

# Brief Announcement: Virtual Stationary Automata for Mobile Networks

Shlomi Dolev, Limor Lahiani  
Department of Computer Science  
Ben-Gurion University  
Beer-Sheva, Israel, 84105  
{dolev, lahiani}@cs.bgu.ac.il

Seth Gilbert, Nancy Lynch, Tina Nolte  
MIT Computer Science & Artificial Intelligence  
Laboratory  
Cambridge, MA 02139  
{sethg,lynch,tolte}@theory.csail.mit.edu

**Categories and Subject Descriptors:** C.2.4 [Computer-Communication Networks]: Distributed Systems

**General Terms:** Algorithms

**Keywords:** Ad-hoc networks, virtual infrastructure, mobile computing, state replication, self-stabilization

The task of designing algorithms for constantly changing networks is difficult. We focus on mobile ad-hoc networks, where mobile processors attempt to coordinate despite minimal infrastructure support. We develop new techniques to cope with this dynamic, heterogeneous, and chaotic environment. We mask the unpredictable behavior of mobile networks by defining and emulating a *virtual* infrastructure, consisting of *timing-aware* and *location-aware* machines at fixed locations, that mobile nodes can interact with. The static virtual infrastructure allows application developers to use simpler algorithms — including many previously developed for fixed networks.

**Virtual Stationary Automata programming layer.** Our programming abstraction consists of a static infrastructure of fixed, timed virtual machines with an explicit notion of real time, called *Virtual Stationary Automata* (VSAs), distributed at known locations over the plane, and emulated by the real mobile nodes in the system. Each VSA represents a predetermined geographic area and has broadcast capabilities similar to those of the mobile nodes, allowing nearby VSAs and mobile nodes to communicate with one another. This programming layer provides mobile nodes with a virtual infrastructure with which to coordinate their actions. Many practical algorithms depend significantly on timing, and it is reasonable to assume that many mobile nodes have access to reasonably synchronized clocks. In the VSA programming layer, the virtual automata also have access to *virtual* clocks, guaranteed to not drift too far from real time.

Our virtual infrastructure differs in key ways from others that have previously been proposed for mobile ad-hoc networks. The GeoQuorums algorithm [2] was the first to use virtual nodes; the virtual nodes in that work are atomic objects at fixed geographical locations. More general virtual mobile automata were suggested in [1]; our automata are more powerful than those in [1] in that ours include timing capabilities, which are important for many applications. Also, our automata are stationary, and are arranged in a connected pattern that is similar to a traditional wired ne-

twork. Finally, our implementation of virtual nodes incurs less communication cost than those in [1,2]. Note that the virtual nodes described in [1,2] could all be implemented using the infrastructure we describe here.

**Emulating the virtual infrastructure.** Our clock-enabled VSA layer is emulated by real mobile nodes in the network. Each mobile node is assumed to have access to a GPS service informing it of the region it is currently in. A VSA for a particular geographic region is then emulated by a subset of the mobile nodes currently populating its region: the VSA state is maintained in the memory of the real nodes emulating it, and the real nodes perform VSA actions on behalf of the VSA. The emulation is shared by the nodes while one node designated as leader is responsible for performing the outputs of the VSA and keeping the other nodes consistent in their VSA state. If no mobile nodes are in the region, the VSA fails; if mobile nodes later arrive, the VSA restarts.

An important property of our implementation is that it is self-stabilizing. Self-stabilization is the ability to recover from an arbitrarily corrupt state. This property is important in long-lived, chaotic systems where certain events can result in unpredictable faults. For example, transient interference may disrupt the wireless communication, violating our assumptions about the broadcast medium. This might result in inconsistency and corruption in the emulation of the VSA. Our self-stabilizing implementation, however, can recover after corruptions to correctly emulate a VSA.

**Applications.** We have considered applications that are significantly simplified by the VSA infrastructure. These include low-level services, such as routing and location management, as well as more sophisticated applications, such as motion coordination and tracking. The key idea in all cases is to locate data and computation at timed virtual automata throughout the network, thus relying on fixed, predictable infrastructure to simplify coordination in ad-hoc networks. It is interesting to note that this infrastructure can be used to implement services such as routing that are oftentimes thought of as the lowest-level services in a network.

[1] Dolev, S., Gilbert, S., Lynch, N., Schiller, E., Shvartsman, A., and Welch, J., “Virtual Mobile Nodes for Mobile Ad Hoc Networks”, *International Conference on Principles of Distributed Computing (DISC)*, 2004.

[2] Dolev, S., Gilbert, S., Lynch, N., Shvartsman, A., Welch, J., “Geo-Quorums: Implementing Atomic Memory in Ad Hoc Networks”, *17th International Conference on Principles of Distributed Computing (DISC)*, Springer-Verlag LNCS:2848, 2003.

**FULL VERSION** is available as a Technical Report, MIT-LCS-TR-979, MIT CSAIL, January 2005.