LABORATOIRE

INFORMATIQUE, SIGNAUX ET SYSTÈMES
DE SOPHIA ANTIPOLIS
UMR 6070

ON THE COMPLEXITY OF BANDWIDTH ALLOCATION IN RADIO NETWORKS WITH STEADY TRAFFIC DEMANDS

Ralf Klasing, Nelson Morales, Stephane Perennes

Projet MASCOTTE

Rapport de recherche
ISRN I3S/RR–2004-40–FR

Décembre 2004

LABORATOIRE I3S: Les Algorithmes / Euclide B – 2000 route des Lucioles – B.P. 121 –
06903 Sophia-Antipolis Cedex, France – Tél. (33) 492 942 701 – Télécopie : (33) 492 942 898
http://www.i3s.unice.fr/I3S/FR/

On the Complexity of Bandwidth Allocation in Radio Networks with Steady Traffic Demands

R. Klasing, N. Morales, S. Perennes *†
MASCOTTE project I3S-CNRS/INRIA/Université de Nice-Sophia Antipolis
2004 Route des Lucioles, BP 93 F-06902 Sophia Antipolis Cedex (France)

Abstract

In this paper we define and study a call scheduling problem that is motivated by radio networks. In such networks the physical space is a common resource that nodes have to share, since concurrent transmissions cannot be interfering. We study how one can satisfy steady bandwidth demands according to this constraint. This leads to the definition of a *call scheduling problem*. We show that it can be relaxed into a simpler problem: The *call weighting* problem, which is almost a usual multi-commodity flow problem, but the capacity constraints are replaced by the much more complex notion of non interference. Not surprisingly this notion involve independent sets, and we prove that the complexity of the call weighting problem is strongly related to the one of the independent set problem and its variants (max-weight, coloring, fractional coloring). The hardness of approximation follows when the interferences are described by an arbitrary graph. We refine our study by considering some particular cases for which efficient polynomial algorithm can be provided: the *Gathering* in which all the demand are directed toward the same sink, and specific interference relations: namely those induced by the dimension 1 and 2 Euclidean space, those cases are likely to be the practical ones.

1 Introduction

Our goal is to study how to allocate bandwidth to connections in a radio network. We address the static case problem in which one wishes to provide some given bandwidth to the networks sites. The goal is to schedule the radio transmissions in order to route some fixed static traffic demands. For each ordered pair of nodes (u, v) we suppose a traffic (flow) bandwidth demand $f(u, v)$ is given, and we wish to route on average about $f(u, v)dt$ units of traffic from u to v during the time interval dt. The originality of the problem comes from the fact that in networks such as radio networks several concurrent transmissions can be performed in parallel during a communication step, but those transmissions must be non-interfering. The communication resource that has to be shared is not a set of links with some capacity as in classical networks. In radio networks the resource is the physical (Euclidean) space. We model the problem by assuming that we are given two relations : the *interference relation* and the *transmission relation*. Given two nodes u, v of the network, we know if the transmission (u, v) can be performed or not (i.e. if v is in the transmission range of u), if so we will call it a *transmission-arc*[1] and we define $E_T \subseteq (V, V)$ as the set of of feasible transmission-arcs. The interference relation is defined on the transmissions-arcs ($E_I \subseteq [E_T, E_T]$), two transmissions (u_0, v_0) and (u_1, v_1) interfere when they cannot be performed at the same time. Note that this generally occurs when u_0 (resp. u_1) is too close to v_1 (v_0). A classical model being to consider nodes in $\mathbb{R}^2$ and to define $E_T = \{(u, v) \in (V, V) \mid d(u, v) \leq d_T\}$ and $E_I = \{[(u_0, v_0), (u_1, v_1)] \in [E_T, E_T] \mid d(u_0, v_1) \leq d_I \vee d(u_1, v_0) \leq d_I\}$ for some fixed $d_T, d_I \in \mathbb{N}$.

*This work was supported by the European community project IST FET CRESCCO (contract no. IST-2001-33135). N. Morales is funded by a CONICYT/INRIA Sophia-Antipolis doctoral grant. Contact emails: {stephane.perennes,ralf.klasing,nelson.morales}@inria.fr.

†This work is part of the CRC CORSO funded by France Télécom R & D, on *Design of Telecommunication Networks, Fault Tolerance and Radio Networks*.

[1]We use this term in order to distinguish from the physical graph arcs or arcs occurring in auxiliary constructions.

This work partially answers a question of J. Galtier and A. Laugier from France Télécom R & D: An Internet provider wishes to design efficient strategies to provide Internet access using wireless devices. Typically, in one village several houses wish to access a gateway and to use multi-hop wireless relay routing to do so, see [2] for a more detailed presentation. In this work we assume that the nodes are synchronized with a small clock drift, and that the traffic pattern is fixed and known, or steady enough so that it can be estimated. We will ensure conflict-free accesses to the radio media, so we assume that the MAC layer protocol has the following property : *conflict free calls are made without significant throughput loss*, for example our schemes can be used above the 802.11 norm.

1.1 Traffic Routing in Interference Graphs – Definitions

We suppose we are given the vertex set V and the set of (feasible) transmissions $E_T \subseteq (V, V)$, and the interference relation $E_I \subseteq [E_T, E_T]$. (Note that transmission-arcs are directed (e.g. $(u, v) \neq (v, u)$) while the interference relation is undirected; for a set S, we use $[S, S]$ to denote the $\binom{|S|}{2}$ pairs of a set and (S, S) to denote the $|S|^2$ ordered pairs.) This induces the *transmission digraph* (V, E_T). We define the *interference graph* as (E_T, E_I). In natural language, the vertex set is the set E_T of transmission-arcs and there is an edge between two transmission-arcs $e = (u, v)$ and $e' = (u', v')$ if they interfere. We define a *Call* $C \subseteq E_T$ as a set of non-interfering transmissions, a call is henceforth an independent set of the interference graph (E_T, E_I). We will denote $\mathcal{C}$ as the set of all possibles calls, $|\mathcal{C}|$ is potentially exponential.

We will study the *fixed power model* [2]. In this model, each node uses the same transmission power, and in each round a node either transmits (at this power level) or it does not. Moreover we will assume that u transmits to only one of its neighbors even if all the vertices in $E_T(u)$ can potentially listen to u simultaneously (see Remark 1).

In this case the instance can be described as follows: We are given for each node $u \in V$ two subsets of V

- the interference set $E_I(u)$
- the transmission set $E_T(u) \subseteq E_I(u)$

The vertex v can transmit to any $u \in E_T(v)$ and induces when it transmits interference at any $u \in E_I(v)$. Formally, we have

$$E_T = \{(u, v) \mid u \in V, v \in E_T(u)\},$$

and two transmission-arcs $e \neq e', e = (u, v), e' = (u', v')$ interfere if

$$v' \in E_I(u) \vee v \in E_I(u').$$

The predicate $v' \in E_I(u) \vee v \in E_I(u')$ means that u produces interference at v', or u' produces interference at v thus preventing reception. In all the practical cases we will have $u \in E_I(u)$ which means that we forbid any node to transmit and receive concurrently.

Remark 1 *Note that when u transmits a message all the nodes in $E_T(u)$ can receive it, if we where studying a broadcast problem we could use this fact to our profit. But in bandwidth allocation problems it is useless to duplicate information since we are routing point to point communications. Still the next situation can occur: during a time τ, u transmits some message m to $v, v' \in E_T(u)$ and later v will forward a part m' of m and v' its complement. In this case one can split the time τ into $\tau_1 + \tau_2 = \tau$ and transmit m' to u during τ_1 and $m \setminus m'$ to u' during τ_2.*

This justifies why we can assume that when u transmits it transmits to only one of its neighbors.

A case of particular interest is the *metric* one. For some numbers $d_T \leq d_I$ and some metric d, the sets $E_T(v)$ (resp. $E_I(v)$) are the set of nodes at distance at most d_T (resp. d_I) from v. The distance can be either the usual distance between the vertices on an underlying graph (*graph case*) or the Euclidean distance (*geometric case*) when one assumes that the nodes are mapped to $\mathbb{R}^n$.

[2] One can easily adapt the model to variable power networks, in which the nodes can adapt their transmission power to different power levels in each round. This would simply change the definition of the interference graph.

