International Global Navigation Satellite Systems Society
IGNSS Symposium 2007

The University of New South Wales, Sydney, Australia
4 – 6 December, 2007

Test results from the next generation of NTRIP

Thomas S. Yan
School of Surveying & Spatial Information Systems, University of New South Wales, Sydney, Australia
Phone +61 2 9385 4189 Fax +61 2 9313 7493 Email thomas.yan@unsw.edu.au

ABSTRACT

Since its acceptance as an RTCM Standard in September 2004, the NTRIP protocol has gained wide support as a method for disseminating GNSS data in real-time over the internet. Most GNSS softwares and hardware released today have built-in support for NTRIP. An important factor that contributed to NTRIP's success is its adoption of existing internet standards and practices. It was developed based on the ubiquitous HTTP and TCP paired with usage of port 80 commonly used by web servers around the world. This paper looks at the current development efforts in improving the NTRIP protocol. One is to achieve full compatibility with HTTP. Another is to adopt the UDP Unicast method which is commonly used for real-time on-demand audio and video delivery over the internet today. Early test results using this proposed method are also presented.

KEYWORDS: NTRIP, HTTP, UDP, GNSS, internet

1. INTRODUCTION

Networked Transport of RTCM via Internet Protocol (NTRIP) is a protocol designed specifically for transport of GNSS data over the internet. It is a generic, stateless, application-level protocol based on the Hypertext Transfer Protocol (HTTP) 1.1 (Weber, 2004). Most of the NTRIP protocol directives are derived from HTTP. Its architecture was originally based on internet radio system such as Icecast in which several data sources are aggregated at one server and mass distributed to a large number of clients. Well aware of the accessibility issues often faced by operators and users with proxy servers and firewalls, NTRIP adopted the usage of firewall/proxy-friendly protocols such as HTTP and TCP. It also allows and encourages the use of port 80 where possible to further alleviate any access and security issues.

A complete NTRIP system consists of three components: NtripClient, NtripServer and NtripCaster. The NtripCaster is the actual HTTP server program while NtripClient and NtripServer act as HTTP clients (Weber, 2004). In practice, NtripServers usually consist of GNSS data sources, typically GNSS receivers while NtripClients are users who wish to utilise the GNSS data provided by these sources. The NtripCaster itself, which acts as a middle man, is usually though not always hosted at a data centre with high network bandwidth capacity.

Version 1.0 of NTRIP was accepted and published as an RTCM Standard in September 2004. For a detailed description of the NTRIP protocol, readers are directed to the document *Standard for Networked Transport of RTCM via Internet Protocol, RTCM Paper 200-2004/SC104-STD, Version 1.0* available from the Radio Technical Commission for Maritime Services (RTCM) website, rtcm.org.

Most GNSS hardware and softwares produced today have built-in support for NTRIP. GNSS receivers designed for Continuously Operating Reference Station (CORS) typically would have built-in NtripServer capability while receivers designed as rover would have built-in NtripClient capability though not always limited to. GNSS softwares such as Leica Spider and Trimble GPSNet have support for the whole system, being able to function as client, server and caster. Support from these hardware and softwares have served to accelerate the acceptance and usage of NTRIP.

As with many aspects of GNSS operations, NTRIP as a working system is continuously and actively being developed. Its wide acceptance and usage amongst the GNSS community has provided a good and dynamic level of feedback on its current strengths and weaknesses which in turn provides its developers with practical and relevant thoughts and ideas for future changes and improvements.

A draft of the next version of NTRIP, here after referred to as Version 2.0, is currently undergoing internal reviews and tests. Two main aims were identified for Version 2.0 (Weber, 2007). The first aim is to achieve full compatibility with HTTP 1.1 for all components of the protocol. This is set with the expectation of an improved compatibility with firewall and proxy servers. The second aim is to add UDP Unicast as an option for both Server-Caster and Client-Caster communication. UDP is widely used to deliver multimedia content over the internet and it has inherent advantage over TCP when it comes to latency which is a crucial issue in real-time applications.

The major changes and additions with regards to the first aim to achieve full HTTP 1.1 compatibility is described in the following section. It should be noted that it is not meant as a comprehensive and complete documentation of the new version but rather written to allow readers to appreciate the changes proposed. As Version 2.0 is very much a work in progress, it should further be noted that changes continued to be made based on test results and hence some descriptions in this paper may become obsolete or deprecated by the time the new version is finalised.

Following this section, a discussion on the second aim to add new UDP option is presented. As with previous section, similar disclaimers are given. It is hoped that these discussions will stimulate constructive feedback and comments from readers which would in turn lead to wide acceptance of the proposed changes.

2. ACHIEVING FULL COMPATIBILITY WITH HTTP 1.1

2.1 General

Most of the changes described here involve replacing non HTTP 1.1-compliant directives in Version 1.0. Several new directives and header fields are introduced to overcome weaknesses that were present in Version 1.0. As much as possible, these new directives and header fields are adapted or derived from HTTP 1.1.

