Computing and maximizing the exact reliability of wireless backhaul networks

David Coudert, James Luedtke, Eduardo Moreno, Konstantinos Priftis

Computing and maximizing the exact reliability of wireless backhaul networks [1]

David Coudert [a], James Luedtke [b], Eduardo Moreno [c] and Konstantinos Priftis [a]

[a] *Université Côte d'Azur, Inria, CNRS, I3S, France*
[b] *University of Wisconsin-Madison, Madison, WI, USA*
[c] *Universidad Adolfo Ibáñez, Santiago, Chile*

Abstract

The reliability of a fixed wireless backhaul network is the probability that the network can meet all the communication requirements considering the uncertainty (e.g., due to weather) in the maximum capacity of each link. We provide an algorithm to compute the exact reliability of a backhaul network, given a discrete probability distribution on the possible capacities available at each link. The algorithm computes a conditional probability tree, where at each leaf in the tree a valid routing for the network is evaluated. Any such tree provides bounds on the reliability, and the algorithm improves these bounds by branching in the tree. We also consider the problem of determining the topology and configuration of a backhaul network that maximizes reliability subject to a limited budget. We provide an algorithm that exploits properties of the conditional probability tree used to calculate reliability of a given network design, and we evaluate its computational efficiency.

Keywords: Network design, Backhaul network, Reliability, Optimization.

1   Introduction

Microwave communications provide a cost-efficient technology for deploying fixed wireless backhaul networks [6]. The capacity of microwave links is mainly determined by the channel bandwidth and the modulation scheme used to transmit data. The reliability of a network is defined as the probability that the network serves all the communication requirements, considering uncertainty in the link capacities that are caused by environmental conditions. In particular, in case of bad weather, adaptive modulation technology (i.e., the ability to change modulation on the fly) enables preserving functionality of a network link by reducing its capacity. This characteristic makes it very difficult to compute the exact reliability of a network [7] and several methods to approximate this value have recently been proposed [1,2,5].

We consider a model as in [1,2,5] in which the available capacity on each link is a random variable given by a discrete probability distribution, which depends on the chosen modulation and the possible weather conditions of the link. The available capacities on the links are assumed to be independent of each other. We provide an algorithm to compute the exact reliability of a backhaul network in this setting. The algorithm computes a *conditional probability tree* (CPT), where at each leaf a valid routing of the traffic in the network is evaluated. In this work we consider the use of Ethernet technology for routing, and so the traffic is routed along a spanning tree that can be changed according to the actual link capacities, but our method is valid for any routing. Any given CPT can be used to compute an upper and lower bound of the reliability, and the algorithm improves these bounds by branching in the tree.

We also consider the problem of determining the optimal network topology and its configuration to maximize the reliability subject to a limited budget. The first optimization algorithm we present is based on combinatorial Bender cuts [3] which are derived using our proposed algorithm for calculating reliability of a given network. We also provide a second topology and configuration optimization algorithm that exploits properties of the CPT used to calculate reliability of a given network design. In particular, any such CPT can be used to build a mixed-integer programming model that yields an upper bound on the maximum reliability. After solving such a model, the optimal solution can then be used to refine the CPT, and the process is repeated until the optimal value is found.

The problems studied in this paper have first been investigated in [1,2] using a chance-constrained programming formulation. However, the proposed

formulation requires using a unique routing, independently of the conditions, and so the usage of network resources is not fully optimized. Later, a mixed-integer linear programming formulation based on column generation has been proposed [5], where the columns are network configurations (a capacity per link) and the routing may change for each configuration. Hence, it provides a better approximation of the reliability of the network compared to previous works. However, its drawback is in the management of the many configurations with very low probabilities and so it unable to compute the exact reliability of a network. The formulation proposed in this paper overcomes the limitations of previous proposals since it can compute the exact reliability of a given network, is able to design networks with maximized reliability for fixed budget, and can be used with any kind of routing.

2   Notation and problem definition

Let $G = (V, E)$ be an undirected graph composed of a set of nodes V and links E, representing the backhaul network. For each pair of nodes in V we denote the required demand as $D_{i,j}$ for $i, j \in V$, $i \neq j$. Let $\mathcal{T}$ be the set of all spanning trees in the network G. Given a spanning tree $T \in \mathcal{T}$, if we route the demand D using only links in T, then each link $e \in E$ has an aggregated required capacity denoted by d_e^T. Note that $d_e^T = 0$ if $e \notin T$. We call T a *routing tree* for G.