Definition 1 *In the metric case we will denote by $\mathcal{I}(G, d_T, d_I)$ the interference graph for distances d_T, d_I.*

We are given also a (directed) bandwidth demand $f : (V, V) \rightarrow \mathbb{R}^+, (u, v) \mapsto f(u, v)$, that expresses the desired average bandwidth from u to v. We will study general traffic patterns but also a specific one : *Gathering* in which all the demands are directed to a single sink. Gathering has some practical importance when one considers a set of network locations equipped with radio devices that need to access some *gateway* that connect the local area network to a high speed network. A slight generalization of the gathering is the *single commodity flow* in which the traffic simply needs to reach one sink among a set of potential ones.

This problem was addressed in the specific case of gathering on line with $d_T = d_I = 1$ in [9]. Modeling interferences with a graph is common but generally it is assumed that $d_I = d_T = 1$, but the problems generally studied assume that the topology is unknown (see as example the work on broadcasting [17]) and have an *anonymous* flavor. Here we focus on how one share efficiently the physical resource when the bandwidth demand dynamic is slow enough to allow a sharp control.

1.2 Our Results

We model the bandwidth allocation problem by defining a *Call Scheduling Problem* and we introduce its natural relaxation the *Call Weighting Problem*. We show that both problems are closely related. This motivates the study of the relaxed variant, namely the Call Weighting Problem.

We then study the complexity of the Call Weighting Problem:

- We show that the general problem is related to computing the maximum weight independent set of the interference graph and prove that it is $\mathcal{NP}$-hard to approximate within $n^{1-\varepsilon}$.
- For *Gathering*, we show that the problem is $\mathcal{NP}$-hard, and we give a 4-approximation algorithm.
- In the case of simple topologies [3] like trees and line we show that the problem is polynomial or admit a Polynomial Time Approximation Scheme, we also provide a small explicit linear program for solving the problem.
- For the gathering problem on the line with transmission at distance 1, we give an explicit formula for the optimum. (This implies a linear-time algorithm for computing the optimum.)
- We give a Polynomial Time Approximation Scheme when the nodes are in $\mathbb{R}^d$.

2 The Scheduling Problem

We assume that the time is divided into slots. During a slot a node can transmit 1 unit of data, the demands are expressed in this unit. The goal is to choose the right sequence of calls in order to be able to route the highest percentage of the traffic demand.

For a time horizon τ, a *Call Scheduling* is a mapping s from $\{0, 1, \ldots, \theta, \ldots, \tau\}$ on the set of calls $\mathcal{C}$ (calls are time-disjoint). Given a call scheduling, one determines a timed flow network as follows : The vertices are the original nodes labeled with a time (we denote as u_θ the vertex representing the node u at time θ). There is a unit capacity arc from u_θ to $v_{\theta+1}$ if and only if (u, v) belongs to $s^{-1}(\theta)$ (i.e. the transmission from u to v is made at time θ), and some infinite capacity $(u_\theta, u_{\theta+1})$. We also assume that for each time slot θ we associate to the bandwidth demand $f(u, v)$ one flow demand from vertex u_θ to the vertex v_τ with $f(u, v)$ units of flows.

The *throughput of* s, denoted by $\gamma(s)$, is defined as

$$\gamma(s) = max\{\gamma \in \mathbb{R}^+, \gamma\tau f \text{ is feasible in the associated timed flow network}\}.$$

$\gamma^*(\tau) = \max_{s \text{ a call scheduling}} \gamma(s)$ will denote the optimum throughput for some time horizon τ. We will mainly be interested in $lim_{\tau\rightarrow\infty}\gamma^*(\tau)$. Note that $\gamma^*(\tau)$ measures the percentage of each bandwidth demand

[3] Assuming interference and transmission relations closely related to the usual metric.

that can be provided if one assumes that a given node produces traffic demands uniformly. We do not require data to be routed within a deadline (the flow produced at time slot θ corresponding to bandwidth demand $f(u,v)$ is from u_θ to v_τ).

Our goal is to find a call scheduling with maximum throughput.

Remark 2 *Note that the problem is quite related to the two following ones :*

- *Given a traffic demand, one wishes to schedule calls in order to route this traffic. This problem is addressed in [1], the main difference is that there the traffic is not to be routed continuously, making the problem harder due to initialization problems.*
- *Given a traffic demand, one wishes to schedule calls in a periodic way so that the traffic demand is routed each τ' slots; the communication pattern is said to be* systolic*. One then wishes to minimize the value of τ'. This version is indeed equivalent to ours. Note that systolic communication was studied for wired networks (see e.g. [15, 8, 14]).*

We first show that this problem can be relaxed to find how to distribute the calls in order to get enough average bandwidth on the arcs to route the traffic. In a sense, when the calls take place is not essential, what does matter is how often.

Call Weighting and Call Scheduling

We relax the call scheduling by removing the time constraints, instead we focus on the average available bandwidth for each call.

Each call scheduling s induces a weight function w_s on the calls defined by $\forall C \in \mathcal{C} : w_s(C) = Card(s^{-1}(C))$, the weight of a call being simply how often it is used during the scheduling. Moreover, to any weight function w on $\mathcal{C}$ (for example w_s) we can associate an *induced capacity* function Cap_w defined on the transmission-arcs E_T by

$$\forall e \in E_T, Cap_w(e) = \sum_{C \in \mathcal{C}, e \in C} w(C).$$

We will say that a *weight function is feasible for a bandwidth demand f* when the multi-commodity flow f is feasible in the induced flow network (with induced capacity function Cap_w). We note that if a scheduling s has throughput $\gamma(s)$ then the flow $\gamma\tau f$ is feasible in the network with capacity function Cap_{w_s}.

Definition 2 *The call weighting problem consists in finding a feasible weight function w on the call set $\mathcal{C}$ such that $\sum_{C\in\mathcal{C}} w(C)$ is minimum. We denote this minimum as W^*.*

Lemma 1 $\gamma^*(\tau) \leq \frac{1}{W^*}$.

Proof: For any call scheduling s with time horizon τ the weight function $\frac{w_s}{\gamma(s)\tau}$ is feasible, and provides a call weighting with cost $\frac{1}{\gamma(s)}$, hence $W^* \leq \frac{1}{\gamma^*(\tau)}$. □

Lemma 2 *For any $\varepsilon > 0$, there exists T_0 such that for any $T \geq T_0$ there exists a call scheduling s such that $\gamma(s) \geq \frac{1}{W^*} - \varepsilon$. This means that $lim_{\tau\to\infty}\gamma^*(\tau) = \frac{1}{W^*}$.*

Proof: Consider an optimal call weighting function w with cost W^* that enables to route statically the flow f. Let k be such that $k \times w$ is integral, and let g be any call scheduling having kw as weight function (simply take the call C exactly $kw(C)$ times and order the calls arbitrarily). The scheduling g lasts kW^* time slots. We look at the call scheduling obtained by repeating the schedule g m times. During each repetition of g we route in the network as one would do when routing the flow kf, but note that some flow units will not be routed since flow units have not yet attained the network node where they should be. The maximum number of period that a flow unit may suffer is the maximum length of a path D in the static flow routing.

During any period $p \geq D$ periods, the pipe-line is initialized and no flow will be missing (except if at least $(p-D)f$ unit of flows have been routed.

Due to this we may only lose the flow kDf. Hence we route at least $f(mk - kD)$ in kmW^* time slots. The throughput is $\frac{1}{W^*}(1 - \frac{D}{m})$, which converges to $\frac{1}{W^*}$ when m grows. □

The above bound is very pessimistic, for most practical cases optimal call weighting are almost integral and small time period achieve an almost optimal throughput (see [2]). According to it the call scheduling and the call weighting problem have equivalent throughput when τ is large and when the node buffers are large enough. We will make this hypothesis and study now the call weighting problem.

Remark 3 *Assuming τ large means that the observation time on which one evaluates the throughput is much larger than the atomic time slot. This means that the duration of a slot must be much smaller than the user perception. The minimum slot duration is limited by factors like reactivity of the radio device, speed of light, clock synchronization, necessity to lose information bits due to protocol overhead. When the user perception is human the atomic slot is several orders of magnitude below the perception time. Nevertheless, if one considers the case of a very reactive application the assumption can pose problems.*

3 Complexity of the Call Weighting Problem

3.1 A Sets & Paths Model for Call Weighting

In this section, we consider the complexity of the call weighting problem assuming that one is given an implicit definition of the call set $\mathcal{C}$. We mainly show that the dual problem is very closely related to the classical flow dual and to independent set weights. In this section $\mathcal{C}$ is not supposed to have any specific property. We first restate the Call Weighting Problem concisely as follows.

