©iStockPhoto.com/William Fawcett

Mobile RTK

Using Low-Cost GPS and Internet-Enabled Wireless Phones

KIMMO ALANEN, LAURI WIROLA, JANI KÄPPI, AND JARI SYRJÄRINNE
NOKIA TECHNOLOGY PLATFORMS

An ever-increasing number of mobile handsets come equipped with GPS and some with inertial sensors. However, these single-frequency units do not exploit the higher accuracy possible with real-time kinematic (RTK) techniques. Now a group of Nokia researchers are developing a software-only RTK solution using the hardware and wireless connections already existing in mobile phones.

Government regulation such as E911 and the promise of location-based services (LBS) are the biggest drivers for integrating positioning capability into mobile phones. The increasing sophistication of applications and refinement of map databases are continually tightening the accuracy requirements for GNSS positioning. In particular, location-based games and features such as "friend finder" sometimes require better accuracy than what is achievable with state-of-the-art network-assisted GPS (A-GPS) platforms.

Cellular standards for GPS assistance data exist for both control plane and user plane protocols. These protocols carry information that help the integrated GPS receiver to improve its sensitivity, speed up signal acquisition, and especially reduce the time to first fix. However, these approved standards do not contain sufficient information for the receiver to do carrier phase positioning.

Until now, no compelling reason existed for adding carrier phase positioning related features into cellular standards so that they could employ real-time kinematic (RTK) techniques. Generally, RTK-enabled devices on the market are expensive and intended primarily for geodetic and survey applications. Also, there has been no real need in the cellular world for the accuracy RTK provides. With evolving LBS applications, however, this situation is changing.

This article describes a solution called mobile RTK (mRTK), a system specifically designed and implemented for the cellular terminal use. Its design incorporates low-cost single-frequency A-GPS receivers, Bluetooth (BT) communications, and inertial sensors. Basically, the technique involves exchanging measurements in real-time between two units — one designated as the reference and the other as the user terminal — and producing the best possible estimate of the baseline between the terminals using RTK techniques. We are developing the solution so that in the future it will be possible to add any other Global Navigation Satellite System (GNSS) measurements in addition to GPS measurements — or even instead of GPS measurements.

Using a simulator, we shall provide data that show it is possible to enable high-precision, carrier phase-based positioning in handsets with minimal additional hardware costs. Further, we shall describe some of the protocol aspects and especially the aspects of adding support for mRTK messaging to already existing cellular standards — GSM and UMTS. We believe that the mRTK solution will bring high performance to the mass market.

Moreover, additional GPS signals, such as L2C and L5, and other GNSSes such as Galileo will become operational in the near future. Consequently, it would be very beneficial to begin incorporating mRTK into the pertinent wireless standards now so that the infrastructure and the service providers will be ready when business opportunities present themselves

mRTK Solution Overview

A plethora of RTK surveying solutions is available on the market today. Generally, they are characterized by the use of both GPS frequencies, L1 and L2, enabling ambiguity resolution in seconds over baselines of up to 20 kilometers, or even 100 kilometers with more time and under good conditions. We must emphasize that this article does not claim to demonstrate similar performance and reliability as high-performance dual-frequency receivers.

We are designing the mRTK solution to work with low-cost, off-the-shelf GPS receivers with certain requirements (for example, the ability to report carrier phase measurements and data polarity). Therefore, performance degradations are expected in terms of time to ambiguity resolution, accuracy, and achievable baseline length.

Double-Difference Solutions. The mRTK solution is based on double-difference measurements to resolve double-difference integer ambiguities, similar to traditional RTK methods that are calculated either from carrier phase measurements alone or from both carrier phase and code phase measurements. The formulation of the single-frequency double-difference ambiguity resolution problem is well documented in the literature and hence, is not summarized here (See "Additional Resources" section at the end of this article for references of appropriate articles on traditional RTK techniques.)

The integer ambiguity resolution in mRTK is based on the Least-Squares Ambiguity Decorrelation Adjustment, or LAMBDA method, developed by Prof. J.G.P Teunisson and colleagues at Delft Technical University, in The Netherlands. The LAMBDA method is well established both theoretically and experimentally, which makes it suitable for the current study. Moreover, a reference implementation is easily available from the developers. (Delft University of Technology, Netherlands, http://www.lr.tudelft.nl.)

