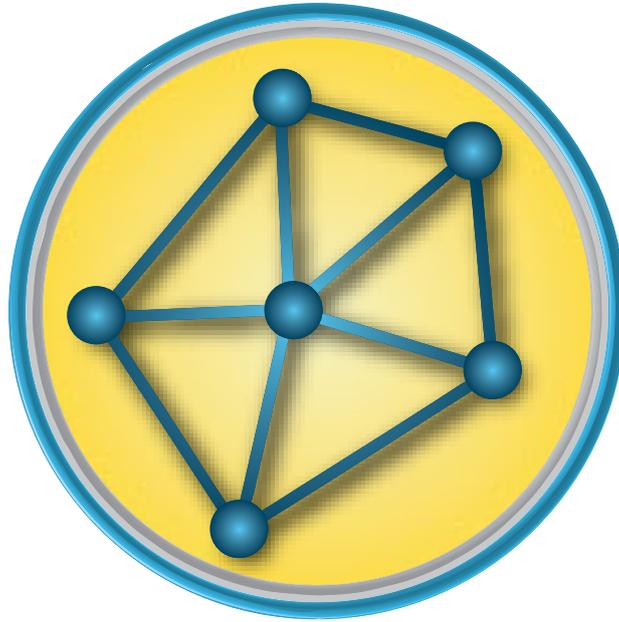


Approaches, Implementations, Brands & Choices

Network corrected real-time is a technological approach to high precision GPS/GNSS positioning that has been theorized about, studied, experimented with, and implemented in various academic, scientific, and commercial forms for nearly a decade. Many of the various approaches share the fruits of past research, algorithms, and technological tools; some which are in common.

In general, all of these various approaches to the same problem seek to simply expand the ability to correct for sources of error common to GPS/GNSS positioning over wider areas, and to deliver a consistency of correction quality through the utilization of networks of well designed, operated, monitored, and maintained reference stations. One outmoded approach is to simply tap each station of a network in traditional single-base mode, but this is impractical on many levels, and would require the stations to be very close together. Network RTK, in its varied guises and approaches, can dramatically extend those baselines.

There is, of course, going to be more than one way to “skin that critter”. Science, academia, publicly funded research, and private industry have delivered handily with a number of solid and proven approaches. The varied approaches existed in academic papers and the journals of science long before someone “productized” them. An approach is not a brand, although some of the specific implementations and the varied nuances of each have sometimes become synonymous with certain brands. It has been hard to wade through the various literature concerning each approach, or to query the respective users of each without encountering some thinly disguised subjective written materials,



RTN-101:

Technological Approaches to Network-based Corrections (Part 7)

or some highly charged rhetoric of loyalty to one or more of the specific implementations of the various approaches.

In this article we will look as generically as possible at four of the most common approaches, and firstly, to contrast these with conventional RTK without looking directly at any specific implementation of any of them. A common question is “How is this approach supposed to work in general terms?” All too often the answers get lost in a torrent of brand-specific nuances and approach variations.

The good news for the user is that, for the most part, the manufacturers of rovers have seen fit to support as many approaches as is practical, and the providers of network software suites often support more than one approach. The network software suite of the network I administer provides implementations of three of the four following network approaches, as well as single-base RTK. This is true of the software suites of a number of the manufacturers. Choice is good.

