SpaceTime Oriented Programming: A New Macro-Programming Paradigm in Data Collection Sensor Networks

Hiroshi Wada and Junichi Suzuki
Department of Computer Science
University of Massachusetts, Boston

Program Insert

This paper proposes a new programming paradigm for wireless sensor networks (WSNs). It is designed to significantly reduce the complexity of WSN programming by providing a new high-level programming abstraction to specify spatio-temporal data collections in a WSN from a global viewpoint as a whole rather than sensor nodes as individuals.

Abstract

SpaceTime Oriented Programming (STOP) is new macro-programming paradigm for wireless sensor networks (WSNs) in that it supports both spatial and temporal aspects of sensor data and it provides a high-level abstraction to express data collections from a global viewpoint for WSNs as a whole rather than sensor nodes as individuals. STOP treats space and time as first-class citizens and combines them as spacetime continuum. A spacetime is a three dimensional object that consists of a two special dimensions and a time playing the role of the third dimension. STOP allows application developers to program data collections to spacetime, and abstracts away the details in WSNs, such as how many nodes are deployed in a WSN, how nodes are connected with each other, and how to route data queries in a WSN. Moreover, STOP provides a uniform means to specify data collections for the past and future.

Using the STOP application programming interfaces (APIs), application programs (STOP macro programs) can obtain a "snapshot" space at a given time in a given spatial resolutions (e.g., a space containing data on at least 60% of nodes in a give spacetime at 30 minutes ago.) and a discrete set of spaces that meet specified spatial and temporal resolutions. (e.g., a list of spaces at every 5 minutes, each containing at least 80% of nodes). Given a snapshot space(s), a STOP macro program can specifies a series of in-network processing such as aggregating multiple data and calculating the maximum, minimum, average and standard deviation values.

A STOP macro program is transformed into a skeleton program in a traditional language such as Java and TinyScript, and application developers implement application logic (e.g., visualization and data analysis) by extending the transformed program. The translated program examines the past and future in a given spacetime. For the past, it checks whether a local spatial-temporal database has enough data to execute queries. If it does, queries are performed immediately. Otherwise, data query messages are dispatched to the network and routed to a certain spatial region (a certain set of nodes) to collect sensor data that meet a certain temporal range. For the data query for the future, data query messages are always dispatched to the network.

Existing macro programming languages such as Regiment, Kairos and SNLong provide certain abstractions to describe spatial relationships and data reduction operations among nodes. Data collection can be expressed without specifying the details of node-to-node communication and data reduction. However, these languages still requires application developers to work at the node level; they need to explicitly write programs to individual nodes. In contrast, STOP allows

developers to write programs to a higher abstraction; spacetime, thereby improving the productivity and maintainability of WSN programming.