The validation of the integer ambiguities is performed by calculating the discrimination ratio, which can readily be calculated based on the results produced by the LAMBDA algorithm. The discrimination ratio is a statistical quantity that describes the relative power of the best and the second-best ambiguity candidate vectors. If the discrimination ratio exceeds a certain threshold, *K*, the best integer ambiguity candidate vector is validated and the fixed baseline solution may be calculated using the ambiguities. The threshold K is commonly set to 2.0 or above.

Assisted Bluetooth GPS (BAG) demonstration platform developed by Nokia for R&D purposes only.

The mRTK solution is also designed to use inertial sensor measurements to detect receiver movement. If both receivers are completely stationary for the entire initialization period, the ambiguity resolving algorithm can assume that the positions of the receivers have not changed between the first and the last epoch and therefore it can reduce the number of unknowns from the equations. This leads to a situation where the ambiguities can be resolved with fewer measurements and therefore also faster.

Two-Step Process. Our design target is to provide the baseline solution as quickly as possible even though some accuracy penalties will occur as a result. The developed mRTK solution works in two phases: *the initialization phase* for solving the ambiguities in real time and *the maintenance phase* for baseline estimation using the ambiguities resolved in the initialization phase.

The speed of the mRTK solution originates from a design that contains several different levels of the initialization phase. The very first baseline estimate is determined simply by calculating the position difference of the two receivers. The uncertainty of the estimated baseline cannot be any better than the uncertainty of either receiver's position solution, but the baseline estimate is made available to the user instantly.

After that, the mRTK solution calculates a baseline estimate using measure-

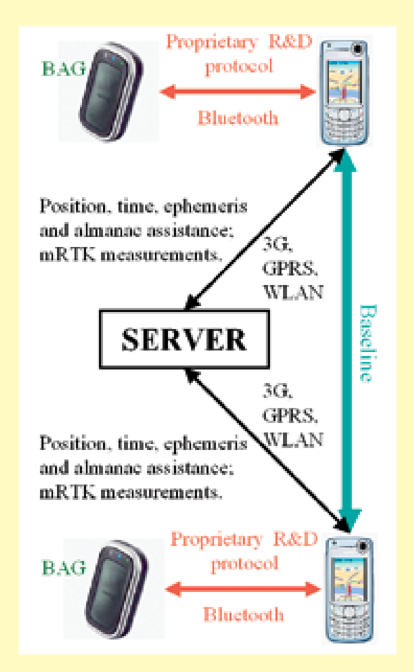

FIGURE 1 Diagram of the demonstration platform.TwoA-GPS-enabledhandsets arepositionedwithrespecttoeachother. A-GPSisconnectedtothecellularterminal viaBluetooth.Theterminalconnectstothe assistance server over any given wirelessstandard.TheserveralsorelaysmRTK measurements

