



***The EU Framework Programme for Research and Innovation H2020  
Research and Innovation Action***

**CENTAURO**

***Deliverable D7.2 Communication Link***

**Dissemination Level: Public**

Project acronym:	CENTAURO
Project full title:	Robust Mobility and Dexterous Manipulation in Disaster Response by Fullbody Telepresence in a Centaur-like Robot
Grant agreement no.:	644839
Lead beneficiary:	PGX - progenoX GmbH
Authors:	Frank Woodcock, Rafal Kiszko, and Max Schwarz
Work package:	WP7 Integration
Date of preparation:	2016-11-30
Type:	Demonstrator
Version number:	1.0

**Document History**

<b>Version</b>	<b>Date</b>	<b>Author</b>	<b>Description</b>
0.1	2016-08-08	Frank Woodcock	First draft
0.2	2016-08-16	Frank Woodcock	Second draft
0.3	2016-10-16	Frank Woodcock, Rafal Kiszko	Revised Version
0.4	2016-11-29	Rafal Kiszko	Updated Version
0.5	2016-11-30	Max Schwarz	Include Bandwidth Management
1.0	2016-11-30	Coordinator	Submitted Version

**Executive Summary**

This deliverable presents the first version of the communication link between the operator station and the CENTAURO robot. The deliverable introduces the core hardware components, their initial integration, measurement results, and software components for data link management.

## Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Objectives</b>	<b>6</b>
<b>3</b>	<b>Performed Work</b>	<b>7</b>
<b>4</b>	<b>Results</b>	<b>12</b>
<b>5</b>	<b>Communication Link Management of Momaro</b>	<b>25</b>
<b>6</b>	<b>Conclusion</b>	<b>32</b>

# 1 Introduction

One of the central objectives of the CENTAURO project is to perform with a mobile robot complex disaster-response tasks while its human operators maintain a safe distance to the robot in order to stay out of harm's way. Hence, a communication link between the robot and the operators is necessary.

The Communication Link is an robot-side wireless communication link which transmits sensory measurements, estimated states, and percepts from the robot to the operators. It also transmits motion commands and action goals from the operators to the robot. This is illustrated in Figure 1.

The CENTAURO robot will be equipped with advanced wireless transceivers, in order to not interfere with its motion with a cable. The opposite wireless access point might be placed directly on the operator station. As the typical operation environment for the CENTAURO robot will be structured sites, wireless connections might be impeded and part of the distance to the robot may be bridged by a communication cable between the operator station and the operator-side wireless access point and the wireless range might be extended by repeaters.

According to the requirements analysis, a need for three communication channels with different properties has been identified:

- a **low-latency** link for small data packages sent with high frequency for force feedback during teleoperation,
- a **high-bandwidth** link for bulk data sent with low frequency, and
- a **high-availability** link to always maintain connection to the robot.

Although the CENTAURO system constantly monitors the quality of the communication, the communication link itself must be equipped with monitoring capability in order to swiftly identify limitations and failures within the communication link. The communication link manages the priorities of data so in case of e.g. bandwidth limitations, latencies or outages, the sent information will be reduced to only the essential amount that is necessary. This ensures to keep control over the robot.

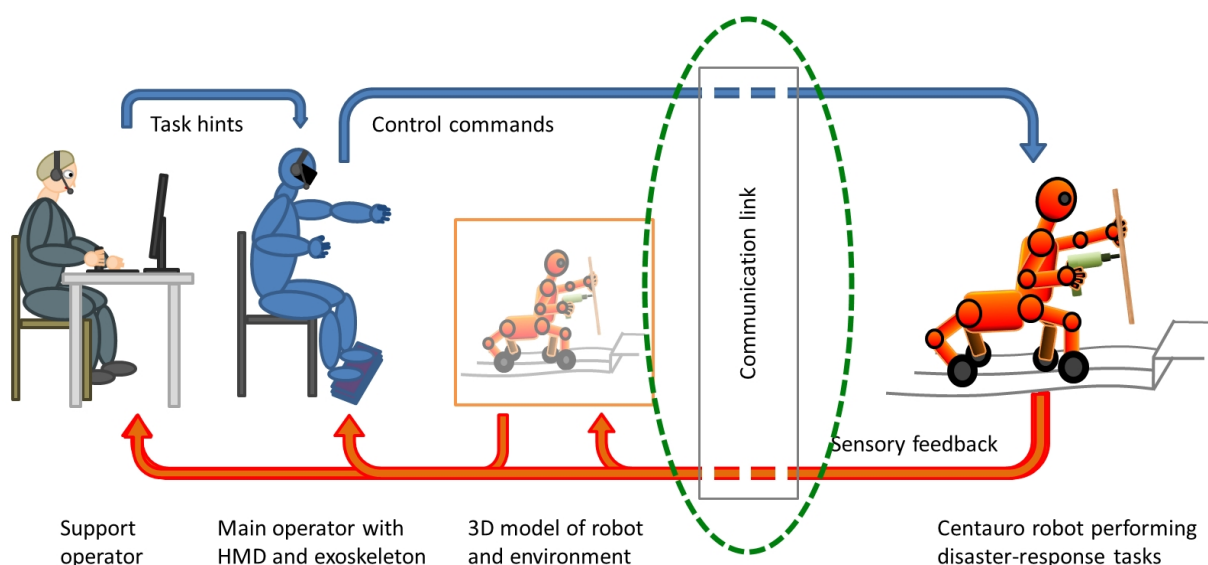


Figure 1: Location of the communication link within the CENTAURO system.

## 2 Objectives

Objective of this deliverable is the communication link, which is the first demonstrator of WP7 – Integration. To this end, the concept of the communication link had to be defined and decided, the hardware components had to be chosen, set up, verified, and evaluated. To properly test and evaluate the chosen components, the typical conditions had to be identified under which the CENTAURO system will have to operate.

In addition to classical requirements like communication distance or bandwidth requirements, there were also operational, logistic, and human-factor demands to be taken into account when deciding about hardware or setting up the test procedures for evaluation.

After deciding the hardware and defining the test procedure for evaluation, the evaluation itself had to be performed.

The result of the evaluation was the demonstration that the identified hard- and software can be used within the CENTAURO system. This is part of the fulfillment of CENTAURO Milestone 2: Core Components.

## 3 Performed Work

### 3.1 Method

We identified requirements towards the communication link on multiple dimensions:

- **Technical:** bandwidth, latency, weight, volume, power consumption, ...
- **Logistical:** availability (COTS vs. development), costs, ease of integration, ...
- **Operational:** Ease of use, reliability, requirements for permits and licenses, ...

In close exchange with the CENTAURO project partners, the principally usable components and technology were defined. A first evaluation of the components by considering the above mentioned needs was performed and components have been decided. The identified components have been procured. The communication link has been assembled and its components have been tested according to a hardware and system test plan specifying the testing scope, tools and requirements.

### 3.2 Communication Link Concept

The CENTAURO communication link consists of three main channels addressing different requirements.

**Low-latency Link** A low-latency connection is needed for force feedback from the CENTAURO robot to the main operator. At regular intervals, e.g. with a frequency of 1000 Hz, small data packages need to be sent from operator to the robot and back. A latency of approx. 1 ms is desirable, while the bandwidth requirements are not very high, e.g. 115 Kbps.

**High-bandwidth Link** For transmission of measurements from high-data-rate robot sensors, such as cameras, RGB-D cameras, and laser scanners, to the operator station, a high bandwidth is needed, i.e. 200 Mbit/s are desirable. These transmissions happen less frequent, e.g. with 10 Hz and, hence, a higher latency of 100 ms is acceptable on this link.

**High-availability Link** Wireless connectivity in structured environments with occlusions, multi-path effects, etc. is error-prone and may fail. In order to always maintain access to the robot even if other data links are interrupted, a high-availability link is needed. Here, good penetration through walls is desired. Neither high bandwidth nor low latency are required.

### 3.3 Communication Setup

After analysis and comparisons, it was decided, that both the low-latency and the high-bandwidth links can be realized using the same WiFi access points from Netgear. The high-availability link can be realized, e.g. through a wireless point-to-point connection. Together with an operator-station side cable and a repeater, this results in the possible setup illustrated in Fig. 2.