Links in the network can operate with different bandwidths or capacities. For a link $e \in E$, we let $C_e = \{c_{e,i} : i = 1, \ldots K_e\}$ denote the set of possible capacities. We assume that $c_{e,i} < c_{e,i+1}$ for $i = 1, \ldots, K_e - 1$. Given a vector of capacities $\hat{c}$ for each link, with $\hat{c}_e \in C_e$ for each $e \in E$, we say that a spanning tree $T \in \mathcal{T}$ is *compatible* with $\hat{c}$ if the routing defined by T is feasible with respect to link capacities $\hat{c}$, i.e., if $d_e^T \leq \hat{c}_e$ for each $e \in E$.

Suppose that the available capacity at each link e of the network is a random variable with a discrete distribution among the available capacities C_e. Once the random capacities are observed, the routing tree can be changed, and hence the network will meet the required capacities if a tree compatible with the observed capacity exists. Hence, we define the *reliability* of the network as the probability that there exists at least one compatible spanning tree:

$$\mathbb{P}_\xi \left(\exists T \in \mathcal{T} : d_e^T \leq \hat{c}_e(\xi) \ \forall e \in E \right),$$

where $\hat{c}_e(\xi)$ denotes the random available capacities.

In Section 3 we study the problem of computing the exact reliability of a

given backhaul network. In Section 4 we study the optimal design of a backhaul network. In this setting, for each link we must choose which technology to install, among a set $\mathcal{K}$ of possible technologies. Each technology in $\mathcal{K}$ has an installation cost, and there is a total budget B for designing the network. In addition, the technology chosen also defines the maximum possible capacity of the link. Then, given the uncertain weather conditions on a link the actual capacity will be the maximum of what is available due to the installed technology, and what the weather conditions allow. The design problem is then to choose a capacity (or none at all) for each link in the network such that the budget is not exceeded and the reliability of the network is maximized. In Section 5 we present results of our computational experience with the proposed methods.

3 Computation of the exact reliability of a network

The idea to compute the reliability of a network is to construct a binary conditional probability tree (CPT), where each node of the CPT has an associated set of lower and upper bounds on the possible capacities for each link in E, representing the possible scenarios associated to that node. At each node, two branches can be created, by imposing a new lower or upper bound on the possible capacity at a selected link. Each node of the CPT has an associated probability equal to the proportion of scenarios included in that node, denoted by p_s for each node $s \in \mathcal{S}$.

A node in the CPT can have three states: *infeasible*, if there is no routing tree in $\mathcal{T}$ compatible with all the capacity scenarios in the node; *feasible*, if there is a $T \in \mathcal{T}$ such that T is compatible with all the scenarios in the node; and *undetermined*, where no $T \in \mathcal{T}$ is compatible with all scenarios but there is a $T \in \mathcal{T}$ compatible with some scenarios in the node.

The algorithm to compute the reliability of a backhaul network constructs this CPT iteratively. Given an *undetermined* leaf of the CPT, it selects one $T \in \mathcal{T}$ compatible with some scenario of the node, and creates two new nodes by imposing a lower and upper bound on the capacities of a particular link of T. In this way, the algorithm splits an undetermined node into several feasible or infeasible nodes. Once all leaves of the CPT are either feasible or infeasible, the reliability of the network is the sum of the probabilities associated to the feasible leaves.

Algorithm 1 presents the details of the algorithm. The algorithm uses two auxiliary functions named EXPLORENODE and SPLITNODE. EXPLORENODE determines the state of a leaf in the CPT. Assume the leaf s is defined by lower

Algorithm 1. Compute the reliability of a backhaul network

1: **procedure** COMPUTERELIABILITY(Conditional probability tree $\mathcal{S}$)
2:     **while** exist *undetermined* leaf s of $\mathcal{S}$ **do**
3:         EXPLORENODE(s)
4:     **return** $\displaystyle\sum_{feasible\ s\in\mathcal{S}} p_s$

5: **procedure** EXPLORENODE($s \in \mathcal{S}$)
6:     **if** no $T \in \mathcal{T}$ is compatible with scenarios in s **then**
7:         s is *infeasible*
8:         **exit**
9:     **else if** there exists a $T \in \mathcal{T}$ compatible with all scenarios in s **then**
10:        s is *feasible*
11:        **exit**
12:     **else**
13:        Choose $T \in \mathcal{T}$ compatible with some scenario in s
14:        SPLITNODE(s, T)