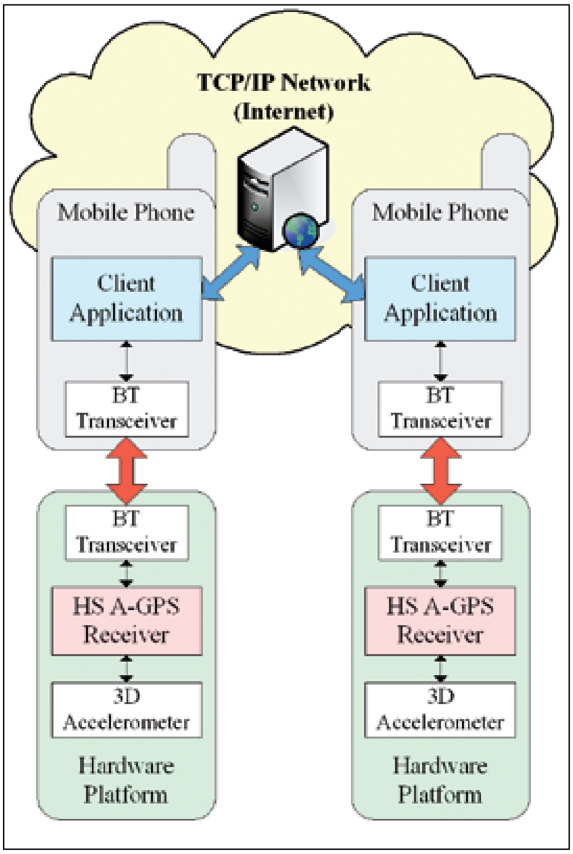

FIGURE 2 Block diagram of the mRTK testing system

ments from a single epoch, incorporating both carrier phase and code phase measurements. However, the uncertainty of this baseline estimate is still in the range of meters due to the noise in the code phase measurements. Usually after 30 seconds, the mRTK solution tries to initialize the baseline by using only carrier phase measurements from two epochs. If the carrier phase data produces as inadequate number of measurements, the mRTK algorithm will also include the code phase measurements and thus achieve a better estimate than the previous baseline.

The final level of initialization is, of course, when the baseline is calculated by using only carrier phase measurements. Once the mRTK solution is able to solve the ambiguities using only the carrier phase measurements and validate the ambiguities using the discrimination ratio, the ambiguities are said to be initialized. The mRTK solution then moves to the maintaining phase and starts to update the baseline using the ambiguities resolved in the initialization phase.

Testing the System

The mRTK performance testing was accomplished using two identical hardware platforms containing 12-channel off-the-shelf high-sensitivity OEM GPS receiver modules and a 3-axis accelerometer. We constructed this test system to determine the physical limitations and requirements for the protocol and messaging aspects.

The hardware platforms have integrated Bluetooth (BT) transceivers and were connected to the mobile terminals via BT connection. The actual position calculation and the mRTK calculation (mRTK application) are performed inside the mobile terminal. The mRTK application is designed to be run in any Symbian Series 60 terminal. Figure 1 shows a diagram of the demonstration platform.

The mRTK application aids the GPS receiver with expected signal and time information, and reference frequency information. Therefore, the hardware platform is considered to be a BT A-GPS receiver and hence the name "BAG". The photograph on page 33 shows this hardware platform (BAG).

The BAG and the mRTK application inside the mobile terminal communicate with each other by means of a proprietary low-level GNSS control interface protocol, which is shown in Figure 2 with red arrows. This protocol contains the previously mentioned aiding information for the GPS receiver and the GPS measurements for the mRTK application. These measurements contain carrier phase measurements, code phase measurements, encoded GPS data bits, and data bit polarity information. The control protocol, of course, also contains the means to control the receiver.

The mRTK applications communicate with each other via a server in the transmission control protocol/Internet protocol (TCP/IP) network (Internet). The mobile terminal has a general packet radio service (GPRS) connection that enables TCP/IP communication. The server, shown in Figure 1, is needed because mobile terminal and network implementations currently do not allow direct connections between two terminals. It also provides, for instance, ephemeris and almanac assistance to the position calculation software running in the terminal.

Communication between the mRTK application and the server is accomplished with a protocol that was specifically designed for mRTK use. This communication is shown with blue arrows in Figure 2. The server also acts as a source for the GPS assistance information providing ephemeris, almanac, reference time, and ionospheric corrections.

Performance

We conducted several experiments using the testing system and a GPS simulator. The simulator was configured to output data from the same eight satellites for both receivers with using several different baseline lengths varying from 0 meters to approximately 5 kilometers , and using scenarios for different GPS weeks.

We chose to characterize the system performance without modeling ionosphere, ephemeris, or satellite clock disturbances. The goal of the simulator tests was to provide information on the best obtainable performance (i.e., under ideal conditions). The future field tests will reveal the real-world performance.

From the perspective of algorithm development, determining the effect of using sensors and the stationary information bit was one of the goals. We tested this effect by making several measurements with different baseline lengths and calculating two mRTK solutions from the same measurements; one solution in which the sensor information is available and one where it is not. Figure 3 shows clearly the benefit of the stationary information when the baseline length is longer than one kilometer.

