A Layered Architecture for Location-based Services in Wireless Ad Hoc Networks [1] [2]

Jonathan Agre, Adedji Akinyemi, Lusheng Ji, Ryusuke Masuoka and Pankaj Thakkar
Fujitsu Laboratories of America
8400 Baltimore Ave., Suite 302
College Park, Maryland 20740
(301) 486-0398
jagre@fla.fujitsu.com

Abstract—The design of a Location Services Module (LSM) that provides a foundation for location-aware services is described. The architecture enhances application software development by providing a simple interface that hides details of the particular underlying location determination technologies. Functions for multiple technology switching, error estimation, location tracking, multiple technology fusion and cooperative location determination are defined. Several self-organizing protocols for cooperative location determination algorithms appropriate for wireless ad hoc networks are described. An example implementation of location-based services for a world wide web browser using an LSM is provided.

TABLE OF CONTENTS

1. INTRODUCTION

Location information is becoming increasingly important in many pervasive computing applications ranging from human-oriented information appliances to distributed sensor networks to robotic colonies. Reliance on accurate local knowledge of location is often critical to a given mission, providing enhanced information to/from the end-devices. Applications incorporating location can provide functions such as navigation aids, geographic contextual information, movement tracking, emergency location, geographically selective communication and coordinated spatial sensor measurements. Location information can also be used to improve system operations of networks by including the spatial distribution of users and assets for communications (e.g., routing) and data storage organization. All of these services depend on the timely availability of position information with known accuracy in varied environments.

There are many location determination methods and technologies that can be employed in these systems, however no single method will operate effectively in all intended environments (e.g., indoors, urban canyons and open terrain) with sufficient accuracy. These technologies employ various media such as optical, acoustic or radio frequency (RF) each have advantages in certain situations. RF methods such as Received Signal Strength (RSS), Angle of Arrival (AOA) or Time Difference of Arrival (TDOA) are potentially desirable as they may be integrated with the required wireless communication devices with minimal cost, especially in an indoor environment [1]. The satellite-based, Global Positioning System (GPS) is the most widely used RF system, providing global outdoor coverage [2]. Emerging higher-rate radios such as 802.11a [3] hold great promise in providing accurate position data due to their inherent timing accuracy. In general, we believe that multiple methods will be required to ensure position determination that operates in varied environments with the desired accuracy.

The systems under consideration consist of a collection of intelligent, heterogeneous devices (clients) that can be arbitrarily spatially arranged and communicate using wireless methods. The degree of mobility is considered as a parameter but some motion is assumed possible. In the ideal situation, client locations can and should be determined locally by the clients, if possible, to reduce infrastructure dependency, although there is an energy tradeoff.

Self-organization is a key objective of the architecture as it is desirable to operate with systems consisting of thousands of devices. It is infeasible to manually configure the location determination processes for a large number of mobile devices in random configurations with random environmental characteristics. The location-based services architecture must support self-organization in several senses: 1) locations should be determined with minimal user

[1] 0-7803-7231-X/01/$10.00/© 2002 IEEE
[2] IEEEAC paper #387, Updated November 19, 2001

inputs; 2) multiple location information sources should be combinable (fused) to increase accuracy; 3) seamless switching between indoor and outdoor operation, and 4) cooperation between neighboring clients to determine position or increase accuracy. Future technologies should also be able to be easily integrated into the design.

2. Location-Based Services

In this paper we use the term *location* to refer to the generic concept of a place or situation occupied by a designated user or object. The objects can be physical, symbolic, with varying spatial extent. We will use the term *position* with reference to a more specific coordinate system. An object's position generally refers to the coordinates of a single point that represents an aspect of the object. For example, an item may be located in a building but its 3-d position (e.g., center of mass) can be given as a point using a latitude-longitude-altitude triple with respect to a reference coordinate system.

Software applications that employ location, termed location-aware services, are being increasingly explored within context-aware mobile applications [4]. Many applications using generic information appliances have been identified [5] for museums, tourist information systems [e.g., 6], advertising, and smart spaces [7]. In addition, knowledge of location provides many advantages for the communications infrastructure such as for geographic-based routing schemes [8].

Location services are being proposed in the cellular telephony arena spurred by the deployment of the wireless Emergency 911 capabilities that will locate cell phone users that dial 911 [9]. The US government has mandated that methods to implemented to locate a caller to within 50 to 100 meters depending on whether there is a GPS unit in the handset or not, respectively. A standard has been defined that encompasses many of the cellular location determination methods under investigation for GSM-based cellular phones and provides a framework for establishing common coordinate systems and error estimation [10]. The Location Interoperability Forum (LIF), another cellular telephony group, is proposing standards across cellular technologies [11]. Other organizations such as the Bluetooth Special Interest Group on Local Positioning Profile are exploring these issues for the Bluetooth personal area network domain. There are many systems and products that are currently commercially available or developed as part of ongoing research efforts. A good survey of these products and their capabilities is given in [12].

3. Location Determination Technologies

In [12], the technologies used to determine location have been classified according to a taxonomy based on properties such as media, method, type of information, scale, cost, accuracy/precision and point of computation. The media refers to the underlying phenomena being sensed and can be acoustic, radio-frequency, electro-magnetic, video, infrared, laser, inertial navigation (gyros), etc. The type of information can be physical position or symbolic information; both can be absolute or relative. Symbolic location is defined to be a position relative to some known entity whose location may or may not be precisely known. For example, a user may be logged into a particular computer, but the geographic position of that computer is not precisely known. One may have other related information that can be combined to find a more precise location from symbolic information. The scale refers to the number of clients that can be supported by the technology.