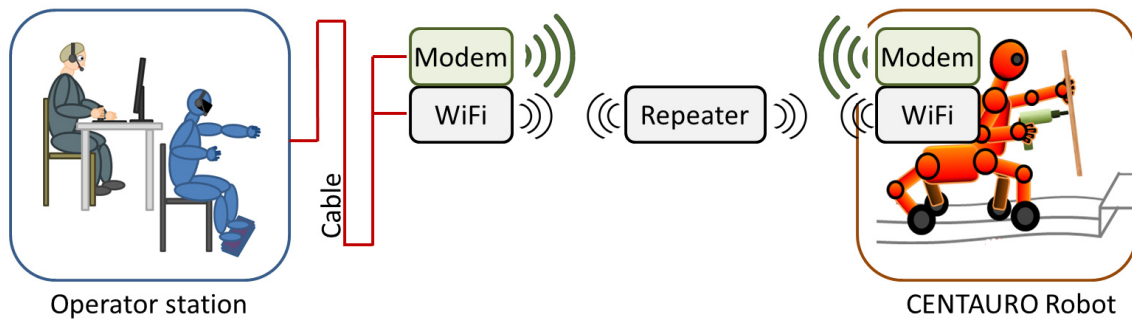


Figure 2: Possible communication setup, where both low-latency and high-bandwidth links are realized by WiFi and the high-availability link is provided by a wireless point-to-point connection (modems).

### 3.4 Selected Components

#### 3.4.1 Netgear WiFi Access point

The selected wireless access point model is Netgear Nighthawk X8-R8500 Tri-Band WiFi Gigabit Router AC5300, shown in Fig. 3. It uses the IEEE 802.11ac standard and is designed to cover large buildings. Its simultaneous Tri-Band WiFi – Tx/Rx  $4 \times 4$  (2.4 GHz) +  $4 \times 4$  (5 GHz) +  $4 \times 4$  (5 GHz) supports implicit and explicit beamforming through four internal and four external antennas. It supports advanced features, such as Smart Connect, which intelligently selects the fastest WiFi band for every device, Load Balancing that distributes WiFi devices ensuring usage of both 5 GHz WiFi bands and MU-MIMO capable-simultaneous streaming of data for multiple devices. Eight high-performance antennas with high-powered amplifiers enhance weak signals and reduce interference. Dynamic QoS prioritizes network traffic by application. This feature can be used to separate the low-latency data link from the high-bandwidth data link.

#### 3.4.2 Netgear WiFi Range Extender

To extend the range of the Netgear WiFi, we use a Netgear Nighthawk AC1900 EX7000 range extender. It supports dual-band WiFi with up to 1900 Mbit/s and is equipped with three antennas. As shown in Fig. 4, we power it with a Lithium-polymer rechargeable battery. The robot can carry it and place it when the wireless signal strength is decreasing.

#### 3.4.3 Amber Wireless Link

For the high-availability link, a long-range low-frequency point-to-point data link from Amber Wireless has been selected: AMB8636. It transmits serial data in the license-free 868MHz band at the maximum allowable transmit power. Fig. 5 shows the module and summarizes its specifications. It transmits data over distances of up to 20 km at a rate of 50 kbs.

The connection to the communicating PCs can be made through:

- Serial port,
- USB port using USB-Serial converter, or
- Ethernet, by using an external hardware Serial to Ethernet converters.

The first two options require Software TCP/IP for using a virtual Ethernet port.





Figure 3: Netgear Nighthawk X8-R8500 Tri-Band WiFi Gigabit Router AC5300.

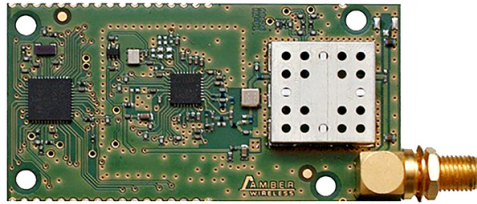


Figure 4: Netgear Nighthawk AC1900 EX7000 range extender with rechargeable battery.

Optionally, the range can be extended or wall penetration improved, by using signal amplifiers, and various antennas with additional signal gain over +30 dBm exceeding overall transmit signal over 70 dBm.

#### 3.4.4 Cable Connection

To bridge larger distances between the operator station and the operator-side wireless access point, 7mm Industrial grade Optical Fiber (LWL) 50/125um OM3 - 300 m, tensile strength 1600 N, and Gigabit Optical-Ethernet converters have been selected, as shown in Fig. 6.



**Specifications**

Performance	Range*	Up to 20 km
	RF data rate	Up to 50 kbps (Gross)
	Interface data rate	Up to 115,2 kbps (UART)
	Output power	Up to 27 dBm
	RF sensitivity	Up to -123 dBm
General	Power supply (VCC)	2,0 - 3,6 V
	Max. Voltage at any Pin	- 0.3V - VCC+0.3 V
	Power consumption	- TX: typ. 500 mA @ 500 mW - RX: typ. 48 mA
	Dimensions	33,5 x 76 x 14,5 mm
	Operating temperature	-40 bis +85 °C
	Weight	Approx. 14 g
RF technology	Antenna	External antenna port (50 Ω)
	Addressing	Up to 255 nodes
	Frequency range	869,4 – 869,65 MHz
	Channel spacing	25 kHz
	Modulation	2-GFSK
Supported topologies	Star, Peer-to-Peer	
Conformity	Europe	EN 300 220, EN 301 489, EN 60950, EN 62479

\* Range stated as assumes line-of-sight. Actual range may vary depending on antenna choice, board integration and environment.



Figure 5: Amber Wireless link AMB8636.

**3.5 Test Setup**

The data link connects two notebook computer running the GNU/Linux Operating System, which represent the robot and the operator station, respectively. Software tools such as: ping, iperf, netperf, nuttcp, and qperf are used to measure performance.

The computers are connected via a LAN cable to the routers/transceivers. Both are located



Figure 6: Optical Fiber cable (LWL).

initially in close line-of-sight (LOS) position. Then the tests are performed sending data back and forward, measuring time delays and integrity of data. Next, the LOS distance was extended step by step, while performance and integrity of data communication has been measured.

For difficult non-line-of-sight (NLOS) situations, a Netgear repeater was used to extend the wireless range. Additional range extender(s) will be used when necessary.

As the high-availability link can realize up to 20 km LOS connections, the focus was on how far walls and obstacles can be penetrated by the radio signal, ensuring reliable connection between the two Amber transceivers. The transceivers were configured for maximum allowable transmit power at 27dBm permitted by the German/EU law, equipped with a standard omnidirectional antenna (Fig. 5). The boards were connected directly to the system via USB cable, emulating a serial interface. The system was configured to connect the two devices via a serial line modem. Once connected, they were visible in the system just like an Ethernet device, where its TCP/IP parameters, routing and other options could be configured and used.

The test and measurements were performed at a remote location, far from cities. The only detectable signal was from our own test equipment. The transmitter was located just outside the house visible on map, and the receiver was a car, with the antenna/router mounted on its roof.

## 4 Results

### 4.1 Initial Experiments with Netgear AC1900 Routers on Momaro Robot

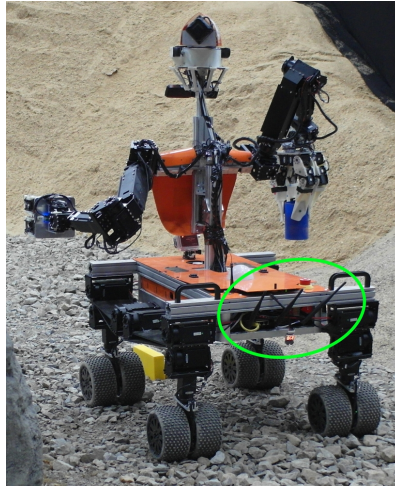


Figure 7: Mobile manipulation robot Momaro with Netgear AC1900 access point.

We performed an initial set of experiments with UBO's mobile manipulation robot Momaro. Its Netgear AC1900 access points have been provided by DARPA for its 2015 Robotics Challenge. Fig. 7 shows that the robot access point is mounted in the rear part of the base. The tests were performed inside the Bonn computer science building. The results shown in Fig. 8 indicate that a low round-trip time of approximately 1 ms can be achieved up to a distance of 60 m, overcoming one firewall in the building. With a repeater, a second firewall can be penetrated, at the cost of adding a bit of latency.