Problem 1 *The* Call Weighting Problem *can be described as follows:*

- *We are given some multi-commodity flow requests $f(u,v)$, $(u,v) \in (V,V)$;*
- *a set of feasible calls $\mathcal{C}$ (this set may be of non-polynomial size), a call being a set of arcs (subset of $E \subseteq (V,V)$).*
- *To a weight function $w : \mathcal{C} \to I\!\!R^+$, is associated an induced capacity $Cap_w : E \to I\!\!R^+$ defined by $Cap_w(e) = \sum_{C \in \mathcal{C} | e \in C} w(C)$.*
- *A weight function w (defined on $\mathcal{C}$) is* feasible *if the flow f is feasible on the graph with vertex set V and capacity function Cap_w.*

Goal : *Find a feasible weight function with minimum total weight.*

As usual in flow problems, we consider the set $\mathcal{P}$ of the dipaths in the transmission digraph. For each ordered pair of vertices $(u,v) \in (V,V)$ we consider the set of dipaths $\mathcal{P}_{uv} \subseteq \mathcal{P}$ connecting u to v. A flow Ω is a positive weight function on the dipaths set, i.e. $\Omega : \mathcal{P} \to I\!\!R^+$ satisfying

$$\forall (u,v) \in (V,V), \sum_{P \in \mathcal{P}_{uv}} \Omega(P) \geq f(u,v).$$

The capacity constraint is particular since the available capacity on the arcs is induced by the weight function w on the call set $\mathcal{C}$.

$$\forall e \in E, \sum_{P \in \mathcal{P} | e \in P} \Omega(P) \leq Cap_w(e) = \sum_{C \in \mathcal{C} | e \in C} w(C)$$

The objective is to minimize the cost function $Obj = \sum_{C \in \mathcal{C}} w(C)$. Hence, we need to decide if the following problem is feasible :

$$\begin{array}{rcc} \forall (u,v) \in (V,V),\ -\sum_{P \in \mathcal{P}_{uv}} \Omega(P) & \leq & -f(u,v) \\ \forall e \in E,\ \sum_{P \in \mathcal{P} | e \in P} \Omega(P) - \sum_{C \in \mathcal{C} | e \in C} w(C) & \leq & 0 \\ \sum_{C \in \mathcal{C}} w(C) & \leq & Obj \end{array}$$

We derive the dual using positive multipliers, let λ_{uv} be the one of the (u,v) flow equation, and $l(e)$ be the one of the capacity equation for the arc e. The derived equation is :

$$\sum_{(u,v)\in(V,V)} \sum_{P\in\mathcal{P}_{uv}} \Omega(P)(-\lambda_{uv} + \sum_{C\in\mathcal{C},e\in P} l(e)) + \sum_{C\in\mathcal{C}}(1 - \sum_{C\in\mathcal{C}|e\in C} w(C)) \leq Obj - \sum_{(u,v)\in(V,V)} \lambda_{uv} f(u,v)$$

This proves that $Obj \geq \sum_{(u,v)\in(V,V)} \lambda_{uv} f(u,v)$ whenever:

$$\begin{array}{rcl} \forall(u,v)\in(V,V), \forall P\in\mathcal{P}_{uv},\ \sum_{e\in P} l(e) - \lambda_{uv} & \geq & 0 \\ \forall C\in\mathcal{C},\ 1-\sum_{e\in C} l(e) & \geq & 0 \end{array}$$

Considering $l(e)$ as inducing a *metric* (i.e. length), and defining for a subset of transmission-arcs $S \subseteq (V,V)$:

$$l(S) = \sum_{e\in S} l(e),$$

we get the intuition of the dual problem : **Maximize** $\sum_{(u,v)\in(V,V)} \lambda_{uv} f(u,v)$, under the constraints $\forall(u,v)\in(V,V), l(\mathcal{P}_{uv}) \geq \lambda_{uv}$ and $\forall C\in\mathcal{C},\ l(C)\leq 1$. Let $d_l(u,v)$ be the distance from u to v according to l, then we can assume that an optimum dual solution satisfies $\lambda_{uv} = d_l(u,v)$.

3.2 Resolution from the Dual Separation

Property 1 *The dual problem consists in finding a metric $l : E \to \mathbb{R}^+$ on the transmission-arcs set maximizing the total distance that the traffic needs to travel ($\sum d_l(u,v)f(u,v)$) and such that the maximum length of a call is 1 ($\forall C\in\mathcal{C}, l(C)\leq 1$). Note that this is almost the classical flow dual. Indeed, if one wishes to find a flow in a network with arc-set E and capacity $c(e)$, $e\in E$, the set of* calls *is simply any set of arcs in which the arc e is repeated less than $c(e)$ times. There exists then a unique maximum length call that is obtained by picking the arc e exactly $c(e)$ times, its length is $\sum_{e\in E} l(e)c(e)$. The dual problem reduces then to the usual multi-commodity flow one : Maximize the traffic length upon the constraint $\sum_{e\in E} l(e)c(e) \leq 1$.*

Definition 3 *Given a weight function $l : E \to \mathbb{R}^+$ the* Maximum Weight Call Problem *consists in finding a call set $C\in\mathcal{C}$ for which $l(C)$ is maximum.*

The next proposition follows from general theorems on separation and optimization given by Grötschel et al [10, 11] in the exact case and by Jansen [16] in the approximate case.

Proposition 1 *If there exists a (polynomial-time) ρ-approximation for Maximum Weight Call there exist a ρ-approximation for Call Weighting.*

Proof: In order to solve the dual problem we only need to be able to separate it. So, given a metric l of the dual we need to decide if it is feasible and if not to output a violated constraint. Since one can check if l is positive and compute $\sum_{(u,v)\in(V,V)} d_l(u,v)f(u,v)$ the problem reduces to check the constraints

$$\forall C\in\mathcal{C}, l(C)\leq 1.$$

To do this one only needs to find a Maximum Weight Call C_0. If its weight is strictly more than one the constraint $l(C_0)\leq 1$ is output otherwise l is feasible. If we have a polynomial scheme to find C_0 we can do it in polynomial time, if we have a ρ-approximation scheme when $l(C_0) > \rho$ we can find in polynomial time C_1 with $l(C_1) > 1$ and output it, otherwise we know that the metric l/ρ is feasible. □

Remark 4 *The above proposition implies that we can solve the dual problem whenever we can find the longest set of calls, hence we can find the value W^*; one may wonder how one can find from this a primal solution.*

One proceeds using a Primal-Dual approach, given a primal solution one looks for a dual solution satisfying the complementary slackness conditions (this means solving a restricted dual problem). If one finds a solution the Primal is optimal and we get an optimality certificate. If not we find a polynomial number of constraints that cannot be satisfied by the dual, if one solves this problem using a classical simplex method one will find an augmentation for the Primal.

3.3 Consequences for Call Weighting in Radio Networks

In the case of radio networks, the set of feasible calls $\mathcal{C}$ is given exactly by the set of all possible independent sets in the interference graph (E_T, E_I), and the maximum weight call is the maximum weight independent set of the interference graph in which the vertex $e \in E$ receives weight $l(e)$. It follows that any ρ-approximation scheme for the maximum weight independent set on the interference graph induces a ρ-approximation scheme for the call weighting problem.

Note that since the result relies on implicit linear programming it may not provide practically efficient algorithms.

3.3.1 Hardness of the General Problem

Since call weighting is related to maximum weight independent set one may expect the problem to be hard to approximate. Indeed the next result shows that *fractional coloring*[4] is a specific case of call weighting.

