



HAL
open science

Completely independent spanning trees for enhancing the robustness in ad-hoc Networks

Axel Moinet, Benoit Darties, Nicolas Gastineau, Jean-Luc Baril, Olivier Togni

► **To cite this version:**

Axel Moinet, Benoit Darties, Nicolas Gastineau, Jean-Luc Baril, Olivier Togni. Completely independent spanning trees for enhancing the robustness in ad-hoc Networks. 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Oct 2017, Rome, Italy. 10.1109/WiMOB.2017.8115791 . hal-01715916

HAL Id: hal-01715916

<https://hal.science/hal-01715916>

Submitted on 23 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Completely Independent Spanning Trees for enhancing the robustness in ad-hoc Networks

Axel Moinet

LE2I, FRE CNRS 2005, Arts et Mtiers
Univ. Bourgogne Franche Comté.
Dijon, France
Email: axel.moinet@u-bourgogne.fr

Benoît Darties

LE2I, FRE CNRS 2005, Arts et Mtiers
Univ. Bourgogne Franche Comté.
Dijon, France
Email: benoit.darties@u-bourgogne.fr

Nicolas Gastineau

LAMSADE - UMR CNRS 7243
Université Paris-Dauphine
Paris, France
nicolas.gastineau@dauphine.fr

Jean-Luc Baril

LE2I, FRE CNRS 2005, Arts et Mtiers
Univ. Bourgogne Franche Comté.
Dijon, France
Email: jean-luc.baril@u-bourgogne.fr

Olivier Togni

LE2I, FRE CNRS 2005, Arts et Mtiers
Univ. Bourgogne Franche Comté.
Dijon, France
Email: olivier.togni@u-bourgogne.fr

Abstract—We investigate the problem of computing Completely Independent Spanning Trees (CIST) under a practical approach. We aim to show that despite CISTs are very challenging to exhibit in some networks, they present a real interest in ad-hoc networks and can be computed to enhance the network robustness. We propose an original ILP formulation for CISTs and we show through simulation results on representative network models that several CISTs can be computed when the network density is sufficiently high. These results tend to reinforce the interest of CISTs for various network operations such as robustness, load-balancing, traffic splitting, ... As an important point, our results show that both the density and the number of nodes have an impact on the number of CISTs that can be found on ad-hoc networks.

Keywords-Completely Independent Spanning Tree, CIST, Integer Linear Programming, Quasi-Unit Disk Graph

I. INTRODUCTION

Maintaining connectivity in communication networks is an important challenge [1], especially in ad-hoc networks due to the absence of centralized structure: nodes act as routers to maintain the network connectivity, but are subject to various effects (interference, radio channel effects, battery limitations, ...) that can lead to spontaneous failures on links and/or nodes. A lot of research investigated the problem of maintaining network connectivity in radio networks under several complementary approaches. The problem of node connectivity was originally discussed by Cheng and Rober-tazzi in 1989 [2]. In many node-to-node communication protocols, network entities discuss with each others using a spanning tree as backbone. Using a spanning tree to organize communications ensures the existence of a path between each pair of nodes while avoiding bridge loops and routing loops. However this approach does not benefit from the advantages of the mesh topologies (i.e. multi-paths), and makes the whole network weak in the event of failures,

especially when occurring on inner nodes on the tree: in such a situation, a new spanning tree requires to be computed in a distributed manner, and point-to-point communications remain unstable until the algorithm converges. One way of performing fault-tolerant communication is to exploit the disjoint paths that exist between pairs of nodes, and to use several disjoint spanning trees. Thus, in case of node failure the system can switch from one tree to another without interruption. The use of multiple disjoint spanning trees finds other applications beyond robustness, i.e. in load-balancing, security, to separate network services, ... The problem of constructing disjoint spanning trees is often motivated by wireless ad-hoc networks [3], [4], because they are used to create a virtual backbone or spine of a wireless ad-hoc network.

Several notions aim to represent disjunction properties for spanning trees in the literature: *independent spanning trees* focus on searching for collections of spanning trees rooted on the same root r , so that for each node x , the paths linking x to r in each tree are pairwise internal-nodes disjoint. Two paths are internal-nodes disjoint if they do not share common vertices excepting the extremities of both paths. Note that independent spanning trees are not always edge-disjoint. In *edge-disjoint spanning trees*, each pair of trees does not share any common edge but can share internal nodes. Many studies have investigated the problem of determining the maximum number of edge-disjoint spanning trees for various topologies, and proposed algorithms for the construction of disjoint spanning trees. In 2001, *Completely Independent Spanning Trees* [5] (CIST) have been defined as a collection of spanning trees, such that for each pair of nodes x and y , the paths linking x to y in each tree are pairwise internal-nodes and edge disjoint.

In this paper, we study *Completely Independent Spanning*

Trees. These trees have been very studied on theoretical aspects. However from our knowledge, no work proposed study of CIST from an operational approach, nor studied the existence of CISTs in realistic networks. This may be due to the strong structural constraints that topologies must satisfy (i.e. vertex-connectivity and edge-connectivity are necessary but insufficient conditions) in order to obtain a collection of CISTs of significant size. We tend to show that despite these strong topological constraints, it is possible to compute and exploit several CISTs in ad-hoc networks if the node density is sufficiently high.

