Proportionally Fair Allocation of End-to-End Bandwidth in STDMA Wireless Networks

Pablo Soldati, Björn Johansson and Mikael Johansson
School of Electrical Engineering, KTH
SE-100 44 Stockholm, Sweden
{firstname.lastname}@ee.kth.se

ABSTRACT

We consider the problem of designing distributed mechanisms for joint congestion control and resource allocation in spatial-reuse TDMA wireless networks. The design problem is posed as a utility maximization subject to link rate constraints that involve both power allocation and transmission scheduling over multiple time-slots. Starting from the performance limits of a centralized optimization based on global network information, we proceed systematically in the development of distributed and transparent protocols. In the process, we introduce a novel decomposition method for convex optimization, establish its convergence for the utility maximization problem and demonstrate how it suggests a distributed solution based on flow control optimization and incremental updates of the transmission schedule. We develop a two-step procedure for finding the schedule updates and suggest two schemes for distributed channel reservation and power control under realistic interference models. Although the final protocols are suboptimal, we isolate and quantify the performance losses incurred by each simplification and demonstrate strong performance in examples.

Categories and Subject Descriptors: C.2.1 [Computer-Communication Networks]: Network Architecture and Design; C.2.2 [Computer-Communication Networks]: Network Protocols.

General Terms: Algorithms, Theory.

Keywords: Ad hoc wireless networks, cross-layer design, congestion control, convex optimization, mathematical decomposition.

1. INTRODUCTION

Although the research on multi-hop wireless networks has a long history [10, 11, 8], with some operational systems already on the market [3], emerging standards such as 802.11s and 802.16 have the potential of allowing wireless networking on a broader scale. The 802.16 WiMAX standard, for example, defines the basic functionalities for establishing mesh networks under centralized or decentralized control. Such networks could potentially be used to create flexible and rapidly deployable wide-area wireless backhaul networks, offering increased wireless coverage at relatively low cost.

A critical problem for traffic engineering in these systems, and in multi-hop wireless networks in general, is the allocation of resources, such as transmission opportunities, radio channels, and transmit powers to links so as to meet performance objectives for the end-to-end transport rates. Recently, a wide variety of optimization methods have been suggested for computing the achievable performance of systems that coordinate multiple layers in the networking stack, see *e.g.* [16] and the references therein. These centralized schemes give insight into the performance benefits of coordinating the different layers of the protocol stack, but are quite far from the distributed resource management protocols needed in practice. Centralized solutions tend to incur large signalling overhead, introduce a single point-of-failure (the network control node) and scale poorly with the number of network nodes. Moreover, many of the centralized schemes are computationally demanding to execute.

Scheduling in wireless networks could either be static, dynamic or a combination thereof. In static scheduling, transmission opportunities are assigned for long durations relative to channel switching time. Typically, these schemes build up a transmission schedule which allows different transmitters access rights in different time slots. This traditional way of operating wireless multi-hop networks (*e.g.*, [3]) is relatively easy to implement and requires moderate signalling overhead. Dynamic allocation schemes, on the other hand, update transmission rights very frequently. These schemes have the potential of allowing the physical layer to be opportunistic and better exploit fading variations. The drawback is that interference-sensitive scheduling in multi-hop networks requires significant signalling overhead and may be hard to execute on a fast time-scale.

Focusing on static scheduling, this paper presents a distributed approach to joint end-to-end bandwidth sharing, transmission schedule construction, and power control in spatial-reuse TDMA (STDMA) wireless networks. Our approach falls into the framework of network utility maximization (*e.g.* [22, 6]), since we formulate the optimal network operation as the solution of an optimization problem and apply mathematical decomposition techniques to find distributed solution mechanisms. As noted by several researchers (*e.g.*, [28, 6, 5]), applying mathematical decomposition techniques to wireless cross-layer optimization problems will typically decompose the problem into three

MobiHoc'06, May 22–25, 2006, Florence, Italy.

subproblems: one source-rate problem to be executed at end-nodes (which can be mapped onto the TCP algorithm) one link-pricing algorithm to be executed by routers (corresponding to an active queue management scheme) and one resource allocation subproblem. Whether the resource allocation problem can be solved distributively or not depends on the channel model and the structure of the resource constraints. Distributed solutions have been found for orthogonal channels with local [28] and network-wide [15] resource constraints, interference-limited channels in the high SINR-regime [6] and contention-based models [29, 5]. All these methods perform a one-shot resource allocation based on the current network state. None of them maintains a transmission schedule with multiple time-slots.

Our development of distributed protocols relies on two main technical contributions. First, noting that the standard approach of dual decomposition can not be used to construct multiple time-slot schedules, we devise a novel mathematical decomposition method that allow us to incrementally construct a schedule that provably converges to the optimal. Based on this decomposition scheme, we derive the basic functionalities that needs to be implemented in sources, queues, and transmitters. The suggested protocols rely on end-to-end bandwidth sharing for a given schedule on a fast time scale and incremental schedule updates based on congestion levels on a slower time-scale. The second technical contribution is a two-step procedure for solving the congestion-weighted resource allocation problem under realistic interference models, including two novel schemes for distributed power control and channel reservation. We demonstrate how these schemes improve significantly over other approaches proposed in the literature. On the methodological side, we proceed systematically from the performance limitations given by a centralized optimization based on global information to two totally distributed, but suboptimal, schemes that rely on local information only. We isolate the performance losses incurred in each simplification step and demonstrate strong performance improvements over existing approaches.

The paper is organized as follows: Section 2 describes the our network model; Section 3 reviews the duality model for TCP due to Low and Lapsley [22] and a centralized cross-layer optimization for STDMA networks by Johansson and Xiao [16]. Section 4 introduces the cross-decomposition approach and establishes its convergence, while Section 5 describes two distributed solutions to joint power control and transmission scheduling for optimizing congestion-weighted throughput. Section 6 discusses how the modules can be implemented and combined into protocols for joint congestion control and resource allocation. The proposed solutions are evaluated and compared on a set of sample networks in Section 7. Finally, Section 8 concludes the paper.

2. NETWORK MODEL

We consider a network formed by a set of nodes located at fixed positions in the plane. Each node is assumed to have infinite buffering capacity and can transmit, receive and relay data to other nodes over wireless links. The network performance depends on the interplay between end-to-end rate selection, power control, and transmission scheduling. A model for these dependencies will be developed next.

2.1 Network topology and network flows

We represent the network topology by a directed graph with nodes labelled $n = 1, \ldots, N$ and links labelled $l = 1, \ldots, L$. A link is represented by an ordered pair (i, j) of distinct nodes. The presence of link (i, j) means that the network is able to send data from node i to node j.

Nodes are assumed to always have data to send to the other nodes, possibly via multi-hop routing. We label the source-destination pairs by integers $p = 1, \ldots, P$ and let s_p denote the end-to-end rate for communication between source-destination pair p. Associated with each pair p is a utility function $u_p(\cdot)$, which describes the utility of the pair to communicate at rate s_p (see [17] for further details). We assume that u_p is increasing and strictly concave, with $u_p \to -\infty$ as $s_p \to 0^+$. Data is assumed to be routed along a single fixed path. The routing can then be specified by a *routing matrix* $R = [r_{lp}] \in \mathbb{R}^{L\times P}$, with entries

$$r_{lp} = \begin{cases} 1 & \text{if data between pair } p \text{ is routed across link } l \\ 0 & \text{otherwise} \end{cases}$$

Letting c_l denote the transmission rate of link l, $c = [c_l]$ be the vector of link-rates, the vector of total traffic across links is given by Rs and the network flow model imposes the following set of constraints on the end-to-end rate vector s

$$Rs \preceq c \qquad\qquad s \succeq 0$$

2.2 Communications Model

The link rates depend on the medium access scheme, channel conditions and the allocation of radio resources, such as transmit powers and time-slots, to the transmitters. In this paper, we consider a system where all links share the same frequency band, so that interference will occur when multiple links transmit simultaneously. Moreover, we will assume that nodes are equipped with omnidirectional antennas and no multi-user detectors, so that nodes are limited to communicate with at most one other node at a time.

