Interference and Topology Control

[Does Topology Control Reduce Interference] *

H. K. Al-Hasani
Computer Networks and Telematics
Department of Computer Science
Freiburg, Germany

Categories and Subject Descriptors
[**Computer networks**]: ad-hoc networks

Keywords
Interference, spanner graph, networks connectivity

ABSTRACT
The concept of ad-hoc networks has first arisen in 1970s. Ad-hoc networks act differently than other wireless networks, for each node can act as a router. For that ad-hoc networks can be used in some interesting applications.
However, if the network wasn't well constructed, it would have an inefficient performance. Interference can be considered to be one of the worst problems, the network can suffer from, as the interference causes energy consuming besides time delay.

This paper consists of five sections namely: Introduction, Goals which the original paper would like to achieve in *Desired Graph*, problems and challenges might be faced in order to establish the desired graph in *Related Issues*. The fourth section will introduce the three suggested algorithms LIFE,LISE,LLISE, and how they claim to hold all the requirments; finally a conclusion and evaluation to the overall paper.

1. INTRODUCTION
A network's interference considered to be high, if many nodes influenced by a communication between any pair of nodes in that network.E.g. in *Figure 1* there are 9 nodes influenced by a communication between x and y including x, y.
Of course interference can't be avoided, but can be optimized. In addition, the interference of a node is the number of discs includes this node. This way the interference of a graph would be the maximum node interference [1] .

*Martin Burkhart, Pascal von Rickenbach, Roger Wattenhofer, Aaron Zollinger

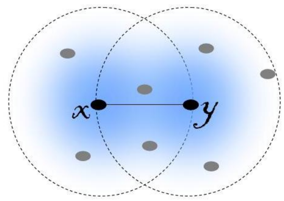

Figure 1: The resulted interferance of *x,y* communicating each other.

1.1 Interference
In order to give a formal definition to the interference, it is important to look first at this communication.
For two nodes *x,y*, we say the are connected, if the edge/link between them is symmetric, i.e. x should be able to send and receive packets from Y and visa versa.
The length of this edge $|x, y|$ is actually the radius of the transmission disc x, and disc y, in other word, these discs refer to the range area of each node. Each disc depends on the energy power of its node. This is it, each node has several discs size, which comes from the distance between it and whom trying to reach. A communication will be established if both radii were identical.
As a consequences, lowering the energy might cause a disconnection.
Given a graph $G(V, E)$, where V denotes the set of graphs vertices, E its edges set; the interference of such graph would be denoted as:

$I_G =^{max}_{e \in E} Cov_e$ where

$$Cov_e = |\{n \in V|\ n\ is\ covered\ by\ D(x, |x, y|)\} \cup \{m \in V \mid m\ is\ covered\ by\ D(x, |\,y, x\,|)\}\,| \hookrightarrow \quad [1]$$

2. DESIRED GRAPH
The desired graph should meet the following points:

1. ***Low interference*** : the paper is trying to minimize the graph interference, which preserves energy, as collisions require retransmission.

2. ***Connectivity*** : If nodes x,y were connected in the graph G, then they should be also connected in the resulted graph G' (either directly or indirectly).

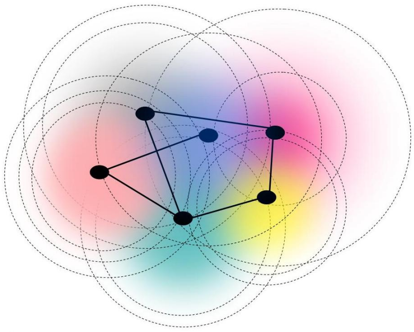

Figure 2: High interference, although it is a low degree graph.

3. ***Spanner graphs*** : a flexable approach. Through spanners graphs one gets more loosely or more tightly graph based on the *stretch factor*. On the other hand, using spanners with the minimum *stretch factor* should ends up with a graph with shortest distance... see *Spanner graph*.

4. ***Planarity*** : where no edges intersect...will be discussed in (**3.4**).

2.1 Spanner graph

Known that a path's length is the sum of all its edges, which are in the euclidean plane, and given a set of vertices V in the euclidean plane, and a set of edges E.
G would be a ***t-spanner graph*** for V if the distance between any pair of nodes x,y is is no larger than the euclidean length time t, where t ***stretch factor*** is the smallest value for the spanner G.
Starting from a graph G, to construct a subgraph G' as a t-spanner graph, the condition above for any pair of nodes (x,y) should then look like

$$|x, y|_{pG'} \leq t \cdot |x, y|_{pG}$$

This is it, the path between these pair of nodes in G' should be equal to at most the shortest path between them in the original graph G time the stretch factor t.
t-Spanners are important, as the stretch factor indicates the graph's performance, i.e. one can tell from the stretch factor, how efficient the routing ,for instance, is going to be.