This work is organized as follows: in Section II we briefly present the notations we use. We present related works on respectively disjoint, edge-disjoint, and completely disjoint spanning trees in Section III. We propose in Section IV an ILP formulation for the problem of computing a fixed size collection of CISTs. Using this formulation, we show in Section V that several CISTs can be found in ad-hoc networks, and that the maximum number of CISTs increases proportionally with the density of the network.

II. NOTATION

Throughout this paper, we use the following notation: we model the network with a connected undirected simple graph $G = (V, E)$ with vertex set V and edge set E . An edge e is an unordered pair of nodes $\{u, v\}$, $u \neq v$ and u is an extremity of e if and only if $u \in e$. Let the functional notations $V(G)$ and $E(G)$ denote respectively the vertex set and the edge set of G . We note by $n = |V|$ the order of G - that is to say its number of vertices -, and by $m = |E|$ its number of edges. For each node $u \in V$, we note $N_G(u)$ the set of adjacent nodes (open neighbourhood) in G , and $J_G(u)$ the set of incident edges $\{u, v\}$, for $v \in N_G(u)$. Let $d_G(u) = |N_G(u)|$ be the degree of u in G .

Let $k \geq 2$ be an integer and T_1, \dots, T_k be *spanning trees* in a graph G . The spanning trees T_1, \dots, T_k are *edge-disjoint* if and only if $\cup_{1 \leq \ell < \ell' \leq k} E(T_\ell) \cap E(T_{\ell'}) = \emptyset$. A vertex u is said to be an *inner vertex* in a tree T if and only if $d_T(u) \geq 2$, and a *leaf* if $d_T(u) = 1$. We denote by $I(T)$ the set of inner vertices of tree T . The spanning trees T_1, \dots, T_k are *internally vertex-disjoint* if $I(T_1), \dots, I(T_k)$ are pairwise disjoint. Finally, the spanning trees T_1, \dots, T_k are *completely independent spanning trees* if they are both pairwise edge-disjoint and internally vertex-disjoint.

We denote by $C_{CIST}(G)$ the cardinality of the largest collection of CISTs that exists on G .

We denote by $d(G) \in [0, 1]$ the density of the graph, i.e. the ratio between the number of its edges and the number of possible edges (the closer the density is to 1, the denser the graph). Formally:

$$d(G) = 2m/(n(n-1))$$

Finally, we note $G[W]$ with $W \subseteq V$ the graph induced by vertices from W : graph $G[W]$ has vertex set W and edge

set the edges from E whose both extremities are in W .

III. RELATED WORK

Disjoint spanning trees have been extensively studied for decades and under different assumptions. Initially, works concentrate on the study of edge-disjoint spanning trees, that is, spanning trees that are pairwise only edge-disjoint. The problem of determining the maximum number of edge-disjoint spanning trees has been studied for various regular topologies, including hypercubes [6], [7], twisted cubes [8] cartesian product of cycles [9] and cartesian product of two graphs [10]. One of the most famous results on edge-disjoint spanning trees is the theorem of Tutte-Nash-Williams (independently proposed in [11], [12]), indicating that there are k edge-disjoint spanning trees in any $2k$ -edge-connected graph. In [13], Roskind and Tarjan presented an algorithm with complexity $O(n^2 \log(n) + n^2 k^2)$ to compute a k -size collection of edge-disjoint spanning trees.

Other related work focused on independent spanning trees, implicitly rooted on a specific node. Independent spanning trees have been studied in several topologies, including chordal rings [14], de Bruijn and Kautz digraphs [15], [16], and product graphs [17].

In addition, some works focus on internally vertex-disjoint spanning trees. These trees are close to Completely Independent Spanning Trees, except that an edge can belong to two trees if it connects a leaf in each of them. Internally vertex-disjoint spanning trees can be expressed in terms of disjoint connected dominating sets. The maximum number of disjoint connected dominating sets in a graph G is the *connected domatic number* [18]. Hartnell and Rall have proven that, except K_4 (which has connected domatic number 4), the connected domatic number of planar graphs is bounded by 3 [19].

In 2001, Hasunama introduced Completely Independent Spanning Trees in [5]. The decision problem of determining whether there are two Completely Independent Spanning Trees in a graph G is NP-hard in general [20]. CISTs have been studied on different classes of graphs, such as underlying graphs of line graphs [5], maximal planar graphs [20], cartesian product of two cycles [21] and complete graphs, complete bipartite and tripartite graphs [22]. Inspired by the Tutte-Nash-William theorem on the edge-disjoint trees [11], [12], Hasunama conjectured in [20] that there are k completely independent spanning trees in any $2k$ -connected graph. The search for CISTs gained increasing interest in 2012 when Péterfalvi disproved the conjecture [23] by proposing a construction of a k -connected graph that does not contain 2 CISTs, for any $k \geq 0$, thus showing the problem was much more difficult than it seemed to be. A similar work showed that even for $2k$ -connected and $2k$ -regular graphs, the maximum number of CISTs is not always k [24].

