Local Algorithms for

Topology Control in Ad Hoc

Networks

Evangelos Kranakis_

School of Computer Science, Carleton University

1125 Colonel By Drive, Ottawa, Ontario, Canada K1S 5B6.

Research supported in part by NSERC (Natural Science and Engineering Research Council of Canada and MITACS (Mathematics of Information Technology and Complex Systems)

and

Jorge Urrutia

Instituto de Matematicas, Universidad Nacional Autonoma de Mexico,

Area de la investigacion cientifica, Circuito Exterior, Ciudad Universitaria,

Coyoacan 04510, Mexico, D.F. Mexico.

_

Local Algorithms for Topology Control in Ad Hoc Networks

Abstract

We survey recent techniques for local topology control in location aware Unit Disk Graphs. including local algorithms for Routing, Traversal, Planar Spanners, Dominating and Connected Dominating Sets, and Vertex and Edge Coloring. In addition to investigating trade-offs for these problems, we discuss open problems that will play an important role in the future development of the subject.

1 INTRODUCTION

Ad hoc wireless networks consist of a collection of hosts of limited power communicating with each other over a wireless medium without any pre-designed or fixed infrastructure. <u>Topology control</u> refers to the problem of maintaining a stable and connected infrastructure among the hosts of an ad hoc network. Effective use of topology control can reduce energy consumption and increase the capacity of the network, due to reduced contention to access the wireless channels. This is accomplished by having nodes in a wireless multi-hop network define collaboratively the network topology by forming the proper neighbor relation under certain criteria.

Despite the fact that communication tasks must be resolved only by consulting nearby hosts, algorithmic solutions must solve global computational tasks, involving, e.g., the computation of dominating and independent sets, vertex and edge colorings, and spanners. Practical considerations inspire the additional requirement for algorithms to be local in the sense that each node of the network should make decisions based only on the information obtained from nodes located a constant (independent of the size of the network) number of steps away from it. Local topology control is vital for heterogeneous network environments affected by mobile hosts, variable neighbor density, and dynamic reconfigurations. Network stability must be attained by devising local algorithms for solving traditional communication tasks like the computation of forwarding tables for routing and broadcasting, node and edge colorings for scheduling and channel assignment.

The present article surveys recent results on local, constant approximation, deterministic algorithms for accomplishing topology control in Unit Disk Graphs (UDGs) with location aware nodes. In more detail, we introduce important background information in Section 2 and discuss routing and traversal in Subsections 3.1 and 3.2, respectively, planar spanners in Subsection 3.3, dominating and connected dominating sets in Subsection 3.4, and vertex and edge colorings in Subsection 3.5. For all the algorithms described we also provide known trade-offs between required locality, processor memory and processing as well as optimality of the object constructed.

2 BACKGROUND

Algorithms devised for traditional wire-line systems are not always adequate in ad hoc networking. In dynamically changing ad hoc networks, participating hosts cannot be assumed to have knowledge of the entire system. In addition, it is often impractical or even impossible to explore the whole network prior to executing an algorithm since by the time the entire system has been examined a new change may have occurred that was not taken into account.

In this setting, <u>locality</u> emerges as an important concept. In local algorithms it is required that the status of a node depends only on the nodes at most a constant number (independent of the size of the network) of edges (hops) away from it. Introduced by (Linial [1992]), this model has the advantage that each node in the network need only be aware of the existence of other parts of the network that are only a constant (usually small) number of hops away from it. Algorithmic design based on locality guarantees stability (changes in the network outside a constant neighborhood do not influence the computation), consistency of solutions regardless of the order of execution, and constant termination time of the proposed algorithm. This approach was further investigated in the work of (Naor and Stockmeyer [1995]) which investigated constant-time solutions for labeling problems and the book of (Peleg [2000]) which proposes a locality-sensitive approach to distributed computing.

