Review of GNSS Formats for Real-Time Positioning

Eldar Rubinov[1], Phil Collier[2], Simon Fuller[3], James Seager[4]

[1]Cooperative Research Centre for Spatial Information, Australia, elrubinov@gmail.com
[2]Cooperative Research Centre for Spatial Information, Australia, p.collier@crcsi.com
[3]Cooperative Research Centre for Spatial Information, Australia, sfuller@thinkspatial.com.au
[4]SeaGIS, Australia, jseager@seagis.com

Abstract

This paper reviews several prevalent formats that are used to transmit GNSS data in real-time. This work has initiated from the research on real-time quality control for Network RTK positioning which aims to independently assess the positioning quality of users performing NRTK surveys. To achieve this purpose, raw observation data from the mobile users needs to be acquired in real-time to be quality assessed. Initially it was intended to use international standard such as RTCM-3 for this purpose, however it was discovered that most current generation GNSS receivers do not support RTCM-3 from a receiver in rover mode. Hence it was necessary to acquire and implement the various binary formats used by different GNSS manufactures. This paper provides a detailed overview of several of these formats. Different formats are examined in terms of their message structure, efficiency, and bandwidth usage. The issue of bandwidth is particularly important as the advent of multiple satellite constellations will see the number of observations (and as a result bandwidth) increase substantially.

1. Introduction

The following review of real-time GNSS formats is the result of the research undertaken by the authors into real-time quality control issues for real-time positioning, as part of the Cooperative Research Centre for Spatial Information (CRC-SI). One of the key ideas driving this research is to provide GNSS users with an independent check on the quality of their positioning in real-time. A major outcome of the research has been the development of the Real-Time Quality Control (RTQC) system. A fundamental requirement of the system is for mobile users to stream their raw observation data to the RTQC server. Once received, the data is analysed, quality assessed, and quality information is sent back in real-time, providing the user with an independent check on the positioning quality. The system is designed to be independent of manufacturer specific processing algorithms, and to work on all brands of GNSS receivers. This is achieved by ensuring that all quality control computations are performed on raw observation data, rather than on derived products such as positions, ionospheric estimates, residuals and so forth (Fuller, 2007).

The design of the RTQC system is shown in Figure 1. The system collects the data from Continuously Operating Reference Station (CORS) networks and mobile users, and quality assesses it in RTQC CORS and RTQC Mobile modules respectively. Two separates modules are necessary due to the fact that quality control computations differ for stationary and moving receivers. Both datasets are then integrated in the RTQC Premium module and a quality indicator is sent to the user in real-time providing them with an independent check on the quality of their positioning.

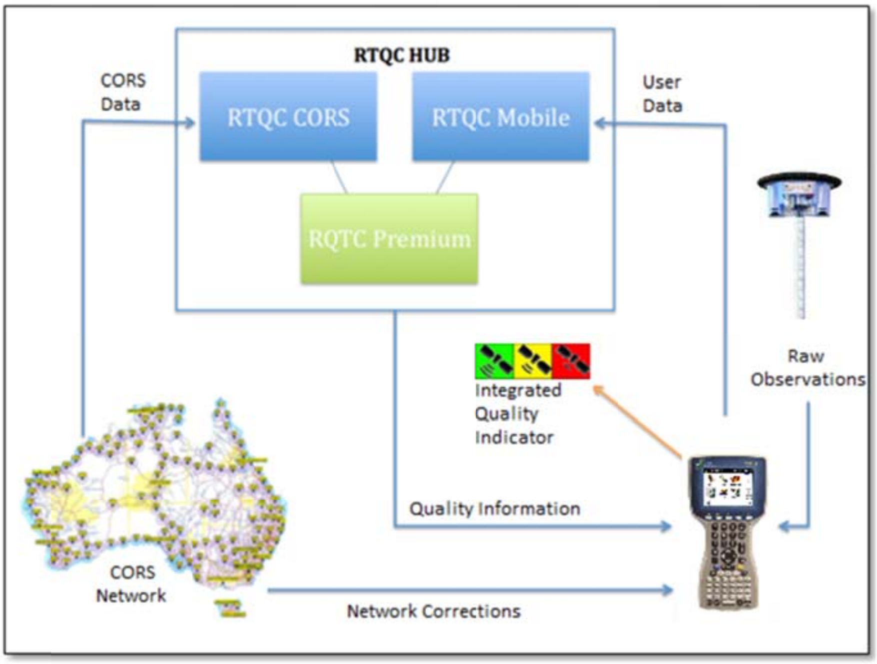

Figure 1: Design of the RTQC System

Initially, the RTQC system was designed to use an international standard such as RTCM-3 (Radio Technical Commission for Maritime Services) for all the data transfer and communications. While this setup has worked fine with data streams from the CORS networks, it was discovered that most current generation GNSS receivers cannot stream RTCM-3 data from a receiver operating in a mobile or "Rover" mode. Rather, with few exceptions, roving receivers tend to either transmit NMEA (National Marine Electronics Association) or their native binary format. NMEA is not acceptable for the purposes of this research as it only provides coordinate information and not the raw observations that are needed for the RTQC quality control computations (NMEA, 2002).