Recently, sufficient conditions have been determined to ensure the existence of two completely independent spanning trees. These conditions are based on the sufficient conditions for hamiltonicity: Dirac's condition [25] and Ore's condition [26]. The Dirac's condition has been generalized to more than two trees [27]–[29] and has been independently improved [28], [29] for two trees. Also, a recent paper has studied the problem on an interesting class of graphs: the class of k -trees, for which the authors have proven that there exist at least $\lceil k/2 \rceil$ completely independent spanning trees [30].

IV. ON THE NUMBER OF CISTs IN AD-HOC NETWORKS: HYPOTHESIS, UPPER BOUNDS AND ILP FORMULATION

Determining the maximum number of CISTs in an ad-hoc network is a difficult problem: from several results, including the construction proposed by Péterfalvi [23], there are k -connected graphs that do not contain two CISTs for any $k \geq 2$ even though the graph density tends to be asymptotically 1 (i.e. a clique graph K_n with an additional pending vertex).

However, we aim to show that despite the strong structural constraints of CISTs and the negative results on some topologies, ad-hoc networks does not suffer from severe restrictions preventing multiple CISTs to be found. Intuitively, only border effects can have negative impact, but they can be reduced when the network density increases. In this work, we propose to verify this hypothesis through simulations.

First, we propose the following remark:

Remark 1: One way to consider searching for a k -size collection of CISTs is to find a k -partition of the network nodes $\{V_1, \dots, V_k\}$ so that for every $i \in \{1, \dots, k\}$ and every $j \in \{1, \dots, k\} - \{i\}$:

- the induced graph $G[V_i]$ is connected,
- let $E_{i,j}$ be the edge set $\{\{u, v\} \in E \mid u \in V_i, v \in V_j\}$. Then there exists two set $E_{i,j}^1$ and $E_{i,j}^2 \subseteq E_{i,j}$ with $E_{i,j}^1 \cap E_{i,j}^2 = \emptyset$, such that for each node $u \in V_i$ there is a node $v \in V_j \mid \{u, v\} \in E_{i,j}^1$, and for each node $v \in V_j$ there is a node $u \in V_i \mid \{u, v\} \in E_{i,j}^2$.

In addition, we have a trivial upper bound for $C_{CIST}(G)$. It is admitted that

$$C_{CIST}(G) \leq \min(\kappa(G), \lambda(G)) \quad (1)$$

where $\kappa(G)$ is the vertex connectivity of G , and $\lambda(G)$ its edge-connectivity. Both $(\kappa(G), \lambda(G))$ can be computed in a polynomial time. Eq. 1 will be used as upper bound when calculating $C_{CIST}(G)$ in Fig. 2-7.

We require some tools to compute $C_{CIST}(G)$ in different networks. Given a network graph $G = (V, E)$ and a positive integer k , we propose an ILP formulation for computing a collection of k CISTs which minimizes the number of internal nodes among all trees if a k -size collection exists, and fails otherwise. As k is an input, we can use a dichotomy

approach and run several times the ILP with different values of k to find the largest number of CISTs.

We define the following variables and constraints $\forall u \in V, \forall e \in E$, and every integer t with $1 \leq t \leq k$:

variables

- we introduce two sets of binary variables x_u^t and x_e^t to describe respectively the internal nodes and the edges of spanning tree t .

$$x_u^t = \begin{cases} 1 & \text{if } u \text{ is an internal node in tree } t \\ 0 & \text{otherwise} \end{cases}$$

$$x_e^t = \begin{cases} 1 & \text{if } e \text{ is an edge in tree } t \\ 0 & \text{otherwise} \end{cases}$$

- To ensure the connectivity of component induced by the selected edges in the same tree t , we use a flow technique: we use variables $x_{(u,v)}^t, \{u, v\} \in E$ to describe the flow in transit on $e = \{u, v\}$ from u to v .

$$x_{(u,v)}^t = \begin{cases} \text{flow transiting from node } u \text{ to it neighbor } v \text{ for tree } t \end{cases}$$

objective

We aim to minimize the number of internal nodes required in any tree. This criterion has a significant influence on the resulting trees, which makes it possible to increase the number of leaves of a tree, decrease the average length of the paths between the nodes and reduce the number of critical points on the network.

$$\min_{1 \leq t \leq k} \sum_{v \in V} x_v^t \quad (2)$$

Note that in this work, the minimization objective is optional, because we focus more on the existence of k CISTs than on the determination of the best trees in the space of feasible solutions.

constraints

For every t with $1 \leq t \leq k$ we define the tree T_t through its $n - 1$ edges, and a flow technique to ensure the connectivity of graph induced by edges: in each tree t , we set a (possibly identical) sink vertex $s^t \in V$. Each vertex in $V - \{s^t\}$ produces an unit of flow that must pass to s^t and through edges of t only. To do this, we define the following constraints:

