

Results of June 2013 Wave Glider HotSpot RF range tests

PRELIMINARY DRAFT - DO NOT DISTRIBUTE

Tom O'Reilly

We conducted at-sea HotSpot tests on June 12 and June 14, 2013. Test objectives included:

- Measure characteristics of Freewave link between Dorado and WG at various ranges
- Measure performance of rsync over Freewave between Dorado and WG at various ranges, including typical transfer rates and error recovery

Chaffey and O'Reilly monitored the test from their vantage point aboard the *Rachel Carson*; Chaffey visually tracked both vehicles from the bridge, while O'Reilly was logged into the WG SMC via Freewave radio from the Carson dry lab.

Test setup

Referring to Figure 1, the following radios were installed on the WG SMC:

- HotSpot Freewave, radio 'C' - links to Dorado MVC Freewave radio, via 'SLIP'
- Console Freewave, radio 'B' - links to a Freewave aboard the Carson (radio 'A'), provides a SMC RF "console" to monitor the test via 'ppp'

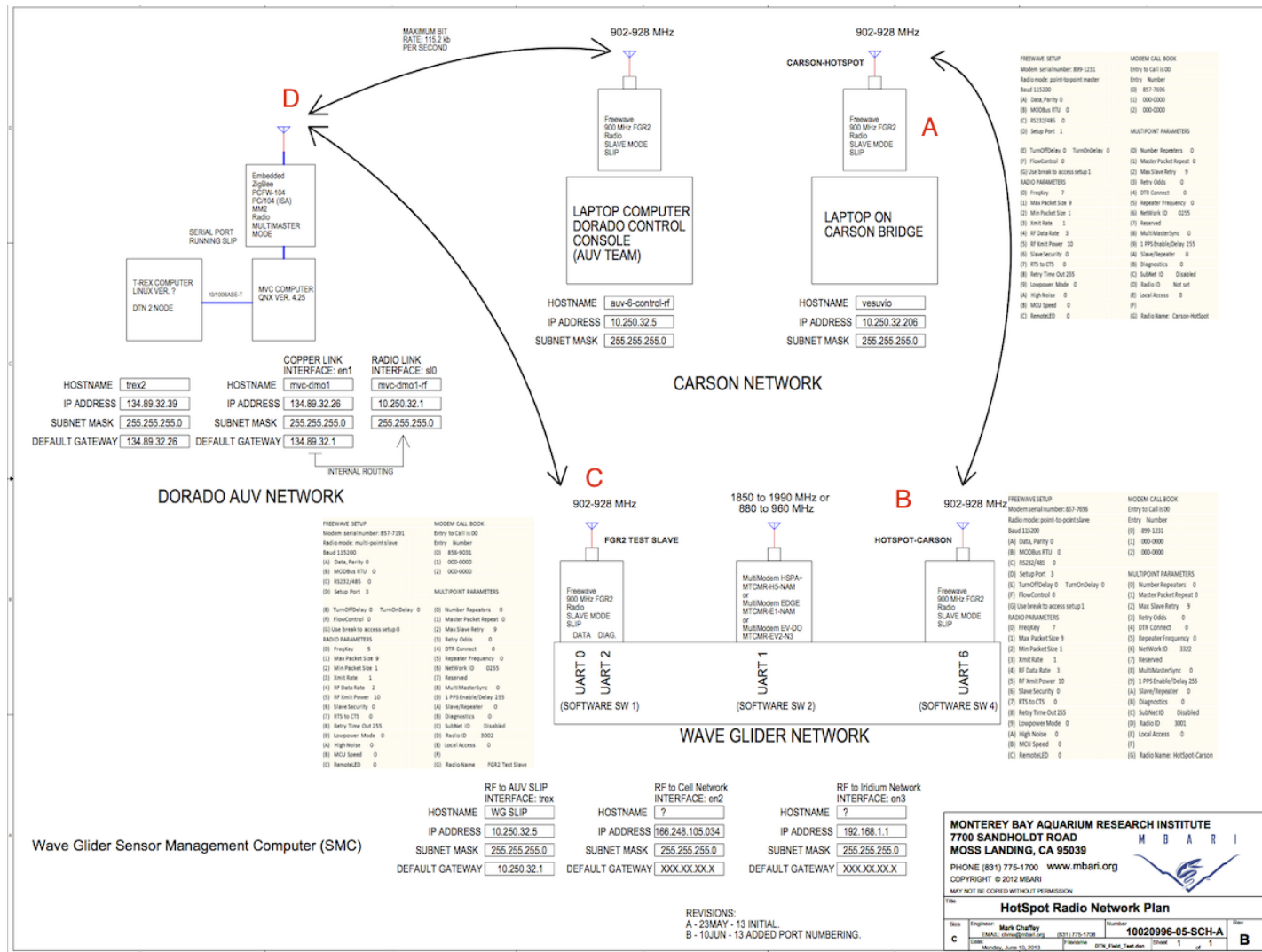


Figure 1: Radios used for June 12,14 tests. Radio A on the Carson communicates with a “console” radio B installed on the WG for this test; this link provides a means to monitor test status from the ship. Radio C (aka “hotspot” radio) on the WG links with radio D on the Dorado MVC, providing the primary data exchange channel.

Freewave radio configuration goes here. All Freewave data rates were configured as 115 KB/sec, and radio power was set to the maximum.

A file-set consisting of 27 ascii files was transferred from the Dorado “trex” computer to the WG SMC during these tests. Each file in the set is 50KB in size (except for one file at 44.739 KB) - the file-set totals 1.34 MB.

IP stack configuration goes here (timeout settings, etc)

In these tests, rync used 'ssh' as the transport, rather than contacting a remote rsync daemon on trex directly. See the [test-syncer](#) script for other rsync options that were used.

The Dorado trex "hotspot" user account was [configured with authentication keys](#) from the WG SMC "hotspot" account, to avoid password prompts during the test rsync transfers.