As a result, proprietary formats from different GNSS manufacturers were acquired. This paper reviews some of the more commonly known formats. Different formats are examined in terms of the information they provide, message structure, and bandwidth. This information will be useful to anyone who has to deal with multiple makes of GNSS receivers and write software for a variety of different real-time data formats. It needs to be stressed that while some manufacturers publicly provide this information, others treat details of their proprietary formats as confidential. In this paper information is provided only about formats that are in the public domain.

As it would be impractical to cover each format in detail, two message types for each format will be considered: the GPS raw observation message, and the position message. These two messages are the basic messages necessary for real time positioning and also the two messages required for quality control computations within RTQC. Most formats accommodate GLONASS observations as well, while some also cater for future constellations such as Galileo. Due to the fact that these messages are similar to the GPS observation messages, they are not covered in this review.

In all cases, the position message has a fixed size, while the observation message size is dependent on the number of satellites observed. Throughout this paper, a typical scenario of ten satellites will be assumed for convenience. Even with these two messages, most formats provide numerous options, such as the ability to transmit concise messages that only include the bare minimum information or expanded messages that include enhanced information such as positioning flags, antenna heights, Signal-to-Noise Ratio (SNR), and Dilution of Precision (DOP) values. The reader will be referred to the full specifications of the various formats where appropriate.

2. Overview of GNSS Formats

2.1 RTCM-3

The first format that will be examined is the RTCM Version 3.x (RTCM-3) format. RTCM-3 is an international standard for streaming real-time GNSS data. It is a compact and flexible format, and is supported by all high precision GNSS receivers. However, most receivers can only stream RTCM-3 from a receiver set up as a “Base” or a reference station, but not as a rover. This was the primary reason that RTCM-3 was not suitable format to use within RTQC to acquire data from mobile users.

The current version of the standard (as at January 2011) is RTCM 3.1 which includes messages to support GPS and GLONASS RTK operations, as well as GPS Network Corrections. Future versions of the standard will be released as new constellations become available (RTCM Standard 10403.1, 2006). The message structure consists of a fixed 8 bit preamble, a 10 bit message length definition, a variable length data message, and a 24 bit checksum. Table 1 shows the structure of the RTCM-3 message.

Table 1: RTCM-3 Message Structure (RTCM Standard 10403.1, 2006)

Preamble	Reserved	Message Length	Data Message	Checksum
8 bits	6 bits	10 bits	n bits	24 bits
0xD3	Not defined	Message Length in bytes	Variable length in bytes	QualComm definition CRC-24Q

A Cyclic Redundancy Check (CRC) or polynomial code checksum is used to check the integrity of the message. It is based on a polynomial function to calculate a value (the CRC) for each block of data. Both the block of data and CRC are transmitted, and the receiver of the transmission can recompute the CRC and compare it against the transmitted value. A mismatch implies a data integrity issue. RTCM-3 uses the Qualcomm Corporation algorithm which is described in detail in the format specification.

In the RTCM-3 format, the standard messages range from 1001-1029. The two messages that will be examined are Message 1004 (GPS carrier phase observables) and Message 1006 (position message). These two messages are extended messages and contain additional information such as antenna heights and SNR values. However there are also concise versions of both messages, being messages 1003 and 1005 respectively. These messages provide the minimum data that is required for real-time operation, and as a result save bandwidth. Table 2 provides the size of various RTCM messages along with their content and bandwidth. The adopted sampling rate is 1Hz or one measurement per second. The message size is given in bytes and includes the message only. The bandwidth is defined as the rate of data transfer and is measured in bits per second. Apart from the message, it also includes the transport layer, which in the case of RTCM-3 is 48 bits (Table 1).