Nodes in wireless networks have limited transmission range and communication between two nodes depends on their Euclidean distance. A standard model of wireless network is the Unit Disk Graph (UDG) which consists of nodes with identical transmission range, say one unit. In this graph, two nodes are adjacent if and only they are within range of each other. Many graph-theoretic problems do not admit local algorithms, even when restricted to the class of UDGs. To overcome this limitation an important assumption concerns location awareness, whereby nodes are assumed to know their geographic positions obtained either from a GPS receiver or from virtual coordinates assigned by another source. Location awareness in conjunction with locality is an important paradigm for the design of efficient algorithms in ad hoc networks. Several networking problems become solvable in the local setting when the network is location aware. In this case the graph is embedded in the plane and each node knows its geographic position. Algorithms for location aware networks are sometimes easier to design and they may lead to better time complexities and/or approximation bounds.

3 ISSUES IN TOPOLOGY CONTROL

In this section we discuss how locality in conjunction with location awareness can be used to provide algorithms suitable for topology control in wireless ad hoc networks.

3.1 Routing

Face routing is a technique that was first proposed in (Kranakis et al. [1999]) in order to discover routes in a geometrically embedded planar subdivision. Given a source s and a destination t it discovers a route by traversing only the faces crossed by the straight line γ formed by the nodes s and t. After each face traversal it advances to a new face of the planar subdivision. It is guaranteed to succeed because each face traversal reduces the (geometric) Euclidean distance of the current position to the target. The important feature of this algorithm is locality in the route discovery process. At each step (see Figure 1)) progress is made along a face of the subdivision and it is irrelevant what happens in the remaining part of the graph as long as it remains connected.

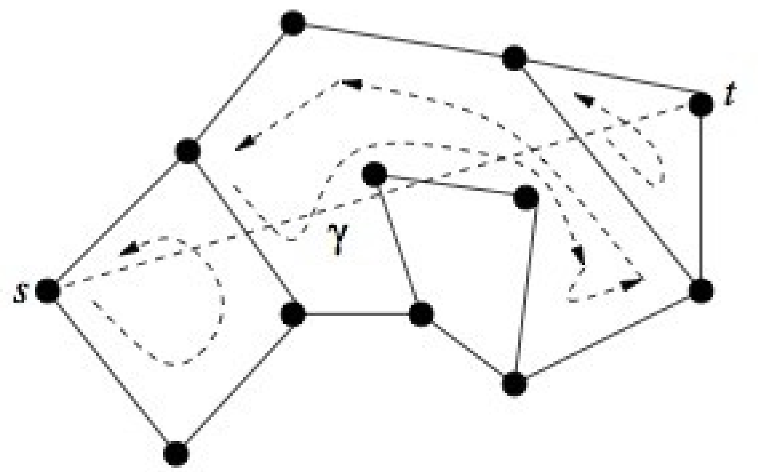

Figure 1: Discovering a route between s and t in a planar graph.

Moreover, to succeed one never has to remember anything more than the straight line from s to t and the current position, information that is easily acquired on-line by a GPS.

Because of the importance of face routing for wireless networking, efforts have been made to extend this result to richer classes of networks. It is therefore worth mentioning (Chavez et al. [2006b]) which discusses route discovery with constant memory in oriented planar geometric networks (Eulerian and Outer-planar), as well as the work of (Kranakis et al. [2006]) which discusses on-line routing in quasi-planar graphs (a class of graphs with distinct faces which allow edge crossings only within faces). There is also some recent work to specific three dimensional representations of graphs. For example (Kranakis et al. [2006]) studies routing in polyhedral geometrically embedded graphs, (Fraser [2007]) extends face routing to geometric graphs of genus one (i.e., embedded on a torus), and (Durocher et al. [2008]) extends face routing to three dimensional graphs delimited by two parallel planes at distance $1/\sqrt{2}$.

3.2 Traversal

Network traversal is a technique widely used in networking for visiting every node of a network, using a small number of steps when required to process the nodes, edges, faces, etc, of a network in some order. For example, it may involve reporting each node, edge, and face of a planar graph exactly once, in order to apply some operation to each. As such it can be used to discover network resources, implement security policies, and report network conditions. Traversal can be used to discover routes between two hosts, but in general it will be less efficient than routing since it cannot guarantee that its discovery process will be restricted to employing only information relevant to routing.