Used command: ping -s 712 -c 10000 -i 0.001 -W 1 momaro

Transmission frequency: 1000 Hz

Packet size: 720 bytes Payload size: 704 bytes (e.g. 44 float values)

Without repeater:

Distance	Walls	Packet loss	Mean RTT	Std. Dev.	Max RTT
1.0 m	0	0.00%	0.716 ms	0.182 ms	3.723 ms
4.0 m	0	0.00%	0.781 ms	0.261 ms	4.231 ms
10.0 m	0	0.00%	0.746 ms	0.267 ms	4.423 ms
50.0 m	0	0.00%	0.936 ms	0.419 ms	7.069 ms
60.0 m	1	0.00%	1.245 ms	0.997 ms	48.268 ms
70.0 m	2	95.00%	106.046 ms	398.652 ms	2020.138 ms

With repeater:

Distance	Walls	Packet loss	Mean RTT	Std. Dev.	Max RTT
4.0 m	0	0.00%	1.095 ms	0.533 ms	13.330 ms
14.0 m	2	0.00%	1.122 ms	0.437 ms	8.601 ms
70.0 m	2	0.06%	3.891 ms	27.591 ms	607.874 ms

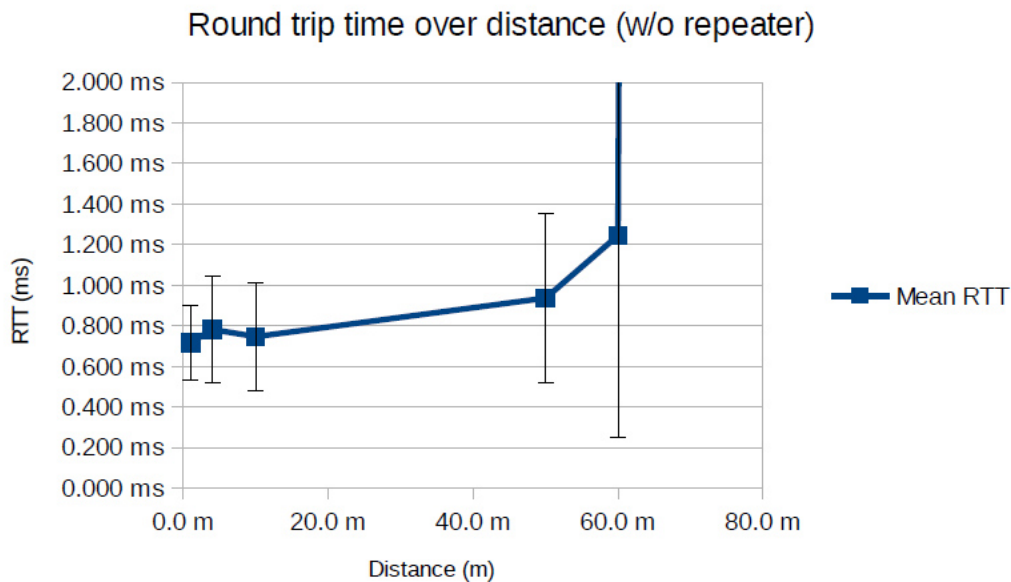


Figure 8: Initial experiments with robot Momaro.

## 4.2 Netgear Nighthawk X8-R8500

The main set of experiments was performed with the pair of the selected Netgear Nighthawk X8-R8500 access points. The following tables list the performed command as well as the resulting measurements.

```
qperf -v IP ADDR tcp bw tcp lat udp bw udp lat
```

Meters		TCP bw	TCP lat	UDP bw	UDP lat
		MB/s	ms	MB/s	ms
2	A	25,8	1,75	18,4	1,77
2	B	15	1,73	17,6	1,78
10		31,1	1,73	39,8	1,7
40		36	1,73	35	1,73
90		21,9	1,75	17,6	1,75
120		20,3	1,73	19,1	1,74
130		22,4	1,77	18,1	1,77
160	A	20,7	1,76	19	1,77
160	B	21,7	1,76	18,2	1,77
200		20	1,78	16,7	1,93
240		15,3	1,8	15,9	1,83
310		18,1	1,8	18,5	1,82
350		1,47	3,46	0,36	2,59
370		19	1,77	12,8	1,78
380		0,03	3,97		
408		0,2	2,54		
410		0,82	3,12	3,31	2,79
435		0,23	2,94		
450		3,8	2,38	1,61	2,39

```
ping -s 712 -c 10000 -i 0.001 -W 1 IPaddr
```

```
712 byte ICMP packet / no. of packets=10000 / sent at rate of 1000Hz / timeout 1s
```

Meters		packet loss	RTTmin	RTTavg	RTTmax	MDEV
		percent	ms	ms	ms	ms
2	A	0	3,093	3,775	8,757	0,560
2	B	0	2,490	3,850	14,630	0,611
10		0	2,610	3,620	7,968	0,459
40		0	2,837	3,738	21,488	0,566
90		0	2,550	3,797	8,210	0,565
120		0	2,297	3,799	15,870	0,585
130		0	2,338	4,257	167,835	6,923
160	A	0	2,612	3,815	7,756	0,567
160	B	0	2,948	3,875	26,138	0,819
200		0	2,671	3,852	7,965	0,577
240		0	2,488	4,017	57,699	1,638
310		0	2,449	3,998	17,383	0,736
350		0	3,181	5,455	133,025	6,600
370		0	2,681	3,911	13,146	0,610
380		63	1242,781	6408,615	11029,640	2190,065
408		75	1994,430	5582,631	9461,407	1658,051
410		7	3,241	2061,924	6747,011	1759,874
435		13	4,307	1355,101	4631,445	1358,111
450		14	3,105	2619,155	7589,769	2493,355

64 byte ICMP packet / Adaptive mode (send next packet as soon as it receives answer) / calculate round trip time / no. Of packets=5

ping -A -c 5 -r -U IPADDR

Meters		packetloss	RTTmin	RTTavg	RTTmax	MDEV
		percent	ms	ms	ms	ms
2	A	0	2,215	2,986	3,806	0,538
2	B	0	3,114	3,728	4,893	0,610
10		0	3,027	3,605	5,769	1,085
40		0	3,171	3,685	4,316	0,378
90		0	3,054	3,139	3,229	0,085
120		0	3,119	3,357	3,685	0,251
130		0	3,079	3,476	4,085	0,391
160	A	0	3,024	3,325	3,770	0,313
160	B	0	3,050	3,419	3,720	0,300
200		0	3,053	3,341	3,783	0,308
240		0	3,027	3,318	4,276	0,487
310		0	3,054	3,088	3,164	0,072
350		0	3,083	3,247	3,583	0,212
370		0	3,119	3,490	4,213	0,447
380		80	13,973	13,973	13,973	0,000
408		20	6,267	9,647	12,554	2,237
410		0	5,556	6,856	8,171	0,914
435		0	6,532	8,041	8,864	0,862
450		0	4,106	7,021	12,674	3,071



ping -A -c 5 -r -U -s 1000 IP ADDR  
 1000 byte ICMP packet / Adaptive mode (send next packet as soon as it receives answer) / calculate  
 round trip time / no. Of packets=5

Meters		packet loss percent	RTTmin ms	RTTavg ms	RTTmax ms	MDEV ms
2	A	0	3,219	3,622	4,291	0,360
2	B	0	3,536	3,809	4,145	0,218
10		0	3,486	3,592	3,711	0,075
40		0	3,543	4,389	5,707	0,785
90		0	3,500	3,847	4,256	0,286
120		0	3,546	3,870	4,158	0,249
130		0	3,504	4,354	6,311	1,110
160	A	0	3,658	3,927	4,136	0,188
160	B	0	3,584	4,299	5,873	0,833
200		0	3,571	4,224	5,068	0,488
240		0	3,508	3,868	4,186	0,281
310		0	3,744	4,431	5,509	0,780
350		0	3,684	4,078	4,890	0,453
370		0	3,617	3,691	3,783	0,105
380		0	27,768	44,785	11,754	33,485
408		0	20,255	34,842	48,892	10,304
410		0	12,149	12,880	13,483	0,479
435		0	9,966	11,066	12,753	1,047
450		0	3,992	5,815	7,322	1,174

```
netperf -L SOURCE IP -H DEST IP -p 5000 -t TCP STREAM  
iperf -c IP ADDR
```