There are three generic location determination methods: 1) Proximity, 2) Triangulation (lateration) and 3) Scene analysis or pattern recognition. Signal strength is often used to determine proximity or sometimes range (through an attenuation model). As a proximity measurement, if a signal is received at several known locations, it is possible to intersect the coverage areas of that signal to determine a "containing" location area. If one knows the angle of bearing (relative to a sphere) and range (distance) from a known point to the target device, then the target location is precisely known in 3-dimensions. Similarly, if one knows the angle of bearing to a target from two known locations, then it is possible to triangulate to determine the x-y coordinates of the target. This can be extended to multi-angulation for higher dimensions. In a similar fashion, if one knows the range from three known positions to a target, then using multi-lateration, it is possible to determine the target location. A GPS receiver uses range measurements to multiple satellites and a multi-lateration algorithm to determine position. Video cameras can be used to describe spatial relationships in scenes using image processing techniques and thereby determine position.

Also, in [12], the location determination methods are also classified according to the point-of-computation: server-based or client-based. However, the location-methods can be further differentiated using the architecture of the implementation to emphasize autonomy. The architecture can vary from infrastructure-dominated to infrastructureless schemes. Infrastructure schemes can be active or passive. In active schemes, there are beacon servers that provide measurement and timing signals that are received by the clients (equipments). The location computations are performed by the clients (as in GPS or LORAN). Beacons may be requested or they may be constantly generated. In passive infrastructure schemes, the beacons (or measurement signals) are generated by the clients and the locations are computed by one or more servers, as in cellular telephone base stations performing pattern matching or angle-of-arrival schemes. The infrastructureless class of systems, where location is performed by clients only, are either independent or cooperative. Independent schemes involve a single client operating from a known position with self-tracking capability such as an inertial navigation (or variants such as star-tracking, compass, etc). Infrastructureless, cooperative schemes, or ad hoc location sensing schemes involve measurements made by multiple clients that must be communicated among clients to determine position. The cooperative schemes cannot determine location individually and must self-organize to perform the communication necessary to share the measurement information. Special emphasis is given in the LSM for cooperative schemes.

4. Location Services Module (LSM)

In order to support these goals, a conceptual design of a Location Service Module (LSM) is described that provides a uniform application programming interface (API) that hides details of the underlying location determination methods and provides useful middleware functions. The LSM is partitioned into a Location Adaptation Layer (LAL) and a Location Dependent Layer (LDL) as shown in Figure 1. The primary reason for the separation is to decouple the application from the advances and evolution of different location technologies and to expose standard forms of location information in order to ease application development.

The LAL controls the location computations and is application dependent, in the sense that it is configurable for a given application, although based on a common set of functions. The primary functions of the LAL are 1: delivery of location data to the upper layers in standard formats through the access interface, 2) coordinate computation, and translation, 3) transparent switching between position technologies, 4) fusion of position data from multiple sources, 5) calculation of location error estimates, 6) tracking and integration of position data for increased accuracy, and 7) coordination of distributed location estimation methods. The LAL interface is defined using current interface languages and standard formats.

The LDL is dependent on the location determination methods that will be employed and provides a generic abstraction to the LAL through conversion routines that hide details of the realtime drivers for multiple attached devices such as for a Global Positioning System (GPS), infrared port (IR), radio-based (RF) or user-derived input (UI). Protocols for communication of position information between neighboring nodes, called cooperative networks (Coop net in Figure 1) are also implemented via this layer.

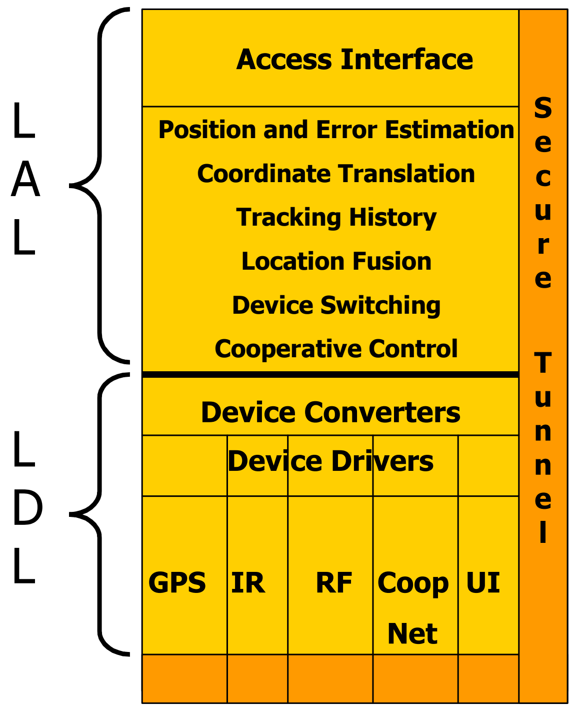


Figure 1: Location Services Module

In addition to the normal API, we see the need for direct access to devices through a secure tunnel facility. This recognizes the varied capabilities of currently available devices. For example, waypoints are often supported by GPS devices, but are not currently supported by the LSM. The LSM is not intended to inhibit use of special features, but rather to support the most common requirements.

5. Location Adaptation Layer

The primary functions of the LAL are described. The LAL is initiated by the LSM at system startup with a set of user-supplied parameters. Many of the LAL routines are configured at startup to execute as independent threads (or process) so that the system status information can be collected as necessary.

Access Interface

The access interface is designed to allow maximum flexibility for application programs to utilize the location data. The interface is defined using the standard world wide web language, Web Services Description Language (WSDL) [13], and conforming to the standard Simple Object Access Protocol (SOAP) interchange protocol [14]. This allows access to the LSM in a uniform fashion from common web tools such as browsers and servers. The interface exposes the other functions of the LAL and supports AAA. However, the AAA is independent and can be tailored for the given application. These functions can be implemented on a client via a light-weight web server (to be described in later section).

The primary function of the access interface is to deliver coordinates to the application in the desired format. A "pull" model is assumed so that the application can request the location information as necessary and control the rate of delivery of location information. The user can also specify the desired accuracy. The LDL will supply the most current information as requested, but allows the device to operate at its own data rate if necessary. A listing of some of the function calls of the interface is given in Table 1.

