Available online at www.sciencedirect.com
SCIENCE DIRECT®

Computer Communications 28 (2005) 1498–1504

computer communications

www.elsevier.com/locate/comcom

Algorithms for ad hoc and sensor networks[☆]

Roger Wattenhofer

ETH Zurich, Zurich, Switzerland

Received 31 August 2004; accepted 15 December 2004
Available online 26 April 2005

Abstract

Wireless and mobiles networks are excellent playground for researchers with an algorithm background. Many research problem turn out to be variants of classic graph theory problems. In particular the rapidly growing areas for ad hoc and sensor networks demand new solutions for timeless graph theory problems, because: (i) wireless devices have lower bandwidth and (ii) wireless devices are mobile and therefore the topology of the network changes rather frequently. As a consequences, algorithms for wireless and mobile networks should have: (i) as little communication as possible and should (ii) run as fast as possible. Both goals can only be achieved by developing algorithms requiring a small number of communication rounds only (so-called *local* algorithm). In the work we present a few algorithmic applications in wireless networking, such as clustering, topology control and geo-routing. Each section is supplemented with an open problem.

© 2005 Elsevier B.V. All rights reserved.

Keywords: Topology control; Interference; Clustering; Dominating sets; Geo-routing

1. Introduction

An ad hoc or sensor network consists of mobile nodes featuring, among other components, a processor, some memory, a wireless radio, and a power source; physical constraints often require the power source to be feeble—a weak battery or a small solar cell.

Ad hoc and sensor networks are emerging areas of research that have been studied intensively for a few years only. Roughly, the researchers investigating ad hoc and sensor networks can be classified into two categories. On the one side there are the systems researchers who build real ad hoc or sensor networks; the Berkeley Motes project [16] is a popular hardware platform marketed by Crossbow (www.xbow.com) that is used in many deployments, but alternative hardware platforms are available as well [5,34]. On the other hand there are the theoreticians who try to understand the fundamentals of ad hoc and sensor networks, by abstracting away a few 'technicalities' that arise in real systems.

Not surprisingly—as in other areas of computer science and engineering—there is no consensus what the technicalities are. Most theoreticians model the networks as nodes (points) in a Euclidean plane; two nodes can communicate if they are within their mutual transmission range, which in an unobstructed and homogeneous environment translates into whether their Euclidean distance is at most the maximum transmission range R. This model is widely known as unit disk graph and—though not quite practical—respected as a first step by practitioners.

More surprisingly, however, most theoreticians make much stronger assumptions. It seems that a majority of papers assumes that the nodes are distributed uniformly at random. At a high node density, such a postulation renders many problems trivial. Also it is not clear that a uniform node density distribution makes sense from a practical point of view. Recently deployed large-scale sensor networks report highly heterogeneous node densities—in 'interesting' areas there are several nodes per square meter, whereas in other ('routing-only') areas nodes are hundreds of meters apart. For mobile ad hoc networks (MANET's), it is often assumed that the nodes move Brownian, a behavior that is not often seen in our macroscopic world.

In this paper we advocate using more realistic *graph theoretical* models. We feel that theoretical research should drop *average-case* assumptions such as uniformly at random distributed nodes and/or Brownian motion, and

---

[☆] A preliminary version of this paper was presented as an invited talk at the 30th Workshop on Graph-Theoretic Concepts in Computer Science [36].

E-mail address: wattenhofer@tik.ee.ethz.ch.

0140-3664/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2004.12.037

instead study *worst-case* distributions and motion models. In this paper we outline a selection of the algorithms that were developed to work also in the non-uniform worst-case.

The paper is organized as follows. In Sections 2, 3, and 4, we sketch a number of algorithmic results in three key areas of ad hoc and sensor networking. In Section 2 we discuss topology control, in Section 3 clustering, and in Section 4 geo-routing, a special but well-studied form of routing. In Section 5 we conclude the paper.

2. Topology control

Since energy is the limiting factor for lifetime and operability of an ad hoc network, researchers have developed a variety of mechanisms and algorithms to conserve energy. These mechanisms and algorithms are often dubbed ‘topology control’.