Let G_{lm} be the direct link gain between the transmitter of link m and the receiver of link l. We use the deterministic fading model $G_{lm} = \beta_{lm} d_{lm}^{-\upsilon}$, where d_{lm} is the physical distance between the transmitter of link m and the receiver of link l, υ is a constant path loss exponent whose value depends on the compliance of territory, and β_{lm} is a normalization constant which reflects the radio propagation properties of the environment, but also allows us to account for the effects of coding and spreading gain (in a CDMA setting), and beam-forming, see *e.g.*, [26, 16].

Let σ_l be the thermal noise power at the receiver of link l, let P_l be the power used by its own transmitter and define the signal to noise and interference ratio (SINR) of link l as

$$\gamma_l(P) = \frac{G_{ll}P_l}{\sigma_l + \sum_{m\neq l} G_{lm}P_m} \tag{1}$$

We view each link as a single-user Gaussian channel with Shannon capacity $c_l(P) = W \log(1+\gamma_l(P))$, where W is the system bandwidth. We assume that transmitters are subject to simple power limits, $0 \leq P_l \leq P_{\max}$ and that links offer a single rate $r_l^{\mathrm{tgt}} = W \log(1 + \gamma^{tgt})$ when the SINR at its receiver exceeds the threshold γ^{tgt}. To be able to see this effect we introduce the following rate allocation policy

$$r_l = \begin{cases} r^{\mathrm{tgt}}, & \text{if} \quad \gamma_l(P) \geq \gamma^{tgt} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Thus, link l may transmit at rate r^{tgt} if the SINR level at its receiver exceed the threshold, and stays silent otherwise. This leads to a finite number of achievable link-rate vectors

$$c^{(k)} = (r_1 \dots r_L) \qquad k = 1, \dots, K$$

where $r_l \in \{0, r^{\mathrm{tgt}}\}$. Although K may be as large as 2^L, interference and the restriction that nodes can communicate with at most one other node at a time drastically reduces the number of link-rate vectors that can be supported. By time-sharing between the achievable link-rate vectors, we can achieve the following polyhedral rate region

$$\mathcal{C} = \left\{ c = \sum \alpha_k c^{(k)} \mid \alpha_k \geq 0,\ \sum_k \alpha_k = 1,\ k = 1, \dots, K \right\}$$

Here, the *time-sharing coefficients* α_k represent the fraction of schedule in which rate vector $c^{(k)}$ is activated. In the following, we will use the short-hand notation $c \in \mathcal{C}$ to denote that c is an achievable long-term average link rate.

3. JOINT CONGESTION CONTROL AND RESOURCE ALLOCATION

3.1 Optimization flow control

A mathematical formulation of a distributed end-to-end flow control scheme over TCP/IP in wired networks has been developed in [17, 22]. It is argued that the optimal network operation solves the network utility maximization problem

$$\begin{array}{ll} \text{maximize} & \sum_p u_p(s_p) \\ \text{subject to} & Rs \preceq c \\ & s \succeq 0 \end{array} \qquad (3)$$

where the variables are collected in the end-to-end rate vector s, while the link capacity vector is assumed to be fixed. If $u_p(\cdot) = \log(\cdot)$, the optimal solution yields a proportionally fair allocation of end-to-end bandwidth [17].

A distributed solution to this problem can be derived via dual decomposition. Introducing Lagrange multipliers λ for the capacity constraints we can form the Lagrangian

$$\begin{aligned} L(s, \lambda) &= \sum_p u_p(s_p) + \sum_l \lambda_l (c_l - \sum_p r_{lp} s_p) = \\ &= \sum_p u_p(s_p) - q_p s_p + \sum_l \lambda_l c_l \end{aligned}$$

where $q_p = \sum_l r_{lp} \lambda_l$ can be interpreted as the total congestion price along route p. Since the Lagrangian is separable in the end-to-end rates s_p, the dual function

$$g(\lambda) = \sup_{s \succeq 0} L(s, \lambda)$$

can be evaluated by letting sources optimize their rates individually based on the total congestion price, *i.e.*, by letting

$$s_p = \arg\max_{z \geq 0} u_p(z) - q_p z \qquad (4)$$

Moreover, the dual problem to (3)

$$\begin{array}{ll} \text{minimize} & g(\lambda) \\ \text{subject to} & \lambda \succeq 0 \end{array}$$

can be solved by the projected gradient iteration

$$\lambda_l^{(k+1)} = \left[\lambda_l^{(k)} + \omega^{(k)} \left(\sum_p r_{lp} s_p^{(k)} - c_l \right) \right]^+$$

where $\{\omega^{(k)}\}$ is a step length sequence and $[\cdot]^+$ denotes projection onto the positive orthant. Note that links can update their congestion prices based on local information: if the traffic across link l exceeds capacity, the congestion price increases; otherwise it decreases. Convergence of the dual algorithm has been established in [22]. The distributed updates of end-to-end rates and link prices can be mapped onto the idealized operations of TCP clients and AQM schemes, and it has been argued that running the appropriate TCP and AQM protocols effectively amounts to letting the network solve the utility maximization problem [22].

3.2 A cross-layer optimization formulation

Our interest is on distributed end-to-end flow control over wireless networks, where the links capacities are not fixed a priori, but depend in a non-trivial way on both MAC scheme and the resource allocation. In the spirit of (3), the associated utility maximization problem can be formulated as

$$\begin{array}{lll} \text{maximize} & \sum_p u_p(s_p) & \\ \text{subject to} & Rs \preceq c & \\ & c \in \mathcal{C} & s \succeq 0 \end{array} \qquad (5)$$

A centralized solution to this problem, combining dual decomposition with a column generation procedure, has been proposed in [16]. Since this procedure will be the basis for our decentralized approach, we will review it briefly below.

Since $\mathcal{C}$ is a convex polyhedron, we can re-write (5) as

$$\begin{array}{lll} \text{maximize} & \sum_p u_p(s_p) & \\ \text{subject to} & Rs \preceq c & \\ & c = \sum_{k=1}^K \alpha_k c^{(k)} & s \succeq 0 \\ & \sum_{k=1}^K \alpha_k = 1 & \alpha \succeq 0 \end{array} \qquad (6)$$

We refer to this optimization problem in variables s and α as the *full master problem*. Although this is a convex optimization problem (maximizing a concave function subject to linear constraints), this formulation is inconvenient for several reasons. First, $\mathcal{C}$ may have a very large number of vertices so explicit enumeration of these quickly becomes intractable as the size of the network grows. Second, even when explicit enumeration is possible, (6) may have a large number of variables and can be computationally expensive to solve directly. To this end, consider a subset $\mathcal{C}^{\mathcal{K}} = \{c^{(k)} \mid k \in \mathcal{K}\}$ of extreme points of $\mathcal{C}$, where $\mathcal{K} \subseteq \{1, \dots, K\}$,

$$\begin{array}{lll} \text{maximize} & \sum_p u_p(s_p) & \\ \text{subject to} & Rs \preceq c & \\ & c = \sum_{k \in \mathcal{K}} \alpha_k c^{(k)} & s \succeq 0 \\ & \sum_{k \in \mathcal{K}} \alpha_k = 1 & \alpha_k \geq 0, \quad k \in \mathcal{K} \end{array} \qquad (7)$$

We call (7) the *restricted master problem*, since this formulation optimizes over the restricted set of rate vectors $c \in \mathcal{C}^{\mathcal{K}}$ where $\mathcal{C}^{\mathcal{K}} \subseteq \mathcal{C}$. The problem (7) is a restriction of the full problem (5) and its solution provides a lower bound u_{lower} of the optimal value u_{opt} of the full problem. The column generation method allows us to sequentially improve the lower bound by adding a new extreme point to $\mathcal{C}^{\mathcal{K}}$, so that this set gradually becomes a better approximation of (the relevant parts of) $\mathcal{C}$, and u_{lower} converges to u_{opt}.