15: **procedure** SPLITNODE($s \in \mathcal{S}, T \in \mathcal{T}$)
16:     **if** T is compatible with all scenarios in s **then**
17:        s is *feasible*
18:        **exit**
19:     **else**
20:        Choose $e' \in E$ such that there is a scenario ξ in s where $\hat{c}_{e'}^{\xi} < d_{e'}^{T}$
21:        Let c' be the minimum capacity in $C_{e'}$ greater than or equal to $d_{e'}^{T}$
22:        Branch leaf s into two new leaves:
23:          $s' : \bar{c}_{e'}^{s'} = c', \bar{c}_{e}^{s'} = u_{e}^{s}, e \neq e', \underline{c}^{s'} = \underline{c}^{s}$
24:          $s'' : \underline{c}_{e'}^{s''} = c', \underline{c}_{e}^{s''} = \underline{c}_{e}^{s}, e \neq e', \bar{c}^{s''} = \bar{c}^{s}$
25:        s' is *undetermined*
26:        SPLITNODE(s'', T)

bounds $\underline{c}_{e}^{s}$ and strict upper bounds $\bar{c}_{e}^{s}$ on the random capacity of link $e \in E$. Then the leaf s is *infeasible* if there is no routing tree in $\mathcal{T}$ compatible with the upper bounds $\bar{c}^{s}$. Conversely, the leaf s is *feasible* if there is a routing tree $T \in \mathcal{T}$ compatible with the lower bounds $\underline{c}^{s}$. Note that this is easy to check because we know the set $\mathcal{T}$ of all routing trees of the network and their capacity requirements. If a leaf is neither *infeasible* nor *feasible*, then the algorithm selects a $T \in \mathcal{T}$ that is compatible with some scenario in the leaf. Since T is not compatible with all scenarios, there exists some scenario in which the capacity of some link $e' \in T$ is smaller than the demand d_{e}^{T}. Thus, the procedure SPLITNODE branches the scenarios in this leaf into new leaves, imposing new upper and lower bounds on the capacities for link e', respectively. Hence, T is compatible with at least one scenario in the new leaf

where a lower bound has been imposed. So, the procedure recursively calls procedure SPLITNODE over this new leaf, eventually obtaining a *feasible* leaf.

At any intermediate step of the algorithm a lower bound on the reliability of the network can be obtained by summing the probabilities associated to all *feasible* leaves. Similarly, an upper bound on the reliability can be obtained by summing the probabilities associated to both *feasible* and *undetermined* leaves. Hence, it is possible to stop the algorithm before it finishes when a desired precision on the reliability is attained.

4 Maximizing the exact reliability of a network

We propose two mixed-integer programming (MIP) based methods for maximizing the reliability of a backhaul network subject to a budget constraint. The first method is a direct application of the combinatorial Bender cuts method proposed in [3]. The second method modifies the algorithm from Section 3 for computing the reliability by solving a MIP problem at each step in order to decide the design and the status of each leaf in the CPT.

We start by defining binary variables x_e^k for each link $e \in E$ and possible technologies $k \in \mathcal{K}$, where $x_e^k = 1$ represents that technology level installed on link e is at least k, and $x_e^k = 0$ otherwise. We assume the if a given capacity k is installed, then it is also feasible to operate the link at any capacity level smaller or equal to that of technology k. Let b_e^k be the incremental cost of installing capacity level k instead of capacity level $k - 1$ on link e, and let B the total available budget. Then, a feasible design of a backhaul network must satisfies the following set of constraints:

$$x_e^{k+1} \leq x_e^k \qquad \forall e \in E, k \in \mathcal{K} \tag{1}$$

$$\sum_{e \in E} \sum_{k \in \mathcal{K}} b_e^k x_e^k \leq B \tag{2}$$

$$x_e^k \in \{0, 1\} \qquad \forall e \in E, k \in \mathcal{K}. \tag{3}$$

Constraints (1) reflect that if the installed technology is at least $k + 1$, then it is also at least k and constraint (2) limits the installation cost to our given budget.

4.1 Combinatorial Bender cuts

In the first method, we introduce a continuous variable θ that represents the reliability of the resulting network. We start with the following "master

problem" mixed-integer program:

$$\text{(MP)} \ \max \ \{\theta : (1), (2), (3), \ 0 \le \theta \le 1\} \tag{4}$$

Given a feasible solution $(\hat{\theta}, \hat{x})$ to (MP), we then use the algorithm COM-PUTERELIABILITY from Section 3 to compute the reliability $\hat{r}$ of the network defined by $\hat{x}$. Given $\hat{x}$ and its computed reliability $\hat{r}$, we add to the problem (MP) the following valid inequality to bound the value of θ:

$$\theta \le \hat{r} + \sum_{e \in E, k \in \mathcal{K} : \hat{x}_e^k = 1} (1 - x_e^k) + \sum_{e \in E, k \in \mathcal{K} : \hat{x}_e^k = 0} x_e^k. \tag{5}$$

The idea is to bound the value of the reliability θ in such a way that if $\hat{x}$ is the optimal solution to the problem, then $\theta = \hat{r}$. We add these cuts each time that a feasible integer solution has been found, until the optimal solution (θ^*, x^*) satisfies that θ^* is exactly the reliability of the network obtained by x^*.