>> By Gavin Schrock, LS

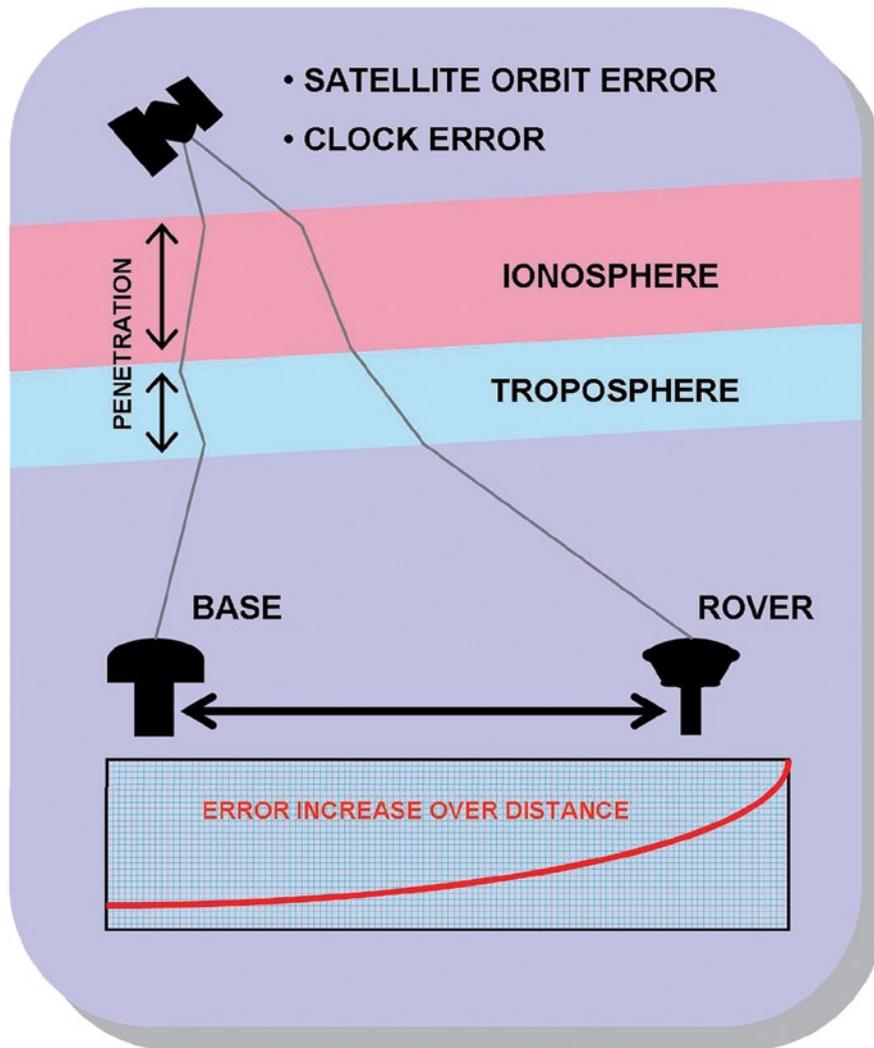


Figure 1 Pseudo-range errors in RTK

Single-base Real-time Kinematic

You may have experienced amazing results with corrections from a single reference station. This is all well and good over a short distance (typically under 10km), but “network RTK” seeks to (and succeeds in) extending this “tether” dramatically (**Figure 1**).

Single-base RTK (otherwise just known to many as “RTK”) uses the modeling capability of just the two receivers (the rover and base) to come up with a single “lump sum” correction that can be applied to the baseline between the rover and base. Errors due to satellite orbit, clock, delay in ionosphere and troposphere are all simply subtracted as a whole from the baseline; fine if the conditions are consistent for all sources of error between your rover and the one base. In some ways this could be described as a one-dimensional view of corrections.

What if you could maintain a consistency of correction value over a wider

area? If your rover was sitting directly between two reference stations, would it be as easy as averaging the correction values from both? No, it is a bit more complicated than that. For instance,