The key for updating $\mathcal{C}^{\mathcal{K}}$ is the Lagrange duality. Let λ be the vector of Lagrange multipliers for the capacity constraints $Rs \preceq c$, and consider the Lagrangian

$$L(s, c, \lambda) = \sum_p u_p(s_p) - q_p s_p + \sum_l \lambda_l c_l$$

where, as in the previous section, $q_p = \sum_l r_{lp}\lambda_l$. Let $(s^\star, \alpha^\star)$ be an optimal solution to (7), so that $c^\star = \sum_{k\in\mathcal{K}} \alpha_k^\star c^{(k)}$, and let $\lambda^\star$ be the associated Lagrange multipliers for the capacity constraints. Since the constraints in (7) are affine, feasibility of the restricted master problem implies strong duality [4, 16] and that u_{lower} can be expressed as

$$u_{\text{lower}} = L(s^\star, c^\star, \lambda^\star) = \sum_p \sup_{s_p\geq 0} u_p(s_p) - q_p^\star s_p + \sup_{c\in\mathcal{C}^{\mathcal{K}}} \sum_l \lambda_l^\star c_l$$

with $q_p^\star = \sum_l r_{lp}\lambda_l^\star$. We can estimate how far u_{lower} is from optimality by generating a corresponding upper bound. This can be done by considering the dual formulation of (5). More specifically, by weak duality, for any $\lambda \succeq 0$ the value

$$g(\lambda) = \sup_{s\succeq 0, c\in\mathcal{C}(P)} L(s,c,\lambda) = \sum_p \sup_{s_p\geq 0} u_p(s_p) - q_p s_p + \sup_{c\in\mathcal{C}} \sum_l \lambda_l c_l$$

provides an upper bound to u_{opt}. In particular, we can consider the bound due to $\lambda^\star$

$$u_{\text{upper}} = \sum_p \sup_{s_p\geq 0} u_p(s_p) - q_p^\star s + \sup_{c\in\mathcal{C}} \sum_l \lambda_l^\star c_l \qquad (8)$$

The difference $u_{\text{upper}} - u_{\text{lower}}$ serves as a measure of the accuracy of the current solution. We consider $(s^\star, \alpha^\star)$ to be the optimal solution of (6) if the difference drops below a predefined threshold. If the current solution does not satisfy the stopping criterion, we can conclude that $\mathcal{C}$ is not well enough characterized by the extreme points in $\mathcal{C}^{\mathcal{K}}$, and a new vertex $c^{(k)}$ must be added before the procedure is repeated.

When adding a new extreme point, we would like to choose a vertex of $\mathcal{C}$ which allows u_{lower} to improve as much as possible. If we view u_{lower} as a function of c, $\lambda^\star$ is a subgradient of u_{lower} at $c^\star$ [4, 16], so it is natural to add the extreme point that solves

$$\begin{array}{ll} \text{maximize} & \lambda^{\star\mathrm{T}} c \\ \text{subject to} & c \in \mathcal{C} \end{array} \qquad (9)$$

We will call this problem the *scheduling subproblem*. Note that this problem is solved at MAC level, and that each extreme point $c^{(k)}$ corresponds to set of links that can transmit simultaneously. Besides, the remaining part of equation (8)

$$\begin{array}{ll} \text{maximize} & \sum_p u_p(s_p) - q_p^\star s_p \\ \text{subject to} & s_p \geq 0 \end{array} \qquad (10)$$

is equivalent to the source subproblem in the optimization flow control, and thus solved on the transport level. We will call this problem the *network subproblem*.

The column generation algorithm can now be summarized as follows. Given an initial set of vertices of $\mathcal{C}$, we solve the restricted master problem to get a lower bound on the performance and the optimal Lagrange multipliers for the capacity constraints. To evaluate the upper bound, we solve the scheduling and network subproblems. If the bounds are sufficiently close, we terminate the algorithm. Otherwise, we add the vertex generated by the scheduling subproblem and start the next iteration. As shown in [16], this algorithm converges to the optimal network configuration (transmission groups, time slot fractions, transmit powers and end-to-end rates) in a finite number of steps. Moreover, an interesting property of the optimal solution is that the optimal schedule is obtained by time-sharing between at most $L+1$ transmission groups (this is a direct consequence of Carathéodory's theorem, cf. [24]).

4. VERTICAL DECOMPOSITION

There are several features of the column generation method that refrain us from making immediate use of it in solving the joint congestion control and resource allocation problem: firstly, the approach relies on a centralized solution to the restricted master problem; secondly, the schedule is assumed to have variable time-slot durations of arbitrary precision (i.e., the durations are not restricted to be integer multiples of a basic time slot length); and third, it assumes that we can solve the scheduling subproblem to optimality, which requires centralized information and is computationally hard. In this section, we will remove the two first hurdles by introducing a distributed optimization scheme that builds up the transmission schedule incrementally, adding a new transmission group in each iteration. Distributed transmission group formation will be discussed in the following sections.

It is important to understand that the classical approach of dual decomposition [22, 17] cannot be used for computing a schedule. The main reason is that the dual function

$$g(\lambda) = \sum_p \sup_{s_p\geq 0} u_p(s_p) - q_p s_p + \sup_{c\in\mathcal{C}} \sum_l \lambda_l c_l$$

is linear in c_l and will return a single transmission group in each iteration. Since each transmission group only activates a few links, many links will have zero rates until the next iteration of the algorithm can be carried out by the system and a new transmission group negotiated. Contrary to this, we will develop a distributed scheme for computing, updating and maintaining a schedule with multiple time-slots.

4.1 A Cross Decomposition Approach

An alternative approach for solving the network utility maximization problem can be developed by re-writing it as

$$\begin{array}{ll} \text{maximize} & \nu(c) \\ \text{subject to} & c \in \mathcal{C} \end{array}$$

where

$$\nu(c) = \{\max\ \textstyle\sum_p u_p(s_p) \mid Rs \preceq c,\ s \succeq 0\}$$

For a fixed link capacity vector c, the function $\nu(c)$ can be evaluated via the optimization flow control algorithm, *i.e.*, by letting the optimization flow control scheme converge. As shown in Appendix, under certain assumptions $\nu(c)$ is differentiable with respect to c with derivative λ, (the equilibrium link price vector for the network flow subproblem). Thus, in order to update the schedule and hence the link capacity vector c, it is natural to add the transmission group computed by the scheduling subproblem

$$\begin{array}{ll} \text{maximize} & \lambda^T c \\ \text{subject to} & c \in \mathcal{C} \end{array} \qquad (11)$$

to the schedule. Effectively, this corresponds to a conditional gradient step in the ascent direction of $\nu(c)$ (see Appendix). The augmented schedule is then applied to the system and the optimization flow control scheme is run until convergence before the procedure is repeated. Contrary to the

Algorithm 1 Cross Decomposition

1: Let $k = k_0$ and $\overline{c}^{(k_0)} \succ 0$.
2: **loop**
3: Evaluate $\nu(\overline{c}^{(k)})$ by solving the optimization flow control problem (3), and let $\lambda^{(k)}$ be the associated equilibrium link prices.
4: Compute a new transmission group $c^{(k+1)}$ by solving the scheduling subproblem (12) for $\lambda^{(k)}$.
5: Augment the schedule with this transmission group and compute the associated $\overline{c}^{(k+1)}$.
6: Let $k = k + 1$.
7: **end loop**

centralized approach, this scheme does not optimize time-slot durations at each iteration, but simply augments the schedule by new transmission groups. To describe the algorithm in detail, let $c^{(k)}$ be the transmission group computed in step k and let $\overline{c}^{(k)}$ denote the average link-rate vector for a schedule consisting of k time-slots of equal length, *i.e.*,

$$\overline{c}^{(k)} = \frac{1}{k}\sum_{t=1}^{k} c^{(t)}$$

For technical reasons (there must exist a $\rho > 0$ such that $\overline{c}^{(k)} \succeq \rho$ for all k) we have to add one more constraint to the scheduling problem. This modified scheduling problem is

$$\begin{array}{ll} \text{maximize} & (\lambda^{(k)})^T c^{(k)} \\ \text{subject to} & c^{(k)} \in \mathcal{C} \\ & \overline{c}^{(k)} \succeq \rho \end{array} \tag{12}$$

in the variable $c^{(k)}$. The positive constant ρ can be chosen to be arbitrarily close to zero. Simulations indicate that the the computed average link-rates $\overline{c}^{(k)}$ never tend to zero, and in practice the additional constraint on $\overline{c}^{(k)}$ appears unnecessary. Indeed, in all examples, we have used the original scheduling subproblem (11) rather than its modified variant.

Our algorithm is summarized in Algorithm 1. An initial schedule can, for example, be constructed by letting $k_0 = L$ and using a pure TDMA schedule.