2.1 Caster Communication

A new general header field “Ntrip-Version” has been proposed in order to handle the different versions of NTRIP. This header field is a special NTRIP extension to HTTP 1.1. This field is applicable to all components of NTRIP system; Client, Server and Caster.

Two other new header fields have been proposed which relate to the NtripCaster components. These are “Server” and “Date”. It is also proposed that GMT is used throughout all NTRIP communication through the use of the new header field “Date”. All NtripCaster responses will always have a sequence of “Ntrip-Version”, “Server” and “Date” fields preceding.

It is also proposed to replace the Version 1.0 response messages of “ICY 200 OK” and “Sourcetable 200 OK” with the HTTP 1.1-compliant response message “HTTP/1.1 200 OK”. The message “ICY 200 OK” was a legacy from the Icecast system as described in the introduction. They are not compatible with HTTP 1.1.

In order to differentiate whether GNSS data or Sourcetable is transmitted, it is proposed that the HTTP entity header field “Content-Type” is used instead. Furthermore it is proposed that new media type “gnss” with subtype “data” and “sourcetable” be registered with the Internet Assigned Numbers Authority (IANA).

An example of a response from a Version 2.0 NtripCaster is given below:

```
HTTP/1.1 200 OK<CR><LF>
Ntrip-Version: Ntrip/2.0<CR><LF>
Server: NTRIP Caster 2.0.1<CR><LF>
Date: 25/Nov/2007:18:30:45: GMT<CR><LF>
Content-Type: gnss/data<CR><LF>
```

2.2 Client Communication

In addition to the new “Ntrip-Version” header field which applies to all NTRIP components, a specific “Host” header field has been added for the NtripClient component. Introduction of this new directive will allow support for virtual hosts. “Host” is part of HTTP 1.1 specifications.

An example of an NtripClient request including these two new header fields is as follow:

```
GET /MountPointA HTTP/1.1<CR><LF>
Host: rtcm-ntrip.org<CR><LF>
Ntrip-Version: Ntrip/2.0<CR><LF>
User-Agent: NTRIP GNSSInternetRadio 2.0.10<CR><LF>
```

```
Accept:*/*<CR><LF>
<CR><LF>
```

A new feature proposed in Version 2.0 is the filtered sourcetable request. When implemented, this feature will allow the NtripClient to send search criteria to the NtripCaster which in turns return a filtered sourcetable. Only contents that matched the search criteria are returned. This allow for reduced traffic and easier access on client's device especially when accessing a large network with high number of mountpoints.

2.3 Server Communication

For Version 2.0, it is proposed that the "SOURCE" directive, which is used by NtripServer to connect to the NtripCaster in Version 1.0, be replaced with the "POST" directive. Unlike "SOURCE", POST is part of HTTP 1.1 and replacing "SOURCE" with "POST" would bring Version 2.0 more aligned with HTTP 1.1.

It is also proposed that user authentication method for Server-Caster connection be updated in Version 2.0. Whereas in Version 1.0 a single encoder password was used for all Servers, in Version 2.0 each Server is allocated its own username and password. The transmission of username and password is enabled via the new header fields "Authorization". This header field is part of HTTP 1.1. Previously in Version 1.0, this field is only applicable to NtripClient.

An example of an NtripServer request message to an NtripCaster incorporating the changes described is as follow:

```
POST /MountPointA HTTP/1.1<CR><LF>
Host: rtcm-ntrip.org<CR><LF>
Ntrip-Version: Ntrip/2.0<CR><LF>
User-Agent: NTRIP ServerCMD 2.0.2<CR><LF>
Authorization: Basic aHvnb2JIbjpodWdvYmVuMTIz<CR><LF>
```

3. UDP UNICAST TRANSPORT OPTION

3.1 Proposed Design

The UDP option is designed with similar principles to multimedia streaming over the internet utilising outband protocol approach. In this approach, Real-Time Streaming Protocol (RTSP) is used on top of TCP for controlling the data stream. It issues commands such as "setup", "play" and "teardown". Real-time Transport Protocol (RTP) on top of UDP is used for data transport. In the case where NMEA message needs to be transmitted, it is sent via RTSP on the stream control channel.

The Real Time Streaming Protocol (RTSP) was developed by the Internet Engineering Task Force (IETF) in 1998 as RFC 2326 (Schulzrinne, 1998). It is a protocol commonly used in streaming media systems which allows a client to remotely control a streaming media server. While RTSP can use either TCP or UDP transports, the proposed design for Version 2.0 specifies for TCP as is commonly the case with many RTSP servers.

RTSP allows interaction using HTTP. In Version 2.0, the HTTP 1.1-compliant directive "GET" is used to download the sourcetable as described in Section 2.2.

Many RTSP servers use the Real-time Transport Protocol (RTP) as the transport protocol for the actual content. This is also what is proposed in Version 2.0. RTP was developed by the Internet Engineering Task Force (IETF) in 1996 as RFC 1889 (Schulzrinne, 1996). It can carry any data but is typically use for those with real-time characteristics such as interactive audio and video. It is built on top of the User Datagram Protocol (UDP).