Table 2: RTCM-3 Binary Messages (RTCM Standard 10403.1)

Message	Message Content	Message Size (bytes)	Bandwidth (bits/s)
1003	L1 & L2 Pseudorange and Carrier Phase	135	1128
1005	Antenna Position	19	200
1004	L1 & L2 Pseudorange, Carrier Phase and SNR	165	1368
1006	Antenna Position, Antenna Height	21	216

As well as standard messages 1001-1029, RTCM-3 also has another set of messages that are reserved for proprietary use. These are messages 4001-4095. A proprietary message number can be obtained by making application to the RTCM. Only one message identifier can be obtained per applicant, but vendors are free to define sub-messages within the allocated message. The CRC-SI made an application to the RTCM and was given message identifier 4082 for the purposes of RTQC research. RTCM message 4082 is used to deliver quality information to mobile users in real-time (Figure 1). In future it is also intended to transmit stochastic model coefficients to provide a more realistic estimates of user positioning quality. Specifications for RTCM Message 4082 can be found in Fuller et al, (2010). The full list of proprietary RTCM messages, as at the time of writing, is given in the Table 3.

Table 3: List of proprietary RTCM-3 messages

Message	Organization	Message	Organization
4095	Ashtech	4088	IfEN GmbH
4094	Trimble Navigation Ltd.	4087	Fugro Pty Ltd.
4093	NovAtel Inc.	4086	inPosition GmbH
4092	Leica Geosystems	4085	European GNSS Supervisory Authority
4091	Topcon Positioning Systems	4084	Geodetics, Inc.
4090	Geo++	4083	German Aerospace Center (DLR)
4089	Septentrio Satellite Navigation	4082	CRC-SI

2.2 RT17

Record Type 17 (RT17) is binary format developed by Trimble Navigation. RT17 was developed in the mid-1990s to use with the Trimble 4000 receiver series. RT17 is a publicly available format and its specifications can be found in the various Trimble reference manuals such as Trimble BD970 GNSS Receiver Module (Trimble, 2010). The structure of RT17 is given in Table 4.

Table 4: RT17 Message structure (Trimble, 2010)

STX	Status	Packet Type	Message Length	Data Block	Checksum	ETX
8 bits	8 bits	8 bits	8 bits	n bits	8 bits	8 bits
02h	Receiver Status	Message Type	Message length in bytes	Message data	mod 256	03h

It can be seen from Table 4 that each RT17 message is sent within a six byte frame – 4 byte packet header and 2 byte packet tail. The checksum is calculated using (Status + Type + Length + Data Block) mod 256. Mod 256 is calculated by summing every byte of the message up to but not including the checksum field itself. This checksum is then transformed into a modulo 256 number for transmission and comparison. Both the raw observation and the position records are sent within a single RT17 message (Message 57h). The raw observation message has the option of being sent in expanded or concise mode. Specifications of the two RT17 messages of interest are given in Table 5.

Table 5: RT17 Binary Messages (Trimble, 2010)

Message	Message Content	Message Size (bytes)	Bandwidth (bits/s)
Observations (expanded)	L1 & L2 Pseudorange , Carrier Phase, SNR, Doppler, Cycle Slip Count, Azimuth, Elevation	859	6920
Observations (concise)	L1 & L2 Pseudorange , Carrier Phase, SNR, Doppler, Cycle Slip Count, Azimuth, Elevation	441	3576
Position	Antenna Position, Solution Type, Number of Satellites	101	856

Compact Measurement Record (CMR) is another format that is being used by Trimble. CMR was released in 1996 quickly becoming an industry standard. The specifications of the CMR format can be found in Talbot (1996). CMR (along with its later version CMR+) is an efficient format in terms of bandwidth usage and as such it is used by various receiver manufacturers along with RTCM. CMR is not included in this review due to the fact that some of its messages refer explicitly to a reference station, which means that it cannot be streamed from a rover. As such it is not applicable for the purposes of RTQC or any other server-side application.

2.3 LB2

Leica Binary 2 (LB2) is a binary format developed by Leica GeoSystems. Several versions of LB2 are available according to the generation of receivers that it is used with. Leica does not change the name of the format, only the version number (LB2 Version 5.0, LB2 Version 5.5 etc.). The LB2 format is protected and not available to the general public. As such it will not be included in this review.

2.4 NovAtel Binary Format

NovAtel binary format is used to communicate and log data from the OEMV family of receivers. The NovAtel Binary format is publicly available and its specifications can be found in the OEMV Firmware Reference Manual (OEMV, 2010). In the NovAtel format each command (or log) has a specific identifier. The two logs that will be examined here are the RANGE log (raw measurements) and BESTXYZ log (position). In addition to the binary form, the format is also available in ASCII and abbreviated ASCII. The structure of the messages is shown in Table 6.