Although DFS (Depth First Search) of the primal nodes and edges or dual faces and edges of the graph is the usual approach followed for implementing traversal, usually it cannot be implemented without using mark bits on the nodes, edges, or faces, and a stack or queue. The traversal technique from (Chavez et al. [2006b]) is applicable to the class of quasi-planar networks (this is a class of subdivisions of the plane in which we allow many edges to cross each other). The general idea of the algorithm is to define a total order on all edges thus giving rise to a unique predecessor for every quasi-face (a closed walk in the subdivision). The predecessor relationship imposes a virtual directed tree. The algorithm will search for the root of this tree and then will report quasi-faces of the graph in DFS order on the tree. For this, a well-known tree-traversal technique is used in order to traverse the tree using O(1) additional memory.

3.3 Planar spanners

Enabling face routing provides important motivation for the design of algorithms that construct spanners (i.e., planar subgraphs) of UDGs. To be useful in an ad hoc network setting, algorithms for constructing spanners should be local. Additional important characteristics of such spanners should include connectivity (basic requirement for message delivery), low degree (eases channel allocation and frequency assignment problems and/or time multiplexing constraints), stretch factor (maximal ratio of the length–hops or Euclidean–of the shortest path in the subgraph with respect to the length of the shortest path in the original graph, and cost (total length of the edges of the subgraph–can also use squares of lengths–as compared to the length of the edge in the MST.

The Gabriel Graph (Bose et al. [2001]) was one of the first such spanners; two nodes keep their link if and only if the disk having as diameter the line with the two nodes as end-points contains no other node from the network (see Figure 2). The Local Minimum Spanning Tree (Li et al. [2004]) produces a planar spanner by having each node construct the minimum spanning tree of its distance k neighborhood; a link between two nodes remains in the spanner if and only if it belongs to the distance k spanning trees of both nodes. An extension of this result to Quasi Unit Disk Graphs (see Barriere et al. [2003]) is given in (Chavez et al. [2006a]). A similar idea also works

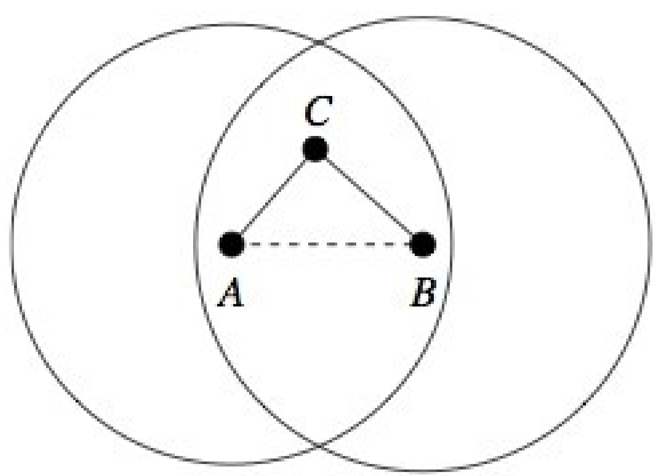

Figure 2: In the Gabriel test nodes A,B forward packets to each other via node C.

for Local Delaunay Triangulations, since in this case a triangle is defined by three vertices whose circle contains no other points from the point-set.

Half-Space Proximal (Chavez et al. [2006c]) is another class of subgraphs of the UDG whereby each vertex determines the closest vertex and excludes all vertices lying on the other side of the bisector of the line formed by these two nodes; it then iterates until no node is left uncovered. The resulting graph is a spanner but unlike the Yao graph, the Half-Space Proximal does not require globally consistent orientation by the nodes. Some of its nice properties include connectivity, constant stretch-factor, and maximum degree five.

3.4 Dominating and connected dominating sets

Consider a graph G. A set D of vertices dominates a vertex u if there is a vertex v in D such that {u, v} is an edge. The set D is called a dominating set for G if it dominates every vertex in G. A set D is called a minimum dominating set for G if it is a dominating set with minimum cardinality. We call a dominating set a connected dominating set if the subgraph induced by its vertices is connected. The minimum dominating set problem is concerned with finding such a dominating set.

Dominating sets are used as a backbone infrastructure and help maintain network stability, power conservation, limit interference, and reduce the number of nodes that contain routing information in ad hoc networks. This is accomplished by organizing the nodes in clusters. One vertex in each cluster takes the role of a leader (often called cluster-head) and the other vertices in the cluster are assigned to this cluster-head (see Figure 3).