- Each tree requires exactly $n - 1$ edges:

$$\forall t, \sum_{e \in E} x_e^t = n - 1$$

- The flow unit produced by each node must reach the vertex s^t while preserving the flow conservation law:

$$\forall t, \forall u \in V - s^t, \sum_{v \in N_G(u)} x_{v,u}^t + 1 = \sum_{v \in N_G(u)} x_{u,v}^t \quad (3)$$

$$\forall t, \sum_{u \in N_G(s^t)} x_{u,s^t}^t = n - 1 \quad (4)$$

- To get a correct flow delivery, we ensure that no flow comes out from the sink:

$$\forall t, \sum_{u \in N_G(s^t)} x_{(s^t,u)}^t = 0 \quad (5)$$

- The flow only passes through the edges selected in tree t :

$$\forall t, \forall \{u, v\} \in E, \quad x_{(u,v)}^t + x_{(v,u)}^t \leq n \cdot x_{(u,v)}^t \quad (6)$$

- Once each tree t is defined, we link the variables x_u^t to x_e^t . Vertex u is an internal node in t iff at least two incident edges are selected:

$$\forall t, \forall u \in V, \forall e_1 \neq e_2 \in J_G(u), \quad x_{e_1}^t + x_{e_2}^t - 1 \leq x_u^t \quad (7)$$

Finally, we add the constraints of not sharing the internal nodes nor the edges between trees:

$$\forall u \in V, \sum_{1 \leq t \leq k} x_u^t \leq 1 \quad (8)$$

$$\forall e \in E, \sum_{1 \leq t \leq k} x_e^t \leq 1 \quad (9)$$

Relaxation, and additional constraints

Note that $x_{e_1}^t$ and $x_{e_2}^t$ are binary values in Eq. 7, and that we aim at minimizing the sum of x_v^t (Eq. 2). Then we can relax the ILP by defining x_v^t as a continuous variable over the interval $[0, 1]$ while guaranteeing that the returned values for variables x_v^t would be binary in the optimal solution. Only variables x_e^t remain binary.

We also add the following optional constraints, inspired by Remark 1. In the simulation we have carried out, these constraints improve the resolution time.

- Each node has a neighbor acting as internal node in each tree:

$$\forall t, \forall u \in V, \sum_{v \in N_G(u)} x_v^t \geq 1 \quad (10)$$

- In addition, each node has an incident edge selected in each tree:

$$\forall t, \forall u \in V, \sum_{e \in J_G(u)} x_e^t \geq 1 \quad (11)$$

V. ON THE NUMBER OF CISTS IN AD-HOC NETWORKS

Recall that our main research hypothesis is that, in ad-hoc networks, density has a significant influence on the number of CISTS, and that collections of CISTS exists when the local node density is large enough to avoid isolated or pending vertices (degree 1). In this section, we generate ad-hoc networks using a realistic model, and then determine by simulation how many CISTS we can calculate on them.

A. Network model

We have simulated ad-hoc networks as follows: a model widely used for ad-hoc networks is the Unit Disk Graph model (UDG), defined as the intersection graph of a family of the same radius disks in the Euclidean plane. This model suffers from its simplicity because the nodes have an identical radius (i.e. transmission range) r and are adjacent to each others as soon as their respective distance is lower than r . They do not take into account the presence of obstacles and interferences, nor the heterogeneous nature of nodes. Quasi-Unit Disk Graph is a graph model considerably closer to reality [31]: two nodes are connected by an edge if their distance is less than or equal to q , q being a parameter between 0 and r . Furthermore, if the distance between two nodes is greater than r , there is no edge between them. In the range between q and r the nodes are connected with a certain probability.

We used the Quasi-Unit Disk Graph for network simulation. We consider a geographic area of 100×100 units. Given a number of nodes and a range $r \in [0, 100]$, we generate random coordinates $x_i, y_i \in [0, 100]$ for every node i so that the network density is evenly distributed in the space. We set $q = 0.6r$, and we link pair of nodes with probability $p = 1$, if their distance d is less than q , or with probability $p = \frac{r-d}{r}$ if their distance is included between $0.6r$ and r .

Note that if we need to update the transmission range from r to r' by keeping the same nodes positions, no probabilistic edge is deleted when the transmission range increases, to maintain cohesion of simulations.

B. simulation protocol

We present simulation results on the number of CISTS computed when some network parameters vary.

We use *IBM ILOG CPLEX Optimization Studio v12.6.3* for ILP resolution on a 8-core 3GHz server machine with 16 GB of RAM. We have restricted the search of a k -size collection of CISTS to 2 hours of CPU time per value k . Thus, the values presented in our simulation results are either the maximum number of CISTS of the network, or the best lower bound achieved so far. Even though the highest collection can not be calculated within a reasonable time for some dense graphs with many nodes, this lower bound provides some valuable information about the relevance of using CISTS in ad-hoc networks.

Fig. 1 shows an example of 3 CISTS obtained with our ILP on a 45-nodes network, each represented with a different color.

C. Impact of the transmission range

First we studied the evolution of the number of CISTS when only the transmission range varies.

We placed 50 nodes uniformly, then we searched for (a lower bound on) the number of CISTS and its upper bound, for transmission range values r in $[20, 25, 30, 35, 40, 45]$.