Table 6: NovAtel Binary Message Structure (OEMV, 2010)

Sync	Header	Message Data	Checksum
24 bits	200 bits	n bits	32 bits
0xAA, 0x44, 0x12	Header Information	Variable length in bytes	32 bit CRC

The BESTXYZ message contains the receiver's best available position and velocity estimates. The message is quite comprehensive and includes Cartesian coordinates, position type, velocity type, velocity vector along each axis, standard deviations for position and velocity, as well as status flags to check whether the data is valid. The RANGE log contains measurements for all tracked satellites. Apart from carrier phase observations, it also contains pseudorange data, SNR, and standard deviations for both code and carrier signals. RANGECMP is a compressed version of the RANGE log. NovAtel binary format caters for both GPS and GLONASS. Amendments for future constellations will be made in due course.

Table 7: NovAtel Binary Messages (OEMV, 2010)

Message ID	Message Content	Message Size (bytes)	Bandwidth (bits/s)
RANGE	L1 & L2 Pseudorange, Carrier Phase, SNR	880	7328
RANGECMP	L1 & L2 Pseudorange, Carrier Phase, SNR	480	4128
BESTXYZ	Antenna Position, Velocity, Base St Id, etc.	112	1152

A distinguishing feature of the NovAtel format is that 'standard' high level programming language data types that align with byte boundaries are used. This makes it very easy to write and debug decoders for this format, as no bit wise manipulation of raw data is required. On the other hand, the size of the messages (and thus bandwidth usage) is significantly increased.

2.5 JPS

JPS is a format being developed and used by Javad GNSS. The format is publicly available and its specifications are given in the GREIS (GNSS Receiver External Interface Specification) reference manual (GREIS, 2009). JPS allows mixing of both ASCII and binary messages in the same format. It also allows foreign messages (such as NMEA) to be mixed with native JPS messages in a single data stream. Another feature of the JPS format is that it allows storing only one kind of measurement in a single message type, and providing wide range of pre-defined messages for different measurements types. This way of representing standard JPS messages is simultaneously flexible and extensible.

C programming language notation is used to describe a binary string as a sequence of bytes. Each message begins with a unique message identifier comprising two ASCII characters. This is followed by the length of message body field, which specifies the length of the message. Message body follows immediately. The message body contains exactly the number of bytes specified by the length field. An extensive library of standard pre-defined messages is available, but non-standard ASCII messages can also be used. The position message [PO] has the following form:

```
struct Pos {30}
{
    f8 x, y, z;          // Cartesian coordinates [m]
    f4 sigma;            // Position SEP6 [m]
    u1 solType;          // Solution type
    u1 cs;               // Checksum
};
```

There are several types of raw observation messages that include different measurement combinations. Some of these are [PC], [P1], [P2], [P3] and [P5], which correspond to CA/L1, P/L1, P/L2, CA/L2 and L5 carrier phases respectively. These are only several examples taken from a much larger set of pre-defined messages. The observation message has the following form:

```
struct CP {8*nSats+1}
{
    f8 cp[nSats];        // CP, [cycles]
    u1 cs;               // Checksum
};
```

The 8-bit checksum is computed starting from the first byte of the message identifier and ending with the byte immediately preceding the checksum field. The algorithm for computing the checksum can be found in the GREIS manual. The JPS binary message summary is given in Table 8.

Table 8: JPS Binary Messages (GREIS, 2009)

Message	Message Content	Message Size (bytes)	Bandwidth (bits/sec)
PC	L1 Pseudorange and Carrier Phase	81	648
P2	L2 Pseudorange and Carrier Phase	81	648
Pos	Cartesian Coordinates, Solution Type	30	240

2.6 ATOM

AshTech Optimized Messaging (ATOM) is a binary GNSS format developed by Ashtech. A distinguishing feature of the ATOM format is the various compression options which can be used to save bandwidth. The structure of the format is given in Artushkin et al, (2008) and the full format specifications can be found in the ATOM reference manual (ATOM, 2010), which can be obtained by request from Ashtech.

ATOM uses RTCM-3 message 4095 (see Table 2) as a transport layer, which means that ATOM messages have the same message structure as RTCM-3 messages. Each ATOM message has the following format:

Table 9: ATOM Message Structure (Artushkin et al, 2008)

Message	Message Sub-Number	Message version	Message Body
12 bits	4 bits	3 bits	0… 8165 bits
0xFFF	ATOM message group	Allowance for firmware upgrades	Variable length

The decision to use RTCM-3 as the transport layer for ATOM development was made to allow third party vendors to decode ATOM easily using standard RTCM-3 decoders. ATOM has several different message groups such as MES (observations), PVT (positions), NAV (navigation data) and others. The most flexible and powerful of these is the RNX (RiNeX) group. RNX is designed to generate raw observations to allow effective and unambiguous conversion to RINEX-3. The variety of GNSS and their signals is almost unlimited in RNX messages, because it uses universal and flexible data identification. The RNX messages can be customized from fully expanded to fully compact, allowing users to choose between message size and data availability. The RNX message group has sub-messages for raw observation data, as well as a sub-message for antenna position. The organization of the ATOM RNX message group is shown in Figure 2.