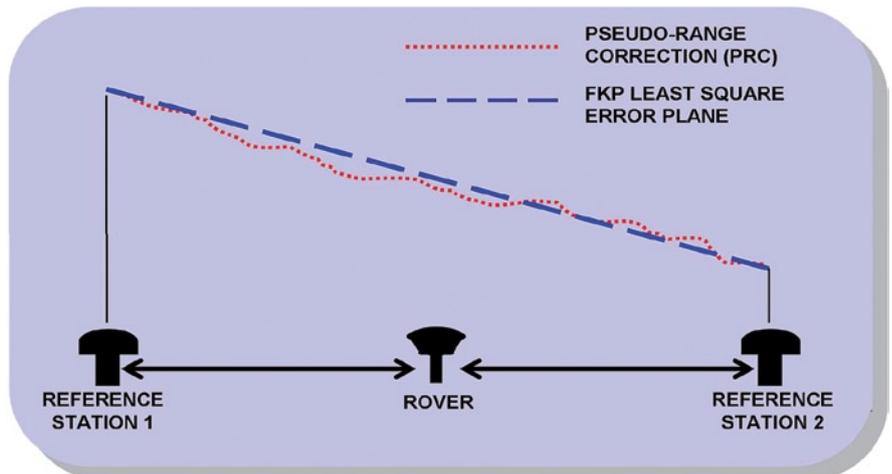


Figure 2 Generation of FKP correction plane

the values for modeling with respects to atmospheric delays are based on the points where the signals penetrate these layers, and in addition these layers are not of uniform thickness or consistency. But if you had data from three or more stations, you could “interpolate” the value for any region inclusive... add that extra “dimension” so to speak.

The single-base RTK approach has been implemented by nearly every manufacturer. It continues to be made available in the respective network software suites as a user option; a supplement to the more sophisticated implementations of network corrected RTK (though I would like to add that most users stop using single-base once they realize the benefits of these newer approaches).

Although the preceding is a gross oversimplification of the contrast between single-base and network RTK, the fundamental goal of seeking to fix such ambiguities and mitigate for the respective sources of error as true to the local vicinity of your rover as possible is common to all of the approaches we will examine.

Flat Plane Correction Parameter – FKP

Area correction parameters in the form of a kind of flat plane projection around a reference station or stations can be generated via the Flächen-Korrektur-Parameter (FKP) approach to network RTK, or as some call it, the “tilt correction”.

The range of values for pseudo-range correction for three or more stations forms a series of planes. The network server derives linear approximations from the complex state vector for the ionospheric and geometrics effects in N-S-E-W directions. The model is, in effect,

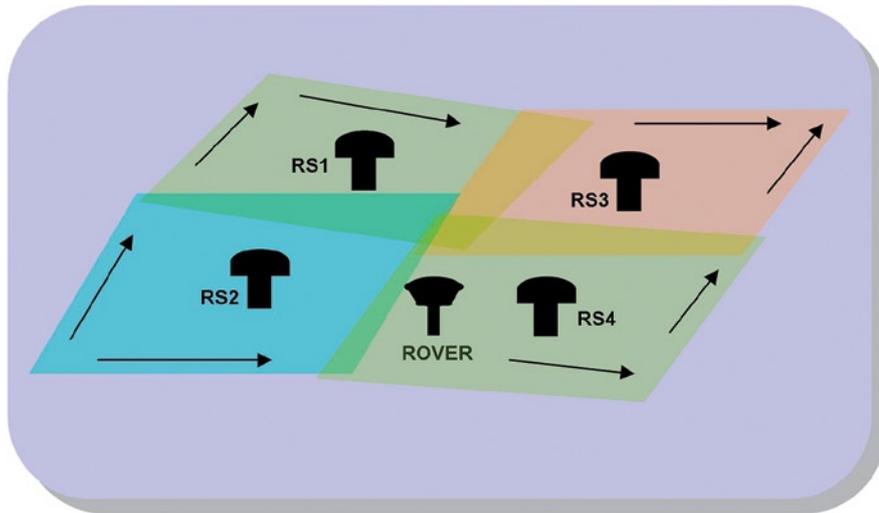


Figure 3 Rover position with respect to FKP planes

a simplified linear model, and therefore is subject to a limited range of effectiveness around each of the reference stations. The aspect and value range is expressed as a mathematical description of the “tilt” of these planes. This “tilt” is transmitted to the rovers. The rover can then apply their position relative to the respective planes to interpolate a correction value for their current location, and as the values are updated constantly and continually broadcast, the rover receives an updated and localized correction (Figures 2 & 3).

By nature, the FKP style of correction can be delivered in a broadcast-only mode, with the rover using its own location to interpolate from the “tilted plane” definitions it receives. Indeed there have been implementations where these corrections have simply been broadcast out over a broad area via radios.