Meters		netperf Throughput Mbits/s	iperf TCPbw Mbits/s	iperf UDPbw Mbits/s
2	A	210,00	199	268
2	B	152,00	133	156
10		333,00	322	357
40		300,00	250	353
90		180,00	197	212
120		172,00	158	176
130		208,00	208	211
160	A	184,00	191	226
160	B	161,00	171	180
200		150,00	158	169
240		138,00	124	169
310		131,00	140	134
350		32,00	12	27
370		147,00	127	107
380		0,74	1,77	
408		0,95	0,69	
410		18,22	45	64
435		2,72	2,6	2,4
450		26,90	29,5	23,5

nuttcp -p5010 -P5011 IPADDR (A-TCP/RTT columns)  
 nuttcp -Ri1000m -p5010 -P5011 -i 1 -T5 IPADDR (B-TCP/RTT columns)  
 nuttcp -u -Ri1000m -p5010 -P5011 -i 1 -T5 IPADDR (UDP column)

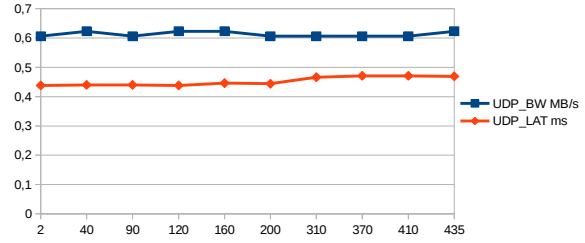
Meters		A-TCP Mbit/s	A-RTT ms	B-TCP Mbit/s	B-RTT ms	UDP Mbit/s
2	A	215	4,38	201	4,35	252
2	B	155	4,31	144	4,43	198
10		316	3,25	315	3,88	383
40		293	4,34	291	4,23	359
90		185	4,54	189	3,78	219
120		170	4,3	179	4,41	163
130		214	4,36	217	4,45	260
160	A	190	4,37	210	4,28	221
160	B	164	3,86	170	6,08	195
200		153	4,43	150	4,28	181
240		124	4,55	132	4,40	148
310		144	4,65	122	3,96	174
350		10	6,78	28	4,99	29
370		75	4,89	119	4,53	160
380		0,61	4,12	0,55	9,31	0,77
408		0,55	6,66	0,77	6,23	0,79
410		57	6,01	7,3	9,08	5,18
435		2,24	3,46	2	7,99	1,32
450		25	6,45	11	6,42	0,81

An additional set of experiments was performed with the pair of Netgear Nighthawk X8-R8500 with modified DD-WRT firmware, as summarized on the next page.

**Netgear AC5300 routers, 2'nd measurements with modified DD-WRT firmware:**

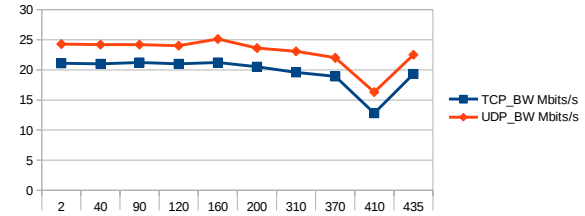
qperf -v IP\_ADDR tcp\_bw tcp\_lat udp\_bw udp\_lat

Meters	TCP_BW MB/s	TCP_LAT ms	UDP_BW MB/s	UDP_LAT ms
2	2,56	0,442	0,606	0,438
40	2,59	0,441	0,623	0,44
90	2,59	0,441	0,606	0,44
120	2,56	0,441	0,623	0,438
160	2,56	0,442	0,623	0,446
200	2,52	0,442	0,606	0,444
310	2,06	0,456	0,606	0,466
370	2,33	0,457	0,606	0,471
410	1,57	0,466	0,606	0,471
435	2,29	0,471	0,623	0,469



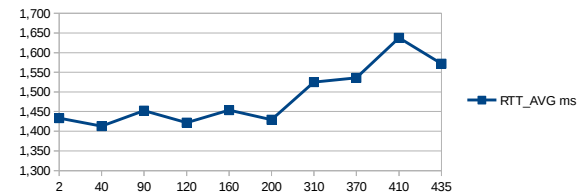
iperf -c IP\_ADDR

Meters	TCP_BW Mbits/s	UDP_BW Mbits/s
2	21,1	24,3
40	21	24,2
90	21,2	24,2
120	21	24
160	21,2	25,1
200	20,5	23,6
310	19,6	23,1
370	18,9	22
410	12,8	16,3
435	19,3	22,5



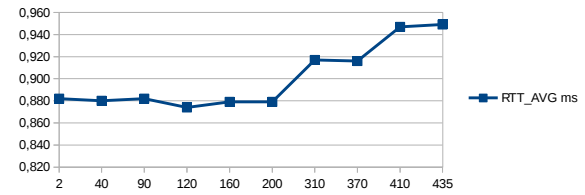
ping -s 712 -c 10000 -i 0.001 -W 1 IP\_ADDR 712 byte ICMP packet / no. of packets=10000 / sent at rate of 1000Hz / timeout 1s

Meters	packet loss percent	RTT_MIN ms	RTT_AVG ms	RTT_MAX ms	MDEV ms
2	0	1,132	1,433	39,442	0,690
40	0	1,169	1,413	4,563	0,208
90	0	1,144	1,452	43,934	0,934
120	0	1,264	1,422	8,839	0,257
160	0	1,154	1,454	48,151	0,938
200	0	1,152	1,429	10,535	0,239
310	0	1,209	1,525	50,473	1,027
370	0	1,144	1,536	49,426	1,104
410	0	1,319	1,638	40,989	0,853
435	0	1,258	1,572	57,300	1,146



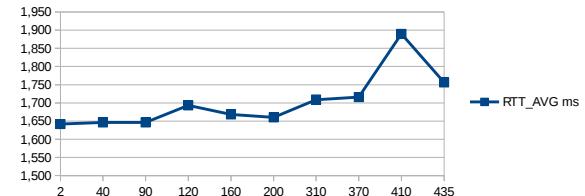
ping -A -c 5 -r -U IP\_ADDR 64 byte ICMP packet / Adaptive mode / calculate round trip time / no. Of packets=5

Meters	packet loss percent	RTT_MIN ms	RTT_AVG ms	RTT_MAX ms	MDEV ms
2	0	0,713	0,882	2,623	0,169
40	0	0,738	0,880	2,658	0,183
90	0	0,695	0,882	2,749	0,173
120	0	0,723	0,874	2,680	0,165
160	0	0,724	0,879	2,724	0,177
200	0	0,690	0,879	2,744	0,172
310	0	0,739	0,917	6,533	0,281
370	0	0,742	0,916	2,813	0,224
410	0	0,750	0,947	4,169	0,294
435	0	0,707	0,949	5,848	0,396



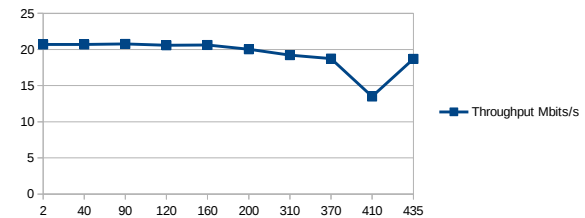
ping -A -c 5 -r -U -s 1000 IP\_ADDR 1000 byte ICMP packet / Adaptive mode / calculate round trip time / no. Of packets=5

Meters	packet loss percent	RTT_MIN ms	RTT_AVG ms	RTT_MAX ms	MDEV ms
2	0	1,449	1,642	3,506	0,229
40	0	1,437	1,646	3,595	0,239
90	0	1,444	1,646	3,550	0,209
120	0	1,429	1,693	3,500	0,227
160	0	1,455	1,668	3,484	0,225
200	0	1,450	1,660	3,718	0,239
310	0	1,449	1,709	4,579	0,322
370	0	1,439	1,716	4,873	0,314
410	0	1,554	1,890	9,820	0,462
435	0	1,446	1,756	10,134	0,540



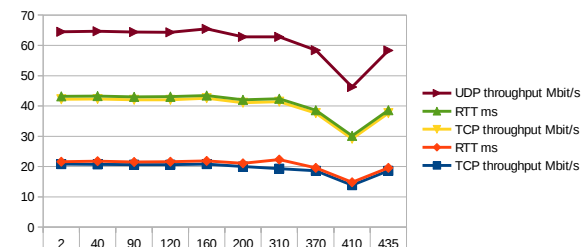
netperf -L SOURCE\_IP -H DEST\_IP -p 5000 -t TCP\_STREAM