This type of decomposition scheme falls into the class of *cross decomposition* methods [13]. Relatively little is known about the convergence of these methods, and no results seems to be available in the literature about the convergence of our scheme. The closest approach for which convergence has been established seems to be *mean value cross decomposition* [14]. In principle, these methods do not make use of the most recent equilibrium link prices in each iteration, but solves a scheduling subproblem based on the average link price vector over all iterations. However, as demonstrated in [12] (and confirmed below), these methods have slower convergence than our proposed scheme.

THEOREM 4.1. *Let $u^\star$ be the optimal value of the centralized cross-layer design problem (5). Algorithm 1 converges in the sense that* $\lim_{k\to\infty} \nu(\overline{c}^{(k)}) \to u^\star$.

PROOF. See Appendix. □

In contrast to the off-line algorithm presented in [16], which converges in a finite number of steps, the convergence of the cross-decomposition is only in the limit. Although we do not have any hard complexity results for the algorithm, it is closely related to Frank-Wolfe methods (*e.g.*, [2]), which require $O(1/\varepsilon)$ iterations to reach an accuracy of ε.

The practical performance of the cross decomposition approach for the sample 10-node/46-link network described in Section 7 is illustrated in Figure 1. While the centralized algorithm with variable time-slot lengths converges in a small number of iterations, the convergence of the cross-decomposition approach is only in the limit. We can also see that the suggested scheme appears to have a faster convergence than the mean-value cross-decomposition.

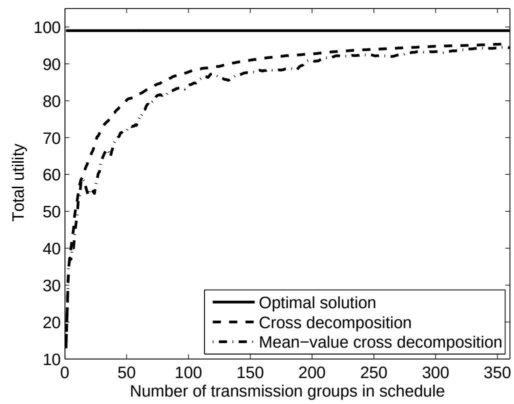

Figure 1: Lower bound trend with a large number of extreme points.

The cross decomposition method suggests the following sequential approach for solving the cross-layer design problem (5): based on an initial schedule with long-term average rate $\overline{c}^{(k_0)} \succ 0$, we run the optimization flow control until convergence (this may require us to apply the schedule repeatedly). We will refer to this as the *data transmission phase*. During a subsequent *negotiation phase* we try to find the transmission group with largest congestion-weighted throughput, and augment the schedule. The procedure is then repeated with this revised schedule. Our theoretical analysis applies to the case when we augment the schedule indefinitely, while in practice one would like to use schedules with limited frame length. We will not discuss these issues in this paper, but refer to [25] for some possible approaches.

5. PRINCIPLES OF DISTRIBUTED TRANSMISSION SCHEDULING

To make the joint congestion and resource allocation fully distributed, the scheduling subproblem must be solved without relying on a global controller and global network information; each device must be able to decide whether to join the time slot under negotiation or not, basing its decision on local information only. Since the scheduling subproblem

$$\begin{array}{ll} \text{maximize} & \sum_l \lambda_l c_l \\ \text{subject to} & c \in \mathcal{C} \end{array}$$

is linear in c_l, an optimal solution can always be found as a vertex of the capacity region, and solving the scheduling subproblem thus amounts to finding the most advantageous transmission group (and the associated power allocation P) in a distributed way. The solutions that we propose share the following two logical steps

1. Find a *candidate transmission group* that attempts to maximize the objective function

$$\sum_l \lambda_l c_l$$

while satisfying the primary constraints of activating at most one incoming or outgoing link to each node.

2. Determine a power allocation that allows simultaneous activation of the most advantageous subset of transmitters in the candidate transmission group, while satisfying the secondary interference constraints

$$\gamma_l(P) \geq \gamma^{\text{tgt}}$$

This may require some links to leave the candidate set. Any set of active transmitters that satisfies both primary and secondary interference constraints is called a *feasible transmission group*.

The idea of the two-step procedure is to separate the combinatorial problem of distributing transmission rights to satisfy the primary interference constraints from the problem of adjusting transmit powers to combat secondary interference.

5.1 Forming candidate transmission groups

As shown in [11], a candidate transmission group with maximum weighted throughput can be found in polynomial time. However, these algorithms require global topology information and, to the best of our knowledge, no distributed algorithm is known that solves the problem exactly (cf. [27]).

We propose to use a greedy scheme where the link with highest priority in a two-hop link neighborhood assigns itself membership to the candidate set. To allow links to make this decision, we assume that the transmitters of each link forwards information about its link price to the receiving node. Each node n maintains the maximum link price $\overline{\lambda}_n$ on its incoming and outgoing links,

$$\overline{\lambda}_n = \max_{l \in \mathcal{I}(n) \cup \mathcal{O}(n)} \lambda_l$$

where $\mathcal{I}(n)$ and $\mathcal{O}(n)$ denote the set of incoming and outgoing links to node n, respectively. Neighboring nodes, in turn, exchange information about their maximum link price $\overline{\lambda}_m$. Based on this information, each node n can decide whether one of its outgoing links can join the candidate set or not. Specifically, if $\overline{l}$ is the link corresponding to $\overline{\lambda}_n$, then node n should let this link join the candidate group if $\overline{l} \in \mathcal{O}(n)$ and if $\overline{\lambda}_n = \max_{m \in n \cup \mathcal{N}(n)} \overline{\lambda}_m$.

Before we proceed to discuss distributed solutions for forming feasible transmission groups, it is instructive to investigate the performance losses that we incur by separating the joint power control and scheduling subproblem into two steps, as well as the performance loss due to the greedy formation of candidate sets. The results of such an investigation for a sample network consisting of 10 nodes and 46 links is shown in Figure 2. We can see a clear performance loss (approximately, 4.7%) from introducing the two-step procedure while the additional loss from using a greedy candidate set formation relatively small (an additional 1.9%)

5.2 Forming feasible transmission groups

Once a link has identified itself as a member of the candidate set, it will start contending for transmission rights. We

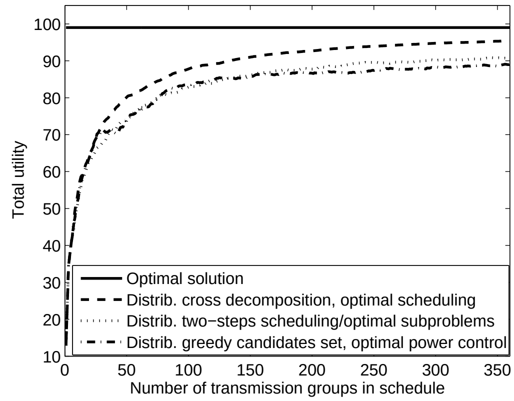

Figure 2: Performance as function of schedule length for optimal scheduling and two-step scheduling with optimal and greedy candidate set formation.

will consider two approaches in which links enter the transmission group one-by-one, adjusting their transmit powers to maximize the number of simultaneously active links. The first approach is based on the scheme proposed by Muqattash and Krunz [23], while the second one exploits the properties of distributed power control with active link protection (DPC/ALP) algorithm of Bambos, Chen and Pottie [1].