Nearly all rover manufacturers have included FKP as a user option along with other styles and formats. There are implementations of this approach included in the respective software suites of Trimble, Geo++, and others.

Non-physical Reference (a.k.a. VRS)

The thrust of this approach is to extend the baseline lengths to physical reference stations by bringing the value and origin of the corrections as close to the rover as possible. For purposes of correction modeling, it is as if there was a reference station in the immediate vicinity of the rover, and hence the term “virtual reference station,” which is a term many may be familiar with as a popular trademarked implementation of this approach.

- The network data is used to compute models of ionospheric, tropospheric, and orbital errors (live high-precision orbit data may be introduced, like Ultra-Rapid Orbit).
- The rover sends its generalized location to the network server to “request” corrections for its vicinity (Figure 4).
- The actual errors on the baselines are derived in centimeter level accuracy using fixed carrier-phase observations.
- Linear or more sophisticated error models are used to predict the errors at the rover location.
- A non-physical (virtual) reference location is derived as an origin for the high confidence modeled values at or

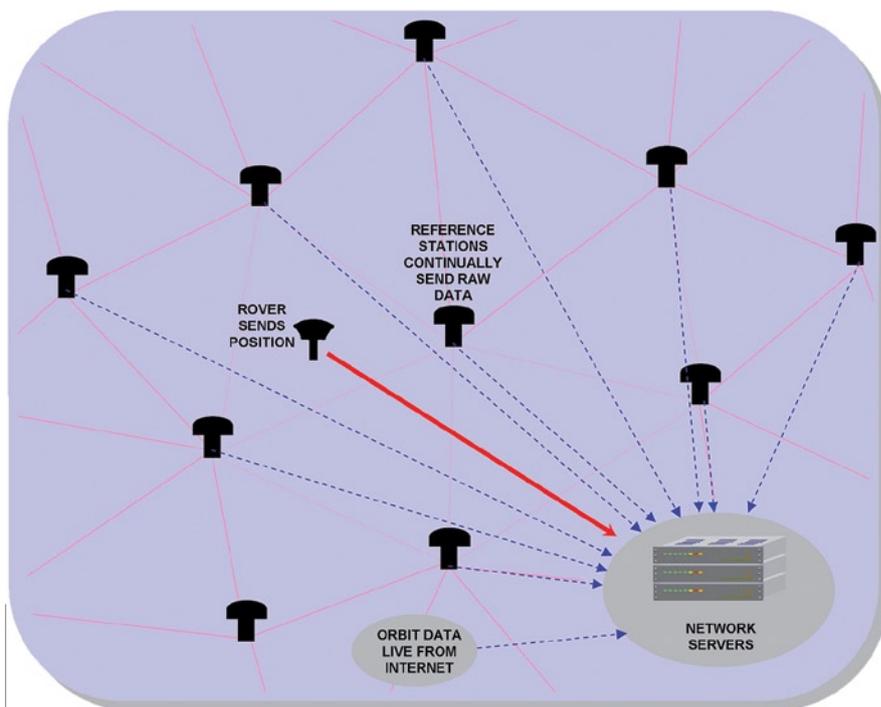


Figure 4 Reference stations continually send raw data, and the rover sends its location to request localized corrections.

One of the most misunderstood aspects of this approach is the notion of a “non-physical” reference station. The solution still provides a hard vector to a physical reference station; it is only the origin for the corrections that is based on the “virtual” location. In a manner of speaking, the rover requests a set of rigorously computed and customized corrections for its location, rather than a simple interpolation from a wider area model.

Here is how it works in the most common implementation of this approach:

- A raw data stream is sent from each reference station (low latency) to the network server Central Processing Center (CPC). All (or clusters of) reference stations may be used.

near the rovers vicinity.

- The observed positions (or vectors) are referenced to a physical station while utilizing the improved corrections from the non-physical station.
- The models are updated constantly, and as conditions may change, or if the rover moves far enough from the non-physical location, and new non-physical location can be initiated automatically and without delay to the rover (Figure 5).
- The rover receives only standard formats (RTCM).