Because the available capacities in each link are incremental, the reliability of the network is determined by the smallest capacity level that is not installed (that is, variables $\hat{x}_e^k$ with value 0). Hence, we can use a strengthened cut by considering only uninstalled capacities, obtaining the following cut:

$$\theta \le \hat{r} + \sum_{e \in E, k \in \mathcal{K} : \hat{x}_e^k = 0} x_e^k. \tag{6}$$

4.2   MIP-driven conditional probability tree

Our second method is derived from the method presented in Section 3 for calculating reliability of a given network. Note that for a given backhaul network with some installed capacities on each link, a CPT obtained from Algorithm 1 is still valid if the available capacities of the links in the network change, but the status of their leaves can also change, obtaining new *undetermined* leaves. This property can be exploited to compute the optimal network design. Let us assume that all available capacities $\mathcal{K}$ are available. Given a CPT, let U be the set of its leaves. Also, for each leaf $u \in U$, let $\mathcal{T}_u$ be the subset of routing trees in $\mathcal{T}$ that are compatible with the upper bound on the capacities imposed by leaf u. Hence, we solve the following MIP model:

$$\max \sum_{u \in U} p_u z_u \tag{7}$$

$$\text{s.t.} \quad (1), (2), (3) \tag{8}$$

$$z_u \leq \sum_{T \in \mathcal{T}_u} y_{T,u} \qquad \forall u \in U \tag{9}$$

$$y_{T,u} \leq x_e^{\bar{k}(T)} \qquad \forall e \in T, T \in \mathcal{T}_u, u \in U \tag{10}$$

$$y_{T,u}, z_u \in \{0, 1\} \qquad \forall T \in \mathcal{T}_u, u \in U \tag{11}$$

For each $u \in U$, the variables z_u represent if the leaf u is *feasible* or *undetermined* ($z_u = 1$) or *infeasible* ($z_u = 0$), so the objective function maximizes an upper bound on the reliability. For each leaf $u \in U$ and $T \in \mathcal{T}_u$, the variable $y_{T,u} = 1$ if the routing tree T is compatible with the available capacities *installed* at leaf u, and $y_{T,u} = 0$ otherwise. This behaviour is obtained by (10), which requires that $y_{T,u} = 1$ only if for each link $e \in T$ the required capacity $\bar{k}(T) = \min_k \{C_{e,k} : d_e^T \leq C_{e,k}\}$ is installed. In other words, $y_{T,u} = 0$ if leaf u is *infeasible* for the capacities defined by x_e^k. Finally, (9) enforces the requirement that if a node is *feasible* or *undetermined* ($z_u = 1$) then at least one routing tree in T_u should be selected that is compatible with node u.

For any given CPT, the MIP model provides an upper bound on the reliability, because z_u can take the value 1 if a leaf u is *undetermined*. Hence, to find the optimal network design, we solve this problem iteratively. Given a CPT, we solve this MIP problem and obtain an upper bound on the reliability. If there is an *undetermined* leaf $v \in U$, then we apply SPLITNODE(v, T) using any T such that $y_{T,v} = 1$. We then remove the variables and constraints associated to leaf u and add the variables and constraints corresponding to the new leaves in the CPT, and re-solve the MIP problem. The algorithm terminates with the optimal solution when no *undetermined* leaf exists.

5 Computational experiments

We evaluate the performance of the algorithm over three networks. First network, named "2-cycles", correspond to two cycles of lengths 4 and 5 sharing two consecutive links. The second network, named "Grid3" is a grid network of 3×3 nodes. Finally, the third network is "Abilene" from SNDlib [8]. The characteristics of these networks are presented in Table 1. For all networks, we assume one unit of demand for every pair of nodes. Each link has three available capacities representing a backhaul wireless link with modulations

Name	$\|V\|$	$\|E\|$	$\|\mathcal{T}\|$	Capacities	Max Reliability
2-cycles	6	7	16	[10, 15, 20]	0.998883
Grid3	9	12	192	[20, 30, 40]	0.999996
Abilene	12	15	249	[32, 48, 64]	0.995316

Table 1

Characteristics of the network instances

16-QAM, 64-QAM and 256-QAM. The distribution probabilities among these capacities is $[0.01, 0.11, 0.88]$ and its costs are $[100, 300, 500]$, respectively. We compute the reliability of each network for 5 levels of budget B, where the first level allows one to install the minimum required capacities to have a reliability larger than 0 and the last level allows one to install the maximum capacity in every link. We run each instance with a time limit of 12 hours.