3. RELATED ISSUES

The aim of presenting the next four topics is to show, what kind of resulted topology we might look at.
It is important to show, that neither sparseness nor planarity the algorithms look for.
Also we might face a graph, which can't be optimized.

3.1 Sparseness

The claim is, by lowering the graph degree, the interference will be lowered as well; which can't stand alone, as it is indicated in *Figure 2*.
For such small graph, constructed with the minimum degree, the interference is Ωn, however. Where n is the nodes number.

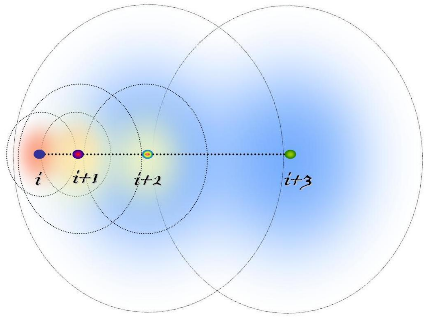

Figure 3: A chain of nodes where $|v_i, v_{i+1}| = 2^i$.

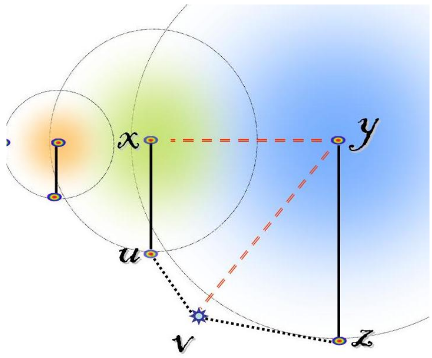

Figure 4: A constant interference for chain of nodes $|v, y| > |z, y| > |x, y|$.

3.2 Nodes chain

Nodes chain refers to a set of vertices, which been introduced in [2], where the distance between them grows exponentially.
Let V be a set of vertices, which their position presented by $(0, x_i) \mid i \in \{1...n-1\}$ and n is number of vertices; i.e. these vertices located on a line.
In this situation if the vetrices v_i, v_{i+1} are communicating, then their disc's radii($|v_i, v_{i+1}|$) is 2^i. Therefore, this communication interfers with all $v_j \mid j \in \{1...i-1\}$[2].
The solution to this scenario is to ask for help from a similar scenario :
Lets assume that V is a set of vertices $\{x, y, z, u, v, ...\}$, $\{x, y\}$ belong to the first scenario, and $\{u, z\}$ belong to the second scenario. The choice of $\{u, z\}$ comes from the assumption that $|y, z|$ is larger than $|x, y|$.
A second selective choice of node v, so that $|v, z| < |z, y|$ and $|v, y| > |z, y|$.
The resulted graph would have a constant interference (*Figure 4*). This result shows how can a lower interference topology differs from some approachs, which would take the greedy strategie, i.e. Nearest Neighbor Forest would choose the nearest neighbor, and that's doesn't solve the problem, as been seen (*for more details and proof [1]*).

3.3 Worst case graph

As mentioned before, there are graphs, their interference cann't be optimized. *Figure 5* is a graph been proposed in [2], shows this case.
From the first look, one can see that $|x, y| > |y, u|$ and u is the closest node to y in x's disc. Any algorithm trys to

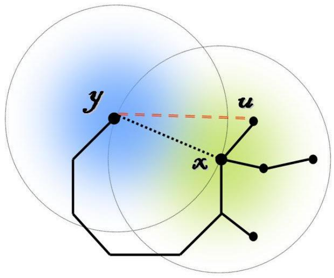

Figure 5: A worst case graph, where $|x,y|$ increases the interference massively.

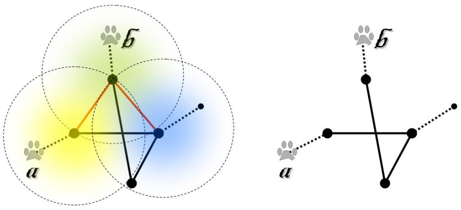

Figure 6: (*left*) Cycles indicate the annoying interference, which can be deleted (*right*) and connectivity remains intact.

establish such a graph will have no choice but to let x, y directly connected. On the other hand there might be an alternative path connect x, y indirectly, this path has the property that the intermidate nodes have constant interference, as they are distributed in such a way.
To make a dicision, whether $|x, y|$ should be remain or to use the other path, is the same dicision of choosing the long safe path or the short expensive one.
Such a scenario shows that for such a graph, although the optimum can be achieved, for other contructions requirments, the algorithm will sacrifice the interference.