The accuracy of the mRTK solution was evaluated using the same measurement set and by calculating the three-dimensional error vector norm ("raw error") from the mRTK solution compared against the true positions configured into the simulator. Figure 4 shows the amount of errors as a function of time spent in processing several different baseline lengths.

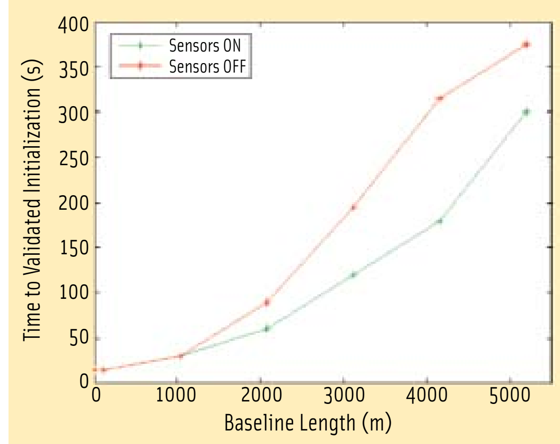

FIGURE 3 Time required for validating the mRTK initialization as a function of baseline length

More detailed results from this mRTK experiment, especially from the algorithm point-of-view, can be found in the paper by Wirola et al listed in the Additional Resources. However, some conclusions can already be drawn from these results. First, the stationary information provides a clear benefit and therefore its inclusion in the protocol is justified. Second, due to the use of single frequency receivers the performance will deteriorate quite rapidly with increasing baseline length. Therefore, we need to exploit all means to keep the baseline length as short as possible.

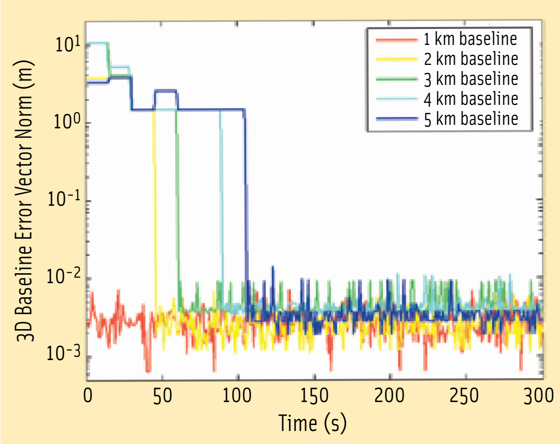

FIGURE 4 Performance of the mRTK solution.

Sizebits (Type)	Unit	Description	Explanation
Time and Position information (once per message)			
32	s	UTC time	UTC time in seconds.
32 (Q8)	m	Position X	Receiver position in the ECEF system
32 (Q8)	m	Position Y	
32 (Q8)	m	Position Z	
1	-	Stationary	Stationary indicator.
31 (Q8)	m	Position uncertainty	Position uncertainty (CEP50)
Measurement information (once per signal)			
16	-	SS ID	Signal ID and Space Vehicle ID. (Table II)
8	-	Polarity	Carrier phase polarity flags: unknown and inverted.
8	-	Cycle Slip indicator	Cumulative loss of continuity indicator.
32 (Q25)	ms	Code phase	Code phase measurement.
32 (Q32)	ms	Code phase STD	Code phase standard deviation.
32 (Q10)	m	Carrier Phase	Accumulated carrier phase measurement.
16 (Q16)	m	Carrier Phase STD	Accumulated carrier phase standard deviation.
32 (Q10)	m/s	Doppler	Doppler frequency for carrier phase extrapolation / interpolation.

TABLE 1. Measurement Message Content

Testing Protocol

The testing protocol used in the mRTK solution was designed specifically for use in research and development and as a reference design for proposed changes to the pertinent cellular standards. The protocol was designed to be as efficient as possible and especially to take advantage of the properties of TCP/IP. As TCP/IP already guarantees that transmitted data are error-free and also preserves the order of the data, our protocol did not need to include extensive error corrections and packet order counts.