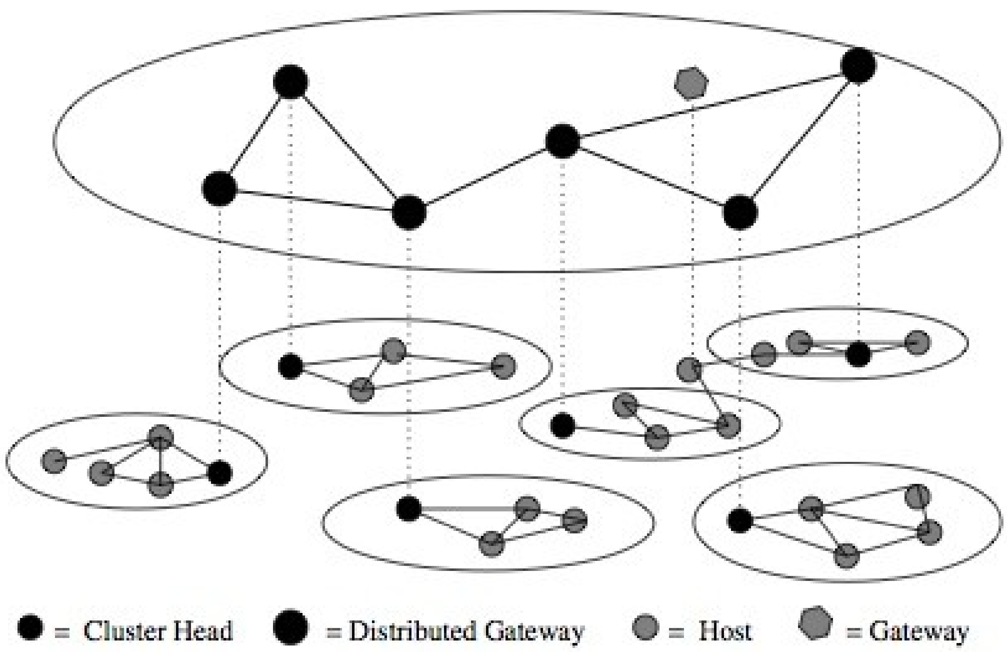

Figure 3: Dominating sets and cluster formation.

Thus the cluster-heads form a dominating set and are responsible for the communication of the members of the cluster. In order to be able to send messages from one cluster to another the cluster-heads form a connected graph which results in a connected dominating set.

Despite the fact that dominating set and connected dominating set problems are NP-hard, polynomial-time approximation schemes have recently been constructed for UDGs. The resulting approximation bounds, although they appear to be better when restricted to the class of UDGs, apply mainly to the non-local setting where all the nodes have complete knowledge of the entire network. Extending the main idea of (Czyzowicz et al. [2008a]) on tiling the plane (Wiese and Kranakis [2008]) give a polynomial time approximation scheme for dominating and connected dominating sets of UDGs.

The main idea of the algorithm is to tile the plane with hexagons each assigned a class number. For some hexagons h we construct a set T_h that contains all nodes in h and the nodes in a certain surrounding area. These sets T_h are disjoint and have certain properties that ensure an approximation ratio which is as close to 1 as we wish. The sets T_h are constructed by iterating over the class numbers of the hexagons. First we cover hexagons of class 1 by computing sets T_h for all hexagons h of class 1. Assume that all hexagons of class i have already been covered. We proceed to cover all hexagons of class i + 1 whose vertices have not been completely covered so far by computing sets T_h for those hexagons. We stop when all vertices in all hexagons have been covered. Moreover, the number of iterations does not exceed the total number of classes. Finally we compute for all sets T_h the minimum dominating set $D(T_h)$. We output as D the union of the sets $D(T_h)$. The processing time that each vertex needs to determine whether or not it is part of the computed set is bounded by a polynomial in the number of vertices which are a constant number of hops away from it.

3.5 Vertex and edge coloring

Graph coloring problems have numerous applications in scheduling and channel assignment. Frequency channel assignment is modeled by a graph in which two vertices are connected by an edge if the broadcasting units of their respective nodes interfere and therefore have to be assigned different channels. Since channels in the frequency band are limited and expensive resources the aim is to minimize the total number of used frequencies.