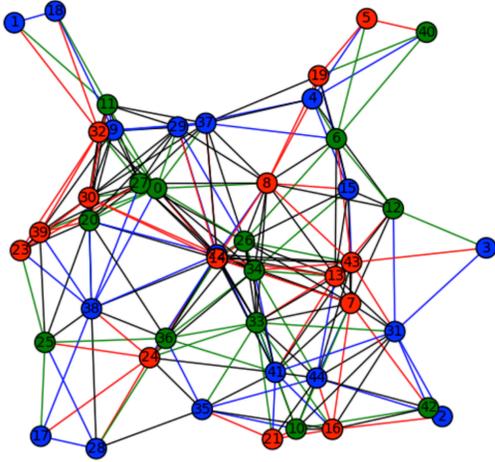


Figure 1: 3 colored CISTs on a 45-nodes network. Here black edges are unused in any tree

We repeated the measurements for 10 different simulated networks. Fig. 2 shows the average number of CISTs found with a 95% confidence interval and the average upper bound in a 50-nodes network per transmission range.

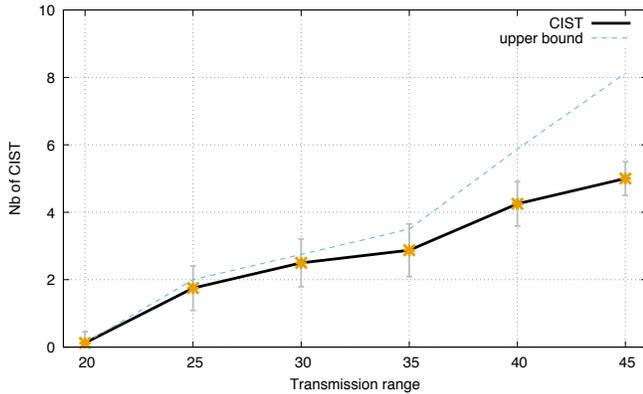


Figure 2: average number of CISTs found and average upper bound ($\min(\kappa(G), \lambda(G))$) in a 50-nodes network

Table I presents a correspondence between the range values presented on Fig. 2 and the average network density (including standard deviation).

Transm. range	20	25	30	35	40	45
Average density	0.10	0.18	0.22	0.26	0.35	0.43
Std deviation	0.016	0.022	0.024	0.029	0.037	0.049

Table I: Average network density per transmission range for a 50-nodes ad-hoc network

We reiterate the same process on 100-nodes networks using the same transmission ranges. The aim is to observe whether the growth of the number of CISTs is similar, regardless of the number of nodes. Fig. 3 shows the average

number of CISTs found for 100-nodes networks with a 95% confidence interval after 10 simulation rounds.

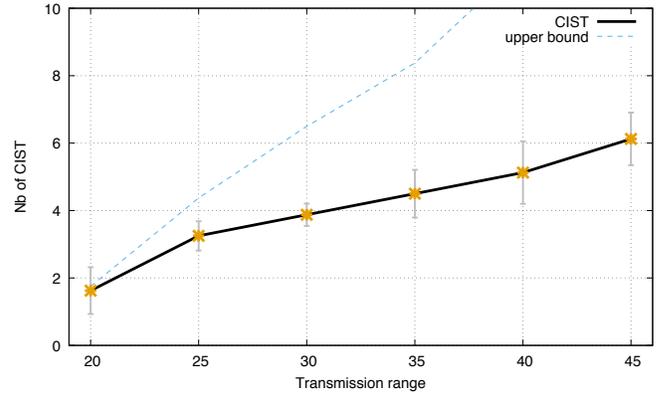


Figure 3: average number of CISTs found and average upper bound in a 100-nodes network

Table II presents a correspondence between the range values shown in Fig. 3 and the average network density (including standard deviation).

Transm. range	20	25	30	35	40	45
Average density	0.11	0.18	0.22	0.27	0.35	0.45
Std deviation	0.005	0.007	0.008	0.010	0.017	0.024

Table II: Average network density per transmission range for a 100-nodes ad-hoc network

In a similar way, Fig. 4 and Table II presents the average number of CISTs found for 150-nodes networks, and the corresponding densities.

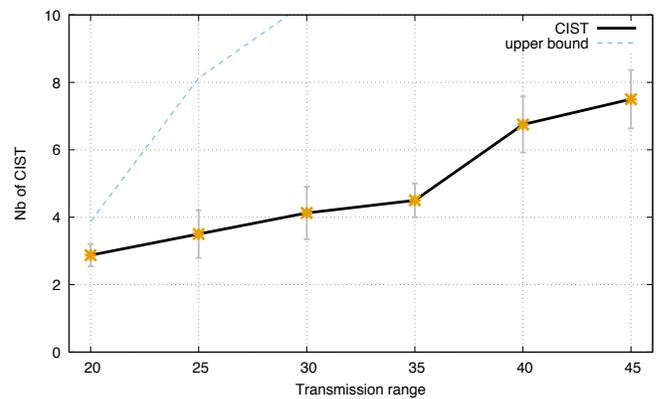


Figure 4: average number of CISTs found, and average upper bound in a 150-nodes network

Remark 2: For transmission range values less than 35 units, the calculated number of CISTs is optimal mostly on networks on 50, 100 and 150 nodes: the ILP admits no solution when we increase the number of CISTs by one.