5.2.1 Channel reservation via RTS-CTS

This scheme, which is based on [23], uses a channel architecture where two non-overlapping frequency bands are used for control and data transmissions, respectively. This assumption allows a terminal to transmit and receive simultaneously over the data and control channels since no interference can occur among them. As in the IEEE 802.11

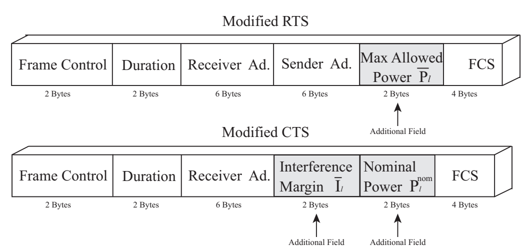

Figure 3: Structure of RTC-CTS packets.

standard, the channel access strategy exploits a reservation mechanisms based on Request-to-Send (RTS) and Clear-to-Send (CTS) packets with the format shown in Figure 3. Thus, by overhearing and decoding control packets, nodes can collect relevant status information about their local surrounding including the number of already active links in a time slot, their transmission power and interference margins (i.e., the additional interference that already active receivers can tolerate). Furthermore, the control packets are sent at maximum power to allow nodes to estimate path gains. Specifically, assume that the transmitter of link l can decode a CTS-message sent by the receiver on link m, and that the received power is P^r_{ml}. The transmitter of link l

can then estimate the path gain G_{ml} as

$$\hat{G}_{ml} = \frac{P^r_{ml}}{P_{\max}}$$

By exchanging RTS-CTS messages between transmitter and receiver, links can decide if they can enter the transmission group without causing too much interference to already active links in their local surroundings. To describe the precise transmitter and receiver behavior, let l be a link in the candidate set, and denote its transmitter and receiver by Tx_l and Rx_l, respectively. Tx_l initiates the reservation by sending an RTS packet which also informs the receiver about the maximum power $\overline{P}_l$ it can use without causing too much interference at active links and without violating its transmit power limits. This value is computed as

$$\overline{P}_l = \min_{m \in \mathcal{L}_s} \{P_{\max}, \frac{\overline{I}_m}{\hat{G}_{ml}}\}$$

Here $\mathcal{L}_s$ denotes the set of links for which the transmitter has overheard RTS-CTS messages, $\overline{I}_m$ is the announced interference margin of link m and $\hat{G}_{ml}$ is the estimated path gain between the transmitter of link l and the receiver of link m, see Figure 4.

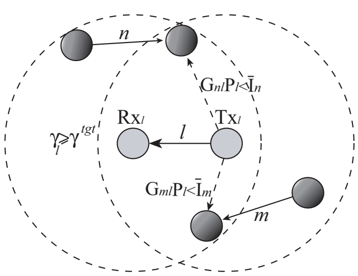

Figure 4: Estimates of interference caused by link l on already active links m and n.

By listening to the control channel and keeping track of local links that have succeeded in reserving transmission rights, Rx_l can estimate the interference $\hat{I}_l$ that it will experience during the data transmission phase and deduce the minimum transmit power $\underline{P}_l$ that Rx_l needs to use as

$$\underline{P}_l = \frac{\gamma^{\mathrm{tgt}}(\sigma_l + \hat{I}_l)}{G_{ll}}$$

Clearly, link l can only receive transmission rights if $\underline{P}_l < \overline{P}_l$. Although one could potentially use the above information for power control, the original scheme by Muqattash and Krunz uses fixed transmit power levels

$$P^{\mathrm{nom}}_l = (1+\epsilon)\frac{\gamma^{\mathrm{tgt}}\sigma_l}{G_{ll}}$$

Here, the second term is simply the transmit power that is needed to sustain the SINR target in the absence of interference and ϵ is an additional factor used to allow for multi access interference. The appropriate value of ϵ depends on the underlying technology; an approach for estimating ϵ under 802.11 and a CA-CDMA scheme is described in [23]. When the power levels are fixed to the nominal levels, $P^{\mathrm{nom}}_l < \underline{P}_l$ means that the multiple access interference in the vicinity of the receiver is grater than the one allowed by the link budget, therefore the transmission over the link can not be activated and Rx_l replies with a negative CTS. If $\underline{P}_l < P^{\mathrm{nom}}_l < \overline{P}_l$, then it is possible for Rx_l to receive the signal of Tx_l while not disturbing any of the ongoing transmissions in its vicinity. In this case, Rx_l replies with a positive CTS message, indicating that the transmitter should use the nominal power P^{nom}_l. In addition, the receiver computes the interference margin $\overline{I}_l$ via the relationship

$$\frac{G_{ll}P^{\mathrm{nom}}_l}{\sigma_l + I_l + \overline{I}_l}\gamma^{\mathrm{tgt}}$$

where I_l is the current interference level at the receiver of link l. A simple calculation gives that

$$\overline{I}_l = \frac{G_{ll}P^{\mathrm{nom}}_l}{\gamma^{\mathrm{tgt}}} - I_l - \sigma_l$$

We would like to emphasize that one of the main limitations of the original channel reservation algorithm is the high number of RTS-CTS control packet that needs to be exchanged in a large network if all links participate in the negotiation. In this case several collisions of control packets could occur and the reservation mechanism becomes ineffective and could even fail [23]. Our approach alleviates this problem by proceeding in two steps, so that only a small subset of links (the candidate set) exchange RTS-CTS control packets. This reduces the load on the control channel and improves scalability of the approach.

There are several simple extensions to the original scheme [23]. As discussed above, one could allow the receiver to suggest an alternative power level if $P^{\mathrm{nom}}_l < \underline{P}_l < \overline{P}_l$. Furthermore, if it is known that the optimal power allocation policy is to use maximum power or stay silent (as is the case in ultra-wideband systems), one could remove the allowed power field in the CTS message. However, none of these extensions will be pursued in this paper.

5.2.2 Channel reservation via DPC/ALP

As an alternative to the RTS-CTS based reservation mechanism, we will explore transmission group formation based on the distributed power control with active link protection (DPC/ALP) algorithm [1]. The DPC/ALP algorithm is an extension of the classical distributed power control algorithms (*e.g.*, [9]) which maintains the quality of service of operational links (link quality protection) while allowing inactive links to gradually power up in order to try to enter the transmission group. As interference builds up, established links sustain their quality while incoming ones may be blocked and rejected channel access.

The original DPC/ALP algorithm was proposed for cellular systems. It exploits local measurements of SINRs at the receivers and runs iteratively over a sequence of time slots. To describe the algorithm in detail, we introduce $\mathcal{A}(j)$ and $\mathcal{I}(j)$ as the set of active and inactive links at time j respectively, and let $\gamma_l(j)$ be the measured SINR on link l at time j. The DPC/ALP algorithm operates by updating the transmit powers $P_l(j)$ at time j according to

$$P_l(j) = \begin{cases} \delta P_l(j-1)\gamma^{\mathrm{tgt}}/\gamma_l(j) & \text{if } l \in \mathcal{A}(j-1) \\ \delta P_l(j-1) & \text{if } l \in \mathcal{I}(j-1) \end{cases} \tag{13}$$

where $\delta > 1$ is a control parameter. Links change status from inactive to active when their measured SINR exceeds the target. Inactive nodes that consistently fail to observe any SINR improvement enters a voluntary drop-out phase and go silent (see [1] for details). Notice that the concepts of local surrounding and local information do not play any role here, since the admission control is based on effective SINR measurements only.

In the cellular system model, links that enter the active set remain active during the full call (or data transfer) duration without regard to congestion levels in inactive links. We will not use the DPC/ALP in this way, but only in the negotiation phase when links in the candidate set contend for transmission rights. The negotiation phase is initialized by letting $\mathcal{I}_0$ equal the candidate set. Links then execute the DPC/ALP algorithm for a short period of time, divided into J mini-slots corresponding to the iteration times above. At the end of such a period we get a feasible transmission group $\mathcal{A}_J^{(k)}$ which is assigned transmission rights for the slot under negotiation. To increase the likelihood of forming a transmission group with high congestion-weighted throughput, we can let links toss a coin to decide whether to start executing the DPC/ALP algorithm or to wait. By letting the waiting probability be a decreasing function of the link congestion level, highly loaded links will tend to start ramping up transmit powers before lightly loaded ones.

The DPC/ALP algorithm was developed for systems with a relatively small number of links and could consume substantial amounts of energy if a large number of links would use it to contend for transmission rights. The two-step procedure makes up for this issue by limiting the number of links that are active in the power negotiation phase.

6. IMPLEMENTATION ASPECTS

The previous sections have derived the basic functionalities needed to align transport and physical layers in S-TDMA wireless multihop networks. These functionalities include an end-to-end rate allocation based on link price feedback, mechanisms for updating and maintaining a transmission schedule including resource allocation to maximize weighted throughput, and an active queue management algorithm to coordinate transport and lower layers. These modules could be implemented in several ways, on different time-scales, and with slightly different overall goals.