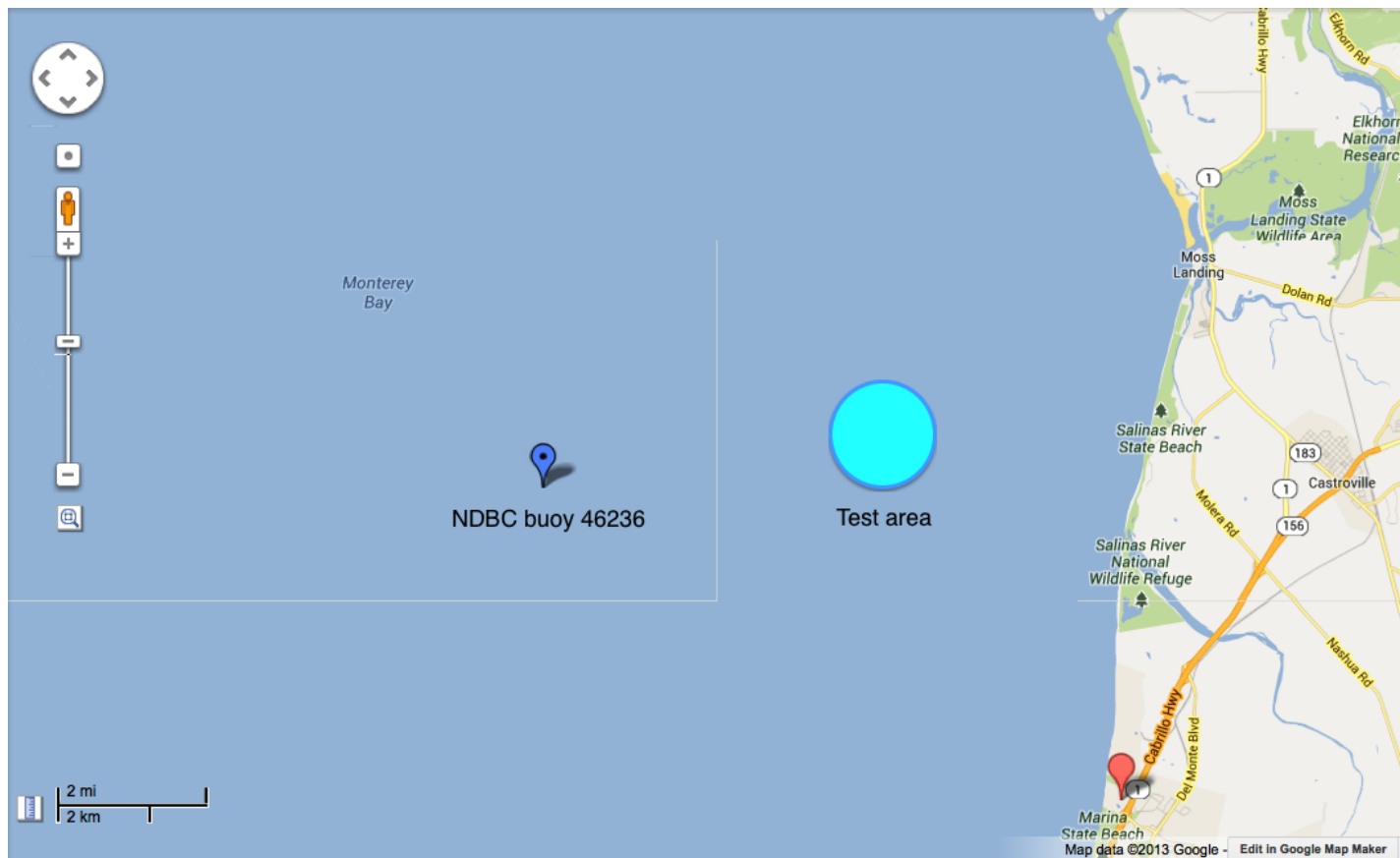


Figure 2: Map showing test area of June 12, 14

Environmental conditions

Wave conditions during the test were measured at NDBC weather buoy 46236, approximately 6 km west of the test site.

June 12, 2013: Weather buoy 46236

time (UTC)	significant wave height (m)	dominant wave period (s)	average wave period (s)	dominant wave direction (deg)
18:48	2	11	6.7	291

19:18	2.2	11	6.7	297
19:48	2.3	9	6.6	296
20:18	2.3	9	6.6	294
20:48	2.4	10	6.4	300
21:18	2.6	10	6.8	296
21:48	2.7	10	6.6	293
22:18	2.5	10	6.5	297
22:48	2.6	10	6.4	293
23:18	2.6	10	6.2	301
MEAN	2.2	10	6.6	296

Average wind speed on June 12 was 10 m/sec = 19 knots, out of the NW.

June 14, 2013: Weather buoy 46236

time (UTC)	significant wave height (m)	dominant wave period (s)	average wave period (s)	dominant wave direction (deg)
15:18	1.8	9	7	300
15:48	1.9	10	6.9	298
16:18	1.8	9	6.9	296
16:48	1.9	11	7	293
17:18	1.7	9	6.7	297
17:48	1.8	8	6.9	297
18:18	1.7	9	6.8	300
18:48	1.7	9	6.7	296
19:18	1.8	9	6.9	294
19:48	1.8	9	6.8	300

20:18	1.9	9	6.7	294
20:48	1.8	9	6.4	298
MEAN	1.8	9	6.8	297

Average wind speed on June 14 was 5.8 m/sec = 11 knots, out of the NW.

Test procedures

At the start of each cruise, script [ping-endpoint](#) was executed on the WG SMC; ping-endpoint sends a 'ping' ICMP packet to a specified endpoint (in this case, the Dorado "trex" computer) once every 5 seconds, with a timeout of 10 seconds. The script continuously time-stamps and logs its results, so ping connectivity between WG and AUV is recorded for the entire cruise.

During each cruise, the AUV was commanded to a series of waypoints; the AUV then drifted on the surface at each waypoint. While the AUV was drifting, the [test-syncer](#) script was executed on the WG SMC, attempting to pull the 1.34 MByte file-set from the AUV trex computer via rsync. The test-syncer script continuously time-stamps and logs its results.

Following the cruises, time-stamped AUV position logs and WG position logs were retrieved from the MBARI AUV operations team and Liquid Robotics, respectively.