In developing our protocol, we considered the traditional RTK protocol of the Radio Technical Commission for Maritime Services (RTCM); however, it appeared too inefficient for a mobile terminal environment using TCP/IP. Even though an RTCM protocol specification exists for transmitting RTCM data over TCP/IP (the Networked Transport of RTCM via Internet Protocol), it did not contain all the parameters that we believe are needed to be transmitted between the terminals. It also appeared complicated to add new forthcoming GNSS systems into RTCM format.

We also considered using the Receiver Independent Exchange Format (RINEX) protocol, but RINEX is text-based and, therefore, requires a lot of processing and would create a lot of overhead to the wireless connection. Because the main goal — and largest challenge — of this experiment was to demonstrate that the necessary information can be also included in cellular standards, we decided it would be beneficial to develop a new protocol from scratch.

From the processing point-of-view, the protocol needed to be as efficient as possible, because the mobile terminal environment has no extra processing resources to waste. Therefore, our design is binary and field-aligned in order to use the data directly in any processor environment. In other words, we determined that all message fields larger than eight bits start from the even offset. This way the field in the message can be directly read with any processor. For instance, an ARM processor is unable to access 32-bit fields from an address that is not dividable by 4. The field-aligned property is something that cannot be accomplished in cellular standards, but that was not seen as a major issue.

The protocol needed to enable the use of multiple simultaneous mRTK sessions. It also needed to contain simple authentication because the server is located in an open Internet environment.

After authentication, the user terminal requests a binding ID from the server. The binding ID given by the server is then sent to the other mRTK terminal via short message service (SMS). Both terminals then bind to the server with the same ID. The binding ID itself is just one way of linking the two terminals together. Of course, a lot of other methods can do the same job.

After the binding is complete, the reference terminal starts to send mRTK measurements back to the user terminal. Table 1 presents the measurement message used. In the testing, measurements were sent at the rate of one message per second. However, the rate of measurements does not have to be fixed. It can vary either way. The mRTK application on the user side then incorporates the received measurements and its own measurements to begin initializing the baseline. After initialization, the mRTK updates the baseline at the rate at which measurements arrive from the reference terminal.

The mRTK measurement message is comprised of two blocks: a *time and position information block* that is present only once per message and *measurement information blocks* that are included once for every measured signal. The time is given as Universal Time Coordinated (UTC) and is therefore independent of any particular satellite system time. The time contains only the integer part of the seconds; so, the measurements must be either extrapolated to or actually measured at an even second. (It doesn't have to be this way; the resolution may be chosen quite freely. However, this choice was made for initial testing.) The position is given with 1/256 meter resolution in order to reduce quantization errors in accurate absolute positioning.

The measurement information block always starts with a signal and space vehicle (SV) identification (ID) field that is basically 12 bits long, even though the testing protocol uses 16 bits due to alignment. The field is a bit mask of two components: the signal ID and SV ID. The different signal ID values are listed in Table 2 and include the satellite system, which is automatically determined from the signal ID. The SV ID portion is, for instance, in the GPS system case the pseudorandom noise (PRN) number.

The pseudorange measurement in the measurement information block is given with seven bits for the integer part in order to avoid ambiguities in the signal's time of flight. The number of bits reserved for the carrier phase measurement field ensures that the field would not roll over more than once between the first and the last epoch in the mRTK initialization. The Doppler field would, however, also work with as few as 22 bits, but, due to alignment, 32 bits were used.

Cellular Protocol Aspects

During the testing protocol design and implementation, several issues emerged concerning the addition of the mRTK feature into cellular protocols. This section lists the considerations and conclusions from those findings.

User-to-user relative positioning is not recommended for control plane systems because it would require a lot of protocol and implementation work to get the binding of two terminals and relaying measurements between two terminals to actually work.

Control Plane. The control plane level would still benefit from the mRTK feature as an accurate absolute positioning method. If the binding between the mobile units is not implemented but the serving base station (BS) is surveyed and paired as a stationary reference to them, the terminal can calculate the baseline to the serving BS (Mobile Station Based (MS-Based) method) and, of course, the reverse: for the network to calculate the baseline from the BS to the terminal (MS-Assisted method).