One natural implementation is in terms of TCP/AQM on the fast time-scale and incremental schedule updates on the slow time-scale. As demonstrated in [21], most of the common TCP/AQM variants can be interpreted as distributed methods for solving the optimization network flow problem (3). The cross decomposition method suggests the following distributed approach for solving the cross-layer design problem (5): based on an initial schedule with long-term average rate $\bar{c}^{(k_0)} \succ 0$, we run the TCP/AQM scheme until convergence (this may require us to apply the schedule repeatedly). We will refer to this as the *data transmission phase*. During a subsequent *negotiation phase* we apply a distributed channel reservation scheme to find the transmission group with the largest congestion-weighted throughput, and augment the schedule. The procedure is then repeated with this revised schedule; see Figure 5. It is important to note that no single node needs to have global knowledge about the transmission schedule: each node only needs to keep track of in what time slots it needs to activate its receivers and transmitters. Further, our theoretical analysis applies to the case when we augment the schedule indefinitely, while in practice one would like to use schedules with

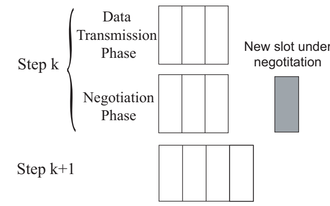

Figure 5: Augmenting the schedule: Each step, k, consists of two phases: a *data transmission phase*, in which data is transmitted until the optimization flow control algorithm converges to yield $\lambda(k)$; and a *negotiation phase*, where transmission rights for a new slot are negotiated on the control channel while transmissions are continuing according to the current schedule on the data channel. When the negotiations are done, the new slot is added to the schedule and the procedure continues with step $k+1$.

limited frame length. We will return to this issue in Section 7 and demonstrate strong performance even when we limit the schedule length and replace rather than add time-slots to the frame.

Similarly to the case of TCP in fixed networks, practical congestion control mechanisms will only work according to the theoretical analysis for long-lived flows. One can argue that the discrepancy is even larger in our case, since the theoretical analysis assumes that the rate allocation converges before the schedule is updated. However, one should note that the resource allocation and queue management schemes act on aggregate traffic. If the aggregate traffic rates between wireless ingress and egress routers react on congestion information similarly to (4), e.g. the aggregate rate decreases with increasing congestion, the complete scheme could be justified as jointly optimizing aggregate end-to-end bandwidth and scheduling in the network (for some appropriate utility functions). For back-haul wireless networks, this appears to be a reasonable assumption.

Alternatively, the proposed solution could be considered as a signalling and negotiation scheme for proportionally fair end-to-end bandwidth allocation between ingress and egress routers. One then associate logarithmic utility functions to each pair of source-destination nodes. Rather than actually updating the end-to-end rates, routers signal rate requests (updated according to the source-rate mechanism in the optimization flow control), while nodes update link prices based on rate requests rather than actual traffic. After a number of iterations, the actual schedule is updated by solving the link-price weighted throughput problem. We will not investigate these schemes further here, but focus on schemes that react on actual congestion levels.

7. EXAMPLES

We are now ready to evaluate the different variants of the joint congestion control and S-TDMA scheduling developed in the previous sections. These distributed schemes will rely on cross-decomposition (TCP/AQM with incremental schedule updates) combined with the heuristic algorithms for distributed transmission scheduling and power control described in Section 5. To achieve proportional fairness, we will assume logarithmic utility functions throughout. We

	Total utility	%
Optimal solution	98.99	−−
Cross decomposition	95.41	−3.62
Mean value cross dec.	94.44	−4.59
Optimal subproblems	90.89	−8.18
Greedy candidate set	89.04	−10.06

Table 1: Performance losses of cross decomposition and two-step procedure.

will isolate and quantify performance losses in detail for a network consisting of 10 nodes and 46 links; statistics from a larger set of network is summarized in the end of the section.

All networks are generated using the procedure described in [16]. Since our channel reservation algorithms are heuristic, secondary interference may cause the SINR to drop below the threshold for certain links. When this happens, we follow our model from Section 2 and allocate the link zero rate, but note that the distributed algorithms would exhibit better performance if links could resort to a lower link rate in this case. It is important to note, however, that the proportionally fair performance objective guarantees non-zero end-to-end rate for all connections if the number of time slots exceeds the number of network links.

7.1 Performance losses due to cross decomposition and fixed time slot lengths

For easy reference, we begin by reviewing the results and performance evaluations from Sections 4 and 5. The base line for our evaluations is the performance of the centralized optimization (5) based on global network information. An optimal schedule can always be constructed that uses (at most) $L+1$ transmission groups, but this is only true if the system can support variable-length time slots. In contrast, the cross decomposition approach allows us to incrementally construct a schedule with fixed time slot lengths. As demonstrated in Section 4, the cross decomposition approach recovers the optimal performance as the number of time slots tend to infinity and appears to converge faster and more stably than the mean-value cross decomposition. Table 1 shows the performances of the centralized approach, along with the cross-decomposition approaches when the iteration is aborted after 360 iterations.

7.2 Performance of the two-step scheduling

Our schemes for solving the joint transmission scheduling and power allocation problem rely on a two-step procedure: first we form a candidate set of links that satisfy the primary interference constraints and then we let these links negotiate transmit powers to arrive at a feasible transmission group.

Table 1 shows the performance of the cross decomposition approach combined with the two-step procedure for transmission scheduling (also in this case we abort the iteration after 360 iterations). We observe a clear performance loss, even when we solve the candidate formation and power allocation subproblems to optimality based on global network information. Moving from optimal candidate set formation to a distributed greedy scheme based on local information gives rise to another, although comparatively smaller, performance loss, see Table 1.

7.3 Performance of the RTS-CTS scheme

We will now consider the performance of the complete distributed solution based on TCP/AQM with incremental schedule updates based on greedy candidate set formation and RTS-CTS-based power allocation and channel reservation. The algorithm performs best when global information is available at each node. However, as shown in Figure 6, even when global information is available, the performance stabilizes after a small number of iterations. One of the reasons for this is the use of fixed transmit power levels which may force links to stay silent even if they could enter the transmission group without causing too much interference on active links. When only very local information is available, collisions of data packets can occur due to erroneous estimation of the external interference level. In this case, the RST-CTS reservation scheme can not ensure an acceptable performance and the total utility trend appears unstable.

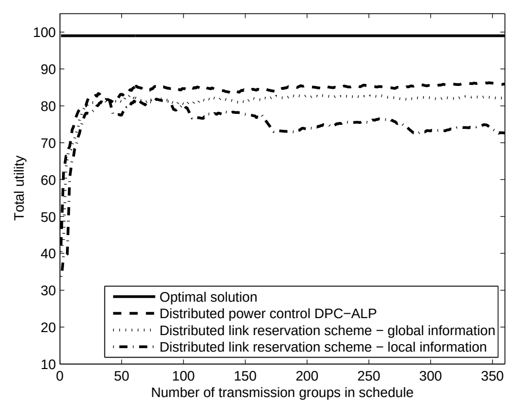

Figure 6: Total utility trend for RTS-CTS reservation mechanism and DPC/ALP.

7.4 Performance of the DPC/ALP approach

Next, we consider the distributed resource allocation when the transmission group formation is performed using the DPC/ALP approach. Figure 6 demonstrates that the distributed approach with DPC/ALP has a stable trend with small, but consistent, performance improvements in each iteration. In general, our numerical experiments have shown a better behavior of DPC/ALP in comparison with the RTS-CTS mechanism, in particular when the latter is forced to use local information only. The precise performance of DPC/ALP and RST-CTS with global and local information are shown in Table 2. The protocol based on DPC/ALP has a total performance loss of 13.4% compared to the centralized solution, while the performance of RTS-CTS lies 17.1% and 26.7% below the optimal.

7.5 Comparison with alternative approaches

Finally, we compare the performance our our schemes with two alternative approaches: the optimal, variable time-slot length TDMA schedule (i.e., the optimal solution when spatial reuse of transmission opportunities is not supported) and cross-decomposition using transmission group formation based on a flow-contention graph (i.e., using a crude model of the secondary interference in the transmission group formation). Basically, in a flow contention graph, links can

	Total utility	%
Optimal solution	98.99	−−
DPC/ALP	85.85	−13.37
RTS-CTS global info	82.06	−17.10
RTS-CTS local info	72.64	−26.62

Table 2: Performance losses of distributed protocols.