Coordinates

The location information can be in several forms: absolute, relative and symbolic. The Absolute information provides 3-d position coordinates in standard format relative to the earth, using the World Geodetic Systems, 1984 (WGS84) as the reference system [15]. Latitude and longitude are defined with respect to the WGS84 ellipsoid representation of the earth. Altitude or elevation is defined as the distance above the ellipsoid (representing the surface of the earth). This is similar to the system used by GPS. However, in deviation from the NEMA 0183 GPS standard [16], the coordinates will be expressed in decimal degrees as signed 32-bit words as currently implemented in certain GPS devices [17](versus 24-bit words),. This provides a representation accuracy to below 1 meter for latitude ($-90^0,90^0$) and longitude ($-180^0,180^0$). In the LSM, the altitude is defined as the distance in increments of .01m.

Relative location is defined as the position with respect to an arbitrary location mark defined as the origin. A relative location coordinate system is characterized by the origin coordinates and a reference system. Several alternative reference systems will be supported as required. Initially, the Euclidean and Spherical references will be included. The primary emphasis in the LSM is to obtain and manipulate absolute 3-D positions, however, as shown in a later section, it is necessary to deal with relative coordinates. Coordinate transformation functions will be used to perform conversions between differing representations of location.

Orientation

Associated with position are three additional coordinates that provide the orientation of the equipment. These are commonly referred to as roll (rotation about longitudinal axis), pitch (oscillation of the longitudinal axis in a vertical motion about the center of gravity) and yaw (oscillation of the longitudinal axis in the horizontal plane). Each is defined as an angle of deviation, ($-180^0,180^0$), in a predefined coordinate system. These are typically used as parameters to a "pointing" function to select a particular object of interest (as opposed to inertial navigation functions). Additional sensors, generally within the device, such as a compass to provide a roll angle or tilt meters for pitch and yaw need to be incorporated in the equipment. The orientation parameters are also defined as signed 32-bit parameters representing decimal degrees with corresponding error parameters.

Table 1: Representative Functions Supported by Access Layer

Position GetPositionAbsolute ()	Get the position of the node in the absolute coordinate system (WGS84).
Position GetPositionRelative ()	Get the position of the node with respect to the coordinate system already fixed.
Position GetPositionSymbolic ()	Get the position of the node as symbolic information, e.g., *You are in room 243*.
Boolean SetPositionAccuracy (accuracy)	Set the desired accuracy for readings. If LSM cannot provide the accuracy it notifies the application. If the desired accuracy can be achieved a true value is returned.
Boolean SetRelativeCoordinates (coord)	Fix the relative coordinate system for GetPositionRelative calls.
Boolean SetSymbolicInformation (info)	Set the symbolic information mode, e.g., Room-mode, so GetPositionSymbolic can return relevant information. The parameter "info" is a GIS data type.
Boolean OrientationSupport ()	Returns true or false based on if LSM can give orientation information or not.
Orientation GetOrientation ()	Get the orientation of the device if supported.

Position Computation and Error Estimation

The position and error estimate computations are performed together using device data and history files. The error metrics are a critical aspect in the development of a useful, generic, middleware service as well as the key performance tool used in the control of the LSM. Accuracy is defined as the displacement of a point from its true position with respect to the reference model. Precision refers to the repeatability of the measurement. In the LSM the accuracy of a location estimate is characterized by both uncertainty and confidence estimates. These values will be determined by this function and made available to the application and to other control functions.

In general, the confidence in a location estimate can be enhanced through integration of multiple position measurements stored in a history file. For example, an ambiguity in a location estimate due to poor configuration of the sensors in a triangulation computation may be eliminated by the integration function. Also, constraints on maximum movement of a device (e.g., maximum velocity) for a maximum time can further improve location knowledge.

The raw position and raw error estimate values are provided by the Device Conversion routines of the LDL in standard formats. If there are multiple devices reporting positions, the fusion routine is used to compute the composite position coordinates and composite error estimate. If a cooperative scheme is employed, the cooperative control routine is invoked that will employ the chosen algorithms to compute the position.

Each of the 3-d coordinates are also accompanied by an uncertainty estimate (32-bits) that together with the coordinates defines an ellipsoid point (with altitude) and an uncertainty ellipsoid about that point [10]. The uncertainty per coordinate is defined as a distance d_i such that each measured coordinate x_i lies within distance d_i of the actual location coordinate. The confidence, p, is the total probability that each location measurement coordinate x_i lies within the error ellipsoid, i.e., within distance d_i of the actual location coordinate.

The overall measurement accuracy is defined by the deviation in each axis as given by the parameter d_i. This metric is not as general as is used for some technologies, but still recognizes that there is differing degrees of accuracy in each axis, such as seen in GPS measurement of altitude. In general, there are many possible error estimation functions and forms that can be defined and that are applicable to various technologies. For example, in cellular telephony an error area can be defined more accurately as an annulus [10]. However, the format used for the LSM is a tradeoff and has been chosen due to its simplicity and ability to encompass many of the other methods. In general, the probability density function for the confidence can be defined by the user in the LSM. The orientation parameters are also accompanied by uncertainty estimates of each value. The monitoring of the changes in errors is a task of the control routine. If the uncertainty exceed the threshold value, then the control routine is invoked to select another device.

Coordinate Transformations

Several basic transformations will be supported as well as user-supplied or LDL-supplied transformation matrices. These will be applied, on demand, to convert to/from relative and absolute or between a few standard reference systems. For example conversions between WGS84 and the Universal Transverse Mercator (UTM) are supported with standard utilities. The Universal Transverse Mercator Coordinate (UTM) system provides coordinates on a world wide flat grid and is frequently used in maps [18].

Storage and Integration Functions

The system maintains the history of positions and orientations. The storage function is controlled by application parameters through defining trigger conditions, such as minimum change in position or a timeout that will trigger a position store. If the device is portable or mobile, then the function will compute the instantaneous velocity of the object based on the rate of change between the last two recorded positions as well as a time-averaged value. A status variable is provided which can be set to indicate whether the device is fixed, portable or mobile.