As baseline length has a huge effect on the performance of the mRTK (as seen, for example, in Figure 4), the length shouldn't exceed two to three kilometers. The cell size, for instance, in the global system for mobile communications (GSM) can be as long as 35 kilometers. Therefore, it doesn't make any sense to survey all the BS locations, because the baseline lengths inside the cells could exceed the performance limits.

However, the use of a virtual reference station (VRS) service, for instance, could solve this issue. The VRS system can be used to calculate a virtual RTK reference station anywhere within the VRS service area, and that "anywhere" can always be close to the user terminal. In this way, the baseline length never becomes too long, and availability of the VRS service eliminates the need to survey all the BS locations.

Current versions of the control plane protocols lack the capability of transmitting the required carrier phase measurements. Also, the accurate BS position cannot be transmitted to the terminal with the current standard. However, activity in the 3GPP standardization process is now under way to include assisted-GNSS data formats in GSM standards. Therefore, the opportunity now exists to include new features such as mRTK with rather minimal effort for improved positioning performance compared to the existing A-GPS implementation.

Control Plane Broadcasts. GSM standards make it possible to deliver GPS assistance from the network to the terminal via broadcast channels. These same channels could be used to serve mRTK reference measurements for accurate absolute positioning of multiple user terminals.

The rate of data that can be transmitted via GSM broadcast channels, however, is rather restricted. So, even though it would be possible to use broadcast channels to simultaneously serve multiple terminals, the cell size in GSM is so large that the single BS would have to broadcast more than one item of reference information in order to get the required performance. As Figure 4 showed, the performance degrades rather rapidly if the baseline length gets long. A GSM cell radius can be as long as 35 kilometers; therefore, the BS would have to broadcast more than 100 reference measurements in order to keep the baseline length less than 3 kilometer. Therefore, in the GSM case, the broadcast solution is most likely not feasible.

For other cellular standards, for instance the Universal Mobile Telecommunications System (UMTS), the cell sizes are relatively small. The radius of one cell is usually less than three kilometers, even though the maximum in UMTS is 6 kilometers. The bandwidth of the broadcast channels is also much higher than in GSM. Therefore, the use of a broadcast solution in the UMTS case shows high potential.

Signal ID	Value	System
Any	0	-
GPS L1 C/A	1	GPS
GPS L2C (data)	2	GPS
GPS L2C (pilot)	3	GPS
GPS L5 (data)	4	GPS
GPS L5 (pilot)	5	GPS
Reservedforfutureuse	6-7	-
GALILEO L1-B (data)	8	Galileo
GALILEO L1-C (pilot)	9	Galileo
GALILEO E5A (data)	10	Galileo
GALILEO E5A (pilot)	11	Galileo
GALILEO E5B (data)	12	Galileo
GALILEO E5B (pilot)	13	Galileo
Reservedforfutureuse	14-15	-
GLONASS L1	16	GLONASS
GLONASS L2C	17	GLONASS
Reservedforfutureuse	18-19	-
QZSS L1 C/A	20	QZSS
Reservedforfutureuse	21-24	-
SBAS L1 C/A	25	SBAS
LAAS L1 C/A	26	LAAS
Reservedforfutureuse	27-31	-

TABLE 2. Signal ID

User Plane Aspects. The testing protocol used during this experiment was already implemented on the user plane level and therefore serves very well, almost as is, in the secure user plane (SUPL) protocol. The same features that were available in the testing protocol can also be included directly in the SUPL protocol, with some modifications. For example, in using VRS services in the user plane, the rough position of the user terminal must be somehow transmitted from the terminal to the VRS service provider at the beginning of the session. Still, implementing this is quite trivial.

Other Aspects. When specifying the mRTK protocol for any carrier specification, several things must be considered. Firstly, the bandwidth requirement was calculated to be roughly 2.3 kbs for 12 signals. In the future, however, the number of available signals will most likely triple due to the forthcoming GNSS satellite systems and modernization of GPS. However, the bandwidth calculation assumes a message rate of 1 Hz and, as was already mentioned, the rate can be less. When compared to the RTCM protocol, which requires (with 12 signals) 1.6 kbs, the bandwidth requirement isn't significantly bigger.