In (Czyzowicz et al. [2008b]) a local algorithm for 7-coloring planar subgraphs of UDGs is presented by using elaborate tilings of the plane. Each vertex can compute its color in a 7-coloring of the planar graph using only information on the subgraph located within at most a constant number (in our case h = 201) of hops away from it. The algorithm does not need to determine locally either what the different connected components are or even what are the local parts of a component connected somewhere far away and the complexity depends on the size of the data acquired within the specified number of hops.

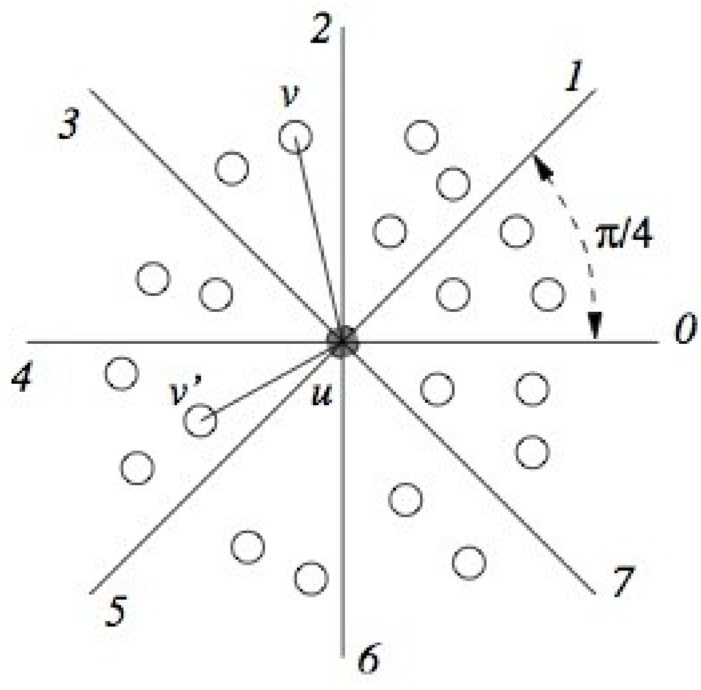

Figure 4: Forming a wedge graph at vertex u with k = 4 and l = 3.

In (Czyzowicz et al. [2007]) a local algorithm is presented for edge colouring (l, k)-edge/wedge subgraphs of UDGs, for all integers l, k. These are geometric graphs (see Figure 4) such that for some positive integers l, k the following property holds at each node u: if we partition the unit circle centered at u into 2k equally sized wedges then each wedge can contain at most l points different from u. An important parameter in the algorithm is the horizon distance d: a given node u never needs to be aware of the location of nodes beyond its horizon, as measured by the euclidean distance from u. The basic 2k + 1 edge coloring algorithm for (l, k)-edge/wedge subgraphs of UDGs, which is presented in this paper, uses a local horizon distance 7.81·lk.

4 CONCLUSION

We discussed techniques for local topology control in location aware Unit Disk Graphs. Our survey included recent local algorithms for Routing, Traversal, Planar Spanners, Dominating and Connected Dominating Sets, and Vertex and Edge Coloring. In addition to investigating trade-offs for the previously mentioned issues, several remaining open problems will play important role in the future development of the subject. These include solutions for three dimensional ad hoc hoc networks, study of power assignments in the physical interference model and the inclusion of realistic models of mobility.

References

L. Barriere, P. Fraigniaud, L. Narayanan, and J. Opatrny. Robust positionbased routing in wireless ad hoc networks with irregular transmission ranges. Wireless Communications and Mobile Computing, 3(2):141–153, 2003.

P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. Wireless Networks, 7:609–616, 2001.

E. Chavez, S. Dobrev, E. Kranakis, J. Opatrny, L. Stacho, and J. Urrutia. Local construction of planar spanners. In LATIN (Valdivia, Chile, March 20-24, 2006), volume 3887 of LNCS, pages 286–297, 2006a.

E. Chavez, S. Dobrev, E. Kranakis, J. Opatrny, L. Stacho, and J. Urrutia. Route discovery with constant memory in oriented planar geometric networks. Networks, 48:7–15, 2006b.