Sensor Fusion

The sensor fusion function will combine the various position estimates provided by differing devices with differing characteristics. For example, if one device has accurate longitude and latitude, while another has accurate altitude, the overall measurement can be improved. In general, the algorithms for sensor fusion become quite complex and are an active area of research. Much work has been reported in the robotics domain on multiple sensor fusion, including collaborative technique [19]. We are currently developing a structure to support fusion algorithms, such as using Bayesian schemes to reduce uncertainty through multiple media measurements and environmental modeling.

LAL Control

The operation of the LAL is controlled by this routine. The control function will turn devices on and off under a variety of conditions and permit the seamless switching between location devices. This control function behavior is illustrated using an example application call as follows. The LAL control is initialized by the application. The application issues a *getPosition* call to the LAL containing the desired accuracy parameters. The LAL control routine will issue a *getDevicePosition* request to the current set of selected Device Conversion routines in the LDL. When the individual device data is returned, the *computePosition* function is invoked which will accept the device data input and the position history. If there are multiple devices, then the sensor fusion function, *fusePositions*, is invoked, otherwise, the position is computed directly. In either case, a new position and error estimate is computed and the history files are updated. If coordinate transformations are required, they are performed and the position and error are returned to the application in the proper format.

The LAL control routine will also invoke the *errorStatus* monitoring routine that can change the status of the devices depending on the errors. For example, a GPS device that mainly operates in an outdoor environment, would be switched when a user moves indoors where another medium such as an infrared system is deployed. If the errors become too great, then there are several options: 1) select a new device as the primary device through the *switch* routine, 2) change the current set of selected devices for sensor fusion (e.g., turn on additional devices), 3) increase the device sensitivity (e.g., cause greater communication among neighbors) and 4) request additional user input. Monitoring of the changes in errors can be supplemented by additional symbolic information such as from a Geographical Information System (GIS). This is not currently implemented in the LAL, but the application can explicity control changes in the selected device set through supplied routines.

The device control function supports the power management functions and the triggering functions for location determination. These include timeout, displacement or maximum error trigger events. A trigger will cause a recomputation of location that was not a direct result of an application call.

Cooperative Control

For schemes involving collection and dissemination of information from neighboring equipments or servers, a cooperative control function is provided. The details of the schemes are handled by the LDL and are described later, but higher-level coordination of the communication procedure is provided by this routine through interaction with the application and the control routine. For example, control over the extent of the cooperation is defined in this routine in response to increased accuracy requirements.

6. Location Dependent Layer

The Location Dependent Layer (LDL) provides a common interface to the LAL on one side and device specific interfaces on the other, through the Device Conversion routine that functions as a Hardware Abstraction Layer (HAL). The LAL operates on the "pull" model, but individual devices may operate on an event or a periodic basis. The LDL will either determine the location on request or simply supply its latest value to the LAL. If there is a latency concern, the LDL can be directed to configure the devices to obtain location values in a proactive and periodic fashion. Various hardware events can be used to trigger a location determination such as a timer, displacement or contact by other devices (e.g., messages from other LSMs).

There is also a direct connection or tunnel that is allowed to the devices that bypasses the LAL and the HAL in order to allow access to special features of a given device, such as waypoints of GPS. It was felt that there are many capabilities of devices that may be useful for specific application, while not of general interest in the development of location-aware services.

The physical devices are plugged into the client and accessed through standard, vendor-supplied drivers. A Device Conversion routine is included that will convert device specific data to the position-error format expected by the LAL (e.g., coordinates and error ellipsoids). The device conversion routine contains specific methods for using each interfaced device. In addition, the conversion routine will supply status information on each attached device.

To date, we have implemented the interface and drivers for a serial or PCM/CIA GPS unit, an infrared link through a USB port and a user interface mode on a laptop computer, a pen-based computer and a PDA. The GPS is typically set to provide a position update once per second and is intended for open outdoor spaces. The error estimates for the GPS are computed from the parameters supplied by the device [2]. The IR system is a proximity system and consists of IR transceivers located on computers inside rooms as well as transceivers located on the client devices. When a client enters a room and points at the computer, a link is established and the computer will supply its location and position information to the client. The accuracy of the IR is derived statically from observed system properties.

A user input mode is also provided that is used to enter a location when there is no location determination technology in place. The user can also resolve ambiguities or errors that arise such as whether the client is in a room or in the hall.

We are currently working on a variety of determination methods using 802.11b wireless LANs including an RF signal strength-based pattern matching, an RF TDOA ranging method and an RF AOA scheme. These are anticipated to form the base of multi-technology fusion and for use in cooperative schemes described next.

Cooperative Networking Schemes

Cooperative schemes are defined within the LDL layer because we consider them to function as a virtual location determination device. From the application point of view, a cooperative scheme will return coordinate-error information in the standard format within the LAL structure. However, from an implementation point of view, the algorithms can be network specific and technology dependent. In general, the cooperative schemes differ from sensor fusion in the sense that the clients are spatially distributed and employing similar location determination measurements (versus differing media measurements on the same client).

There are a variety of ways to determine or improve a position estimate through the sharing of information with other clients. One of the simplest forms of cooperative location, communication proximity, is for a node to request the positions of other entities in the current communication neighborhood, using a broadcast request message. Depending on the transmission range, a set of regions (ellipsoids) can be collected by the source and used to estimate its own location. The accuracy depends on the density of responding devices, the environmental characteristics (e.g., obstructions) and the radio channel (e.g., multipath effects and noise). For short range radios, such as Bluetooth (e.g., 10m) or the IR medium, knowledge of even a single neighbor can provide significant accuracy. Variations of this approach for ad hoc networks are described in [20,21] in which neighbors can be used to define an average location value. In general, the more devices that can respond with their position, the more accurately the source position can be determined.