The second aspect that we should consider is the real-time requirement of the mRTK. Actually, there aren't any strict real-time requirements. The user application just has to buffer its own measurements for the delay that is caused by the carrier of the reference measurements and that delay can be several seconds. Even tens of seconds shouldn't cause any major problems technically. The only considerable effect is on the user who directly experiences the delay.

Thirdly, the testing protocol assumed that the carrier (in the testing case, TCP/IP) guaranteed that the absence of transmission errors and stability of the message order. Of course, these assumptions do not apply in all possible wireless carriers. Therefore, when specifying this protocol in such a carrier that does not guarantee these assumptions, they must be addressed.

Finally, the biggest issue is the class marking of user equipment and network capabilities. In the future, more satellite based navigation systems and civil GNSS signals will become available. Most likely some terminals will not contain the ability to measure all possible signals. Therefore, regardless of the carrier, the class marking of the terminal's measuring capability must be solved somehow. This also applies to class marking for VRS service capability and the signals in that service.

Future Work

This article has introduced a new concept called mobile Real-Time Kinematics and shows that RTK-like features are possible using low-cost components and existing cellular communication carriers. Even though a lot of development work remains on the mRTK algorithm side, the biggest challenge still involves cellular carriers and their standardization. Of course, even after standardization, the development of the infrastructure would require a huge effort.

Future work with the existing testing protocol includes more testing, especially field testing, and testing with different signal conditions and satellite constellations. The testing protocol itself should be modified with new features such as the VRS service. Using VRS, the baseline can always be kept very short, and accurate absolute positioning is available everywhere using mRTK.

One of the ideas that also need to be further developed is peer-to-peer protocols. In those protocols the mRTK measurements would be transmitted directly from one terminal to another without the use of a server in between. As an example, this kind of protocol could be embedded into voice-over-IP (VoIP), in which the data channel for the voice encoding is already open and could easily accommodate other data transmissions that do not have strict real-time requirements, such as mRTK. Other peer-to-peer protocol means would exist, for instance, in WLAN, where the terminals are connected to the same subnet and would be able to open direct connections to each other.

The solution we have presented holds a lot of potential. Especially with the forthcoming satellite systems (e.g., Galileo and modernized GPS), the solution will significantly improve the accuracy of positioning in the mobile terminal. Nonetheless, the standardization of the mRTK features will require a lot of joint effort among terminal and network manufacturers and cellular operators.

Acknowledgments

This article is based in part on two papers, "Bringing RTK to Cellular Terminals Using a Low-Cost Single-Frequency AGPS Receiver and Inertial Sensors," by L. Wirola, K. Alanen, J. Käppi, and J. Syrjärinne, and "Inertial Sensor Enhanced Mobile RTK Solution Using Low-Cost Assisted GPS Receivers and Internet-Enabled Cellular Phones," by K. Alanen, L. Wirola, J. Käppi, J. Syrjärinne, presented at the IEEE/ION PLANS 2006 conference, © 2006 IEEE.

Manufacturers

The mRTK prototype platform uses the iTrax03/16 GPS OEM receiver manufactured by **Fastrax Ltd.,** Vantaa, Finland. The accelerometer is an LIS3L02DQ from **STMicroelectronics**, Geneva, Switzerland. A GSS7700 GPS/SBAS simulator from **Spirent Communications,** Paignton, Devon, United Kingdom.

Additional Resources

Protocols & Standards

[1] 3GPP TS 04.31 and 44.031, Location Services (LCS); Mobile Station (MS) - Serving Mobile Location Centre (SMLC) Radio Resource LCS Protocol (RRLP), http://www.3gpp.org/

[2] 3GPP TS 04.35 and 44.035, Location Services (LCS); Broadcast network assistance for Enhanced Observed Time Difference (E-OTD) and Global Positioning System (GPS) positioning methods, http://www. 3gpp.org/

[3] 3GPP TS 25.331, Radio Resource Control (RRC) protocol specification, http://www.3gpp.org/