Meters	Throughput Mbits/s
2	20,71
40	20,69
90	20,77
120	20,57
160	20,63
200	20,01
310	19,23
370	18,74
410	13,50
435	18,70



nuttcp -p5010 -P5011 IP\_ADDR  
nuttcp -Ri1000m -p5010 -P5011 -i 1 -T5 IP\_ADDR  
nuttcp -u -Ri1000m -p5010 -P5011 -i 1 -T5 IP\_ADDR

Meters	TCP throughput Mbit/s	RTT ms	TCP throughput Mbit/s	RTT ms	UDP throughput Mbit/s
2	20,76	0,83	20,62	0,96	21,31
40	20,72	1,04	20,58	0,92	21,39
90	20,6	0,91	20,53	1,00	21,33
120	20,62	0,93	20,48	1,02	21,26
160	20,76	1,14	20,63	0,91	22,04
200	20,01	1,05	19,97	0,97	20,85
310	19,26	3,06	19,08	0,96	20,49
370	18,59	1	18,02	1,02	19,76
410	13,86	0,94	14,38	0,96	16,1
435	18,55	1	18,11	0,95	19,7



### 4.3 High Availability Link (Amber wireless)

ping -s 1 -c 500 -A -W 1 -i 0.1 IP ADDR

9 byte ICMP packet / Adaptive mode (send next packet as soon as it receives answer) / no. Of packets=500 - Latency time not available at this packet size.

Meters	packet loss percent
2	0
40	0
90	9
120	0
160	0
200	0
310	25
370	3
410	2
435	13
635	87
785	86
830	34
945	68

ping -s 16 -c 500 -A -W 1 -i 0.1 IP ADDR

24 byte ICMP packet / Adaptive mode (send next packet as soon as it receives answer) / no. Of packets=500 / 1 second timeout

Meters	packet loss percent	RTTmin ms	RTTavg ms	RTTmax ms	MDEV ms
2	0	71	73	75	0,810
40	0	71	73	76	0,866
90	20	71	73	75	0,823
120	0	71	73	96	2,233
160	0	71	73	142	3,658
200	0	71	73	98	2,438
310	30	71	73	76	0,828
370	9	71	73	75	0,726
410	5	71	73	76	0,861
435	33	71	73	76	0,880
635	96	71	73	75	0,918
785	84	71	73	75	0,847
830	53	71	73	76	0,872
945	80	72	74	76	0,809

ping -s 110 -c 200 -A -W 1 -i IP ADDR

118 byte ICMP packet / Adaptive mode (send next packet as soon as it receives answer) / no.  
Of packets=200 / 1 second timeout

Meters	packet loss percent	RTTmin ms	RTTavg ms	RTTmax ms	MDEV ms
2	0	151	153	155	0,826
40	0	151	153	155	0,929
90	32	151	153	155	0,774
120	0	151	153	156	0,984
160	4	151	153	181	3,372
200	0	151	153	155	0,890
310	57	151	153	154	0,907
370	24	151	153	155	0,793
410	15	151	153	155	0,772
435	62	150	153	155	0,872
830	67	151	152	154	0,856

ping -s 246 -c 100 -A -W 1 IP ADDR

254 byte ICMP packet / Adaptive mode (send next packet as soon as it receives answer) / no.  
Of packets=100 / 1 second timeout

Meters	packet loss percent	RTTmin ms	RTTavg ms	RTTmax ms	MDEV ms
2	0	255	257	259	0,711
40	0	254	256	259	0,922
90	90	255	256	257	0,876
120	1	255	256	258	0,801
160	17	255	257	259	0,730
200	6	255	257	258	0,874
310	97	257	258	258	0,716
370	63	254	276	977	116,862
410	74	256	257	259	1,019
435	98	255	257	258	1,178
830	94	255	256	257	0,768

ping -s 504 -c 50 -A -W 1 -i IP ADDR

512 byte ICMP packet / Adaptive mode (send next packet as soon as it receives answer) / no.  
Of packets=50 / 1 second timeout

Meters	packet loss percent	RTTmin ms	RTTavg ms	RTTmin ms	MDEV ms
2	0	ms	ms	ms	ms
40	0	454	456	458	1,202
90	98	454	456	458	0,856
120	4	467	467	467	0,000
160	32	454	456	458	0,906
200	2	454	456	458	0,971
370	82	454	456	468	2,006
410	74	454	461	481	8,443
435	100	454	461	487	11,133

ping -s 1016 -c 25 -A -W 1 -i IP ADDR

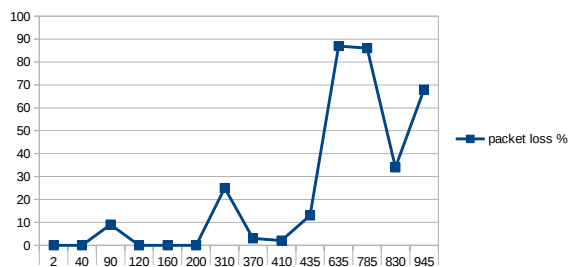
1024 byte ICMP packet / Adaptive mode (send next packet as soon as it receives answer) / no.  
Of packets=25 / 1 second timeout

Meters	packet loss percent	RTTmin ms	RTTavg ms	RTTmax ms	MDEV ms
2	0	838	843	846	1,919
40	0	840	842	844	1,630
90	100				
120	12	840	842	843	1,367
160	40	841	844	852	3,319
200	24	840	842	843	1,348
370	96	879	879	879	0,000
410	96	843	843	843	0,000
435	100				

### Amber AMB8636 transceiver measurements:

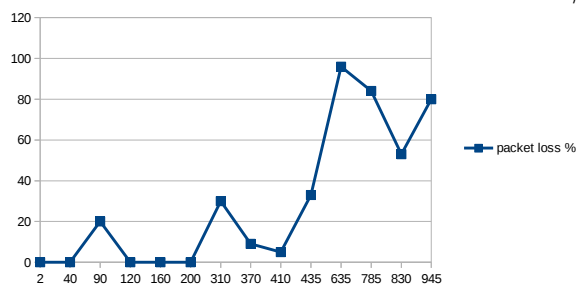
Ping 9 byte packet

Meters	packet loss %
2	0
40	0
90	9
120	0
160	0
200	0
310	25
370	3
410	2
435	13
635	87
785	86
830	34
945	68



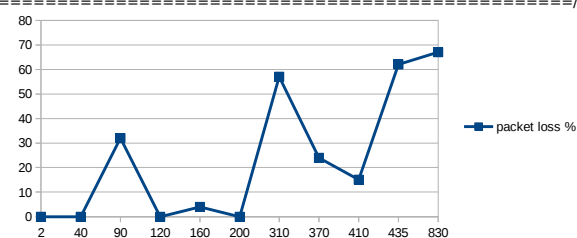
Ping 24 byte packet

Meters	packet loss %	RTT_MIN ms	RTT_AVG ms	RTT_MAX ms	MDEV ms
2	0	71	73	75	0,810
40	0	71	73	76	0,866
90	20	71	73	75	0,823
120	0	71	73	96	2,233
160	0	71	73	142	3,658
200	0	71	73	98	2,438
310	30	71	73	76	0,828
370	9	71	73	75	0,726
410	5	71	73	76	0,861
435	33	71	73	76	0,880
635	96	71	73	75	0,918
785	84	71	73	75	0,847
830	53	71	73	76	0,872
945	80	72	74	76	0,809



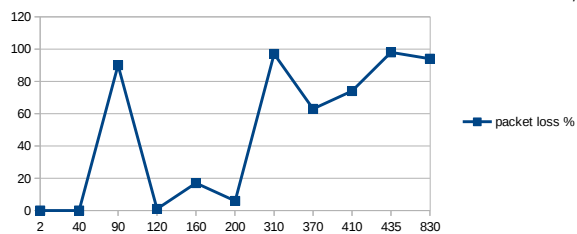
Ping 118 byte packet

Meters	packet loss %	RTT_MIN ms	RTT_AVG ms	RTT_MAX ms	MDEV ms
2	0	151	153	155	0,826
40	0	151	153	155	0,929
90	32	151	153	155	0,774
120	0	151	153	156	0,984
160	4	151	153	181	3,372
200	0	151	153	155	0,890
310	57	151	153	154	0,907
370	24	151	153	155	0,793
410	15	151	153	155	0,772
435	62	150	153	155	0,872
830	67	151	152	154	0,856