Proximity schemes are often enhanced by using received signal strength (RSS) to provide additional indications of the distance from the source. These schemes rely on propagation loss estimates often derived from a careful mapping of the environment characteristics. Several schemes employing this method have been proposed [22,23]. In [22] the effects of using RSS-based triangulation or multilateration, in a multihop, peer-to-peer network is described. An algorithm for using more than the minimum number of ranges to compute a position is described and the impact on the accuracy of the number of ranges used is shown to be significant.

Several scenarios for cooperative location determination using ranging will be described in terms of the communication protocols. The availability of standard communication capabilities such as reliable sockets and reliable multicast is assumed. We also consider a set of homogeneous nodes that can perform a similar ranging measurement between other client nodes or servers within their radio coverage area. The combination of communication and ranging is synergistic as ranging algorithms typically involve some form of control, coordination and clock synchronization provided by the communication channel.

We consider a collection of nodes distributed in a 3-d spatial area and that their communication capability forms a connected network graph (e.g., each node can communicate with any other node through one or more paths). We further assume that if a node can communicate with a neighboring node, then it is possible for the pair to determine the distance (range) between them. In general, a set of four non-coplanar nodes can determine a relative 3-d coordinate system, given the ranges to each other. If another node is connected to at least 4 nodes with established positions (assuming non-coplanar), it is then possible for that node to execute a *multi-lateration algorithm* that will determine its position in the relative coordinate system. The procedure can be applied in an iterative fashion to compute the positions of other nodes in the network graph that can be 4-connected to non-coplanar nodes with known locations. In general, it is not possible to exactly determine the positions of all nodes in the network graph, as this depends on the graph topology and the initial marking (selection of initial four nodes to define the coordinate system). We note that it is possible to use estimates of the location, derived from other techniques, such as proximity measures, to continue the propagation of position information [20], however, we do not consider these technique in this scheme.

Further, if we assume that there are certain nodes that know their absolute position, it is possible to compute a transformation from the relative coordinate system to the absolute. If a set of four nodes that know both their absolute and relative positions exist in the network graph, the precise transformation matrix can be constructed and disseminated to the remaining nodes [22].

In such a network graph, one can either compute locations in a proactive fashion so that the node locations that can determined are computed in one invocation of the algorithm; or; a node that requires its location will compute it on demand in a reactive fashion, possibly limiting the scope of the computation to a local neighborhood. The primary tradeoff involves complexity and potentially greater communication overhead in the proactive case but with smaller latency when a position request is received. Also, if there is a high degree of node mobility, then the frequency in which the proactive algorithm needs to be executed may be too great to achieve desired accuracy and latency. Several scenarios for cooperative location determination are described to illustrate the protocols involved. We first describe a relatively straight-forward environment using an 802.11 wireless LAN. Then, two ad hoc cases are described.

Scenario 1: 802.11b Wireless LAN:

A basic multi-lateration procedure can be implemented in a single 802.11 wireless LAN with the inclusion of special server nodes that know their absolute location and can perform ranging measurements. A source node will use the broadcast message facility to issue a *Request-Range* message. It must then reserve a slot through negotiation with the access point. The source will transmit a measurement signal in the reserved slot. All of the special nodes that can receive the measurement signal will determine the range and respond with *Report-Range* messages that contain the range and the position of the special nodes. If the requestor collects at least three ranges, then it can compute its position using multi-lateration. As 802.11 is a single-hop network, there are no further steps necessary.

If many situations that are envisioned for the future, the client will be able to contact multiple access points. The source node must issue the *Request-Range* message in each service set of each access point and then follow the above procedure, until it has received sufficient range data for an accurate calculation.

Scenario 2: Static Ad Hoc Networks

Assume for now that the nodes are not mobile. A basic proactive algorithm for a wireless broadcast system is proposed that proceeds in two phases. In phase 1, four nodes will establish a relative coordinate system. In phase 2, the coordinate system is propagated out into the network in an iterative fashion.

We assume that each node has its unique ID. The relative coordinate system is established using a procedure outlined in [24] that requires the range values between the nodes. In order to establish a coordinate system a set of four nodes that are fully connected must be identified. (Identifying the set of initial nodes that maximizes the number of computable positions in a graph is beyond the scope of this paper.) Assume that four fully-connected nodes, $N_0 - N_3$, are chosen to form the coordinate system and that they are in the broadcast range of each other (e.g., 1-hop neighbors). N_0 will take the coordinates (0,0,0) and initiate the procedure by issuing a *Request-Coordinate-Initialize* message to N_1, N_2 and N_3. N_0 will broadcast a range signal and N_1, N_2, N_3 will determine their range to N_0. Each will reply to N_0 with a *Report-Range* message. N_1 will assume the coordinate (d_{10}, 0,0), where d_{10} is the range between N_1 and N_0. Next, the other nodes in turn will determine their remaining range values, i.e., N_1 to N_2, N_3 and then N_2 to N_3. Using the Laws of Sines and of Cosines, N_2 will assume the coordinate ($d_{20}\cos\theta_1$, $d_{20}\sin\theta_1$,0) where θ_1 is determined as the angle putting N_2 in the positive-y half-plane at the point that is the correct distance from N_0 and N_1. In a similar, fashion, N_3 is placed in the positive-z half-plane. Once the relative coordinate system is established with these nodes, the process moves to phase 2.

If the distribution of nodes can be controlled, these four initiation nodes can be preconfigured. When such a pre-configuration is not feasible, phase 1 can also be executed in a distributed fashion. In this case, any node in a fully connected set with at least 4 nodes can initiate the procedure. It assumes the role of N_0. If multiple nodes within the same fully connected set try to initiate a coordinate system, the contention may be resolved by comparing the initiator's ID (that will remain associated with the coordinate system). The node within higher ID will yield and participate in the coordinate system established by the lower ID node. Then this node selects 3 nodes from its fully connected neighbor set to become N_1-N_3. The assignment is notified to these 3 neighbors by *Request-Coordinate-Initialize* message. If a node is already playing a role in another coordinate system, it will reject the assignment. The N_0 then needs to select another candidate from its neighbor set. If N_0 can not collect enough neighbors to assist in its coordinate system, it will yield and issue *Request-Coordinate-Terminate* to dismiss those who already agreed to participate. After the initial 4 nodes are determined, the procedure described in the previous paragraph is executed to establish the relative coordinate system.