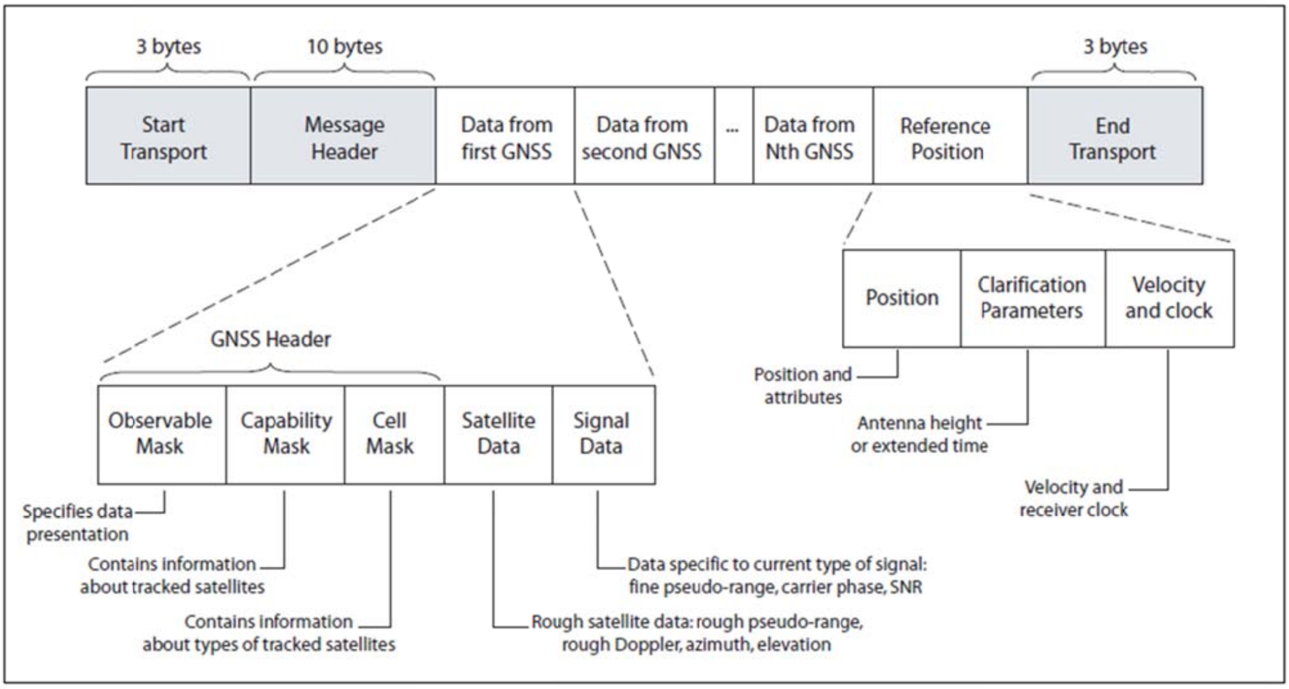

Figure 2: ATOM RNX Message Structure (ATOM, 2010)

RNX messages have an extra feature, they can generate the same observation data in different forms, thereby allowing some trade-off between data quality and message throughput. These different forms of data presentation are available through the so-called **SCN,x** scenario where the integer x stands for the scenario number. The data can be presented with several levels of compression – compact (SCN,100), standard (SCN,4) and full (SCN,0). Table 10 shows the RNX binary messages.

Table 10: ATOM binary messages (ATOM, 2010)

Message	Message Content	Message Size (bytes)	Bandwidth (bits/sec)
RNX Obs (SCN,100)	Compact Representation	80[1]	768
RNX Pos (SCN,100)		16	
RNX Obs (SCN,4)	Standard Representation	158[1]	1416
RNX Pos (SCN,4)		19	
RNX Obs (SCN,0)	Full representation	415[1]	3600
RNX Pos (SCN,0)		35	

The combined bandwidth is shown for both messages, as both the position and the observations are transmitted within a single message.