Proposition 2 *Fractional coloring is a specific case of the call weighting problem on a graph G with distances $d_T = 1, d_I = 2$. Approximating the call weighting problem within $n^{1-\epsilon}$ on a graph G with $d_T = 1, d_I = 2$ is $\mathcal{NP}$-hard.*

Proof: Consider an undirected graph H. For each $v \in V(G)$ add a vertex v' connected only to v. Let the bandwidth demand be $f(v, v') = 1, v \in V(H)$, and the transmission and interference distance be 1 and 2. Then each call weighting must induce a capacity of 1 on all the arcs (v, v'). W.l.o.g. we can assume that calls contain only arcs of the kind $(v, v'), v \in H$. Since the interference distance is 2, calls are in bijection with the independent sets of H ($C \in \mathcal{C}$ if and only if $C = \bigcup_{v \in E_I} (v, v')$ for E_I an independent set of H).

It follows that call weighting on G with $d_T = 1, d_I = 2$ and fractional colorings of H are in bijection, and that the bijection preserves the cost.

The hardness result follows from the difficulty to approximate the fractional chromatic number [19]. □

3.3.2 Call Weighting and Fractional Coloring

We assume that the routing of the traffic is given, fixed or unique; either because it is imposed as an additional constraint or because of the topology (e.g. in trees).

The problem then reduces to *fractional coloring*. Indeed the routed flow induces a weight function on the transmission-arcs: $\forall e \in E_T$ the value of $load(e) = \sum_{P \in \mathcal{P} | e \in P} \Omega(P)$ is known. One then must find a minimum weight function on $\mathcal{C}$ under the constraint $\sum_{C \in \mathcal{C} | t \in C} w(C) = C_w(e) \geq load(e)$. This is exactly finding the fractional chromatic number of the interference graph (E_T, E_I) when each transmission-arc $e \in E_T$ considered as a vertex of (E_T, E_I) is repeated $load(e)$ times.

Note that an application of this approach is e.g. contained in Section 4.1.2, where one essentially fractionally colors the interference graph.

3.3.3 Hardness of the Gathering Problem

Proposition 3 *The call weighting problem on a graph G with distances $d_T = 1, d_I = 2$ is $\mathcal{NP}$-hard even if restricted to a gathering instance.*

Proof: We consider a graph $G = (A, X)$ and an integer N, we denote by m the size of the maximum independent set of G. We build an associated graph G' as follows : we start with a copy of G and we connect each vertex $x \in A$ to a new vertex x' with an edge, and we denote $A' = \bigcup_{x \in A} \{x'\}$. We add a new sink vertex s and $\forall x \in A'$ we add the edge $[x', s]$. Finally we add a set B of N new vertices and connect them to all the vertices in A. See Figure 1.

Consider the gathering instance with $f(x, s) = 1, x \in B$, each unit of traffic must travel along at least one arc (A, A') and one transmission-arc (A', s). When a transmission-arc in (A', s) is used no transmission-arc

[4]To fractionally color a graph, one simply finds a minimum weight cover of the vertex set using independent sets; if one requires the solution to be integral one gets the usual vertex coloring problem, see [11].

of the kind (A, A') can be used. Moreover, if some set of transmission-arcs $\bigcup_{x \in W \subseteq A}\{(x, x')\}$ is used then W is an independent set of the original graph G. It follows that:

$$W^* \geq \frac{N}{m} + N\,.$$

(An alternative proof would be to claim that $l(x, x') = \frac{1}{m}$ and $l(x', s) = 1$ is a feasible dual solution with cost $N(1 + \frac{1}{m})$, see Property 1.)

Consider now a maximum independent set I of the original graph. We gather the traffic as follows : we divide B into $k = N/(m-1)$ parts $B_1, B_2, \ldots, B_k$. At time 0 the vertices of B_1 transmit their information to I and at time 1, I forwards it to $I' = \cup_{x \in I}\{x'\}$. During m time units the vertices of I' will forward m traffic units to the sink, during this time when x' sends to s we keep accumulating traffic in I by performing communication between $B_2, B_3, \ldots, B_{m+1}$ and $I \setminus \{x'\}$.

After $m + 2$ time units we have $m - 1$ units of information per vertex of I we then perform a call (I, I') and restart forwarding information from I' to the sink while accumulating traffic in I (as long as all the traffic from N has not been forwarded).

Globally the scheme ensures that except for the first two slots we are forwarding information to the sink m times and refilling the set I' during one slot with m units of flow. Doing so, I will never run out of traffic to forward and the scheme will last $N + \frac{N}{m} + 2$ time slots. Hence

$$W^* \leq \frac{N}{m} + N + 2\,.$$

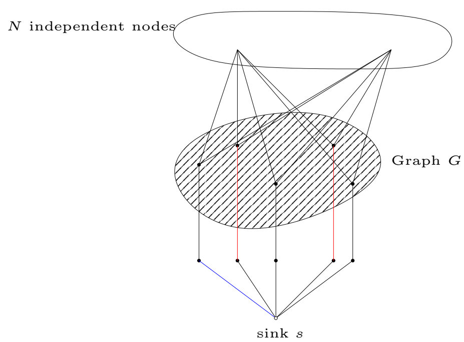

Figure 1: Hardness of a gathering instance.

□

In the proof above, as long as $1 = o(m)$ and $m = o(n)$ we have $m = (1 + o(1))(\frac{n}{W^* - N})$. Note that this means that deciding if $W^* \leq N + \frac{N}{\log N}$ is $\mathcal{NP}$-hard, this does not exclude a $(1 + \varepsilon)$-approximation but excludes an FPTAS.

Note that the gathering problem seems much "easier" than the general case, this is mainly due to the particular simplicity of single sink flows. In the case of multi-commodity flows the structure of the edge length in the dual problem is not well characterized. In the case of single sink flow the dual problem consists in associating to any node $u \in V$ a distance to the sink $d(u, s)$ and to any arc (u, v) the length $max\{0, d(u, s) - d(v, s)\}$.

Proposition 4 *The call weighting problem on a graph G with distances d_T, d_I restricted to a gathering instance admits a polynomial-time 3-approximation when $d_I \neq 2d_T$ and a 4 approximation otherwise.*

Proof: Consider a node x at distance l from the sink s in the graph G, and a shortest path P in the transmission digraph with length $l' = \lceil \frac{l}{d_T} \rceil$. Let the vertices of P be numbered $x = 1, 2, \ldots, l' = s$. The transmission-arcs $(i, i+1)$ and $(j, j+1)$ with $j \geq i$ are non-interfering when $jd_T - d_I > id_T + d_T$, that is $j - i \geq$

$\lceil \frac{d_T+d_I+1}{d_T} \rceil$. Let $k = \lceil \frac{d_T+d_I+1}{d_T} \rceil$, then the transmission-arcs $(i, i+1), (i+k, i+k+1), \ldots, (i+ak, i+ak+1), \ldots$ do not interfere. It follows that one can cover all the transmission-arcs $(i, i+1)$, and consequently the path P using $\min(k, l')$ calls. So, we can route the traffic unit from x to s using that weight; doing it for all the vertices leads to a simple Call weighting with total weight

$$\sum_{x \in V} \min\{\lceil \frac{d_T + d_I + 1}{d_T} \rceil, \lceil \frac{d(x,s)}{d_T} \rceil\}\,.$$

Now, consider the ball of radius $\lfloor \frac{d_I+1}{2} \rfloor$ centered at the sink, and assign a length of $l(e) = 1$ to any transmission-arc starting in this ball and directed toward the sink (i.e. an arc (u, v) with $d(s, u) > d(v, s)$). Since two transmission-arcs $(u_1, v_1), (u_2, v_2)$ directed toward the sink interfere whenever $d(u_1, s) + d(u_2, s) \leq d_I + 1$, this is a valid dual solution. According to this metric we have $d_l(x, s) = \min \left\{ \lceil \frac{d(x,s)}{d_T} \rceil, \left\lceil \frac{\lfloor \frac{d_I+1}{2} \rfloor}{d_T} \right\rceil \right\}$ and the cost of any call weighting is at least

$$\sum_{x \in V} d_l(x, s)\,.$$

To compare the upper and lower bounds, we only need to consider nodes that are the most expensive for the upper bound : $ub = \lceil \frac{d_T+d_I+1}{d_T} \rceil$ (i.e. $d(x, s) \geq d_T + d_I + 1$) and for which the lower bound cost is only $lb = \lceil \frac{\lfloor \frac{d_I+1}{2} \rfloor}{d_T} \rceil$. Note that $lb \geq 1$, so when $d_I \leq 2d_T - 1$, we have $\frac{d_T+d_I+1}{d_T} \leq 3 \leq 3lb$. When $d_I \geq 2d_T + 2$, we have $3lb \geq 3 \cdot \frac{d_I}{2d_T} \geq \frac{d_T+d_I+1}{d_T}$. Last when $d_i = 2d_T + k, k \in \{0, 1\}$ we get $ub = 4$ and $lb = 1$ if $k = 0$ and $lb = 2$ when $k = 1$.