For 2-cycles instances, the MIP-driven CPT algorithm solves the problem for all budgets in less than a minute, requiring between 34 to 81 iterations to maximize the exact reliability. For the two other networks, Figure 1 shows the relative gap [2] of the computed reliability over time. We find that with lower budgets the algorithm computes the exact solution, and for higher budgets it obtains a near-optimal solution. In fact, for most of the cases, a relative gap less than 1% is attained within one hour of computation.
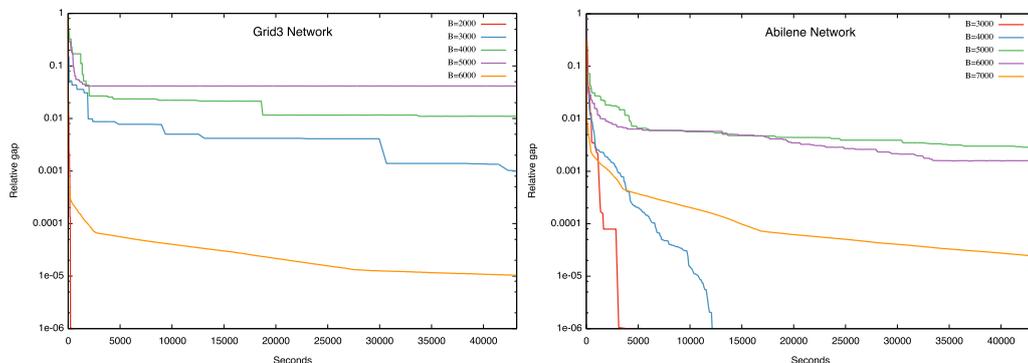

Fig. 1. Reliability gap versus time for Grid3 (left) and Abilene (right)

In comparison, using combinatorial Bender cuts we can solve the problem for 2-cycles instances in similar times, but for the two larger instances its behaviour is unreliable. For all but two instances, the upper bound attained after the time limit is still 1.0, showing that the included cuts still cannot provide a good estimation of the reliabilities.

---

[2] The relative gap is computed as the difference between the upper bound and the lower bound, divided by the upper bound of the reliability.

6 Conclusions and future work

We propose a methodology to compute the exact reliability of a backhaul wireless network, and to optimize its design given a budget. To our knowledge, there is no other methodology available to solve this problem.

A drawback of the proposed methodology is the requirement to know all available spanning trees of a network, because the number of trees grows exponentially with the size of the graph. However, we can easily modify the model solved at each iteration in order to find a routing tree for each leaf of the CPT in the model. This problem can become very large for a large CPT, but its structure is suitable for applying a decomposition method. Another idea to be explored is to focus on the most probable states [4] of the CPT, which might reduce considerably the size of the resulting tree.

References

[1] Claßen, G., D. Coudert, A. Koster and N. Nepomuceno, *A Chance-Constrained Model & Cutting Planes for Fixed Broadband Wireless Networks*, in: *Internationale Network Optimization Conference (INOC)*, Lecture Notes in Computer Science **6701** (2011), pp. 37–42.

[2] Claßen, G., A. Koster, D. Coudert and N. Nepomuceno, *Chance-constrained optimization of reliable fixed broadband wireless networks*, INFORMS Journal on Computing **26** (2014), pp. 893–909.

[3] Codato, G. and M. Fischetti, *Combinatorial Benders' cuts for mixed-integer linear programming*, Operations Research **54** (2006), pp. 756–766.

[4] Colbourn, C. J. and D. D. Harms, *Evaluating performability: Most probable states and bounds*, Telecommunication Systems **2** (1993), pp. 275–300.

[5] Kodjo, A., B. Jaumard, N. Nepomuceno, M. Kaddour and D. Coudert, *Dimensioning microwave wireless networks*, in: *IEEE International Conference on Communications*, 2015, pp. 2803 – 2809.

[6] Lehpamer, H., "Microwave transmission networks: planning, design, and deployment," McGraw-Hill, 2010.

[7] Nepomuceno, N., "Network optimization for wireless microwave backhaul," Ph.D. thesis, Université Nice Sophia Antipolis (2010).

[8] Orlowski, S., R. Wessäly, M. Pióro and A. Tomaszewski, *SNDlib 1.0 survivable network design library*, Networks **55** (2010), pp. 276–286.