Transm. range	20	25	30	35	40	45
Average density	0,10	0,17	0,21	0,25	0,33	0,42
Std deviation	0,004	0,005	0,005	0,009	0,012	0,013

Table III: Average network density per transmission range for a 100-nodes ad-hoc network

However this optimality is not presented on Fig. 2 , 3, and 4, as bounds refer to values of Eq. 1 only.

D. Impact of the number of nodes

In a second part, we try to check whether the number of nodes can influence $C_{CIST}(G)$ or not. We performed a new set of simulations according to the following process: we generated sets of random networks with different number of nodes but equivalent density by adjusting the transmission range, and computed the number of CISTs on them. We execute 10 times the measurements. Fig. 5 show the variation of the average number of CISTs on the number of nodes for a density network between 0.9 and 0.12

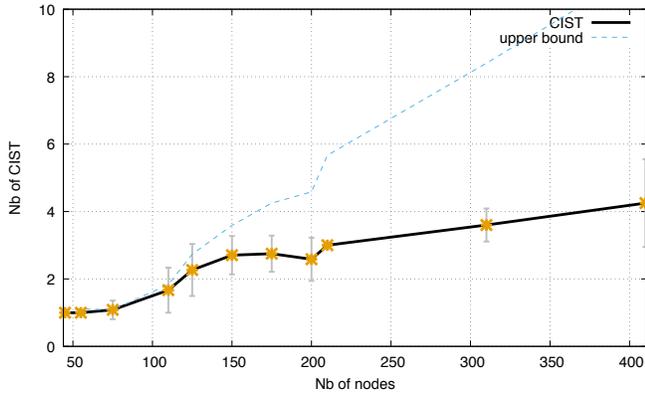


Figure 5: average number of CISTs found when the number of nodes varies, for a network density between 0.9 and 0.12

We have realized this scenario for two other density ranges, respectively medium density (between 0.15 and 0.18) presented on Fig 6, and high density (between 0.25 and 0.28), presented on Fig. 7). Results are presented with an interval confidence of 0.95%.

E. Simulation results and interpretation

The results presented in Fig. 2, 3, 4 , confirm the main research hypothesis: the number of CISTs increases in ad-hoc networks when the network density increases, making CISTs an appropriate solution to improve the robustness of these networks. In particular, the results indicate that there is good hope of finding 3 CISTs for transmission range greater than 35 units, regardless of the number of nodes. As a positive result, we note that there is a very high probability of finding at least 2 CISTs even for small densities on 50-nodes networks. This definitively confirms the relevance of CISTs in such networks.

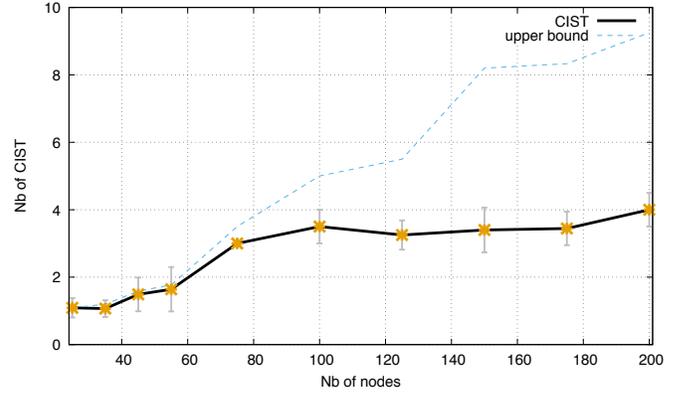


Figure 6: average number of CISTs found when the number of nodes varies, for a network density between 0.15 and 0.18

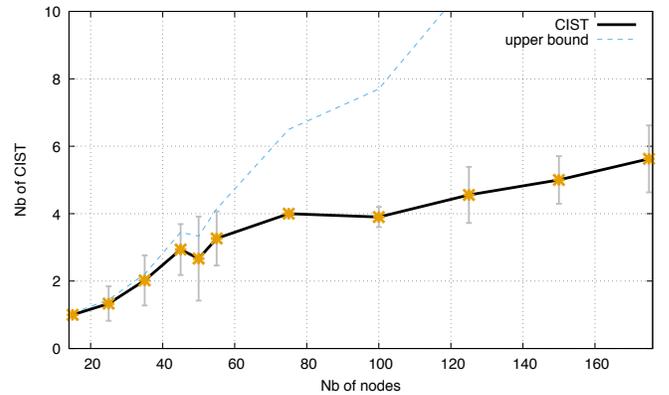


Figure 7: average number of CISTs found when the number of nodes varies, for a network density between 0.25 and 0.28

More surprisingly, presented results, especially the second set of conducted simulations (Fig 5, 6, and 7), tend to show that density is not the only parameter that can influence C_{CIST} : for analogous densities, the number of nodes seems to have a positive impact on the number of CISTs: for example, there is good hope to find 4 CISTs in 50-nodes networks when the transmission range reaches 40 units, while it can be reduced to 30 units in 150-nodes networks. A high number of nodes even allows to get up to 8 trees for high transmission ranges (Fig. 4). While we have demonstrated a strong correlation between $C_{CIST}(G)$ and $d(G)$; this additional result suggests that there might be another measurement that could benefit from a higher correlation with $C_{CIST}(G)$ on ad-hoc networks.