The ratio between the lower and the upper bound is at most 3, but when $d_I = 2d_T$ then it has value 4. □

Note that when $d_I >> d_T$ the above result tends toward a 2-approximation. This result 2-approximation and also holds for the call scheduling problem (see [2]). So far, we miss a more exact (in-)approximability result for gathering, we believe that the problem admits a PTAS.

Conjecture 1 *There exists a Polynomial-Time Approximation Scheme for Call Weighting restricted to gathering instances.*

4 Call Weighting – Some Particular *Easy* Cases

Since the general problem is hard to approximate we study simple topologies for which polynomial time exact or approximation algorithms exist.

4.1 The Case of the Line

We study the problem on a Line (Path) P_n with nodes $1, 2, 3 \ldots, n$ and with the edges $[i, i + 1]$. We assume that nodes transmit at distance d_T and interfere at distance d_I (i.e. the interference graph is $\mathcal{I}(P_n, d_T, d_I)$). The graph distance and the l_1-distance coincide. The results that we derive are easily generalized to points in $\mathbb{R}$ that would not be regularly located. We start with an example so that the reader can get an idea of the construction.

4.1.1 An Example – Gathering on a Path with $n = 2p + 1$ Vertices

Our goal is to illustrate the problem with a very simple example. We consider the line with transmission distance $d_T = 1$ and interference distance $d_I = 2$ and the following traffic : each node v requests 1 unit of bandwidth to a unique sink. For $d_T = 2$, an optimal solution is displayed in Figure 2 when the sink is in the middle.

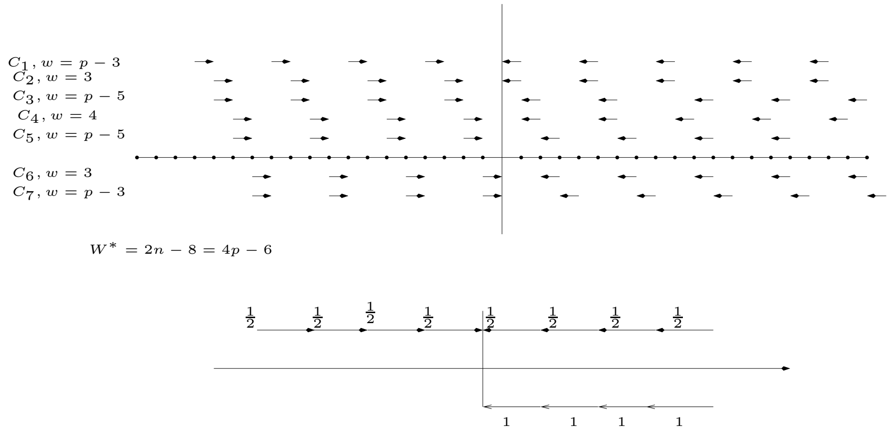

Figure 2: Gathering to the middle of a path of $n = 2p+1$ vertices with $d_T = 1, d_I = 2$. The top of the figure shows a scheme using 7 calls with the weights as displayed in the figure, and with total weight $2n-8$. The bottom of the figure provides two optimal dual solutions.

4.1.2 $d_T = 1$, W^* is Equal to the *Load* for a Single Sink

When $d_T = 1$, the transmission arcs are the usual arcs of the dipath. There is then only one simple dipath between two nodes, so the routing is fixed (forced), and the problem reduces to fractional coloring of the associated *interference* graph (see Section 3.3.2). Moreover, the interference graph $\mathcal{I}(P_n, 1, d_I)$ is almost an interval graph; indeed if it was one we could stop the study at this point as fractional coloring of interval graphs has been well-studied [20].

We will use the following notations : We assume that the sink is some node s, and we consider the two parts of the path obtained by removing s, the left (resp. right) part contains q_{left} (resp. q_{right}) nodes. We number the nodes in each semi-dipath (left and right) $0, 1, 2, 3, \ldots$ starting from s. Since the routing is unique, the transmissions are only of the form $(i, i-1)$. Let l_i denote the arc $(i, i-1)$ in the left dipath and r_i the arc $(i, i-1)$ on the right dipath. Finally, let $f(l_i)$ (resp. $f(r_i)$) denotes the demand of the left (right) node i. The load of the arc l_i (resp. r_i) will be denoted $L(i)$ (resp. $R(i)$). We have $L(i) = \sum_{j \geq i} f(l_j)$ and $R(i) = \sum_{j \geq i} f(r_j)$.

The two following subgraphs of the interference graph : P_{left} (resp. P_{right}), induced by the transmission arc in the left (resp. right) path will be of importance.

From the simple structure of the interference graph, the next proposition follows:

Proposition 5 *For a single flow traffic, W^* can be computed in linear time using a simple load argument; in the case of the gathering pattern, we have*

$$\max_{x \in \mathbb{R}} \{ \sum_{k \in [x - \frac{d_I+1}{2}, x + \frac{d_I+1}{2}]} R(k) + \sum_{j \leq \frac{d_I+1}{2} - x} L(j) \} .$$

Proof: Remember that the node l_i (resp. r_i) must be covered $L(i)$ (resp. $R(i)$ times).

According to our notation, the edges of the interference graph are then as follows :

- We have one edge between l_i (resp. r_i) and l_j (resp. r_j) when $|i - j| \leq d_I + 1$.
- There is an edge between l_i and r_j if and only if $i + j \leq d_I + 1$.

From this relation, we remark that the subgraph P_{left} (resp. P_{right}) is an interval graph[5] that can be represented by choosing to associate to l_i (resp. r_i) the interval $[i - \frac{d_I+1}{2}, i + \frac{d_I+1}{2}]$.

[5] A graph is an interval graph if its node can be represented by intervals of $\mathbb{R}$ and if there exists an edge between two nodes if and only if the associated intervals intersect [20].

We remark also that the interference graph can be cut at the sink node s and we find there two *clique cuts* (i.e. a clique that disconnect the graph), see Figure 3. We denote the first clique K_l, it contains the arc l_1 and its neighborhood inside the left dipath and the other K_r contains r_1 and its neighborhood inside the right dipath. Hence, one can first color P_{left} and then infer the coloring on P_{right}. It follows that the chromatic number of P_{left} is its maximum load, since the load of the arcs increases from left to right this value is $\chi(K_l)$. After coloring P_{left} we need to color the nodes of the right dipath, at this stage we need to take into account the fact that some nodes (transmission-arcs) are intersecting K_l, indeed we are simply coloring the subgraph induced by $P_{right} \cup K_l$.

This can be easily done by introducing new nodes and keeping the interval structure[6]. Indeed to the node l_j we associate the interval $[-\infty, \frac{d_I+1}{2} - j]$ and for r_i the situation is unchanged : we use the interval $[i - \frac{d_I+1}{2}, i + \frac{d_I+1}{2}]$.