Test results

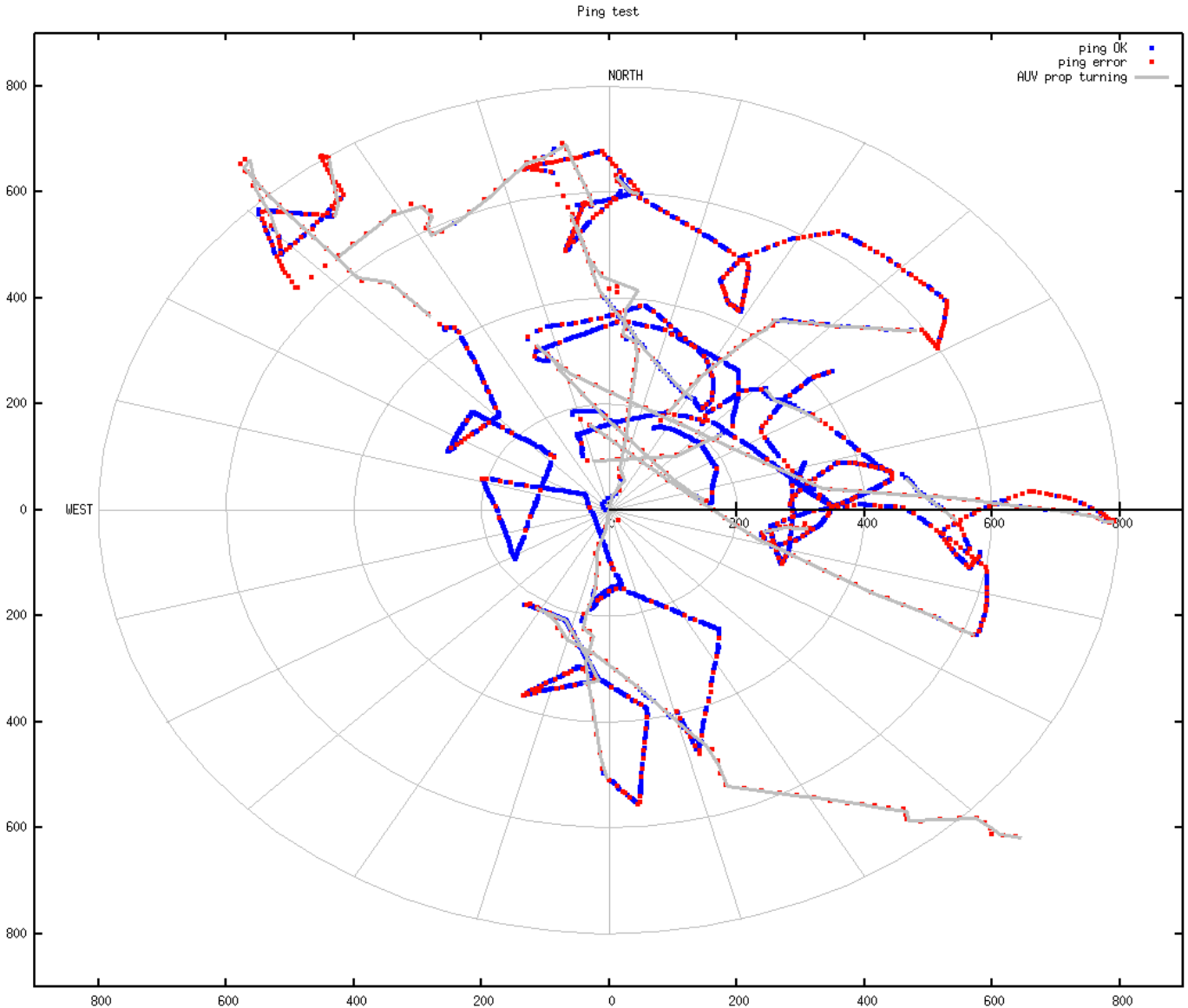


Figure 3: AUV bearing and range from WG, showing results of 'ping' test. Blue dots indicate return packet received from AUV; red dots indicate no response received. Gray tracks mark locations where AUV prop was turning and antenna may have been submerged.

Ping returns were sometimes received at a range of approximately 800 meters. Note that a successful 'ping' does not guarantee link quality sufficient to support TCP/IP-based 'rsync'. Figure 4 shows ping packet loss versus WG-AUV range; as expected packet loss generally increases with range.

