Geospatial Authentication

Stennis Space Center, Mississippi

A software package that has been designed to allow authentication for determining if the rover(s) is/are within a set of boundaries or a specific area to access critical geospatial information by using GPS signal structures as a means to authenticate mobile devices into a network wirelessly and in real-time. The advantage lies in that the system only allows those with designated geospatial boundaries or areas into the server.

The Geospatial Authentication software has two parts — Server and Client. The server software is a virtual private network (VPN) developed in Linux operating system using Perl programming language. The server can be a stand-alone VPN server or can be combined with other applications and services. The client software is a GUI Windows CE software, or Mobile Graphical Software, that allows users to authenticate into a network. The purpose of the client software is to pass the needed satellite information to the server for authentication.

This work was done by Stacey D. Lyle of Geospatial Research Innovation Design for NASA's Stennis Space Center.

Mars Science Laboratory Workstation Test Set

NASA's Jet Propulsion Laboratory, Pasadena, California

The Mars Science Laboratory developed the Workstation TestSet (WSTS) is a computer program that enables flight software development on virtual MSL avionics. The WSTS is the non-real-time flight avionics simulator that is designed to be completely software-based and run on a workstation class Linux PC. This provides flight software developers with their own virtual avionics testbed and allows device-level and functional software testing. The WSTS has successfully off-loaded many flight software development activities from the project testbeds. Flight software developers can now instantiate as many virtual testbeds as there are available computer resources and also enables device level fault injections that are difficult to achieve on real avionics testbeds.

The WSTS provides peripheral component interface (PCI)-card-level simulation of avionics hardware, enabling testing of all but the lowest layers of the flight software. The WSTS utilizes shared-memory and synchronization provisions of POSIX in a Linux environment to provide high-resolution simulation with synchronization of the interaction between simulation and the flight software.

This program was written by David A. Henriquez, Timothy K. Canham, Johnny T. Chang, and Nathaniel J. Villaume of Caltech for NASA's Jet Propulsion Laboratory.

Computing Bounds on Resource Levels for Flexible Plans

New algorithm entails less computation than previous algorithms.

Ames Research Center, Moffett Field, California

A new algorithm efficiently computes the tightest exact bound on the levels of resources induced by a flexible activity plan (see figure). Tightness of bounds is extremely important for computations involved in planning because tight bounds can save potentially exponential amounts of search (through early backtracking and detection of solutions), relative to looser bounds.

The bound computed by the new algorithm, denoted the resource-level envelope, constitutes the measure of maximum and minimum consumption of resources at any time for all fixed-time schedules in the flexible plan. At each time, the envelope guarantees that there are two fixed-time instantiations — one that produces the minimum level and one that produces the maximum level. Therefore, the resource-level envelope is the tightest possible resource-level bound for a flexible plan because any tighter bound would exclude the contribution of at least one fixed-time schedule. If the resource-level envelope can be computed efficiently, one could substitute looser bounds that are currently used in the inner cores of constraint-posting scheduling algorithms, with the potential for great improvements in performance.

What is needed to reduce the cost of computation is an algorithm, the measure of complexity of which is no greater than a low-degree polynomial in N (where N is the number of activities). The new algorithm satisfies this need. In this algorithm, the computation of resource-level envelopes is based on a novel combination of (1) the theory of shortest paths in the temporal-constraint network for the flexible plan and (2) the theory of maximum flows for a flow network derived from the temporal and resource constraints. The measure of asymptotic complexity of the algorithm is $O(N \cdot O(\text{maxflow}(N))$, where $O(x)$ denotes an amount of computing time or a number of arithmetic operations proportional to a number of the order of x and $O(\text{maxflow}(N))$ is the measure of com-