For two communicating ad hoc nodes u and v, the energy consumption of their communication grows at least quadratically with their distance. Having one or more relay nodes between u and v therefore helps to save energy. The primary target of a topology control algorithm is to abandon long-distance communication links and instead route a message over several small (energy-efficient) hops. For this purpose each node in the ad hoc network chooses a ‘handful’ of ‘close-by’ neighbors ‘in all points of the compass’ (we are going to fill in the details later). Having only near neighbors not only helps reducing energy but also interference, since fewer nodes are disturbed by high power transmissions. Clearly nodes cannot abandon links to ‘too many’ faraway neighbors in order to prevent the ad hoc network from being partitioned or the routing paths from becoming non-competitively long. In general there is a trade-off between network connectivity and sparseness.

Let the graph $G=(V, E)$ denote the ad hoc network before running the topology control algorithm, with V being the set of ad hoc nodes, and E representing the set of communication links. There is a link (u, v) in E if and only if the two nodes u and v can communicate directly. Running the topology control algorithm will yield a sparse subgraph $G_{tc}=(V, E_{tc})$, of G, where E_{tc} is the set of remaining links. The resulting topology G_{tc} should have a variety of properties:

(i) Symmetry. The resulting topology G_{tc} should be symmetric, that is, node u is a neighbor of node v if and only if node v is a neighbor of node u. Asymmetric communication graphs are unpractical, because many communication primitives become unacceptably complicated [32].

(ii) Connectivity/Spanner. Two nodes u and v are connected if there is a path from u to v, potentially through multiple hops. If two nodes are connected in G, then they should still be connected in G_{tc}. Although a minimum spanning tree is a sparse connected subgraph, it is often not considered a good topology, since close-by nodes in the original graph G might end up being far away in G_{tc} (G being a ring, for instance). Therefore the graph G_{tc} is generally not only being asked to be connected, but a spanner. For any two nodes u and v, if the optimal path between u and v in G has cost c, then the optimal path between u and v in G_{tc} has cost $O(c)$.

(iii) Sparseness/Low Degree/Low Interference. The remaining graph G_{tc} should be sparse, that is, the number of links should be in the order of the number of nodes. More ambitiously, one might even ask that *each node* in the remaining graph G_{tc} has a low (constant) degree. Since a low degree alone does not automatically imply low interference (after all nodes might choose few but very far away neighbors!), some researchers have started studying topology control algorithms that concentrate on the interference issue.

(iv) In addition to the properties (i)–(iii) one can often find secondary targets. For instance, it is popular to ask the remaining graph to be planar in order to run a geometric routing algorithm, such as GOAFR [28].

Since connectivity and sparseness run against each other, topology control has been a thriving research area.

The currently best algorithms feature an impressive list of properties. Wang and Li [35] present the currently most promising proposal—a distributed topology control algorithm that computes a planar constant-degree distance-spanner. (As opposed to energy-spanners as considered in earlier work [37,17].) However, the distributed algorithm might be quite slow; in an unlikely (but possible) worst-case instance it will run for a linear number of steps. Also, like many others this algorithm makes strong assumptions: first, all the nodes need to know their exact positions, by means of a global positioning system (GPS) for example. Second, the algorithm assumes that the world is flat and without buildings (a perfect unit disk graph, so to speak). These assumptions make the algorithm unpractical.

In an almost ‘retro’ approach [38] recently presented the XTC algorithm that works: (i) without GPS and (ii) even in a mountainous and obstructed environment. Surprisingly the XTC algorithm features all the basic properties of topology control (symmetry, connectivity, low degree) while being faster than any previous proposals.

All known topology control algorithms including [35] and XTC [38] do not explicitly address interference, but argue that the sparseness or low degree property will take care of it.[1] In [9] it has recently been shown that

---

[1] Meyer auf der Heide et al. [29] are a notable exception who study interference explicitly, however, not in the context of topology control, but in relation to traffic models. They show that there are worst-case ad hoc networks and worst-case traffic, where only one of the performance parameters congestion, energy, and dilation can be optimized at a time.

Download English Version:

https://daneshyari.com/en/article/447467

Download Persian Version:

https://daneshyari.com/article/447467

Daneshyari.com