To conclude about the fractional chromatic number, we simply need to find the maximum load on the interval graph that is :

$$\max_{x \in \mathbb{R}} \sum_{k \in [x - \frac{d_I+1}{2}, x + \frac{d_I+1}{2}]} R(k) + \sum_{j \leq \frac{d_I+1}{2} - x} L(j).$$

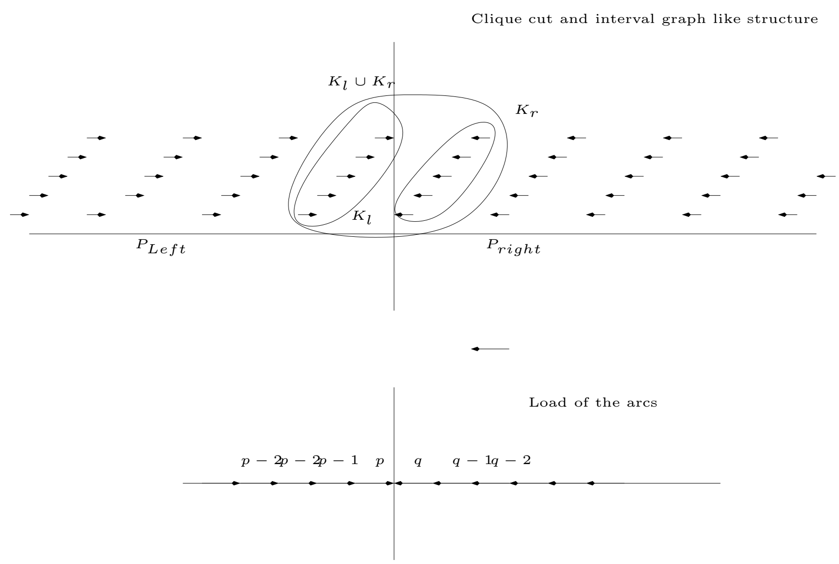

Figure 3: The interval graph structure, the 2-clique cut, and the central clique. The load of the arcs is given for the case of uniform traffic demands.

.

□

By following the convention $[u, v] = \emptyset$ if $u > v$ and defining $\mathbb{1}_k = 1$ if k is odd and 0 if not, we characterize the uniform case in the next result.

[6] This implies that the chromatic number of the graph is the one of $K_l \cup K_r$.

Corollary 1 *For the line with $n = p+1+q, p \geq q$ nodes, the minimum call weighting cost for the gathering at a sink located at $p+1$ is*

$$W^* = \max\{ \sum_{i=1}^{\min(d_I+2,p)} (p-i+1) + \sum_{i=1}^{\min(\max(0,d_I+1-p),q)} (q-i+1), \sum_{l=\lfloor \frac{p+q-d_I+1}{2} \rfloor}^{p} l + \sum_{l=\lfloor \frac{p+q-d_I+1}{2} \rfloor}^{q} l + 1\!\!1_{p+q-d_I+1} \lfloor \frac{p+q-d_I+1}{2} \rfloor \}$$

In particular, if $p, q \geq d_I + 2$ and $\Delta = p - q \geq 0$

$$W^* = \begin{cases} (d_I+2)p + O(1) & \text{if } \Delta \geq d_I + 2, \\ (d_I+2)q + O(1) & \text{if } \Delta < d_I + 2. \end{cases}$$

Therefore $W^* = \Omega((d_I + 2)p)$.

Proof: In the case of uniform traffic demands $L(i) = p - i + 1$ (resp. $R(i) = q - i + 1$) and for $i \in]-\infty, 0] \cup [p+1, +\infty[, L(i) = 0$ (resp. $i \in]-\infty, 0] \cup [q+1, +\infty[, R(i) = 0$). According to Proposition 5 the cost is, $W^* = \max_{x \in \mathbb{R}} \{\sum_{i \in [x - \frac{d_I+1}{2}, i + \frac{d_I+1}{2}]} R(i) + \sum_{j \leq \frac{d_I+1}{2} - x} L(j)\}$. Note that the value of $SL(x) = \sum_{j \leq \frac{d_I+1}{2} - x} L(j)$ increases when x decreases. Now, consider the value of $SR(x) = \sum_{i \in [x - \frac{d_I+1}{2}, i + \frac{d_I+1}{2}]} R(i)$. Since $R(i)$ decreases with i, if $[x - \frac{d_I+1}{2}, x + \frac{d_I+1}{2}]$ is not empty and do not contains 1 then the value of $SR(x)$ will increase if we choose $x - 1$. Since $SL(x)$ will increase too we conclude that $[x - \frac{d_I+1}{2}, i + \frac{d_I+1}{2}]$ is either empty either of the form $[1, k]$. The summation interval for SL is then $[1, k']$ with $k + k' \leq d_I + 2$. Assuming that $p \geq q$ we get as maximum value of

$$W^* = \max_{0 \leq k \leq d_I+2} \sum_{i=1}^{k} L(i) + \sum_{j=1}^{d_I+2-k} R(j). \tag{1}$$

Depending on the values this maximum is attained for one of the two possible situations:

- Either we get the maximum amount of terms if the left sum $A = \sum_{i \in [1, d_I+2]} L(i)$ and some additional right terms $\sum_{j \in [1, d_I+1-\min(p, d_I+2)]} R(j)$;
- Or, when the terms in the left are getting small it is better to introduce right terms. Let us study the last term $B = \max \sum_{i \in [1,\ldots k]} R(i) + \sum_{j \in [1, d_I+1-k]} L(j)$, this sum is the load of $d_I + 2$ consecutive arcs, then it looks like $(p + (p-1) + \ldots p') + (q + (q-1) + \ldots q')$ with $p' = p - k + 1, q' = q - (d_I + 1 - k) + 1$, which is maximum only if $p' = q' = x$ or $p' = x, q' = x + 1$. It follows that $x = \lfloor \frac{p+q-d_I+1}{2} \rfloor$, hence

$$B = \sum_{l \in [x,p]} l + \sum_{l \in [x,q]} l$$

To study the order, we go back to (1) and calculate it. Now, assume first that $\Delta \geq d_I + 2$, so the best choice is to keep the $d_I + 2$ arcs from the longest side (left). We have then that

$$W^* = \sum_{i=1}^{d_I+2} p - i + 1 = (d_I + 2)p + O(1).$$

On the other hand, the loads decrease with Δ and as this happens, it becomes more convenient to discard arcs from the left side and take some from the right one. We can bound them term by term to get that

$$\begin{aligned} W^* &\geq (q - \lfloor (d_I+2)/2 \rfloor) + \ldots + (q-1) + q + \\ & \quad q + (q-1) + \ldots + (q - \lfloor (d_I+2)/2 \rfloor) + \\ & \quad 1\!\!1_{d_I+2}(q - (d_I+3)/2) \\ &= 2 \sum_{i=1}^{\lfloor (d_I+2)/2 \rfloor} q - i \; + \; 1\!\!1_{d_I+2}(q - (d_I+3)/2) \\ &\geq (d_I+2)q + O(1). \end{aligned}$$

Therefore the result follows.

□

The above formula can be made more explicit if one assume conditions on d_I, p, q. The explicit formula can be found in [1], in which the specific problem of the line is studied exactly in the non systolic model. Here we give the order of W^*. For the gathering problem in the line with $n = p + 1 + q$ nodes in the uniform case, if $\Delta = p - q \geq 0$ and the sink located at p.

4.1.3 Arbitrary Traffic Pattern on the Line

According to Proposition 1, it is enough to prove that the maximum weight independent set can be solved in polynomial time on $\mathcal{I}(P_n, d_T, d_I)$. This can be easily performed by a standard left right dynamic program. Assume that some independent arcs $(u_i, v_i), i \in I$ have been chosen with all senders $v_i, i \in I$ before node x. In order to keep generating an independent set one needs only to remember three values : the leftmost sender, the leftmost receiver and the cost of the solution : $\max_{i \in I} u_i, \max_{i \in I} v_i$ and $\sum_{i \in I} l(u_i, v_i)$. This provides us with an implicit linear program method to solve the dual problem in polynomial time and suggests like in [3, 4] that we may find a direct approach using dynamic programming.

This approach generalizes to trees, but the knowledge of the closest sender and receiver does not suffice to encode the sub-solutions. Nevertheless, since in a non-interfering pattern one cannot find more than one transmission-arc using an edge of the graph (in a direction or another) sub-solutions can still be encoded using a polynomial size *trace* (i.e. encoding). Note that the approach that we use is close to the one of Erlebach and Jansen to find edge-disjoint paths on trees [6, 5].