[1] These values are approximate only due to the complex nature and variability of computing these messages. The numbers are based on an example in the ATOM Reference Guide.

2.7 SBF

Septentrio Binary Format (SBF) is developed by Septentrio Satellite Navigation. Format specifications can be found in the SBF Reference Guide (SBF, 2007) available by request from Septentrio. In the SBF, the data is arranged in binary blocks referred to as SBF blocks. Each block consists of a sequence of numeric or alphanumeric fields of different types and sizes. The total block size is always a multiple of four bytes. Every block begins with an 8-byte header, which is followed by the block body. The SBF header format is given in Table 11.

Table 11: SBF Message Structure (SBF, 2007)

Sync	Checksum	Message ID	Length
16 bits	16 bits	16 bits	16 bits
0x24, 0x40	CRC-CCITT polynomial	Block type and contents	Size of SBF block

Each block has an unambiguous ID for easy identification. The raw observation block is called MeasEpoch(v2) and has a block ID of 4027 and the position block is called PVTCartesian(v1) and has a block ID of 5903. The details of these blocks are given in Table 12.

Table 12: SBF binary messages (SBF, 2007)

Message	Message Content	Message Size (bytes)	Bandwidth (bits/sec)
MeasEpoch	L1 & L2 pseudorange and carrier phase, Doppler, SNR	320	2728
PVTCartesian	Antenna Position, Velocity, Time, Receiver Clock	76	720

The MeasEpoch block contains all measurements of all tracked signals for a particular epoch. Each measurement is stored in a sub-block of the MeasEpoch block. To decrease the block size, all measurements are referenced to one “master” measurement set. For example L2 is referenced to L1. The PVTCartesian block contains the Cartesian position, velocity and time solution, as well other parameters such as receiver clock bias and receiver clock drift.

3. Bandwidth Comparison

A comparison has been made between the formats to determine the most efficient format with respect to bandwidth usage (Figure 3). The graph is based on a ten minute time interval using the figures used throughout the paper which is ten satellites streaming both the observation and position messages at 1Hz interval. The bandwidth is shown in kilobits per second (Kbps).