For 50-nodes networks, the vertex and edge connectivity is an interesting upper bound, showing that restrictions on CISTs have a reduced impact on their number compared to edge-disjoint spanning trees. However, this bound becomes less relevant for high transmission ranges in 100 and 150-nodes networks, and some effort must be made to propose better calculable bounds in reasonable time.

VI. CONCLUSION AND PERSPECTIVE

Despite the high difficulty of finding CISTs, and some negative theoretical results for dense networks, CISTs appear to be an interesting choice in ad-hoc networks for improving the robustness and other common network operations such as load balancing.

Here we have focused our efforts on providing proof of concepts of the relevance of CISTs in ad-hoc networks. Now we are trying to extend our work from an operation research perspective: our ILP can be improved to get better results in less computation time. In particular, we intend to propose an improved formulation using column generation [32] to see the gain in term of computation time and to obtain new results by studying other parameters on larger networks.

An interesting research perspective focuses on the gap between the number of CISTs and the number of Edge-Disjoint Spanning Trees: we plan to investigate whether there is a strong relationship between the two structures, and whether correlate them on particular networks.

However, CISTs remain very sensitive to minor local issues or border effects that can induce, i.e. low-degree nodes. These nodes greatly reduce the number of calculable CISTs even if all the other nodes benefit from a high degree. An interesting perspective would be to extend the definition of CISTs by including a bounded number of nodes and/or edges on which the combinatorial constraints can be relaxed. A main interest would be to compute more *almost* CISTs, while bounding the number of nodes that cannot satisfy the constraints specific to CISTs. Another use would help the network designer to find critical network points, to focus on the node requiring extensive supervision, or to advise the placement of additional nodes. Our ILP formulation can be easily extended to accommodate such an improvement. An additional study could focus on how the number of CISTs increases when the number of relaxed nodes or edges varies.

Other perspective could lead to efficient heuristics for calculating collections of CISTs, and to evaluate the results obtained with respect to the optimum.

ACKNOWLEDGMENT

The authors would like to thank the Regional Council of Burgundy who founded this work.

REFERENCES

- [1] M. Jain and S. Chand, "Issues and Challenges in Node Connectivity in Mobile Ad Hoc Networks : A Holistic Review," *Wireless Engineering and Technology*, vol. 07, no. January, pp. 24–35, 2016. [Online]. Available: <http://www.scirp.org/journal/PaperDownload.aspx?DOI=10.4236/wet.2016.71003>
- [2] Y. C. Cheng and T. G. Robertazzi, "Critical Connectivity Phenomena in Multihop Radio Models," *IEEE Transactions on Communications*, vol. 37, no. 7, pp. 770–777, jul 1989. [Online]. Available: <http://ieeexplore.ieee.org/document/31170/>
- [3] S. Guha, S. Khuller, and D. Sets, "Approximation Algorithms for Connected Dominating Sets," *Algorithmica*, vol. 20, pp. 374–387, 1998. [Online]. Available: <http://link.springer.com/article/10.1007/PL00009201>
- [4] P. Wan, K. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," *INFOCOM 2002. Twenty-First ...*, vol. 00, no. c, pp. 1597–1604, 2002. [Online]. Available: <http://ieeexplore.ieee.org/xpls/abs/all.jsp?arnumber=1019411>
- [5] T. Hasunuma, "Completely independent spanning trees in the underlying graph of a line digraph," *Discrete Mathematics*, vol. 234, no. 1-3, pp. 149–157, 2001.
- [6] B. Barden, R. Libeskind-Hadas, J. Davis, and W. Williams, "On edge-disjoint spanning trees in hypercubes," *Information Processing Letters*, vol. 70, no. 1, pp. 13–16, 1999.
- [7] I. Havel, "On Hamiltonian circuits and spanning trees of hypercubes," *Časopis pro pěstování matematiky*, vol. 109, no. 2, pp. 135–152, 1984. [Online]. Available: <http://eudml.org/doc/19336>
- [8] M. C. Yang, "Constructing edge-disjoint spanning trees in twisted cubes," *Information Sciences*, vol. 180, no. 20, pp. 4075–4083, 2010.
- [9] H. Wang and D. M. Blough, "Multicast In Wormhole-switched Torus Networks Using Edge-disjoint Spanning Trees," *jpdc*, vol. 61, no. 9, pp. 1278–1306, sep 2001. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0743731501917514>
- [10] S.-C. Ku, B.-F. Wang, and T.-K. Hung, "Constructing edge-disjoint spanning trees in product networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 14, no. 3, pp. 213–221, 2003.
- [11] C. S. J. A. Nash-Williams, "Edge-disjoint spanning trees of finite graphs," *Journal of the London Mathematical Society*, vol. s1-36, no. 1, pp. 445–450, 1961. [Online]. Available: <http://doi.wiley.com/10.1112/jlms/s1-36.1.445>
- [12] W. T. Tutte, "On the Problem of Decomposing a Graph into n Connected Factors," *Journal of the London Mathematical Society*, vol. s1-36, no. 1, pp. 221–230, 1961. [Online]. Available: <http://doi.wiley.com/10.1112/jlms/s1-36.1.221>
- [13] J. Roskind and R. E. Tarjan, "A Note on Finding Minimum-Cost Edge-Disjoint Spanning Trees," *Mathematics of Operations Research*, vol. 10, no. 4, pp. 701–708, 1985.
- [14] Y. Iwasaki, Y. Kajiwara, K. Obokata, and Y. Igarashi, "Independent spanning trees of chordal rings," *Information Processing Letters*, vol. 69, no. 3, pp. 155–160, 1999. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0020019098002051>
- [15] Z. Ge and S. L. Hakimi, "Disjoint Rooted Spanning Trees with Small Depths in deBruijn and Kautz Graphs," *SIAM Journal on Computing*, vol. 26, no. 1, pp. 79–92, 1997. [Online]. Available: <https://doi.org/10.1137/S0097539793244198>