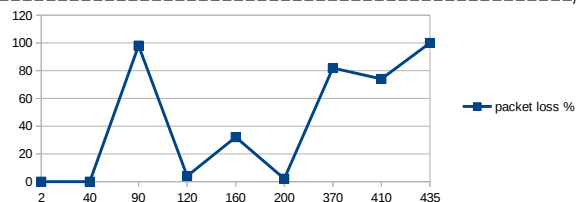
Ping 254 byte packet

Meters	packet loss %	RTT_MIN ms	RTT_AVG ms	RTT_MAX ms	MDEV ms
2	0	255	257	259	0,711
40	0	254	256	259	0,922
90	90	255	256	257	0,876
120	1	255	256	258	0,801
160	17	255	257	259	0,730
200	6	255	257	258	0,874
310	97	257	258	258	0,716
370	63	254	276	977	116,862
410	74	256	257	259	1,019
435	98	255	257	258	1,178
830	94	255	256	257	0,768



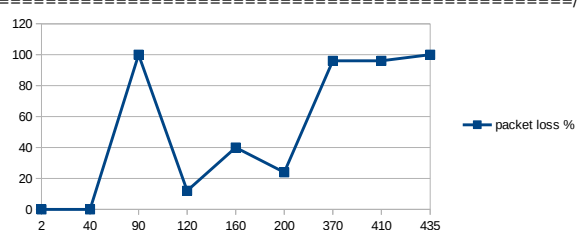
Ping 512 byte packet

Meters	packet loss %	RTT_MIN ms	RTT_AVG ms	RTT_MAX ms	MDEV ms
2	0	454	456	458	1,202
40	0	454	456	458	0,856
90	98	454	456	458	0,856
120	4	467	467	467	0,000
160	32	454	456	458	0,906
200	2	454	456	458	0,971
370	82	454	456	468	2,006
410	74	454	461	481	8,443
435	100	454	461	487	11,133



Ping 1024 byte packet

Meters	packet loss %	RTT_MIN ms	RTT_AVG ms	RTT_MAX ms	MDEV ms
2	0	838	843	846	1,919
40	0	840	842	844	1,630
90	100				
120	12	840	842	843	1,367
160	40	841	844	852	3,319
200	24	840	842	843	1,348
370	96	879	879	879	0,000
410	96	843	843	843	0,000
435	100				





## 5 Communication Link Management of Momaro

### 5.1 UBO ROS-Multimaster Communication

Since its participation in the DLR SpaceBot Cup 2013 [6], UBO develops a robust software module (`nimbro_network`) for communication between multiple ROS masters over unreliable and high-latency networks, which has been released <sup>1</sup> under BSD-3 license. It provides transport of ROS topics and services over TCP and UDP protocols. Since it does not need any configuration/discovery handshake, it is ideally suited for situations where the connection drops and recovers unexpectedly. Compared to custom-engineered protocols for specific data types or competitions, this module makes fast development possible, as topics can be added on-the-fly in configuration files, without developing specific transport protocols.

Several specific transports or compression exist, such as a ROS log transport, `tf` snapshotting or H264 video stream compression.

For large messages, a transparent BZip2 compression can be enabled. Automatic rate limiting with configurable upper and lower bounds ensures that bandwidth limits are met.

`nimbro_network` also allows forward error correction (FEC), i.e. augmenting the sent packets with additional packets allowing content recovery from arbitrary subsets of sufficient size of transmitted packets. Depending on the message size, a Reed-Solomon codec [2] or a LDPC-Staircase codec [3] is chosen.

Note that in principle ROS offers built-in network transparency. Since this functionality heavily relies on the TCP protocol for topic discovery and subscription, even when the “UD-PROS” transport is chosen, this is unsuitable for unreliable and high-latency networks.

The developed ROS-multimaster communication software has been used for the DARPA Robotics Challenge (DRC) and the DLR SpaceBot Cup 2015 for management of the communication link.

### 5.2 DARPA Robotics Challenge Communication Architecture

One constraint during the DRC was the limited communication between the operator station and the robot, which was enforced to simulate degenerated communication as may occur in a real-world mission. The uplink from the operator station to the robot was limited to 9600 bit/s at all times. The downlink from the robot to the operator station was limited to 300 Mbit/s outside of the building during the driving tasks, the door task, and the stairs task. Inside the building (defined by the door thresholds), the downlink was limited to 9600 bit/s, interleaved with one second long bursts of 300 Mbit/s bandwidth. These burst became more frequent during the run and the blackouts vanished completely after 45 minutes into the run. As usual, the wireless communication link does not guarantee packet delivery, so communication systems had to deal with packet loss.

To cope with this degraded communication, sensor information cannot be transferred unselected and uncompressed. The main idea of the developed communication system is to transfer stationary information about the environment over the high-latency high bandwidth channel, while we use the low-latency low-bandwidth channel to transfer frequently changing data. Both are then combined on the operator station to render immersive 3D visualizations with low latency for the operators.

The UBO DRC communication architecture is shown in Fig. 9. The main topology was formed by the DARPA requirements, which placed the Degraded Communications Emulator

<sup>1</sup>[https://github.com/AIS-Bonn/nimbro\\_network](https://github.com/AIS-Bonn/nimbro_network)

(DCE) between the operator crew and the robotic system. To allow buffering and relaying of data over the high-bandwidth link, we made use of the option to include a separate field computer, which is connected via Ethernet to the DCE on the robot side. The key motivation here is that the wireless link to the robot is unreliable, but unlimited in bandwidth, while the link over the DCE is reliable, but limited in bandwidth. Placing the field computer directly after the DCE allows exploitation of the characteristics of both links.

On the operator side of the DCE, the operator station is the central unit. Since the robot operator crew consists of more than one person, we have the option to connect multiple specialists’ notebooks to the operator station. Finally, the computer running our telemanipulation interface is also directly connected to the operator station. Since we use the ROS middleware for all software components, separate ROS masters run on the robot, the field computer, and the operator station. The communication between these masters can be split into three channels, which will be explained below.

### 5.2.1 Low-bandwidth Channel

The low-bandwidth channel is a bidirectional channel between the operator station and robot (blue (1) in Fig. 9). It uses the low-bandwidth link of the DCE and is therefore always available. Since the bandwidth is very limited, we do most compression on the field computer, where we can be sure that packets sent to the operator station are not dropped, which would waste bandwidth.

Since the low-bandwidth link over the DCE was the main live telemetry source for the operator crew, we spent considerable effort on compressing the data sent over this link in order to maximize the amount of information the system provides. The transmitter running on the field computer sends exactly one UDP packet per second. The bandwidth is thus easily controlled by limiting the UDP payload size. Since the amount of data is much less in the other direction, the transmitter on the operator station sends operator commands with up to 5 Hz. Payload sizes in bits are given in Tab. 1.

For low-level compression of floating point numbers as well as 3D/4D vectors and quaternions, we developed a small helper library, which is freely available<sup>2</sup>. It employs techniques

<sup>2</sup>[https://github.com/AIS-Bonn/vector\\_compression](https://github.com/AIS-Bonn/vector_compression)

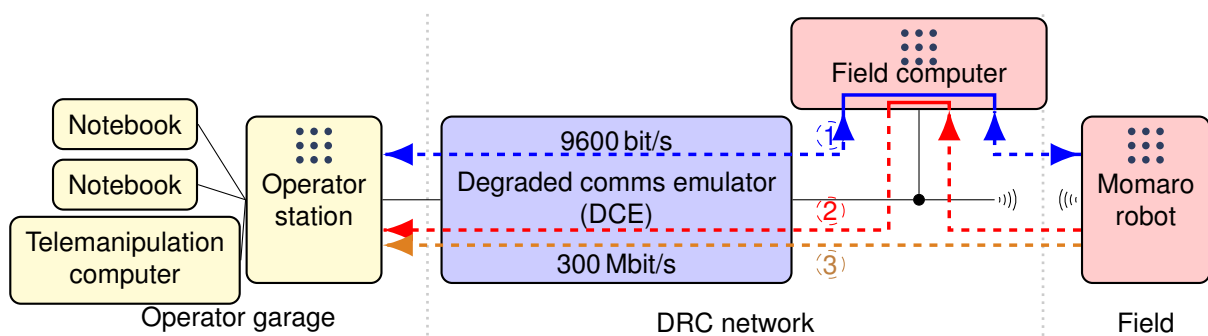


Figure 9: Communication architecture used at DARPA Robotics Challenge. Components in the vicinity of the operators are shown in yellow, DARPA-provided components in blue, components in the “field”-network in red. Solid black lines represent physical network connections. Dashed lines show the different channels, which stream data over the network (blue (1): low-bandwidth, red (2): bursts, brown (3): direct imagery). The ROS logo (⋮) indicates a ROS master.