This implementation of the non-physical reference approach requires bi-directional communications (*i.e.*, the rover must notify the network servers of its location to initiate the request). With the minimal bandwidth needed for server-side processing, and considering that most networks have been designed for authenticated use and the wide availability of bi-directional communications, this requirement has not proven to be an obstacle. This approach is also capable of producing “zone corrections” for specific pre-defined geographic areas, and in such implementations there are broadcast-only options (*e.g.*, corrections zone pre-defined for a specific job-site with corrections relayed from the internet to a base-style radio).

The non-physical approach is the most widely implemented and utilized network correction approach at this time. As it is server-side processing, and the correction product is simply standard, non-proprietary formats (*e.g.*, RTCM2.3, 3.0, CMR, etc.), it has been widely operable for nearly every rover capable of accepting real-time corrections.

There are implementations of this approach in the respective network software suites of Trimble, Topcon, and others.

Network Message (RTCM 3.1-Network)

This is an approach which utilizes client-side processing. Indeed the notion of a “network message” was researched and theorized for many years before the fairly recent implementations. The scenario posed is “What if the rover could receive the observation data from a group of stations, then process and analyze the data onboard to mitigate for sources of error?” This could be implemented as a “broadcast-only” approach, though the most common implementations have found advantage in refinements possible through bi-directional communications.

Certainly as the processing power of receiver/collector gear has improved, the practicality of such an ambitious approach has come to fruition. There have been further steps taken to mitigate for the possible hazards of trying to send all of the data from each station: the most common implementation of this approach is the “master-auxiliary” method for reducing the transmissions to a manageable level. Full data is sent from a “master” station; from the other “auxiliary” stations in a “cluster” or “cell”, only offsets in data relative to the master need be sent (reducing the total data sent).

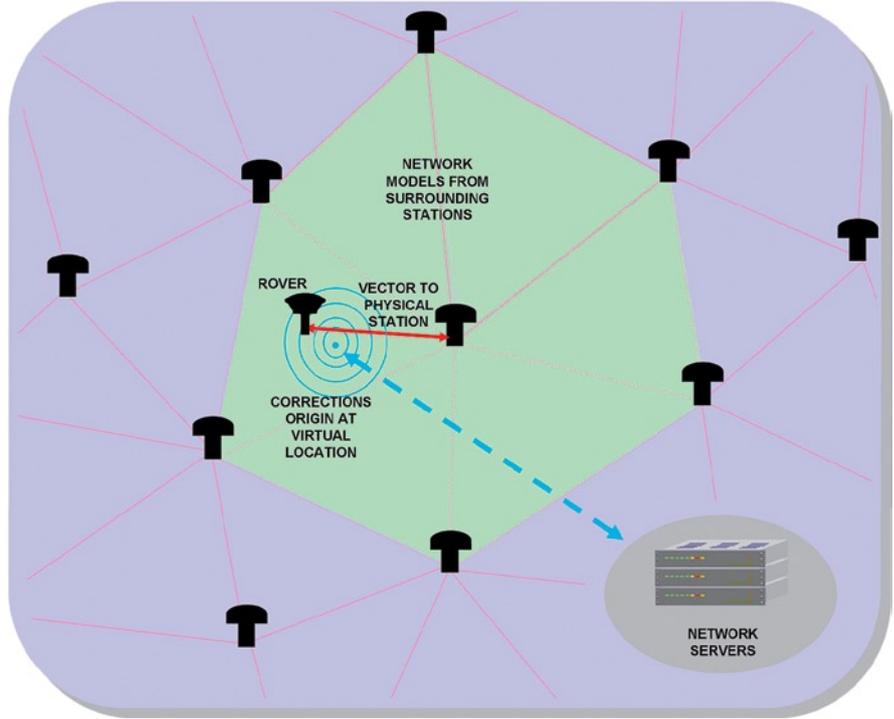


Figure 5 Continually updated localized corrections are sent to the rover.