3.4 Planar graph

An example given in [1], in which one can successfully construct an interference-optimal graph (here it is a tree), and the resulted topology is not planar.
In respect to interference, the weight of the edges referes to the number of nodes, which are influenced by the communication of that edge.
In *Figure 6* where the original graph on the left, and the resulted interference-optimal tree on the right side.
The red edges have the largest weight, for a and b are small group of nodes. Therefore disposing these nodes will reduce the interferance...(*for more details and proof [1]*).

4. SUGGESTED ALGORITHMS

Unfortunatly, only the first algorithm could be traced, as the other two algorithms require the knowledge of the stretch factor in advanced, therefore a brief look at them will be given[1].

[1]These algorithms will be also discussed in the conclusion

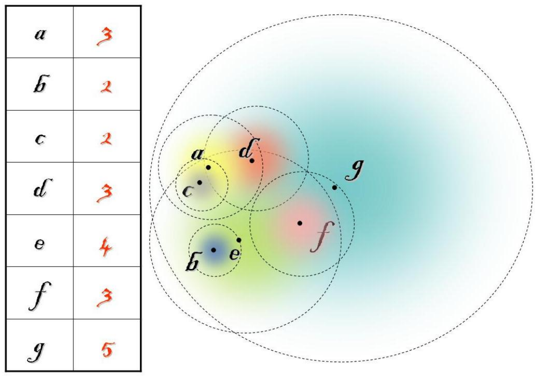

Figure 7: A set of nodes $\{a, b, c, d, e, f, g\}$ and their related maximum transmission radius,which been chosen randomly.

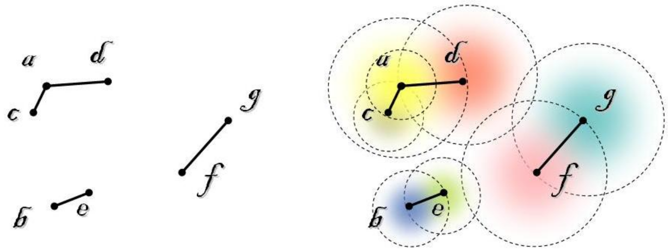

Figure 8: The constructed forest G_{LIFE}(*left*), and its related interference (*right*).

4.1 Low Interference Forest Establisher (LIFE)

As the name indicates, this algorithm constructs several spanning trees. One main advantage is that the input is only the set of vertices, so that no specific knowlesge is required. *Figure 7* shows the input of this algorithm, while *Figure 8* shows its result G_{LIFE}.
In the given example, vertices number, vertices locations and transmissions value been choosen randomly[2] .
The $PROOF$ given in [1] shows that the resulted forest is MSF, as MSF considered to minimize the maximum weight of the edges:
Given G^* which is another MSF for the same set of vertices. If G^* has the maximum weight edge e^* and could be replaced by an edge e from G_{LIFE} which has a lower weight, then G^* cann't be a MSF[3].
The running time for this algorithm is conditionaly $O\left(n^2 log\ n\right)$, for the algorithm needs to know for each vertex, whether it is connected to the current nominated vertex in the final graph, or not. This dicision is needed for setting an edges between these vertices.
In best cases the running time would be $O\left(n^2\right)$.

4.2 Low Interference Spanner Establisher (LISE)

From above, the connectivity condition is violated, i.e. these spannening trees are not one spanning graph.
For LISE input: A set of vertices, and the stretch factor.

[2]These parameters been altered several times, for clarification reason.
[3]LIFE and its Proof will be discussed in the Conclusions section

When LISE starts constructing the interference-optimal, it looks for the shortest path between the current vertex and the nominated vertex, and checks whether they are connected already, or not.
The Spanning property is held, as the comparison is based on t (the stretch factor) time the shortest path
$|x, y|_{P^{G_{LISE}}} < t. |x, y|$ .. [♣]

If this comparison returns false, then other low weighted edges (low coverage) will be inserted to the desired graph G_{LISE}, if these edges are not exit in G_{LISE}.
To prove the interference optimality of the constructed G_{LISE}, it is suffecint to show first that only edges with low coverage are included.
Again [1] proofs it by following the algorithm and takes two edges f, e with the relation $Cov_f \geq Cov_e$ as an example, the proof shows that, if Cov_f been included in G_{LISE} while e is not, then $Cov_f \geq Cov_e$ doesn't hold any more.
Moreover, it is required to prove that G_{LISE} is a ***t-spanner graph***, which is easy to see, as the condition in the algorithm to include the shortest path combined with the stretch factor...*See* [♣].
At last, the same approach to prove the interference optimality of G_{LISE} is used: Assuming there are other two graphs $G^*, \tilde{G}$, which are a t-spanner interference-optimal for the same set of vertices, and G_{LISE} is not optimal. Deriving the new situation in a way to show $G^*, \tilde{G}$ are related to each other and to G_{LISE}, that's it is impossible to have such optimality without G_{LISE} being optimal as well...(*for more details and proof [1]*).
LISE needs polynomial running time in the number of networks nodes.