UDP, in contrast to TCP, does not guarantee reliability or ordering in the way that TCP does. Datagrams may arrive out of order, appear duplicated, or go missing without notice. Avoiding the overhead of checking whether every packet actually arrived makes UDP faster and more efficient (Postel, 1980). Time-sensitive applications often use UDP because dropped packets are preferable to delayed packets. UDP's stateless nature is also useful for servers that answer small queries from huge numbers of clients.

Consistent with the approach taken with the default HTTP/TCP delivery option, most of the NTRIP directives for the UDP Unicast option are derived from and designed to be compatible with RTSP. Since these directives were not part of Version 1.0, no changes can be presented for comparison. Furthermore, since most of these directives are still under testing, it would avoid any confusions in the community not to have them published before they become more stable. For these and the sake of brevity, the author has opted not to include them in this paper.

3.2 UDP Unicast Test

To verify the behaviour of the system under the new option, a series of tests were carried out. They were setup to assess a variety of issues such as proxy compatibility, data completeness, latency of messages and effects on real-time applications. These tests are still ongoing as of the writing of this paper however for readers' interest some early results are presented here. It should be noted though that while the results presented here showed the expected behaviour of the proposed design, the tests are still continuing and no final conclusion is yet to be drawn.

3.2.1 UDP Unicast Test Platform

The test platform utilised a Leica MC500 GPS receiver to produce observation data in Leica raw binary format at 1Hz. This stream is then picked up by two separate NtripServers running on the same server. This NtripServer is a draft version that supports Version 2.0. One NtripServer uploads the stream onto the test NtripCaster using the default HTTP/TCP option while another NtripServer uploads the same stream onto the test NtripCaster using the new UDP option. The test NtripCaster is hosted by BKG. As with the NtripServer, this NtripCaster is also a draft version which supports Version 2.0.

Both mountpoints are then picked up by a Leica GPS Spider software running at UNSW. The computer running Spider is connected directly via serial port to the same receiver producing the observation data. In this way, Spider is able to measure the latency of each stream coming back from the NtripCaster.

3.2.2 UDP Unicast Latency

Latency is measured epoch by epoch, in this case at 1Hz rate. They are averaged out throughout the duration of the test. Two average values were obtained from two separate tests with different test periods. The first value was obtained from a shorter test period of approximately 20 minutes. The second value was obtained from a longer test period of approximately 24 hours. In both tests, the reduction of latency obtained from using the UDP option is found to be fairly close.

Test length	TCP latency	UDP latency	TCP-UDP	Percentage
20 mins	1.16s	0.81s	0.35s	30%
24 hrs	1.23s	0.91s	0.32s	26%

Table 1. Latency of TCP compared to UDP

While the improvement can be seen as fairly significant, what is also important in real-time applications is the ability to transfer the data below the output rate for them to be used optimally. In this test, the latency that occurs in the TCP connection consistently exceeds the output rate rendering the received epoch data out-of-sync. By using the UDP Unicast option it was possible to reduce the latency to below the data output rate and hence apply the epoch data effectively.

3.2.3 UDP Unicast Data Completeness

During a test period over six consecutive days (261-266), the NTRIP connection using TCP transfers the observation data completely without any loss. NTRIP connection using UDP however showed some dropped epochs. On day 262, two data gaps each of 16s and 10s in length were recorded. On day 263, a single data gap of 19s in length was recorded. On day 264, two data gaps each of 15s and 23s in length were recorded.

Day	Length of gap	Percentage of data loss
262	26s	0.03%
263	19s	0.02%
264	38s	0.04%

Table 2. Data completeness with UDP

As described earlier, some data loss using UDP is to be expected due to the design nature of UDP which makes no effort to guarantee the delivery of data packets over the network.

4. CONCLUSIONS

Details for a new version of NTRIP have been discussed. Several proposed changes to NTRIP Version 1.0 were presented. These changes were proposed to make NTRIP fully compatible with HTTP 1.1 and hence better compatibility with firewalls and proxy servers. A new transport option using UDP Unicast was also presented. Early test results showed expected behaviour from the proposed design. More tests are currently still ongoing to satisfactorily demonstrate the impact and benefits from these changes and new design.

ACKNOWLEDGEMENTS

The author would like to acknowledge the invaluable support and contributions received from Dr. Georg Weber from the Federal Agency for Cartography and Geodesy (BKG), Germany.

REFERENCES

Postel J (1980) User Datagram Protocol. http://tools.ietf.org/html/rfc768. Cited 30 Sept 2007.

Schulzrinne H, Rao A, Lanphier R (1998) Real Time Streaming Protocol (RTSP). http://tools.ietf.org/html/rfc2326. Cited 30 Sept 2007.

Schulzrinne H, Casner S, Frederick R, Jacobson V (1996) RTP: A Transport Protocol for Real-Time Applications. http://tools.ietf.org/html/rfc1889. Cited 30 Sept 2007.

Weber G (2004) Networked Transport of RTCM via Internet Protocol Version 1.0. http://igs.bkg.bund.de/root_ftp/NTRIP/documentation/NtripDocumentation.pdf. Cited 30 Sept 2007.

Weber G (2007) Towards Ntrip Version 2.0, BKG, Frankfurt