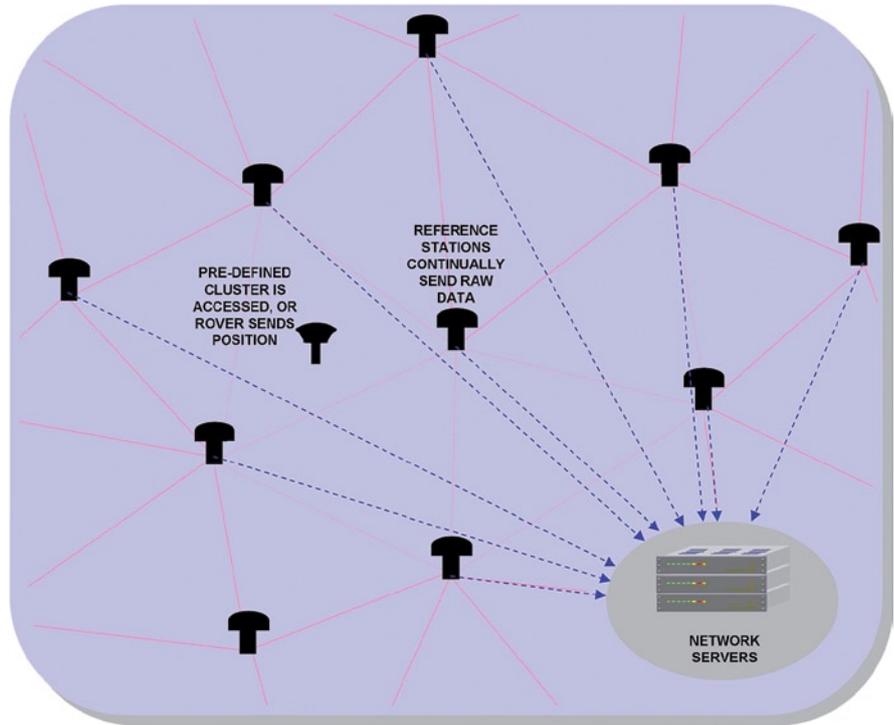


Figure 6 Reference stations continually send raw data.

The master station need not be the closest to the rover, and would likely not be if the implementation has predefined clusters or cells (as would be in a broadcast-only implementation). But with a bi-directional implementation, the master-auxiliary relationships could be re-tasked to always make the closest the master.

Here is how it works in a common implementation:

- Reference stations continually transmit raw data (low latency) to the network servers.
- Network estimation process to reduce stations to a common ambiguity level (Figure 6).

- Either a pre-defined cluster or cell of stations (e.g., as few as 3, or as many as, say, 30) is accessed, or optionally the rover sends its position, and a set of stations is selected based on proximity to that location.
- Derivation of RTCM 3.1 Network Messages for the master station, and offset messages for each auxiliary station are broadcast or transmitted to the rover (**Figure 7**).
- Rover (client-side processing) computes the high precision rover position.

algorithms that seek to fix ambiguities at the single-epoch level. The RTD (Real-Time Dynamics) suite of solutions, a product line of Geodetics Inc., and mainly a by-product of the research and development Dr. Yehuda Bock, has sought to implement these and other algorithmic solutions in either client or server side processing (or a combination of both).

The “reverse RTK” treats the rover as a sort of “guest” station on the network; the rover’s raw data is sent to the central server and its coordinates are computed

needs, and further for applications where the rover could send its resultant positioning to other applications or recorded on the server (also possible in some variations of the other approaches).

As with the other approaches, the solutions seek to utilize the benefits of a fully constrained and monitored reference network, and to apply this positional integrity and the ability to utilize these stations to model and fix ambiguities experienced by both rovers and reference stations. The widest implementation of this suite of solutions is in southern California, where it originated and continues to be developed.

Conclusions?

It would be a great disservice to the current users of RTN, or those about to step into this potentially wonderful amenity to try to compare “pluses and minuses” of the various approaches. There is no tremendous weight of empirical data to fully support the supremacy of any approach. Indeed, you can find conflicting opinions even within the academic communities on these approaches. I have heard explanations of the approaches vary wildly even amongst technical staff of the same vendor, let alone their respective sales staff. This is a new subject for a lot of these folks too, and there are good sincere folks in academia and private industry working to improve these approaches on a continual basis.