[4] 3GPP2 TSG-C C.S0022-0, Location Services (Position Determination Service), http://www.3gpp2.org/Public_html/specs/tsgc.cfm

[5] 3GPP2 TSG-XX.P0024-0V0.9, IP-Based Location Services

[6] OMA-TS-ULP-V1_0-20050719-C, User Plane Location Protocol, http://www.openmobilealliance.org/release_program/supl_v1_0.html

[7] NTRIP, Networked Transport of RTCM via Internet Protocol, http://www.rtcm.org/

[8] RINEX, the Receiver Independent Exchange Format Version 2.10, Werner Gurtner, Astronomical Institute, University of Berne, http://www.ngs.noaa. gov/CORS/Rinex2.html

[9] RTCM-104, The Radio Technical Commission for Maritime Services, http://www.rtcm.org/Publications

[10] de Jonge, P., and C. Tiberius, The LAMBDA method for integer ambiguity resolution: implementation aspects, Publications of the Delft Geodetic Computing Centernumber 12, Delft Universiteitsdrukkerij TU Delft, 1996.

[11] Kim, D., and R.B. Langley, "GPS Ambiguity Resolution and Validation: Methodologies, Trends and Issues", 7th GNSS Workshop – International Symposium on GPS/GNSS, Seoul, Korea, November 30 – December 2, 2000.

[12] Leick, A., GPS Satellite Surveying, 3rd edition, Hoboken, NJ: John Wiley & Sons, 2004.

[13] Richert, T., and N. El-Sheimy, "Ionospheric Modeling: The Key to GNSS Ambiguity Resolution", GPS World, June 2005, pages 35-40.

[14] Teunissen, P.J.G., "A new method for Fast Carrier Phase Ambiguity Estimation", Proceedings IEEE Position, Location and Navigation Symposium PLANS'94, Las Vegas, NV, April 11-15 1994, pages 562-573.

[15] Teunissen, P.J.G., and P.J. de Jonge and C.C.J.M Tiberius, "The least-squares ambiguity decorrelation adjustment: its performance on short GPS baselines and short observation spans", Journal of Geodesy, number 71, pages 589-602.

[16] Teunissen, P., and P. de Jonge, C. Tiberius, "On the Spectrum of GPS DD-Ambiguities", Proceedings of ION GPS 1994, Salt Lake City, Utah, September 20-23 1994, pages 115-124.

[17] Tiberius, C.C.J.M., and P.J. de Jonge, "Fast Positioning Using the LAMBDA-method", Proceedings of the 4th International Symposium on Differential Satellite Navigation Systems DSNS'95, Bergen, Norway, April 24-28 1995.

[18] Wirola, L., and K. Alanen, J. Käppi, and J. Syrjärinne, Bringing RTK to Cellular Terminals Using a Low-Cost Single-Frequency AGPS Receiver and Inertial Sensors, IEEE/ION PLANS 2006 Conference, 24-27 April 2006, San Diego, CA

Authors

Kimmo Alanen received his M.Sc. degree from the Tampere University of Technology, Finland, with a major in software engineering. He joined the Nokia Corporation in 1997 and has been working with positioning research for the last eight years. He is currently undertaking postgraduate studies in research on GNSS assistance protocol enhancements.

Lauri Wirola received his M.Sc. degree from the Tampere University of Technology, with a major in electrophysics. Shortly after completing his thesis, he started working in the area of positioning at Nokia Technology Platforms in 2005. His present research interests include RTK and A-GNSS standardization.

Jani Käppi received his M.Sc. degree from the Tampere University of Technology. He worked at the Institute of Computer and Digital Systems at the Tampere University of Technology doing research in the area of personal positioning before joining Nokia Corporation in 2002. He is currently researching the development of sensor-enhanced positioning systems.

Jari Syrjärinne received his M.Sc. degree and his doctoral degree, both from Tampere University of Technology, majoring in digital signal processing and applied mathematics. Between 1996 and 1998 he worked for Tampere University of Technology Signal Processing Laboratory in the areas of data fusion and target tracking, and since 1999 he has been with the Nokia Corporation. He is currently involved in work on A-GNSS and algorithms for hybrid positioning. IG