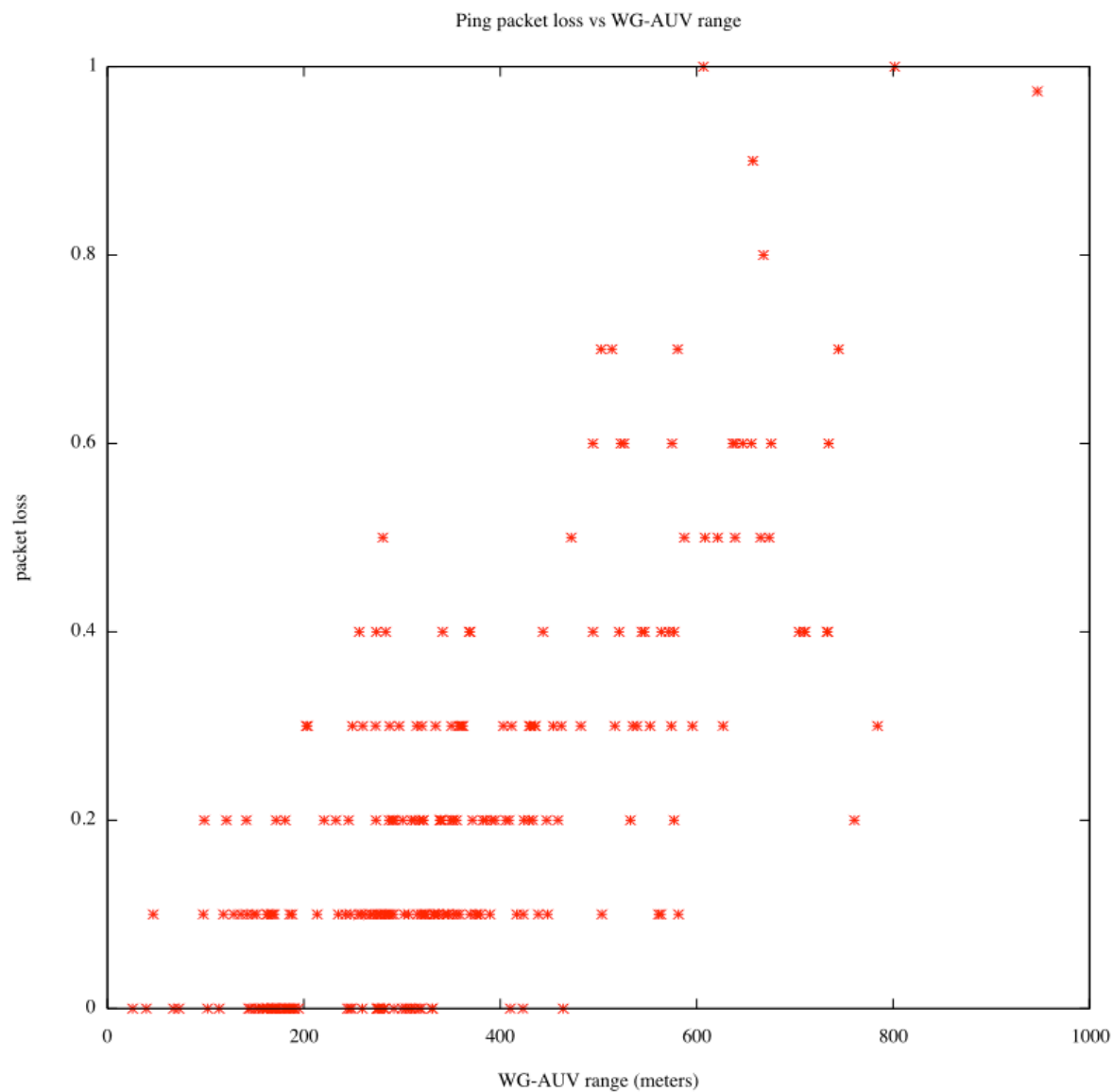


Figure 4: Ping packet loss versus WG-AUV range. Ping was issued from WG to AUV every 5 seconds, with 10 second timeout. Packet loss was computed for every 10 pings issued.

Transfer of the test files from AUV to WG was attempted at ranges out to 800 meters, using the [test-syncer](#) script (which uses rsync). Note that rsync timeout was set to 60 seconds or all of these tests. Figure 5 shows the results of those tests.