- [16] T. Hasunuma and H. Nagamochi, "Independent spanning trees with small depths in iterated line digraphs," *Discrete Applied Mathematics*, vol. 110, no. 2-3, pp. 189–211, 2001.
- [17] K. Obokata, Y. Iwasaki, F. Bao, and Y. Igarashi, *Independent spanning trees of product graphs*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 338–351. [Online]. Available: http://dx.doi.org/10.1007/3-540-62559-3{_}27
- [18] B. Zelinka, "Connected domatic number of a graph." *Math. Slovaca*, vol. 36, no. 4, pp. 387–392, 1986.
- [19] B. L. Hartnell and D. F. Rall, "Connected Domatic Number in Planar Graphs," *Czechoslovak Mathematical Journal*, vol. 51, no. 1, pp. 173–179, 2001. [Online]. Available: <http://link.springer.com/10.1023/A:1013770108453>
- [20] T. Hasunuma, "Completely Independent Spanning Trees in Maximal Planar Graphs," in *Graph-Theoretic Concepts in Computer Science, 28th International Workshop, WG 2002*. Cesky Krumlov: Springer Berlin Heidelberg, 2002. [Online]. Available: http://link.springer.com/10.1007/3-540-36379-3{_}21
- [21] T. Hasunuma and C. Morisaka, "Completely independent spanning trees in torus networks," *Networks*, vol. 60, no. 1, pp. 59–69, 2012. [Online]. Available: <http://dx.doi.org/10.1002/net.20460>
- [22] K. J. Pai, S. M. Tang, J. M. Chang, and J. S. Yang, "Completely Independent Spanning Trees on Complete Graphs, Complete Bipartite Graphs and Complete Tripartite Graphs," *Smart Innovation, Systems and Technologies*, vol. 20, pp. 107–113, 2013.
- [23] F. Péterfalvi, "Two counterexamples on completely independent spanning trees," *Discrete Mathematics*, vol. 312, no. 4, pp. 808–810, 2012.
- [24] B. Darties, N. Gastineau, and O. Togni, "Completely independent spanning trees in some regular graphs," *Discrete Applied Mathematics*, vol. 217, pp. 163–174, sep 2017. [Online]. Available: <http://arxiv.org/abs/1409.6002>
- [25] T. Araki, "Dirac's condition for completely independent spanning trees," *Journal of Graph Theory*, vol. 77, no. 3, pp. 171–179, 2014.
- [26] G. Fan, Y. Hong, and Q. Liu, "Ore's condition for completely independent spanning trees," *Discrete Applied Mathematics*, vol. 177, pp. 95–100, nov 2014. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0166218X14002662>
- [27] H. Y. Chang, H. L. Wang, J. S. Yang, and J. M. Chang, "A note on the degree condition of completely independent spanning trees," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E98A, no. 10, pp. 2191–2193, 2015.
- [28] T. Hasunuma, *Minimum Degree Conditions and Optimal Graphs for Completely Independent Spanning Trees*. Venice: Springer International Publishing, 2015. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-29516-9{_}22
- [29] X. Hong and Q. Liu, "Degree condition for completely independent spanning trees," *Information Processing Letters*, vol. 116, no. 10, pp. 644–648, 2016.
- [30] M. Matsushita, Y. Otachi, and T. Araki, "Completely independent spanning trees in (Partial) K-Trees," *Discussiones Mathematicae - Graph Theory*, vol. 35, no. 3, pp. 427–437, 2015.
- [31] L. Barrière, P. Fraigniaud, L. Narayanan, and J. Opatrny, "Robust Position-Based Routing in Wireless Ad Hoc Networks with Unstable Transmission Ranges," *Wireless Communications and Mobile Computing*, vol. 3, no. 2, pp. 141–153, 2003. [Online]. Available: <https://doi.org/10.1002/wcm.108>
- [32] M. E. Lübbecke, "Column generation," *Wiley Encyclopedia of Operations Research and Management Science*, pp. 1–19, 2011. [Online]. Available: <http://www.or.rwth-aachen.de/research/publications/colgen.pdf>