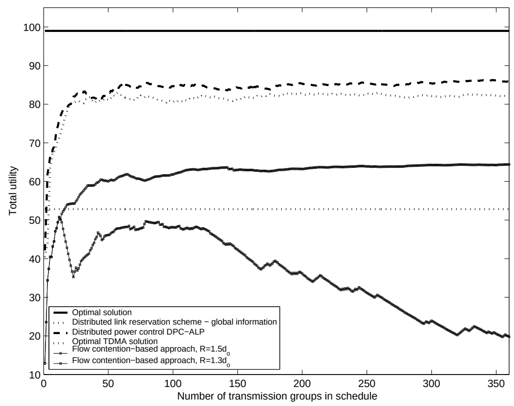

Figure 7: Comparison of total utility for optimal and distributed schemes, compared to the alternative schemes of optimal TDMA and flow contention-based scheduling for various values of the carrier sensing range R ($R = 1.3d_0$ and $R = 1.5d_0$), where d_0 corresponds to the maximum distance between a pair of node in order to establish a link.

be simultaneously activated as long as they are not within each other's carrier sensing range [5, 29, 19]. The comparison shown in Figure 7, reveals that the suggested schemes improve significantly over the state-of-the art solutions. It also demonstrates the significance of taking interference into account when scheduling links: the graph-based algorithm neglects secondary interference when making its scheduling decisions. As a consequence it tries to activate too many links, which results in too high interference levels and allows very few (if any) links to meet their SINR targets.

7.6 Performance for a set of sample networks

We have evaluated the performance losses of distributed scheduling with limited schedule length (the cross decomposition approach with optimal solution to subproblems) and of the distributed scheme based on DPC-ALP for a set of 10 randomly generated 10-node networks. In this case, the performance loss of the cross decomposition ranged from $2 - 15\%$ with an average of around 11.7%. The distributed solution based on DPC-ALP came within $13 - 20\%$ of the optimal (centralized variable slot-length) performance, with an average performance loss of 18.7%.

7.7 Rolling Horizon Optimization

The optimization problem described in Section 4 augments the schedule indefinitely and requires nodes to maintain a transmission schedule of ever-increasing length. An alternative and more practical approach would be to maintain and update a schedule with finite length, say M transmission groups, and to sequentially replace the oldest ones in a rolling horizon fashion. We refer to this approach as *rolling horizon optimization*. Figure 8 shows the performance of a rolling horizon optimization with both optimal centralized transmission group formation and the DPC-ALP scheme. The rolling horizon is fixed to $M = 100$ time slots. As we can see, in the first case the rolling horizon scheme approaches the optimal performance with a remarkable improvement in rapidity. This can be intuitively explained by the fact that the initial TDMA slots are quickly replaced by new STDMA transmission groups. In the second case we also got a significant improvement in comparison with the original algorithm. Table 3 summarizes the results.

	Total utility	%
Optimal solution	98.99	−−
Optimal resource allocation	97.66	−1.34
Distributed DPC/ALP	89.40	−9.68

Table 3: Performance losses using a rolling horizon optimization.

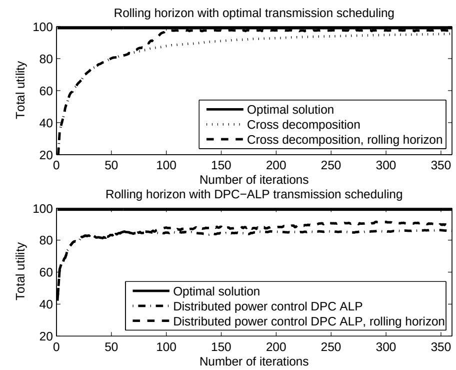

Figure 8: Rolling horizon optimization performance.

8. CONCLUSIONS

We have considered the problem of joint congestion control and resource allocation in spatial-TDMA wireless networks. By posing the problem as a utility maximization problem subject to link rate constraints which involve both transmission scheduling and power allocation, we have proceeded systematically in our development of transparent and distributed protocols. In the process, we have introduced a novel decomposition method for convex optimization, established its convergence for the utility maximization problem, and demonstrated how it suggests a distributed solution based on TCP/AQM on a fast time-scale and incremental updates of the transmission schedule on a slower time-scale. We have developed a two-step procedure for finding the schedule updates, and suggested two schemes for distributed channel reservation and power control under realistic interference models. Although the final protocols are suboptimal, we have isolated and quantified the performance losses incurred in each simplification step and demonstrated strong improvements over the state-of-the art solutions.

There are many natural extensions to our work. The equilibrium results presented in this paper should be complemented with an analysis of the dynamical properties of the distributed solution in the presence of network delays. The algorithms have been derived under assumptions of saturated traffic, *i.e.*, considering bit-level rather than flow-level performance. Although bit-level optimal policies tend to work well also for stochastic traffic (see, *e.g.*, [20]), a queueing-theoretic analysis of the stability properties of the scheme should be conducted. Further, detailed models of the protocols suggested by our theoretical analysis should be developed and their practical performance should be evaluated in network simulations under more realistic radio environments. In particular, it would be interesting to compare the practical performance of solutions relying on dynamic and static scheduling. On the one hand, dynamic scheduling allows the physical layer to be opportunistic and could harvest statistical multiplexing gains. On the other hand, the interplay between opportunistic schedulers and transport layer is non-trivial and careless combinations could result in lower performance than non-opportunistic use of the channel [18]. Finally, we have only considered transmission group formation for single-rate links. Since most modern radio technologies offer rate adaption, it would be interesting to derive distributed methods for scheduling variable-rate links. The computational experiments in [16] indicate that the performance benefits of variable link rates could be substantial.

9. ACKNOWLEDGMENTS

This research was sponsored in part by the Swedish Research Council, Wireless@KTH and the European Commission through the projects EuroNGI, Hycon and Runes.

10. REFERENCES

[1] N. Bambos, S. C. Chen, and G. Pottie. Channel access algorithms with active link protection for wireless communication networks with power control. *IEEE/ACM Trans. on Networking*, 8(5):583–597, October 2000.

[2] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[3] R. Bittel, E. Caples, C. D. Young, and F. Loso. Soldier phone: an innovative approach to wireless multimedia communications. In *Proc. of the IEEE Milcom*, volume 3, pages 903–907, October 1996.

[4] S. Boyd. *Convex Optimization*. Cambridge University Press, 2004.

[5] L. Chen, S. H. Low, and J. C. Doyle. Joint congestion and media access control design for ad hoc wireless networks. In *Proceedings of the IEEE Infocom*, Miami, FL, March 2005. To Appear.

[6] M. Chiang. Balancing transport and physical layers in wireless multihop networks: Jointly optimal congestion control and power control. *IEEE Journal on Selected Areas in Communications*, 23(1):104–116, January 2005.

[7] J. C. Dunn. Conditional gradient algorithms with open loop step-size rules. *Journal of Mathematical Analysis and Applications*, 62:432–444, 1978.

[8] T. ElBatt and A. Ephremides. Joint scheduling and power control for wireless ad-hoc networks. In *Proc. of the IEEE Infocom*, New York, NY, June 2002.

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

[10] N. Funabiki and Y. Takefuji. A parallel algorithm for broadcast scheduling problems in packet radio networks. *IEEE Trans. on Communications*, 41(6):828–831, June 1993.

[11] B. Hajek and G. Sasaki. Link scheduling in polynomial time. *IEEE Trans. Information Theory*, 34(5):910–917, September 1988.

[12] K. Holmberg. Linear mean value cross decomposition: a generalization of the Kornai-Liptak method. *European Journal of Operational Research*, 62:55–73, 1992.

[13] K. Holmberg. Primal and dual decomposition as oragnizational design: Price and/or resource directive decomposition. Technical Report LiTH-MAT-R-94-03, Dpt. of Mathematics, Linköping Institute of Technology, Linköping, Sweden, 1994.

[14] K. Holmberg and K. C. Kiwiel. Mean value cross decomposition for nonlinear convex programs. *Optimization Methods and Software*, 21(3):401–417, June 2006.

[15] B. Johansson and M. Johansson. Primal and dual approaches to distributed cross-layer optimization. In *16th IFAC World Congress on Automatic Control*, July 2005.

[16] M. Johansson and L. Xiao. Cross-layer optimization of wireless networks using nonlinear column generation. *IEEE Transactions on Wireless Communications*, 5(2):435–445, February 2006.

[17] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of Operations Research Society*, 49(3):237–252, March 1998.