4.1.4 Arbitrary Traffic Pattern on Trees

Proposition 6 *If T is a tree, the Maximum Call problem can be solved in polynomial time on $\mathcal{I}(T, d_T, d_I)$.*

Proof: Consider a tree T rooted at a node x with degree Δ. The subtrees $T_k, k \in S$ rooted at x are indexed by a set $|S| \leq \Delta$ and have vertex sets $V_k, k \in S$. We denote $V' = \cup_{k \in S} V_k$. For the subtree T_k one needs to represent a partial solution corresponding to a call C in a compact way. For this we consider the set $I_k = (V_k, V_k) \cap C$ of *internal* transmission-arcs (i.e. performed inside the subtree), and the set E_k of external transmission-arcs (i.e. performed between V_k and $V \setminus V_k$). If one looks how a partial solution interacts with the global one, it turns out that there can be at most one external transmission-arc. Hence one can encode E_k by storing in a variable e_k the node $u \in V_k$ from which or to which the external transmission arc will finally be directed (this uses a space of $2|V| + 1$). The set I_k impacts on the global solution only according to

- the distance from the root of the sender in I_k that is $s_k = \min\{d(x, u), (u, v) \in I_k\}$;

- the distance from the root of the closest receiver in I_k which is $r_k = \min\{d(x, u), (u, v) \in I_k\}$;

Last, the local solution impacts also by the value of $l_k = |I_k|$.

It follows that a sub solution can be encoded with the 4 values (s_k, r_k, l_k, e_k), so it is possible to use one table T_k per subtree containing all the quadruples corresponding to one solution. One builds then the table for T as follows : one considers the Cartesian product of the $|S|$ tables and each $|S|$-tuple $\{(s_k, r_k, l_k, E_k), k \in S\}$

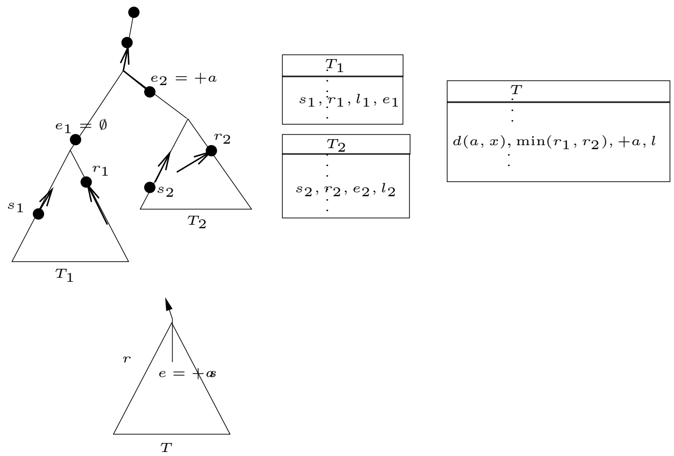

Figure 4: Example of table merging

of entries *merged* in an intuitive way. The process (see Figure 4) is tedious but simple, basically one needs to associate two by two the external transmission-arcs, to decide if the root will be involved in an external transmission-arc and to check that the interference rules are satisfied. We let the reader convince himself that the tables contain enough information to build the global table.

The problem is that even if the individual tables have sizeat most $M = d_I^2 2|V||V|$, the Cartesian product may have an exponential size M^Δ when the degree grows. this problem can be solved by merging tables two by two in order to generate a table for the union of the subtrees $T_1, T_2, \ldots T_j$. Again one can encode the partial solution in a group of subtrees by keeping only exactly the same 4 numbers.

□

Corollary 2 *If T is a tree the maximum weight independent set problem on $\mathcal{I}(T, d_T, d_I)$ admits an FPTAS.*

Proof: The algorithm is the same as in the unweighted case, except that we have to record the weight of sub-solutions in the tables, we denote $l(e)$ the weight of some transmission-arc e and $L = \max_{e \in E} l(e)$. For a given triple (r, s, e) we keep only the best solution. As always, the load function l is encoded on a number of bit polynomial in $\log n \log \varepsilon$.

□

Corollary 3 *The call weighting problem on trees with distances d_T, d_I admit an FPTAS.*

Proof: Follows from Proposition 1 and Corollary 2. □

Explicit small linear programs for the line. We show now how to solve for the line the primal problem with an explicit (polynomial-size) linear program. For this we use the fact that due to the small tree-width independent sets can be expressed by simple constraints, in a way similar to the one used to fractionally color circular arc graph (see the work by Tucker [20] or Kumar [18]). Basically, one only needs to express some compatibility constraint on each small cut. On interval graphs, the convex hull of the independent sets is defined by writing *the load of each part of the line if at most* 1. For circular arc graphs Tucker gave a flow formulation to solve the fractional coloring of those graphs and mainly remarked that the convex hull of independent sets could be expressed by simple flow equations.

Proposition 7 *The call weighting problem on a line with distances d_T, d_I for an arbitrary traffic can be solved in polynomial time using a linear program with $\Theta(n^2 d_I^2)$ variables and nd_I constraints.*

Proof: We represent an arc (u, v) by the directed segment (u, v). First we notice that if A is an independent set then its segments are non-intersecting and can be ordered from the left to the right. One then easily checks that a set of disjoint ordered segments $\{(u_0, v_0), (u_1, v_1), \ldots, (u_k, v_k)\}$ is an independent set if and only if $\forall i, (u_i, v_i)$ and (u_{i+1}, v_{i+1}) are non-adjacent (i.e. adjacency only needs to be checked for each two successive segments).

We represent this constraint by using the next auxiliary flow network:

- The network contains a source s and a sink p.
- The vertices are the potential arcs (transmissions) (u, v).
- (u, v) is connected with an arc with infinite capacity to (u', v') if (u', v') and (u, v) are not interfering and (u', v') is to the right of (u, v).
- Any vertex (u, v) is connected to the source (resp. sink) with an arc $start(u, v)$ (resp. $stop(u, v)$).

Consider now a flow function with value f from s to p, and decompose this flow into a set of weighted dipaths. Each dipath is a sequence q of transmission-arcs ordered from let to right that are by construction not interfering. Hence it corresponds to an independent set of transmission-arcs.

It follows that a flow function with value f corresponds to a weight functions on the calls with sum f; the cost of a solution will hence be represented by the value f of the flow.

It remains to express conditions so that the call weighting associated to a flow will be feasible. Note that, according to the above construction, the induced capacity on the transmission-arc (u, v) is exactly the amount of flow that crosses the vertex (u, v). It follows that the induced capacity $Cap_w(u, v)$ on the transmission-arc (u, v) can be expressed by adding $2d_T n$ (one per transmission-arc) variables and constraints. At that stage of the formulation we are back to a classical multi-commodity flow problem since the capacity function $Cap_w(u, w)$ has been expressed (see Problem 1). One completes the linear program by introducing a general flow problem with demand $f(u, v)$ from u to v and capacity $Cap_w(u, v)$ on the transmission-arc (u, v). □

Remark 5

- *In the special case $d_T = 1$, the second flow problem can be skipped since the routing being unique one knows what is the necessary induced capacity for transmission-arc (u, v). One simply computes the values $load(u, v)$ and adds the following constraints : the flow going through (u, v) is at least $load(u, v)$.*
- *One gets the solution by decomposing the flow into paths, each path giving rise to some weighted call.*

Polynomial size linear program for trees. Here again one can use the same approach as for the line. We mainly mimic the dynamic program but associating to it a linear program that we build dynamically. In a bottom-up approach we will keep for each subtree group a table of entries and for each entry (s, r, E) one variable $X_{s,r,E}$ that counts the total weight assigned to independent sets having this entry as trace (i.e. encoding).

One needs then to express, using linear constraints, the values of the group formed by combining two groups. The most immediate approach simply consists in looking at the set of possible patterns $\mathcal{P}$ made of 2 independent sets traces $(s_1, r_1, E_1), (s_2, r_2, E_2)$ that can be merged into a set having trace (s', r', E'). One introduces then one new variable x_P per pattern and equations stating that $x_{s,r,E} = \sum_{P \in \mathcal{P}, (s,r,E) \in P} x_P$. Note that the induced capacity of each transmission-arc can easily be expressed from the other variables.

Remark 6 *Note that this linear formulation is not compact, the number of patterns being around $(2nd_Id_T)^3$. One can decrease this value by using a more sophisticated dynamically generated linear program, but we are currently unable to obtain something as compact as in the line case.*

As one expected, the results stated for trees generalize too the case of bounded tree-width graphs. One simply need to record information with respect to each node of the small cut. The size of the tables will be like $a = (n^2 d_I d_T)^T$ for a tree-width T to solve the maximum independent set problem on $\mathcal{I}(G, d_T, d_I)$. One can also build explicit linear programs with like a^3 variables.