Table 1: Average bit rates of topics transmitted over the low-bandwidth link.

Robot → Operator			Operator → Robot		
Channel/Topic	Rate	Avg. Bit/message	Channel/Topic	Rate	Avg. Bit/message
H.264 Camera image	1 Hz	6000	Arm control	5 Hz	96
Joint positions	1 Hz	736	Joystick command	5 Hz	56
Base status	1 Hz	472	Generic motion <sup>1</sup>	-	144
3D Contour points	1 Hz	250	Motion play request	-	80
Transforms	1 Hz	136			
Audio amplitude	1 Hz	8			
<b>Sum per 1s</b>		7602			<sup>2</sup>

Topics with rate of “-” are transmitted only on operator request.

<sup>1</sup> Generic transport for all kinds of keyframe motions. Here: one frame using Cartesian EEF pose.

<sup>2</sup> Summation is not applicable here, since the total bit rate depends heavily on operator action.

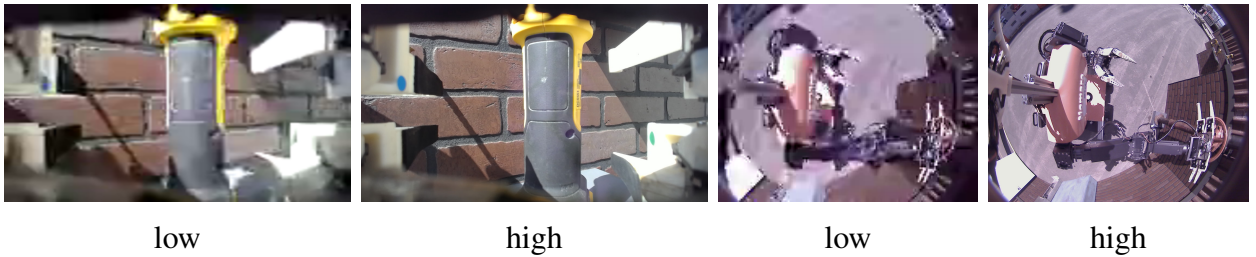


Figure 10: Comparison of webcam images over low- and high-bandwidth channels. The left two images were captured by the right hand camera, looking at the drill tool. The right two images show the overhead view while the robot is grasping the valve.

originally developed for compressing geometry data for transfers between CPU and GPU: Quaternions are compressed using the hemi-oct encoding [1], while 3D vectors are compressed using a face-centered cubic packing lattice. The lattice approach offers better average discretization error than naive methods which discretize each axis independently.

Since visual information is of crucial importance to human operators, we also transmit a low resolution video stream. As Momaro is equipped with a variety of cameras, an operator needs to select the camera whose output should be sent over the low-bandwidth link. The selection of the camera depends on the currently executed task and is also often changed during a task. Note that all camera images are also transmitted over the high-bandwidth link. The purpose of low-bandwidth imagery is merely to provide low-latency feedback to the operators. The selected camera image is compressed at the field PC using the H.264 codec. Before compression, the image is downsampled to  $160 \times 120$  pixels. Furthermore, we use the periodic intra refresh technique instead of dedicated keyframes, which allows specifying a hard packet size limit for each frame. While the compression definitely reduces details (see Fig. 10), the camera images still allow the operators to make fast decisions without waiting for the next high-bandwidth burst image.

Measured joint positions are discretized and transmitted as 16 bit integers (8 bit for the most distal joints in the kinematic tree). The joint positions are used for forward kinematics on the operator station to reconstruct poses of all robot parts. A small number of 3D rigid body transformations are sent over the network, including the current localization pose, odometry,

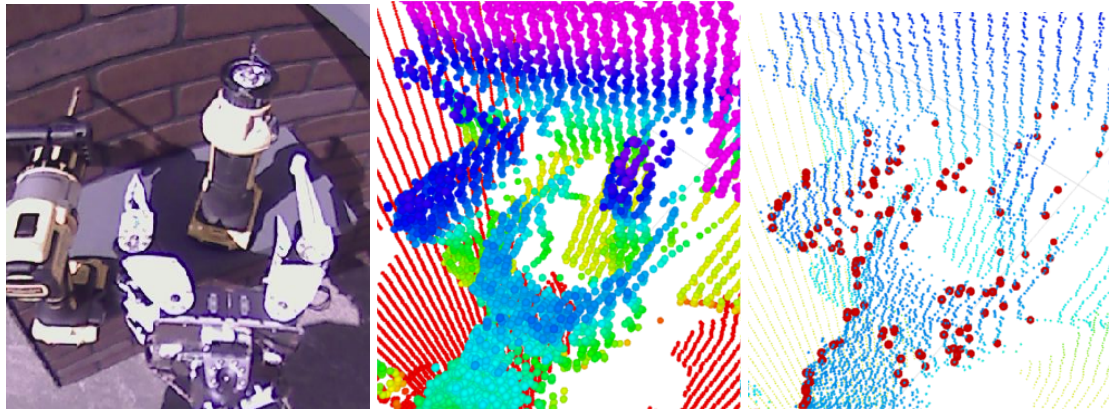


Figure 11: 3D contour points for a typical manipulation task (grasping the drill). Left: The overhead camera image. Middle: The raw laser scans (color encodes height from ground). Right: The resulting contour points (red).

and IMU information. The transforms are sent as 3D vector and quaternion pairs, compressed using the library mentioned above.

Up to 125 3D *contour points* are compressed and sent to the operator for display. These contour points are extracted from the laser scans and are meant to outline the contour of the end-effector and objects in its direct vicinity. By transmitting contour points over the low-bandwidth channel, the operator is provided with live sensory feedback from the laser scanner during a manipulation task. Fig. 11 shows the extracted contour points from a typical manipulation task. In order to minimize the number of points that are transmitted, we detect measurements on the manipulator and the close-by object by applying a combination of filters on the raw laser scans in a given scan window extracted from the last three 2D laser scans.

Telemetry from the robot base includes the current support polygon, estimated COM position, emergency stop status, infrared distance measurement from the hand, and the maximum servo temperature. Finally, the low-bandwidth link also includes the measured audio amplitude of the right hand camera microphone, which allows us to easily determine whether we succeeded in turning the drill on.

### 5.2.2 High-bandwidth Burst Channel

Since the connection between robot and field PC is always present, irrespective of whether the DCE communication window is currently open, we use this connection (red (2) in Fig.9) to transfer larger amounts of data to the field PC for buffering. The field computer sends a constant 200 MBit/s stream of the latest received ROS messages to the operator station. This maximizes the probability of receiving complete messages during the short high-bandwidth communication windows inside the building.

The transferred data includes:

- JPEG-compressed camera images from all seven cameras on board, plus two high-resolution cut-outs of the overhead camera showing the hands,
- compressed<sup>3</sup> point cloud from the ego-centric 3D map,
- ROS log messages,

<sup>3</sup>The point clouds were compressed using the PCL point cloud compression.

- servo diagnostics, and
- miscellaneous diagnostic ROS topics.

The 3D data received in the communication bursts is shown to the operators and transformed into a fixed frame using the low-latency transform information received over the low-bandwidth channel.

### 5.2.3 High-bandwidth Direct Imagery

During the outside tasks, the high-bandwidth link is always available. This opens the possibility of using streaming codecs for transmitting live imagery, which is not possible in the inside mode, where communication blackouts would corrupt the stream. Thus, an additional high-bandwidth channel using the *nimbros\_network* module carries H.264 encoded camera streams of the main overhead camera and the right hand camera. The streams use an increased frame rate of 5 Hz to allow low-latency operator control. These camera streams are used during the drive task for steering the car. The channel is shown in brown (3) in Fig. 9. Further details are provided by Schwarz et al. [5].