[18] T. Klein, K. Leung, and H. Zheng. Enhanced scheduling algorithms for improved TCP performance in wireless IP networks. In *IEEE Globecom*, Dallas, TX, November 2004.

[19] J.-W. Lee, M. Chiang, and R. Calderbank. Fair and efficient mac for wireless ad-hoc networks: optimization framework, optimal algorithms, and performance comparison. *submitted to IEEE Trans. on Mobile Computing*, April 2005.

[20] X. Lin and N. B. Shroff. The impact of imperfect scheduling on cross-layer rate control in wireless networks. In *Proc.of the IEEE Infocom*, Miami, FL, March 2005.

[21] S. H. Low. A duality model of TCP and queue management algorithms. *IEEE/ACM Trans. on Networking*, 11(4):525–536, August 2003.

[22] S. H. Low and D. E. Lapsley. Optimization flow control I: Basic algorithm and convergence. *IEEE Transactions on Networking*, 7(6):861–874, 1999.

[23] A. Muqattash and M. Krunz. CDMA-based MAC protocol for wireless ad hoc networks. In *Proc.of the ACM MobiHoc'03*, pages 153–164, Annapolis, MD, USA, June 2003.

[24] B. Radunovic and J.-Y. L. Boudec. Rate performance objectives of multi-hop wireless networks. *IEEE Trans.*

on Mobile Computing, 3(4):334–349, Oct.-Dec. 2004.

[25] P. Soldati. Distributed cross-layer optimized flow control for STDMA wireless networks. Master's thesis, Royal Institute of Technology, November 2004.

[26] S. Toumpis and A. Goldsmith. Capacity regions for wireless ad hoc networks. *IEEE Trans. on Wireless Communications*, 2(4):736–748, July 2002.

[27] M. Wattenhofer and R. Wattenhofer. Distributed weight matching. In *18th Annual Conference on Distributed Computing*, Amsterdam, The Netherlands, October 2004.

[28] L. Xiao, M. Johansson, and S. Boyd. Simultaneous routing and resource allocation via dual decomposition. In *Proc. of the 4th Asian Control Conference*, pages 29–34, Singapore, September 2002.

[29] Y. Yi and S. Shakkottai. Hop-by-hop congestion control over a wireless multi-hop network. In *Proceedings of the IEEE Infocom*, volume 4, pages 2548–2558, Hong Kong, March 2004.

APPENDIX

Convergence of the cross decomposition method

To prove convergence of the algorithm we need the following additional assumptions: (1) all links are bottlenecks, (2) the routing matrix has full row rank, and (3) that the scheduling subproblem can be solved to optimality. The first assumption can be fulfilled by requiring that all links have at least one flow using only that link.

First we consider some basic properties of $\nu(c)$ defined as

$$\nu(c) = \max_{Rs \preceq c,\, s \succ 0} \sum_p u_p(s_p)$$

Also define the dual function, $g(c, \lambda)$, as

$$g(c,\lambda) = \max_{s \succ 0} \sum_p u_p(s_p) - \lambda^T(Rs - c)$$

LEMMA .1. *For every positive capacity vector, there exists a strict interior point $\bar{s}$ to (5) and strong duality holds.*

PROOF. A strict interior point $\bar{s}$ satisfying $\bar{s} \succ 0$ and $R\bar{s} \prec c$ can be constructed by setting the elements in $\bar{s}$ to

$$\bar{s}_p = \min_l \frac{c_l}{L} - \epsilon > 0, \quad p = 1, \ldots, P$$

where ϵ is a small positive constant. Since a strict interior point exists, Slater's condition for constraint qualification is satisfied, hence strong duality holds [2]. □

LEMMA .2. *$g(c, \lambda)$ is concave and continuous in c for all $\lambda \succeq 0$*

PROOF. Follows since $g(c, \lambda)$ is linear in c. □

LEMMA .3. *$\nu(c)$ is concave, and subgradient to $\nu(c)$ at c is given by the optimal Lagrange multipliers, $\lambda^\star$, for the capacity constraint in the definition of $\nu(c)$.*

PROOF. See [2]. □

LEMMA .4. *The optimal Lagrange multipliers $\lambda^\star$ for the capacity constraint in the definition of $\nu(c)$, with $c \succ \rho \succ 0$, are bounded and belong to a convex and compact set $\Lambda(\rho)$.*

PROOF. Follows analogously to Problem 5.3.1 in [2]. □

LEMMA .5. *The optimal Lagrange multipliers $\lambda^\star$ for the capacity constraint defining $\nu(c)$ are unique.*

PROOF. The Karush-Kuhn-Tucker (KKT) conditions are

$$u'(s^\star(c)) = R^T\lambda^\star(c)$$
$$Rs^\star(c) = c$$

The latter expression holds by the assumption that all links are bottlenecks. The observation that the source rate $s^\star(c)$ is unique since the objective functions are strictly concave together with the fact that R has full row rank implies that $\lambda^\star(c)$ is unique. □

THEOREM .6. *$\nu(c)$ is differentiable for all $c \in C, c \succ \rho \succ 0$, and the derivative is $\lambda^\star(c)$.*

PROOF. $\nu(c)$ can now be expressed as

$$\nu(c) = \min_{\lambda \in \Lambda(\rho)} \max_{s \succ 0} \sum_p u_p(s_p) - \lambda^T(Rs - c) = \min_{\lambda \in \Lambda(\rho)} g(c, \lambda)$$

where the set $\Lambda(\rho)$ is compact (Lemma .4), the optimal Lagrange multipliers $\lambda^\star(c)$ are unique (Lemma .5), and $g(c, \lambda)$ is continuous and concave in c for all $\lambda \in \Lambda(\rho)$ (Lemma .2). Furthermore, for every optimal Lagrange multiplier $\lambda^\star$, $g(\cdot, \lambda^\star)$ is differentiable at c, i.e.,

$$\frac{\partial g(c, \lambda^\star)}{\partial c_i} = \frac{\partial}{\partial c_i} \sum_p u_p(s_p^\star) - \lambda^{\star T}(Rs^\star - c) = \lambda_i^\star$$

Now Danskin's theorem [2] gives the desired result. □

REMARK 1. *Since a ρ fulfilling $c \succ \rho \succ 0$, can be found for all $c \in \mathcal{C}, c \succ 0$, the theorem above gives that $\nu(c)$ is differentiable for all $c \in \mathcal{C}, c \succ 0$.*

LEMMA .7. *The derivative of $\nu(c)$ is continuous*

PROOF. Using the KKT conditions and the implicit function theorem, it can be shown that λ is differentiable with respect to c. This means that λ is continuous in c, and the derivative is therefore continuous in c. See [2] for details. □

THEOREM .8. *Let $u^\star$ be the optimal value of the centralized cross-layer design problem (5). Algorithm 1 converges in the sense that $\lim_{k\to\infty} \nu(\bar{c}^{(k)}) \to u^\star$.*

PROOF. The update rule for $\bar{c}^{(k)}$ can be re-written as

$$\bar{c}^{(k+1)} = \frac{k}{k+1}\bar{c}^{(k)} + \frac{1}{k+1}c' = (1 - \omega_k)\bar{c}^{(k)} + \omega_k c'$$

where $\omega_k = 1/(k+1)$ and c' is found as the solution to the congestion-weighted throughput problem

$$\begin{array}{ll} \text{maximize} & \lambda^T c' \\ \text{subject to} & c' \in \mathcal{C} \\ & \bar{c}^{(k)} \geq \rho \end{array}$$

Since λ is a gradient of $\nu(\bar{c}^{(k)})$ at $\bar{c}^{(k)}$, this is a conditional gradient method with an open loop step-size rule [7] that satisfies

$$\omega_{k+1} = \frac{\omega_k}{\omega_k + 1}, \qquad \omega_0 = 1$$

Since $\lambda^\star$ is continuous and the domain is compact (since $\bar{c}^{(k)} \in \mathcal{C}$ and $\bar{c}^{(k)} \geq \rho$), the derivative of $\nu(\bar{c}^{(k)})$ is uniformly continuous. By Theorem 1 in [7], $\lim_{k\to\infty} \bar{c}^{(k)} = c^\star$ and $\lim_{k\to\infty} \nu(\bar{c}^{(k)}) = u^\star$. Thus, Algorithm 1 converges to the optimal solution. □