While the algorithm is initiated in a distributed fashion, it is possible that there are multiple relative coordinate systems being established in the whole network. This can be detected in phase two when a node receives the flooded *Compute-Position* message and resolved by the following steps. When a node receives a *Compute-Position* message from a node in a different coordinate system than it is currently in, and the N_0's ID is lower than the current coordinate system origin's ID, the node will erase any information about its current coordinate system and join the newly learned coordinate system. It will then act as a source node as described below in phase 2 of the protocol.

In phase 2, node N_0 will initiate the process by issuing a *Compute-Position* message to the network will be broadcast throughout the entire network using a flooding algorithm. Upon receiving a *Compute-Position*, each node will attempt to determine its position as follows:

1. If the node knows its position, then it responds with *Report-Position* message containing their relative coordinates.
2. If the node, does not know its position, it will broadcast a *Request-Position* message to its broadcast group containing a timestamp and sequence number.
3. Destination nodes that have knowledge of their position will acknowledge the request with a *Report-Position* message containing their relative coordinates. Other nodes will store the request in a pending list. Nodes will maintain a list of the relative positions of their 1-hop neighbors.
4. If the source node collects at least four 1-hop neighbors that have reported their positions, it will broadcast a *Request-Range* message to the responding group containing a timestamp for transmission and information on its local clock parameters needed to synchronize clocks.
5. Destination nodes with known position will acknowledge the request with a *Ready-to-Measure*

message containing its local clock information and prepare to make a measurement. Other nodes will ignore the message.

6. The source node will broadcast a measurement signal
7. Destination nodes will receive the measurement signal and compute propagation delay and thus range. Destination position and range-to-source will be sent to the source in a *Report-Range* message.
8. Once a node has received replies from at least 4 nodes, the relative position can be computed using a multi-lateration algorithm. After a node has determined its coordinates, it will broadcast its coordinates in a *Report-Position* message, if it has previously received any *Request-Position* messages. If the new *Report-Position* message causes any awaiting nodes who previously sent out *Request-Position* messages to collect enough responses, these nodes will then resume their location determination procedure from step 4.

In this way, the local relative coordinates will be propagated and if there is sufficient connectivity will allow the node positions to be determined.

When a node receives the *Compute-Position* message from an origin and it knows its absolute location, it will generate a *Report-Absolute-Position* message and the *Report-Absolute-Positions* will be forwarded to N_0[3]. If there are at least four nodes that have knowledge of their absolute position, then a coordinate transformation matrix can be constructed. After a period of waiting time (long enough for all nodes in the network obtaining their relative locations), this matrix is then broadcast to the network in a *Report-Coordinate-Transform* message.

If a new node joins the network it must wait for another round of position computations to be started by the distinguished node. Alternatively it may act as a source node in phase 2 and poll its neighbors to find its location. However, the node may fail if it can not collect enough report-position messages and then must wait for another round of position computation. In step 2 above, rules for deciding whether a node's position data is stale need to be incorporated. If a node has moved, then it must invalidate its position and recompute its position in a subsequent round. In general, if a relatively long period of time has elapsed between position updates, then it is reasonable to invalidate.

Scenario 3: Dynamic Ad Hoc Networks

Under conditions which cause the network to change frequently, the proactive method is not efficient and a reactive scheme may be more appropriate. In a reactive system, any node that requires its position can initiate the *Compute-Position* procedure. This would also be a similar two-phase protocol: checking for a relative coordinate system and establishing one if needed and then computing positions. In this case, however, there may be multiple relative coordinate systems established and position computations proceeding simultaneously. This requires that each coordinate system be associated with a unique ID and each compute position request be relative to this coordinate system. Each node would retain a list of relative coordinate systems to which it belongs. If a node desires its position, it first polls its neighbors to determine whether there are four belonging to the same coordinate system. If there are, then it establishes range and thus position as in phase 2 earlier. If not, then it invokes the establish coordinate procedure of phase 1 with itself as the source node and a unique ID.

The flexibility to allow any node to initiate the computation may cause considerable overhead if changes occur too frequently. In large networks, it may be desirable limit the spread of the Compute-Position message to limit the network overhead. In a fashion similar to routing schemes, a heuristic method based on a time-to-live field may be employed to provide a compromise between network overhead and reasonable position coverage. During the phase 2 procedure, if there are an insufficient number of neighboring positions reported, then

1. The source node will issue a *Compute-Position* message containing the source, a sequence number and a Time-to-Live (TTL) value. If the TTL is set to one then the *Compute-Position* request will only be executed within the broadcast group of the source node.
2. If a node receives a *Compute-Position* request that has a positive TTL and it does not know its position, then it will begin a position-determination procedure, as defined above (steps 2-8) with this node as the source.
3. If the node is able to determine its position, then it will send a *Report-Position* message to its broadcast group.
4. If the node is not able to compute its position, it will generate a *Compute-Position* message with the original source and the TTL decremented
5. After sufficient time, if the original source node can not determine its position, it can either generate another Compute-Position message with a larger TTL count or estimate its position using available information. This is controlled by the LAL.

In order to assure termination, a node must detect and ignore requests that result from its own requests. This is accomplished by including the original requestor ID with

[3] In networks that do not support routing, routes to N_0 can be constructed by each node remembering the previous hop when they receive the broadcast *compute-position* message and using that as the next hop towards N_0.

the *Compute-Position* messages. A hop count field can also be included that will serve to further restrict the flow of Compute-Position messages.

It is worth noting that this process can lead to the entire network being engaged in location determination if there are few nodes that know their locations. Also, depending on the characteristics of the radio-link topology graph and the number and location of the nodes that know their positions (either absolute or relative), the location procedure may not result in enough neighbors with knowledge of their position to compute an accurate position upon termination.