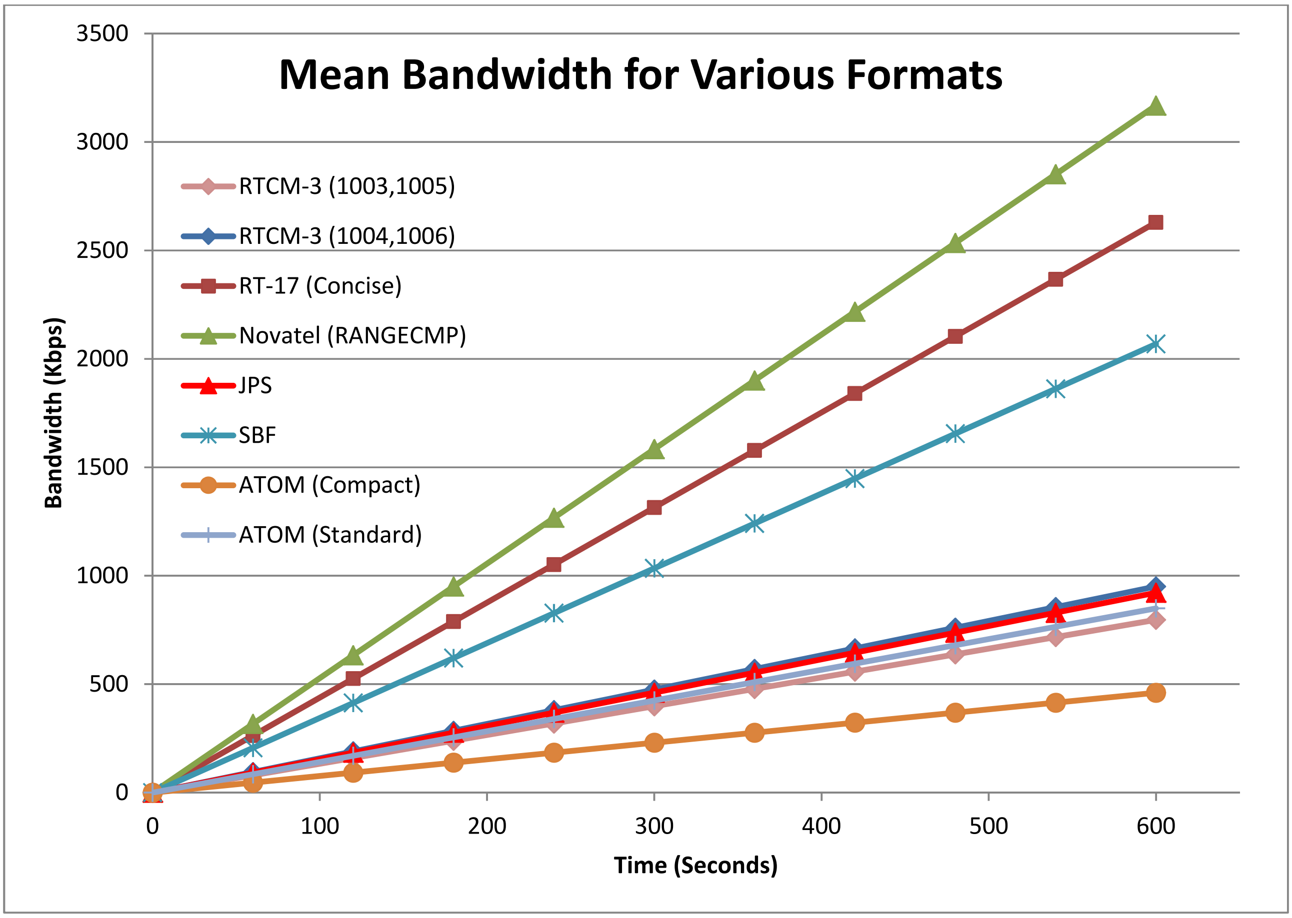

Figure 3: Bandwidth Comparison

It can be seen that the ATOM compact messages (SCN,100) have the most minimal bandwidth of all the formats examined in this paper. This is achieved by providing the minimal information only and using various compression strategies. On the other hand NovAtel, SBF and RT-17 consume the most amount of bandwidth, as a result of being the most comprehensive messages. JPS and ATOM standard messages (SCN,4) consume similar amount of bandwidth relative to the RTCM-3 standard.

So what do these numbers mean in a real-world scenario? This paper is written with a server-side application (such as, but not limited to RTQC) in mind where mobile users need to stream real-time data back to the server. The only option to stream data from the field, while performing the survey, is to use a wireless communication channel either from the survey equipment itself or via a mobile phone. There are several types of wireless communication technologies available which can generally be grouped into three generations – 1G, 2G, and 3G. 1G is the analog technology which was introduced in the 1980s and was superseded by the digital telecommunications in the early 1990s. 2G is a second generation wireless technology that was introduced in 1991. 2G provides relatively low data transmission rate through various standards such as Code Division Multiple Access (CDMA) and Global System for Mobile communication (GSM). 2.5G was an enhancement to 2G and used General Packet Radio Service (GPRS) standard. Whilst 2G has been superseded by the newer technologies, it is still used in many networks in different parts of the world. Finally 3G is the third generation wireless communication technology which supports high speed data capacities and is intended for demanding data transfer applications. The current version is 3.5G which supports HSPA (High Speed Packet Access) standard. HSPA improves existing technologies by reducing latency and increasing the system uplink and downlink capacity (ACMA, 2007). For an extensive review of the various wireless communication standards the reader is referred to Wang et al, (2009). Table 13 provides the specifications of the various wireless data communication technologies:

Table 13: Wireless Data Communication Standards (Wang et al, 2009)

Technology	Maximum Downlink	Maximum Uplink
GSM	9.6 Kbps	9.6 Kbps
GPRS	80 Kbps	20 Kbps
HSPA	14.4 Mbps	5.76 Mbps

The figures presented in Table 13 provide the maximum values for all users of the service. The true bandwidth will never be that high and will depend on such factors as number of users, how much signal is available, distance from the nearest tower, temporal changes in the environment and so on. The uplink speed is a lot slower than downlink due to the fact that networks are designed to be asymmetric, as the users mostly require to download fast than to upload fast. The author did an empirical test by measuring the internet speed on a typical survey controller used in the field enabled with a 3.5G modem. Out of ten attempts the average downlink speed was 124.5Kbps while the lowest was 41Kbps. On the other hand the average uplink speed was 75.1Kbps with the lowest recorded speed of 34.1Kbps. It can be seen that these figures are a lot lower than those quoted in Table 13.

