

A New Task-Based Approach for Supporting Distributed Applications on Mobile Ad Hoc Networks¹

P. Basu, W. Ke, and T.D.C. Little

Department of Electrical and Computer Engineering
Boston University, Boston, Massachusetts 02215, USA
(617) 353-9877
tdcl@bu.edu

MCL Technical Report No. 07-10-2001

Mobile ad hoc networks (MANETs) have received significant attention recently owing to the increasing popularity of tetherless computing and the rapid growth of wireless networking technology. In this paper, we present a novel application framework for *dynamic* execution of distributed tasks on potentially mobile devices which form a MANET. In this framework, logical patterns of interaction between simpler components of a distributed application are exploited for the discovery of suitable devices/resources on which the application is actually executed. Our application framework is adaptive to network partitions caused by mobility of devices as it enables users to obtain a required service from several devices (providing the same service) over the lifetime of an application. This work helps enable new applications in diverse areas such as smart offices/homes, sensor networks, distributed computing, and disaster relief.

A distributed application is represented as a *task* comprised of smaller, less complex *sub-tasks* that need to be performed on different types of computing resources/devices with specialized roles. Dependencies are induced by logical patterns of data flow between different categories of devices responsible for performing the aforementioned sub-tasks. Such dependencies yield an abstract *task graph* (TG) representation for a given application. We focus on the problem of efficiently executing distributed tasks on a MANET by means of dynamic discovery and selection of devices/resources for completion of simpler sub-tasks. The discovery of devices is carried out with respect to the underlying TG representation from among a multitude of candidate devices offering similar functionality. We refer to this process as *task embedding*.

¹*ACM Computer Communication Review*, January 2002. This work is supported in part by the NSF under Grant No. ANI-0073843.

²A weight function $w : E_T \rightarrow \Re^+$ associated with each edge can signify characteristics such as data rate, relative importance in task etc.

A task graph $TG = (V_T, E_T)$ is composed of a set of *nodes*, V_T , and a set of *edges*, E_T . The nodes represent *classes* of devices, resources or services (e.g. Printer, Speaker, Camera etc.) needed for processing data related to the task, whereas the edges represent necessary logical associations between different nodes for performing the task². The process of *embedding* a task graph TG onto a MANET $G = (V_G, E_G)$ involves *mapping* or *instantiation* of logical TG *nodes* and *edges* onto physical devices/resources and shortest paths between them, respectively. This is equivalent to finding a pair of mappings $\varphi : V_T \rightarrow V_G$ and $\psi : E_T \rightarrow P_G$, where the *class* of $v \in V_T$ is the same as that of $\varphi(v)$, and P_G is the set of all source-destination paths in G . The quality of a mapping (φ, ψ) can be measured by a metric called *average dilation* which is the average over all edges in TG of the length in hops of the shortest path between two mapped devices in the network, whose logical counterparts in TG were connected by an edge. Minimizing dilation is one of the primary goals of our embedding algorithm since a low value of dilation indicates that most resources were found *near* the devices that needed those resources.

Embedding of *general* TGs onto *general* network graphs appears to be computationally hard but we have developed an optimal algorithm for a special case of TGs, namely *trees*. However, this optimal algorithm for tree TGs has a high time complexity. Hence, we developed a distributed greedy heuristic algorithm whose operation can be summarized as follows: the algorithm starts from the root node of TG and conducts a search through TG along the edges of its maximum weight spanning tree $MaxST_{TG}$ in a breadth first order. Every TG node that is visited is mapped to some physical device in the network that possesses the capabilities to perform the sub-task specified by that node. The above process is coordinated in a distributed fashion as the instantiation of a child node in $MaxST_{TG}$ is the responsibility of its already instantiated parent.

Although MANET routing protocols dynamically adapt to changes in network topology due to mobility of devices, route failures can sometimes disrupt a currently executing task. In more severe cases, mobility can cause network partitions too. A primary focus of this work is to make the previously described algorithm adaptive to such events. We achieve this by *periodically* monitoring the logical associations between the instantiated devices by exchanging *soft state*, possibly over multiple MANET hops, and by triggering recovery mechanisms if these associations are severed. If an instantiated child is disconnected from its parent, the latter triggers a re-instantiation process to discover a replacement device with similar capabilities. Then, the application is resumed after a few necessary state synchronization steps, if possible.

In this paper, we introduced the concept of task graph based application modeling for MANETs. We briefly described a distributed algorithm for efficient discovery of resources required for executing a given distributed task. We also described how the algorithm can adapt to mobility of devices and survive network partitions. The aforementioned algorithms have been presented in detail (including extensive simulation results) in [1]. More information about the project can be found at reference [2].

References

- [1] P. Basu, W. Ke, and T.D.C. Little, “A Novel Task-based Approach for Supporting Distributed Applications on Mobile Ad Hoc Networks,” *BU MCL Technical Report TR-072201*, July 2001.
- [2] MANET Research Summary, <http://hulk.bu.edu/projects/adhoc/summary.html>.