There are plenty of happy users of any and all of the approaches and their implemented variations. The best advice for you, the user, is to evaluate to your own criteria. What is the most important factor in deciding? Is it looking for the most common? The longest track record? The broadest choice of approaches? Other unrelated gear features? Head-to-head tests are hard to pull off as it is hard to control or even anticipate all of the possible variables. There might not be one all-inclusive “magic bullet,” but we certainly live in exciting times where so much effort by so many is being continually applied to improve and perfect the amenity of RTN. *AS*

Gavin Schrock is a surveyor in Washington State where he is the administrator of the regional cooperative real-time network, the Washington State Reference Station Network. He has been in surveying and mapping for more than 25 years and is a regular contributor to this publication.

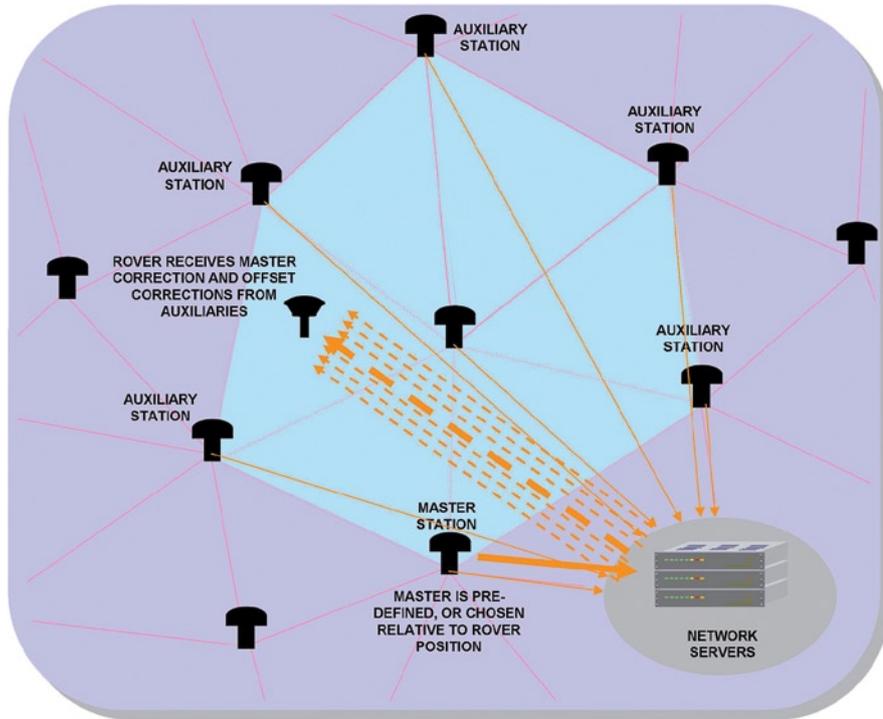


Figure 7 Server sends correction data for master station and correction offsets for auxiliary stations; rover completes processing

Early implementers of this approach include Leica Geosystems (with V2.0 of their SPIDER suite of network software), Trimble (since V2.5 of their GPSNet/RTKnet suite), and others. On the rover end of the equation, the various manufacturers have not adapted so rapidly, with only a handful of brand lines with enough “oomph!” to handle the client-side processing, though some variations have been implemented to allow a wider range of gear to be able to take advantage of at least part of this approach.

A Suite of Dynamic Solutions

Several other approaches have been studied and implemented to include a sort of “reverse RTK” and to apply

by the server with respect to multiple base stations and the sent back to the user. This approach reduces the processing requirements at the rover and, like other server-side approaches, allows for the introduction of external modeling into the network solution. These can include ultra-rapid ephemerides, continental models for ionospheric and tropospheric conditions, geoid models, and crustal motion models.

In the epoch-by-epoch mode, a mainly client-side mode, the raw data from a number of reference stations, their positions, and optionally precise ephemeral data are sent to the rover where further processing is applied to seek fixes for each respective observational epoch. This is envisioned for dynamic positioning