In the RTQC scenario (Figure 1), a mobile user will have three communications channels activated. Two downlink channels to receive network RTK corrections and quality information from the RTQC, as well as one uplink channel to stream raw observation data to the RTQC server. The network RTK corrections will typically be transmitted in RTCM-3 format as either Virtual Reference Station (VRS) messages (Vollath et al, 2000), or Master-Auxiliary Concept (MAC) messages (Euler et al, 2001). According to Janssen (2009), the average theoretical bandwidth for the VRS corrections is 1.8Kbps. The bandwidth for MAC depends on the number of reference stations involved, and for a case of six stations, the will be around 5.6Kbps. The RTQC quality information, transmitted via RTCM Message 4082, is also dependent on the number of reference stations. For the same scenario of six reference stations it will require 0.3Kbps of bandwidth (Fuller et al, 2010). In total, the RTQC user will require 2.1Kbps of downlink bandwidth using VRS, and 5.9Kbps using MAC. The uplink bandwidth will depend on the format used, and could be as high as 8.5Kbps using the NovAtel RANGE messages (Table 7), or as low as 0.8Kbps using compact ATOM messages (Table 10). According to these figures

RTQC user should be well within the bandwidth limits if using GPRS technology or above, but the required bandwidth might not be adequate if using 2G technologies such as GSM.

4. Conclusion

In this paper, an overview of several GNSS formats was presented with a particular emphasis on server-side applications where real-time information is required from rover receivers. Only two basic messages, raw observations and position, were examined, and the reader was referred to the full format specifications where applicable. The formats were examined in terms of message structure, design philosophy, and bandwidth requirements. The bandwidth was compared with the existing wireless communication technologies to see whether the various formats are applicable for use in demanding real-time applications such as RTQC.

References

ACMA (2007), 'Communications Infrastructure and Services Availability in Australia 2006-2007', *Australian Communication and Media Authority.*

Artushkin I, Boriskin A, Kozlov D (2008), 'ATOM: Super Compact and Flexible Format to Store and Transmit GNSS Data', *Proceedings of ION GNSS 2008*, Savannah, Georgia, pp. 1895-1902.

ATOM (2010), *Ashtech Optimized Messaging – Version 1.12*, Ashtech.

Euler H-J, Keenan CR, Zebhauser BE, Wübbena G (2001), 'Study of a simplified approach in utilizing information from permanent reference station arrays', *Proceedings of ION GPS 2001*, Salt Lake City, Utah, pp. 379-391.

Fuller S, Collier PA, Seager J (2007), 'Real-Time Quality Control for CORS Networks and Mobile Users', *Proceedings of IGNSS Symposium 2007,* Sydney, Australia.

Fuller S, Rubinov E, Collier PA, Seager J (2010), 'Integrated Quality Indicators and Stochastic Modelling for Real-Time Positioning: Overview and Implementation', *Proceedings of FIG Congress 2010, Sydney, Australia.*

GREIS (2009), *GNSS Receiver External Interface Specification – Version 3.1.0*, Javad GNSS, San Jose, California.

Janssen V (2009), 'A comparison of the VRS and MAC principles for network RTK', *Proceedings of IGNSS Symposium 2010*, Gold Coast, Australia.

NMEA (2002), *NMEA 0183 Interface Standard – Version 3.01*, New Bern, North Carolina.

OEMV (2010), *OEMV Family Firmware Reference Manual, OM-20000094, Revision 8,* NovAtel Inc, Calgary, Canada.

RTCM (2006), *RTCM Standard 10403.1 - Differential GNSS (Global Navigation Satellite Systems) Services – Version 3,* Arlington, USA: Radio Technical Commission for Maritime Services.

SBF (2007). *SBF Reference Guide – Version 1.0*, Septentrio Satellite Navigation, Leuven, Belgium.

Talbot NC (1996), 'Compact Data Transmission Standard for High-Precision GPS', *Proceedings of ION GPS 1996,* Kansas City, Missouri, pp. 861-871.

Trimble (2010), *Trimble BD970 GNSS Receiver Module* - *Version 4.10, Revision A*, Trimble Navigation Limited, Sunnyvale, California.

Vollath U, Buecherl A, Landau H, Pagels C, Wagner B (2000), 'Multi-base RTK positioning using virtual reference stations', *Proceedings of ION GPS 2000*, Salt Lake City, Utah, pp. 123-131.

Wang C, Feng Y, Higgins M, Looi M (2009), 'Communication Infrastructure Study for Precise Positioning Services in Regional Queensland' *Proceedings of IGNSS Symposium 2010*, Gold Coast, Australia.

ACKNOWLEDGEMENTS

This work was conducted within the CRC for Spatial Information, established and supported under the Australian Government's Cooperative Research Centres Programme. The authors would like to thank all members of the research team as well as the industry and government sponsors of the project for their on-going support. Additionally the authors would like to thank Olivier Casabianca, Sergei Organov and Noor Raziq for their input into the paper.