7. APPLICATIONS

The Time-User-Location Information Processing (TULIP) System represents an initial design of a location-aware service system consisting of one or more servers that provides location-context sensitive world wide web (WWW) pages to clients with standard web-browsers and some additional software. The system is based on a WWW model and all information is provided through HTML customized for each client. A client (and a server) are assumed to be running their own LSM that will provide location information.

The system operates as shown in Figure 2 where a client with a wired or wireless connection to the Internet uses its browser to log onto a TULIP server, available on the web. The login procedure will also generate a special user profile that is sent to the server. After login, the server will initiate a separate side-channel socket connection to the client and will request the client location, as determined through its LSM. Once the server has received the requested location data, it will generate web pages that contain hyperlinks based on that location. These pages are derived from a spatial database with geographic information and knowledge from the user profile. For instance, the web pages for nearby shops will be displayed as a user moves down a street. The side channel socket is maintained and will updated periodically according to the LSM settings. Each change in position will be followed by transmission of updated web pages.

A basic TULIP design is currently implemented and operates in the local area using GPS for outdoor positioning and IR and user inputs for indoor positioning. The client platforms include a Compaq iPAQ PDA running Microsoft CE and a Fujitsu CT 500 Pen Computer running Microsoft Windows 2000. The TULIP server is implemented on a PC running Linux. It uses an 802.11b wireless LAN on the clients connecting to an access point on a wired LAN. The server is connected to the LAN. A demonstration application of the basic capabilities has been implemented in a local points-of-interest guide. The system is primarily limited by the WLAN communication range.

There are many interesting issues concerning what level of information to display that require further study and experimentation. For instance, if a user is in a neighborhood, does one want to know information about the state or just the local neighborhood. This seems to be highly situation dependent, so that a web interface that allows navigation between levels is a practical approach. Other applications being considered include tourist information systems, shopping malls, theme parks and emergency response systems.

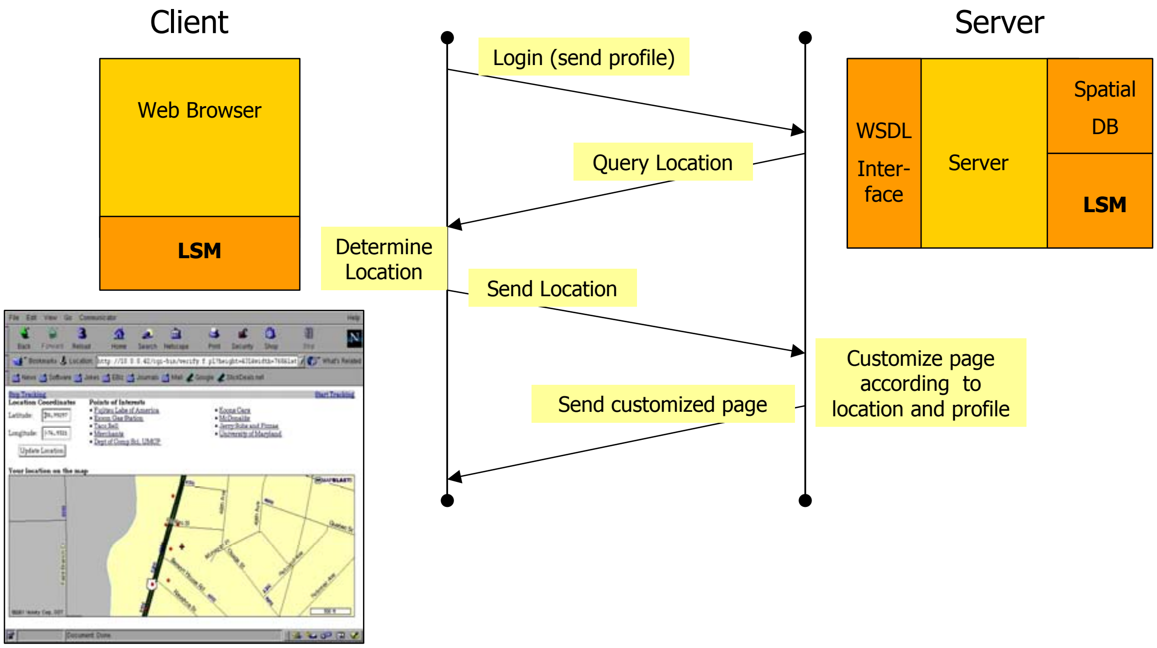


Figure 2: TULIP System Operation

8. Conclusion

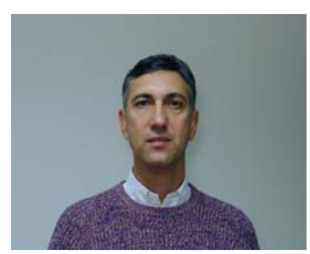
In summary, we have presented a general description for a location services middleware layer that simplifies development of location-aware services. The middleware is partitioned into two sublayers that effectively hides the details of the specific location determination technology from the application. Special emphasis is placed on distributed or cooperative position determination algorithms. A protocol appropriate for ad hoc networks of intelligent client devices with ranging capability was described. Lastly, an example of a location-aware service system, called TULIP, was described.

Our research activity in location-based services is continuing in several areas. Location determination methods that are compatible with wireless LAN devices are being investigated. Development and experimentation of the LSM and TULIP are continuing in several application domains including a museum information system and in rapidly deployable communication systems. In particular, fusion algorithms and error estimation methods are being pursued. Protocols for cooperative location determination algorithms will be evaluated and verified. The tradeoff between the proactive and reactive schemes will be explored. In the long term, we see collections of enhanced information appliances that can randomly collect and self-organize to compute their mutual locations.