## 5.3 DLR SpaceBot Cup Communication

Communication between the ground station and a planetary rover is typically very limited – in particular it has high latency due to the speed of light and the large distances involved. The SpaceBot Cup addressed this limitation by imposing several constraints on the network link:

- Packets were delayed by 2 s in each direction, as expected to occur on a lunar mission,
- the uplink from the ground station to the robot could only be opened for 5 min at a time, and
- the 60 min schedule included two 4 min windows where uplink communication was not possible (e.g. due to planetary occlusions).

Furthermore, the UBO system uses a wireless data link inside the arena, which introduces packet loss.

The main idea of our communication system is to minimize latency by exploiting the different characteristics of the local wireless link inside the arena and the simulated inter-planetary network.

### 5.3.1 SpaceBot Communication Architecture

The UBO SpaceBot Cup communication architecture is shown in Fig. 12. The DLR-provided network emulator is the central element limiting all communication between robot and operator crew. To be able to exploit the different link characteristics, we place an additional field computer between the network emulator and the robot. Thus, it is connected to the network emulator via a reliable Ethernet connection, and communicates directly with the robot over WiFi. As the WiFi link is unreliable, but has low latency, while the network emulator link is reliable, but has high latency, this places the field computer in an ideal position to exploit both link characteristics.

As the network emulator allows communication only through a single port per direction, we use the Linux `tun` interface to create a network tunnel over two ports. For UDP tunneling,

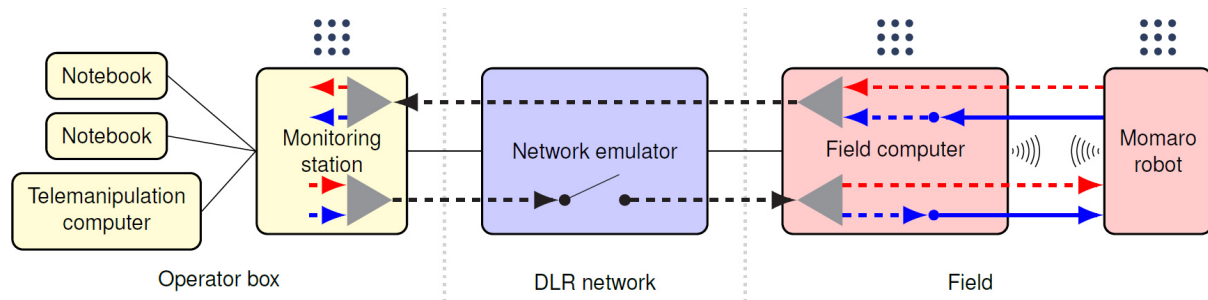


Figure 12: DLR SpaceBot Cup communication architecture. Components in the vicinity of the operators are shown in yellow, DLR-provided components in blue, components in the “arena”-network in red. Solid black lines represent physical network connections. Thick lines show the different channels, which stream data over the network (dotted: UDP, solid: TCP). The ROS logo (⋮) indicates a ROS master. UDP tunnel endpoints are designated by triangles. Streaming links (Sec. 5.3.2) are colored red, message links (Sec. 5.3.3) are shown in blue.

we adapted code from the `quicktun` project<sup>4</sup>. The tunnel wraps all packets in UDP packets, transmitted over the two designated ports. This allows us to use multiple communication channels without interference.

Since we use the ROS middleware for all software components, separate ROS masters run on the robot, the field computer, and the ground station. Multiple operator computers can be connected to the ROS master running on the ground station to provide additional views and means for intervention.

### 5.3.2 Streaming Data

Most high-bandwidth data from the robot is of *streaming* type. The key feature here is that lost messages do not lead to system failures, since new data will be immediately available, replacing the lost messages. In this particular application, it even would not make sense to repeat lost messages because of the high latencies involved. This includes

- video streams from the onboard cameras,
- transform information (TF),
- servo diagnostic information (e.g. temperatures),
- object detections, and
- other visualizations.

In the uplink direction, i.e. commands from the operator crew to the robot, this includes e.g. direct joystick commands.

Consequently, we use the `nimbro_network` UDP transport for streaming data (red in Fig. 12). The transport link between robot and field computer uses the FEC capability of `nimbro_network` with 25% additional recovery packets to compensate WiFi packet loss without introducing new latency.

<sup>4</sup><http://wiki.ucis.nl/QuickTun>

### 5.3.3 Message Data

Other data is of the *message* type, including

- Laser pointclouds,
- SLAM maps,
- SLAM transforms,
- ROS action status messages, and
- ROS service calls.

Here, a message loss might be costly (e.g. SLAM maps are only generated on every scanner rotation) or might even lead to system failure (e.g. loss of a ROS action state transition). Therefore, the TCP transport is used for this kind of messages over the WiFi link to eliminate the possibility of packet loss. The link over the network emulator is still implemented with the UDP protocol, since there is no packet loss here and the high latencies prohibit TCP handshakes. The message links are colored blue in Fig. 12. Details on the use of the described communication link are given by Schwarz et al. [4].

## 6 Conclusion

The test results show that with the factory router firmware in bridged mode, communication via the Netgear Nighthawk X8-R8500 is stable up to 300 m, with an average latency less than 5 ms in a Line-Of-Sight environment. The sustained bandwidth is over 100 Mbits/s.

We run a set of second measurements, with modified router firmware (open source DD-WRT), where we can see a considerable improvement in latency, reaching round-trip-times below 1 ms at the cost of throughput. The achievable communication range remained the same. Additional experiments in the Bonn computer science building confirmed that the communication range of the Netgear Nighthawk X8-R8500 is larger than the one of the older model present in Momaro. These routers utilize tri-band and beam-forming technology, to maximize their range, minimize latency through QoS packet prioritization, and may be used in bridge mode, eliminating one network layer.

The Amber wireless modules proved to allow for remote login to the system using encrypted secure shell at strong signal levels. As the signal decreases, the capability to transmit larger packet also decreases. Although we were able to communicate with the remote system, results are not fully satisfactory and need to be improved.

UBO developed ROS-based software, described in Sec. 5, for communication over unreliable and slow data links, which has been tested successfully at the DARPA Robotics Challenge and the DLR SpaceBot Cup. It is a matter of future work in WP7 – Integration, to adapt this software to the specific communication requirements of the CENTAURO robot teleoperation.

## References

- [1] Zina H Cigolle, Sam Donow, and Daniel Evangelakos. A survey of efficient representations for independent unit vectors. *Journal of Computer Graphics Techniques*, 3(2), 2014.
- [2] Jérôme Lacan, Vincent Roca, Jani Peltotalo, and Sami Peltotalo. Reed-solomon forward error correction (FEC) schemes. Technical report, 2009.
- [3] Vincent Roca, Christoph Neumann, and David Furodet. Low density parity check (LDPC) staircase and triangle forward error correction (fec) schemes. 2008.
- [4] Max Schwarz, Marius Beul, David Droeschel, Sebastian Sch $\tilde{A}$  $\frac{1}{4}$ ller, Arul Selvam Periyasamy, Christian Lenz, Michael Schreiber, and Sven Behnke. Supervised autonomy for exploration and mobile manipulation in rough terrain with a centaur-like robot. *Frontiers in Robotics and AI*, 3:57, 2016.
- [5] Max Schwarz, Tobias Rodehutsors, David Droeschel, Marius Beul, Michael Schreiber, Nikita Araslanov, Ivan Ivanov, Christian Lenz, Jan Razlaw, Sebastian Sch $\ddot{u}$ ller, David Schwarz, Angeliki Topalidou-Kyniazopoulou, and Sven Behnke. NimbRo Rescue: Solving disaster-response tasks with the mobile manipulation robot Momaro. *Journal of Field Robotics*, 2016.
- [6] J $\ddot{o}$ rg St $\ddot{u}$ ckler, Max Schwarz, Mark Schadler, Angeliki Topalidou-Kyniazopoulou, and Sven Behnke. NimbRo Explorer: Semiautonomous exploration and mobile manipulation in rough terrain. *Journal of Field Robotics (JFR)*, 2015.



# Annex: Outdoor Experiment Location Maps



2 Meters



10 Meters



40 Meters



90 Meters



120 Meters



130 Meters



160 Meters A



160 Meters B



200 Meters



240 Meters



310 Meters



350 Meters



370 Meters



380 Meters



408 Meters



410 Meters





435 Meters



450 Meters