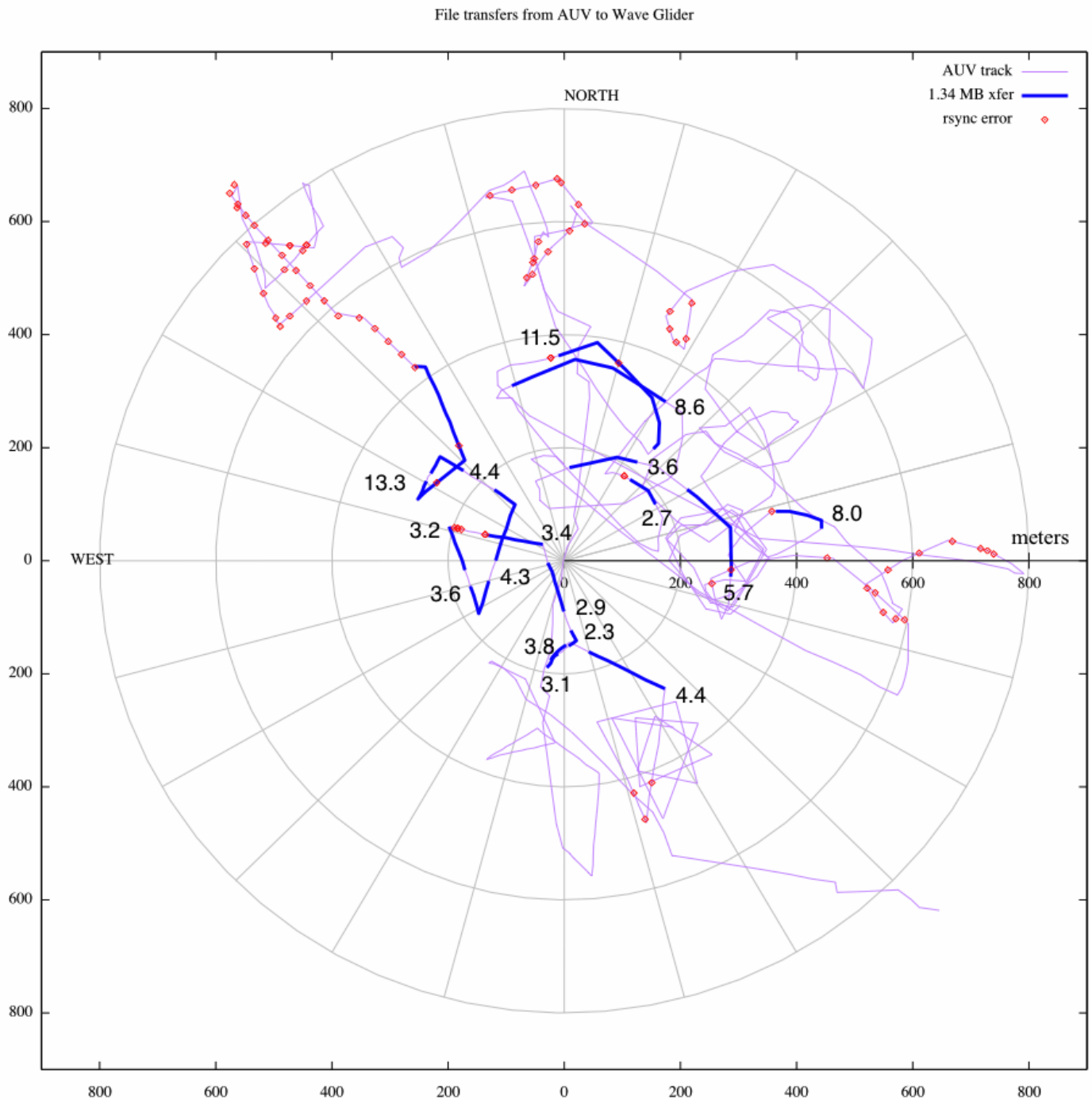


Figure 5: AUV bearing and range from WG, showing results of rsync tests. Blue segments show successful transfers of 1.34 MByte test file set. Numbers next to the blue segments show total duration of the transfer, in minutes. Red dots show location of rsync errors. Rsync timeout was set to 60 seconds for these tests.