References

[1] K. Pahlavan, X. Li, "An Overview of Wireless Indoor Geolocation Techniques and Systems," Proceedings of IEEE Mobile Wireless Communication Networks 2000, May 2000.
[2] B. Hofmann-Wellenhof, H. Lichtenegger, J. Collins, *Global Positioning System: Theory and Practice*, Fourth Edition, Springer-Verlag Wien, New York, NY, 1997.
[3] *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-speed Physical Layer in the 5 GHz Band*, ISO/IEC 8802-11:1999/Amd 1:2000(e), IEEE Std 802.11a-1999, IEEE Computer Society, 1999.
[4]G. Chen and D. Kotz, "A Survey of Context-Aware Mobile Computing Research," Technical Report TR2000-381, Department of Computer Science, Dartmouth College, Hanover, NH.
[5] J. Spohrer, "Information In Places," IBM Systems Journal, Vol. 38, No. 4, 1999, pp.602-628.
[6] N. Davies, K. Chevrest, K. Mitchell, A. Efrat, "Using and Determining Location in a Context-Sensitive Tour Guide," IEEE Computer, Aug. 2001, pp.35-41
[7] S. Pradhan, C. Brignone, J. Cui, A. McReynolds, Mark T. Smith, "Websigns: Hyperlinking Physical Locations to the Web," IEEE Computer, Aug. 2001, pp.42-48.
[8] B. Karp, H. Kung, "Greedy Perimeter Stateless Routing for Wireless Networks," Proceedings of the Sixth ACM/IEEE Conference on Mobile Computing and Networking (MobiCom 2000), Boston, MA, August, 2000, pp. 243-254.
[9] Enhanced 911 (E911), US Federal Communications Commission, www.fcc.gov/e911.
[10] *Digital Cellular Telecommunications System (Phase 2+); Universal Geographic Area Descriptor (GAD)* (GSM 03.32 Version 7.1.0 Release 1998, European Telecommunications Standards Institute, Sophia Antipolis, France, 2000.

[11] Location Interoperability Forum (LIF), http://www.locationforum.org/
[12] J. Hightower, G. Borriello, "Location Systems for Ubiquitous Computing," IEEE Computer, Aug. 2001, pp.57-65.
[13] E. Christensen, F. Curbera, G. Meredith, S. Weerawarana, "Web Services Description Language (WSDL) 1.0", IBM Corp., September, 2000, www2.software.ibm.com/developer/papers.nsf/dw/web-papers-bytitle?OpenDocument&Count=100.
[14] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. Nielsen, S., Thatte, D. Winer, "Simple Object Acccess Protocol (SOAP) 1.1," W3C Note, May, 2000, http://www.w3.org/TR/SOAP
[15] *Department of Defense World Geodetic System 1984 (WGS84)*, 3rd Edition, NIMA TR8350.2, January 2000, (www.wgs84.com).
[16] *NMEA 0183 Interface Standard*, www.nmea.org/0183.html.
[17] Garmin International Corp., www.garmin.com.
[18] *The Universal Grids: Universal Transverse Mercator (UTM) and Universal Polar Stereographic (UPS)*, TM 8358.2, The National Imagery and Mapping Agency (NIMA), http://www.nima.mil/GandG/pubs.html
[19] D. Fox, W. Burgard, H. Kruppa, S. Thrun, "A Probabilistic Approach to Collaborative Multi-Robot Localization," Autonomous Robots, Vol. 8, No. 3, June, 2000, pp.325-344.
[20] N. Bulusu, J. Heidemann, D. Estrin, "GPS-less Low-Cost Outdoor Localization for Very Small Devices," IEEE Personal Communications, Oct. 2000, pp.28-34.
[21] L. Doherty, K. Pister, L. El Ghaoui, "Convex Position Estimation in Wireless Sensor Networks," Proceedings of IEEE Infocom 2001, Anchorage, AL, April 2001.
[22] J. Beutel, "Geolocation in a Pico-Radio Environment," MS Thesis, ETH Zurich and University of California-Berkeley, Dec., 1999.
[23] P. Bahl, V. Padmanaban, "RADAR: An In-Building RF-Based User Location and Tracking System,'Proceedings of. IEEE Infocom 2000, Tel Aviv, Isreal, March, 2000, pp.775-784.
[24] R. Mukai, R. Hudson, G. Pottie, K. Yao, "A Protocol for Distributed Node Location," UCLA Report, Jan. 2000.



Jonathan Agre is the Director of the Fujitsu Laboratories of America in College Park, Maryland. He is currently leading a research team in the area of Pervasive Computing. His research interests include wireless protocols, location-based services and cooperative sensor networks. Dr. Agre received his Ph. D. in Computer Science from the University of Maryland in 1981.

Adedji Akenyemi is currently an intern at the Fujitsu Laboratories of America in College Park, Maryland and a graduate student in the Electrical and Computer Engineering Department of the University of Maryland. He received his BSEE and his BS in Electrical and Computer Engineering and Computer Science, respectively, from the University of Maryland in 2001. His research interests include Voice over IP and wireless protocols.

Lusheng Ji is currently a member of the research staff at the Fujitsu Laboratories of America in College Park, Maryland. He is currently working on wireless edge extension networks. He received his Ph. D. in Computer Science from the University of Maryland in 2001. His research interests include routing protocols, MANETs (Mobile Ad Hoc Networks) and wireless protocols.

Ryusuke Masuoka is a senior researcher at the Fujitsu Laboratories of America in College Park, Maryland. He is currently working on Pervasive Computing and applications to bioinformatics. He received his Ph. D. from the University of Tokyo in 2000 and his research interests include agent systems, simulated annealing and neural networks.

Pankaj Thakkar was a summer intern at the Fujitsu Laboratories of America in College Park, Maryland and is now a Ph. D. candidate in the Computer Science Department of the University of Maryland. He received his Bachelors in Technology majoring in Computer Science and Engineering in 2000 from the Indian Institute of Technology, Delhi (INDIA). His research interests include operating systems, wireless networks like 802.11 and ad hoc networks, and Pervasive Computing.