukuri, Power controlled multiple access (PCMA) in wireless communication networks, in: *Infocom 2000*, Tel Aviv, Israel (2000).

[7] D. Bertsekas, *Dynamic Programming* (Prentice-Hall, 1987).

[8] B.E. Collins and R.L. Cruz, Transmission policies for time varying channels with average delay constraints, in: *Proc. of Allerton Conference on Communications, Control and Computers*, Monticello, IL (1999).

[9] G.J. Foschini and Z. Miljanic, A simple distributed autonomous power control algorithm and its convergence, IEEE Transactions on Vehicular Technology 42(4) (1993) 641–646.

[10] G.J. Foschini and Z. Miljanic, Distributed autonomous wireless channel algorithm with power control, IEEE Transactions on Vehicular Technology 44(3)(August 1995) 420–429.

[11] S. Hanly, Capacity and power control in a spread spectrum macro diversity radio networks, IEEE Transactions on Communications 44 (1996) 247–256.

[12] D. Mitra, An asynchronous distributed algorithm for power control in cellular radio systems, in: *Proc. of 4th WINLAB Workshop*, Rutgers University, NJ (1993).

[13] T.S. Rappaport, *Wireless Communications, Principles and Practice* (Prentice-Hall, 1996).

[14] S. Ross, *Introduction to Stochastic Dynamic Programming* (Academic Press, 1983).

[15] J. Rulnick and N. Bambos, Mobile power management for wireless communication networks, Wireless Networks 3 (1997).

[16] H. Taub and D. Schilling, *Principles of Communication Systems* (McGraw Hill).

[17] J. Whitaker, *Radio Frequency Transmission Systems* (Intertext Publications, McGraw Hill Book Series).

[18] R. Yates, A framework for uplink power control in cellular radio systems, IEEE Journal on Selected Areas in Communications 13(7) (1995) 1341–1347.

[19] R. Yates and C.Y. Huang, Integrated power control and base station assignment, IEEE Transactions on Vehicular Technology 44(3) (1995) 638–644.

[20] J. Zander, Distributed cochannel interference control in cellular radio systems, IEEE Transactions on Vehicular Technology 41(3) (1992) 305–311.

Nick Bambos received his Ph.D. degree in electrical engineering and computer science in 1989 from the University of California at Berkeley. From 1989 to 1995, he was an Assistant and later Associate Professor in the Electrical Engineering Department of the University of California at Los Angeles. Since 1996, he has been Associate Professor at Stanford University in the Department of Electrical Engineering and the Department of Management Science Engineering. His current research interests are in modeling, analysis and design of communication networks and distributed systems.
E-mail: bambos@stanford.edu

Sunil Kandukuri received the Bachelor of Engineering (B.E.) degree from Andhra University, India, in 1995, and the Master of Science in Electrical Engineering (M.S.E.E.) from Stanford University, USA, in 1998. He is currently a Ph.D. candidate at Stanford University. His current research interests include wireless communications, ad hoc networks, optimization, and stochastic modeling.
E-mail: sunilkn@stanford.edu