4.3 Local Low Interference Spanner Establisher (LLISE)

The algorithm LLISE is applied on each edge e (by one of its incident) to find the interference optimal path, and thats what meant by locality, i.e. to find interference optimal path between (x, y) with respect to the relation $|x, y| \leq t\,|x, y|$. LLISE has three steps:

1. Collect $(\frac{t}{2})$*neighborhood.*
2. Compute minimum interference path for e $\{e \in E\}$
3. Inform all edges on that path to remain in the resulting topology. $\hookrightarrow$ [1]

The second step computes the interference optimal path between (x, y), this path been choosen from all eligible sorted edges among the $(\frac{t}{2})$ *neighbors* of e. Informing the other vertices to stick to this path, should guarantee the interference optimality.
(*In [1] a Lemma been introduced to prove that the resulted optimal path is always found and no other optimal paths this algorithm couldn't find between* $(\frac{t}{2})$ *neighbors of* e).

5. CONCLUSIONS

The paper **Does Topology Control Reduce Interference** consists of two parts as been shown. Suggesting the three given algorithms gave a big clue, how the presented concepts can be implimented.
Unfortunatly, these algorithms are expencive to see them in reality. Both LISE and LLISE assume the knowledge of the network besides the stretch factor.
To find the stretch factor, some algorithms vary the computation coast, e.g. for positive edge weights graph finding the stretch factor requires $O(n^3m + n^4 log\ n)$ *n and m are vertices number, edges number respectivly* , or $O(n^3)$ as in [10].On the other hand some papers proved it is NP-Complete for certain kind of graphs.
Hence the paper doesn't give a hint how this stretch factor computed in first place, the algorithms results which been given cann't really be counted *(although these results are very interessting)*, also the running time for these algorithms.
On the other hand, tracing LIFE wasn't that awarding, as one would assume...in so many situations, the constructed forest was actually just set of disconnected nodes. Moreover, setting the condition of choosing the edge with minimum coverage was't accurate, for that requires the knowledge of the network. For instance LIFE cann't avoid a scenario which is similar to the worst case, i.e when the nodes are indirectly connected.
Regarding to the Proof, showing that G^* is not MSF doesn't gaurentee that G_{LIFE} is MSF. Furthermore, the assumption, constructing a minimum spanning tree alone gaurentees low interference is not sufficient. MST shows bad behavior in so many cases (*see nodes chain*).

---

6. FINAL WORDS

This abstract highlighted the very informative paper of *Martin Burkhart, Pascal von Rickenbach, Roger Wattenhofer and Aaron Zollinger* with the help of the supervisor Prof. Dr. Christian Schindelhauer, and the References.
In the mentioned paper, a formal definition to the interference problem been given besides other concepts been established.
Probably the authors didn't mean to give a solution, but tended to show how the solution might wanted to be.

---

7. REFERENCES

1. ***Does Topology Control Reduce Interference***
Martin Burkhart, Pascal von Rickenbach, Roger Wattenhofer, Aaron Zollinger
2. ***Energy, Congestion and Dilation in Radio Networks***
Friedhelm Meyer auf der Heide, Christian Schindelhauer, Klaus Volbert, Matthias Grünewald
3. ***Analysis of Interference in Ad-Hoc Networks***
Martin Burkhart
4. ***Spanner Graph based Topology Aggregation Schemes for Large Networks***
Aniruddha S. Diwan, P. Basker, Kumar N. Sivarajan
5. ***Constructing Interference-Minimal Networks***
Marc Benkert, Joachim Gudmundsson, Herman Haverkort, Alexander Wolff
6. ***Improved Randomized Algorithms for Path Problems in Graphs***
Surender Baswana

7. ***Geometric Spanners***
Joachim Gudmundsson, Giri Narasimhan, Michiel Smid

8. ***A Robust Interference Model forWireless Ad-Hoc Networks***
Pascal von Rickenbach, Stefan Schmid, Roger Wattenhofer, Aaron Zollinger

9. ***Low-Interference Topology Control for Wireless Ad Hoc Networks***
Kousha Moaveni-Nejad and Xiang-Yang Li

10. ***Improving the stretch factor of a graph by edge augmentation***
Mohammad Farshi, Panos Giannopoulos, Joachim Gudmundsson

11. ***Location-based Routing in Ad hoc Networks***
Lecture notes

12. ***GRAPH SPANNERS***
Lecture notes by: S.Nithya

13. ***On the Locality of Distributed Sparse Spanner Constructions***
Lecture notes by: B. Derbel, C. Gavoille, D. Peleg, L. Viennot