E. Chavez, S. Dobrev, E. Kranakis, J. Opatrny, H. Tejeda, and J. Urrutia. Half-space proximal: A new local test for extracting a bounded dilation spanner. In OPODIS 2005, volume 3544 of LNCS, 2006c.

J. Czyzowicz, S. Dobrev, E. Kranakis, J. Opatrny, and J. Urrutia. Local edge colouring of Yao-like subgraphs of unit disk graph. In 14th SIROCCO (June 6-8, 2007, Livorno, Italy), volume 4474 of LNCS, 2007.

J. Czyzowicz, S. Dobrev, T. Fevens, H. Gonzalez-Aguilar, E. Kranakis, J. Opatrny, and J. Urrutia. Local algorithms for dominating and connected dominating sets of unit disk graphs. In LATIN 2008, April 07-11, Buzios, Brazil, LNCS, 2008a.

J. Czyzowicz, S. Dobrev, H. Gonzalez-Aguilar, E. Kranakis, J. Opatrny, and J. Urrutia. Local 7-coloring for planar subgraphs of unit disk graph. In TAMC 2008, April 25-29, Xi’an, China. LNCS, vol. 4978, pp. 170-181, 2008b.

S. Durocher, D. Kirckpatrick, and L. Narayanan. On routing with guaranteed delivery in three dimensional ad hoc wireless networks. In ICDCN, pages 546–557, 2008.

M. J. Fraser. Local routing on tori. In Proceedings of 6th ADHONOW, volume 4686 of LNCS, pages 1–14, 2007.

E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In Proceedings of the Canadian Conference on Computational Geometry (CCCG-99), pages 51–54, Aug. 15–18 1999.

E. Kranakis, T. Mott, and L. Stacho. Online routing in quasi-planar and quasi-polyhedral graphs. In 2nd IEEE PerCom Workshop on Pervasive Wireless Networking (PWN06), 2006. Discrete Applied Mathematics, Volume 156, Issue 18, 28 November 2008, pp. 3430-3442.

X.-Y. Li, Y. Wang, and W.-Z. Song. Applications of k-Local MST for topology control and broadcasting in wireless ad hoc networks. In INFOCOM- 04, Piscataway, NJ, USA, Mar. 7–11 2004.

N. Linial. Locality in distributed graph algorithms. SIAM Journal on Computing, 21(1):193–201, Feb. 1992.

M. Naor and L. Stockmeyer. What can be computed locally? SIAM Journal on Computing, 24, 1995.

D. Peleg. Distributed Computing: A Locality-Sensitive Approach. SIAM, 2000. Monographs in Discrete Mathematics and Applications.

J. Urrutia. Local Solutions for Global Problems in Wireless Networks. Journal of Discrete Algorithms, 5(3): 395-407, 2007.

A. Wiese and E. Kranakis. Local PTAS for dominating and connected dominating set in location aware unit disk graphs. In 6th Workshop on Approximation and Online Algorithms. LNCS, 2008.

KEY TERMS AND THEIR DEFINITIONS

1. Eulerian Graph: A directed graph such that for every vertex its in-degree equals its out-degree.

2. GPS: Geographic Positioning System.

3. Local Algorithm: A communication algorithm whereby messages need only propagate a constant number of hops independent of the size of the network.

4. Location Aware Network: A wireless network where all hosts know their geometric location.

5. Minimum Spanning Tree: A spanner of a graph which has no cycles and has minimum weight among all such spanners.

6. Outer-planar Graph: A planar graph all of whose vertices lie on a cycle.

7. Planar Graph: Geometric representation of a graph so that no two edges cross.

8. Planar Face: A cycle in a planar graph with no internal edges.

9. Planar Spanner: A connected planar subgraph of a graph which spans all the vertices of the graph.

10. Stretch Factor of a Spanner: The worst case ratio of the length of a minimum path between two nodes in the graph divided by the length of a minimum path between these two same nodes in the spanner.

11. Traversal: An exploration technique for visiting every node (or link) of a network.

12. UDG (Unit Disk Graph): A graph consisting of wireless hosts with identical transmission range.