The 1.34 MByte filesset was successfully transferred from the AUV to the WG 17 times using the [test-syncer](#) bash script hosted on the WG. test-syncer invokes rsync to pull data files. As can be

seen in Figure 5, rsync errors sometimes occurred during fileset transfer - however the test-syncer script retries in case of error, and was able to recover the entire fileset despite these errors.

Mean transfer time: 4.6 minutes

Farthest successful transfer range: 450 meters, transferred in 8.0 minutes

Mean transfer time for range < 200 meters: 3.2 minutes

Mean transfer time for 200 < range < 450 meters: 7.4 minutes

When rsync errors occurred, they were one of the following:

- error in rsync protocol data stream (code 12) - 46% of all errors
- timeout in data send/receive (code 30) - 54% of all errors

Next steps

1. Using the 'ping' test results, we believe we can realistically simulate at-sea conditions by suitable attenuation of the antenna in the lab. This will allow us to tune the 'rsync' mechanism in the lab.

2. In these tests we characterized the WG-to-AUV Freewave link, testing rsync functionality with the test-syncer script. In the near future we will test shore-WG-AUV link, with a script that implements endpoint-to-endpoint transfers. We plan to implement and demonstrate the following for the Fall 2013 CANON campaign: a) Demonstrate transfer of data files from Dorado AUV to shore and SEACON AUV (University of Porto) to shore, via Freewave and 3G modem. b) Demonstrate transfer of command files from shore to Dorado AUV, via 3G modem and Freewave.

3. Note that the test-syncer script uses rsync with 'ssh' as transport, rather than contacting a remote rsync daemon on trex directly. The encryption associated with 'ssh' transport adds overhead to the link, and the unencrypted [rsync daemon mode](#) may be preferable for the Hotspot application. There are two known disadvantages to the daemon mode approach: 1) a configuration file is required on the endpoint (e.g. AUV) and the endpoint must arrange to start the daemon, and 2) the link is not secure.

4. The Multi-Tech 3G modem has been mechanically/electrically integrated with the SMC. We will implement software to enable rsync between SMC and shore; we plan to test this feature during our August sea date.

Appendix

ping-endpoint listing:

```
#!/bin/bash

# Continually ping target endpoint
if [ $# -ne 1 ]; then
    echo usage: $0 endpointIP
    exit 1
fi

LOGFILE=/opt/hotspot/logs/pinglog.txt
WAIT=5
TIMEOUT=10

while :
do
    date | tee -a $LOGFILE
    ping -c 1 --timeout $TIMEOUT $1 | tee -a $LOGFILE
    sleep $WAIT
done
```

test-syncer listing:

```
#!/bin/bash
# Pull files from connected endpoint
if [ $# -ne 3 ]; then
    echo usage: endpointIP outgoing-subdirectory-name cache-subdir-name
    exit 1
fi

echo "starting test"

LOGFILE=/opt/hotspot/logs/rsynclogs.txt
date | tee -a $LOGFILE

username=hotspot
# Connected IP is first argument
connEndpoint=$1
# Hotspot subdirectory is second argument
subdir=$2

# Destination cache subdirectory is third argument
```

```
destsubdir=$3

HOTSPOT_DIR=/opt/hotspot/
OPTS='--timeout 60 -P -arzpuv --copy-dirlinks --partial
--partial-dir=partial-dir --stats'

# Ensure that destination directory exists
mkdir -p $HOTSPOT_DIR/cache/$connEndpoint/$destsubdir 2>&1 | tee -a
$LOGFILE

# Pull files from source to destination in infinite loop (^C to end
test)
while :
do
    date | tee -a $LOGFILE
    echo Pull files from $connEndpoint to hotspot cache | tee -a
$LOGFILE

    rsync $OPTS $username@${connEndpoint}::$HOTSPOT_DIR/outgoing/$subdir
$HOTSPOT_DIR/cache/$connEndpoint/$destsubdir 2>&1 | tee -a $LOGFILE

    if [ -e $HOTSPOT_DIR/cache/$connEndpoint/$destsubdir/$subdir ];
    then
        echo "got files:"
        ls -l $HOTSPOT_DIR/cache/$connEndpoint/$destsubdir/$subdir
    fi

done
```