4.2 PTAS for Grid or Euclidean Graphs for Fixed d_T, d_I

A construction similar to the one used to show the non approximability of the general problem implies that the call weighting problem is $\mathcal{NP}$-hard even when restricted to the 2 dimensional grid. The difficulty is coming from the hardness to find a maximum independent set, to color or fractionally color unit disk graphs. Nevertheless the maximum weight independent set problem on the interference graph is easy to approximate using a standard locality idea combined with shifting (see e.g. [13, 12, 7]). We do the same for $\mathcal{I}(G_n, d_T, d_I)$, where G_n is the 2-dimensional square grid of n nodes.

Proposition 8 *If G is a 2 dimensional grid, for any $k \in I\!N$, call weighting can be solved with precision $1+\frac{1}{k}$ in polynomial time (with a multiplicative factor of 2^{k^2} in the running time).*

Proof: Assume that I^* is the optimal independent set and let w^* be its weight. We associate to each call (u,v) the disk of u and radius d_I. Let $k > 1$ be a fixed integer and define A_k as the set of points with $x = kpd_I$ or $y = kpd_I$, and let $A_{k,i} = A_k + (id_I, id_I)$. Note any disk representing an interference (u,v) intersects only one of the sets $A_{k,i}$. (See Figure 5.) Since there are k such sets, it follows that there exists some j such that the weight of the elements of I^* that intersect $A_{k,j}$ is less than w^*/k.

So to solve the problem with precision $\frac{1}{k}$ we simply need to solve it for each i when the disk intersections $A_{k,i}$ are removed. But in this case the call in different cells of the grid cannot interfere, and we simply need to solve the problem in each cell locally. Since each cell is of size kd_I it contains at most $\theta(k^2)$ independent transmissions, so the maximum independent set in each cell can be computed in $\Theta(2^{k^2})n^2$ time.

The result follows from Proposition 1. □

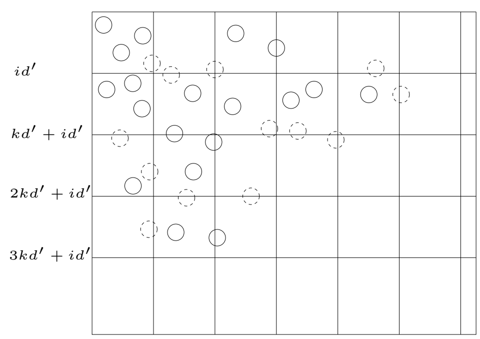

Figure 5: PTAS on the grid. Dashed circle are erased and the other calls can only interfere if they are in the same cell.

Corollary 4 *If G is a d dimensional grid, for any $k \in I\!N$, call weighting can be solved with precision $1+\frac{1}{k}$ in polynomial time (with a multiplicative factor of 2^{k^d} in the running time).*

Proof: Follows using the shifting method, we find at most k^d in each kD_I cell. □

Remark 7

- *If one considers instead of a grid point in R^d and a metric interference graph we obtain the same result.*
- *If instead of assuming that transmission-arc and interference relation are defined according to balls of radii d_T, d_I one assumes some irregular transmission-arc or interference areas, the technique can still be applied as long as the shape are too far from balls. Two conditions need to be fulfilled : $I(u) \subseteq T(u)$ and some locality is preserved, $T(u) \subset Ball(u, r_1) \subseteq I(u) \subseteq Ball(u, r_2)$ with $\frac{r_2}{r_1}$ bounded (where $Ball(u,r)$ denotes the balls around u of radius r).*

5 Conclusion

We have characterized the complexity of the call weighting problem and given algorithms to solve it on the most practical topologies. Some questions remain open :

- Whereas the algorithms presented for the line and trees are working well in practice (D. Coudert nicely implemented them and they work really fast), the PTAS given for the 2-dimensional grid is purely theoretical, and the one for bounded tree-width graphs is unpractical as soon as the tree-width grows. Can one get more practical algorithms for those cases?
- Is the problem $\mathcal{NP}$-hard in the 2-dimensional grid in the case of gathering ?
- Can one get simple good approximation (better than 3) for the gathering problem in the general case ? Can one get PTAS for the gathering problem in the general case ?
- Is it possible to give purely combinatorial approximation algorithms that would not use linear programming or gradient methods?

Acknowledgments

We wish to thank Jérome Galtier and Alexandre Laugier for introducing us to the *Internet in Village* problem. A special thanks to David Coudert for helpful discussions and for implementing the explicit linear program on the line efficiently.

References

[1] Bermond, J.-C., Corrêa, R., and Yu, J. Gathering in radio networks, the one dimensional case. Manuscript, 2004.

[2] Bermond, J.-C., Klasing, R., Morales, N., and Perennes, S. Gathering traffic in radio networks, the grid and path cases. In preparation, 2004.

[3] Caragiannis, I., Ferreira, A., Kaklamanis, C., Pérennes, S., Persiano, P., and Rivano, H. Approximate constrained bipartite edge coloring. In *27th International Workshop on Graph-Theoretic Concepts in Computer Science (WG'01)* (Boltenhagen, Germany, June 2001), A. Brandstädt and V. Le, Eds., vol. 2204 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 21–31.

[4] Caragiannis, I., Ferreira, A., Kaklamanis, C., Pérennes, S., and Rivano, H. Fractional path coloring with applications to WDM networks. In *Proc. of the 28th ICALP* (Crete, Greece, jul 2001), F. Orejas, P. G. Spirakis, and J. van Leeuwen, Eds., vol. 2076 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 732–743.

[5] Erlebach, T. Maximum weight edge-disjoint paths in bidirected tree. In *Communication and Data Management in Large Networks. Workshop of INFORMATIK'99* (1999), pp. 13–19.

[6] Erlebach, T., and Jansen, K. Maximizing the number of connections in optical tree networks. In *Proceedings of the Ninth Annual International Symposium on Algorithms and Computation (ISAAC'98)* (1998), vol. LNCS 1533, Springer Verlag, pp. 179–188.

[7] Erlebach, T., Jansen, K., and Seidel, E. Polynomial-time approximation schemes for geometric graphs. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms, SODA'01* (2001), pp. 671–679.

[8] Flammini, M., and Pérennes, S. Lower Bounds on systolic gossiping. *Information and Computation*. To appear.

[9] Gasieniec, L., and Potapov, I. Gossiping with unit messages in known radio networks. In *Proceedings of the IFIP 17th World Computer Congress - TC1 Stream / 2nd IFIP International Conference on Theoretical Computer Science* (2002), Kluwer, B.V., pp. 193–205.

[10] Grötschel, M., L.Lovász, and Schrijver, A. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica 1* (1981), 169–197.

[11] Grötschel, M., Lovász, L., and Schrijver, A. *Geometric Algorithms and Combinatorial Optimization*, second corrected ed. Springer-Verlag, 1993.

[12] Harry B, H. I., Marathe, M. V., Radhakrishnan, V., Ravi, S. S., Rosenkrantz, D. J., and Stearns, R. E. NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. *Journal of Algorithms 26*, 2 (1998), 238 – 274.

[13] Hochbaum, D. S., and Maass, W. Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM 32(1)* (1985), 130–136.

[14] Hromkovic, J., Klasing, R., Pardubská, D., and J.Waczulik. Effective systolic algorithms for gossiping in cycles. *Parralel Processing Letters 8*, 2 (1998), 197–205.

[15] Hromkovic, J., Klasing, R., Pardubská, D., Unger, W., and Wagener, H. The complexity of systolic dissemination of information in interconnection networks. *R.A.I.R.O. Theoretical Informatics and Applications 28*, 3-4 (1994), 303–342.

[16] Jansen, K. Approximate strong separation with application in fractional graph coloring and preemptive scheduling. *Theoretical Computer Science 1-3(302)* (2003), 239–256.

[17] Kowalski, D. R., and Pelc, A. Time of deterministic broadcasting in radio networks with local knowledge. *Siam Journal on Computing 33*, 4 (2004), 870–891.

[18] Kumar, V. Approximating circular arc coloring and bandwidth allocation in all-optical ring networks. In *Proc. of the International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX'98)* (1998), vol. 1444 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 147–158.

[19] Lund, C., and Yannakakis, M. On the hardness of approximating minimization problems. *Journal of the ACM 41* (1994), 960–981.

[20] Tucker, A. Coloring a family of circular arcs. *SIAM Journal of Applied Mathematics 29*, 3 (1975), 493–502.