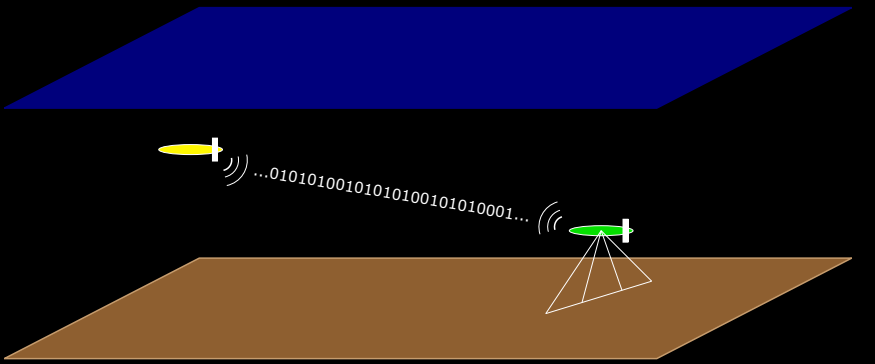
Goby-Acomms (v2)

(including pAcommsHandler)

Slow link networking for robotic marine platforms



Toby Schneider

MIT/WHOI Joint Program

Henrik Schmidt

MIT Laboratory for Autonomous Marine Sensing Systems

Plus contributions from Goby Developers Team

launchpad.net/~goby-dev

Problem

Given:

1. Many marine communications links (often acoustic) are extremely rate-limited (as low as 32 bytes / minute).
2. Marine robots are increasingly used collaboratively.
3. Research at sea demands robust & reconfigurable software.

How can we design a networking framework that satisfies these constraints?

Modular design with emphasis on efficiency instead of abstraction.

Review of Goby-Acomms v1

Version 1 of pAcommsHandler provides:

- A robust and field-tested MOOS interface to the Goby-Acomms acoustic networking libraries:
  - libdccl: highly compact object-based data marshalling using a configurable XML structure and preset algorithms.
  - libqueue: priority queuing of DCCL or CCL messages to maximize receipt of desired messages (overall and timeliness)
  - libamac: basic TDMA medium access control
  - libmodemdriver: abstracted interface to the WHOI Micro-Modem.

New since MOOS-DAWG ‘10

- Release series:
  - 0.0 (deprecated): basically what was presented last year.
  - 1.0 (stable): revamped internal messaging
    much improved MOOS ProcessConfig block
    support for WHOI ranging messages (REMUS LBL beacons, two-way ping, one-way time of flight)
  - 2.0 (experimental pre-release): new DCCL + other breaking changes (*expected released fall 2011*).

New since MOOS-DAWG ‘10

- Build / unit testing (CTest/CDash):
  http://gobysoft.org/CDash/

- Wiki:
  http://gobysoft.org/wiki

- Ubuntu .deb packages (back to 10.04 LTS “lucid lynx”):
  ppa:goby-dev/ppa (releases)
  ppa:tes/goby-daily (nightly build)

Version 2 Overview

libdccl: mostly complete rewrite:

- Google Protocol Buffers instead of XML
- Custom codecs
- Variable size messages
- Smaller header

libqueue: cleanup and support for DCCL v2.

libmodemdriver and libamac: No changes for 2.0 (all development happening in 1.0 series).

Numerous software quality assurance improvements.

DCCL v2: Protobuf

Protocol buffers are “a language-neutral, platform-neutral, extensible way of serializing structured data for use in communications protocols, data storage, and more.”[1]

Why Google Protocol Buffers (protobuf) instead of XML for DCCL?

- Compile-time (static) type checking
- Widely used messaging scheme
- Cleaner syntax
- Richer representation (embedded / repeated fields)

[1] http://code.google.com/apis/protocolbuffers/docs/overview.html

DCCL v2: format comparison

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<message_set>
  <message>
    <name>Example</name>
    <size>32</size>
    <id>1</id>
    <layout>
      <bool>
        <name>B</name>
      </bool>
      <enum>
        <name>E</name>
        <value>cat</value>
        <value>dog</value>
        <value>mouse</value>
      </enum>
      <string>
        <name>S</name>
        <max_length>4</max_length>
      </string>
      ...
    </layout>
  </message>
</message_set>
```

Protobuf

```
import "goby/protobuf/dccl_option_extensions.proto";
message Example {
  option (dccl.id) = 1;
  option (dccl.max_bytes) = 32;
  optional bool B = 4;
  optional EEnum E = 5;
  enum EEnum{
    cat = 0;
    dog = 1;
    mouse = 2;
  }
  optional string S = 6 [(dccl.max_length)=4];
}
```

DCCL v2: Custom codecs

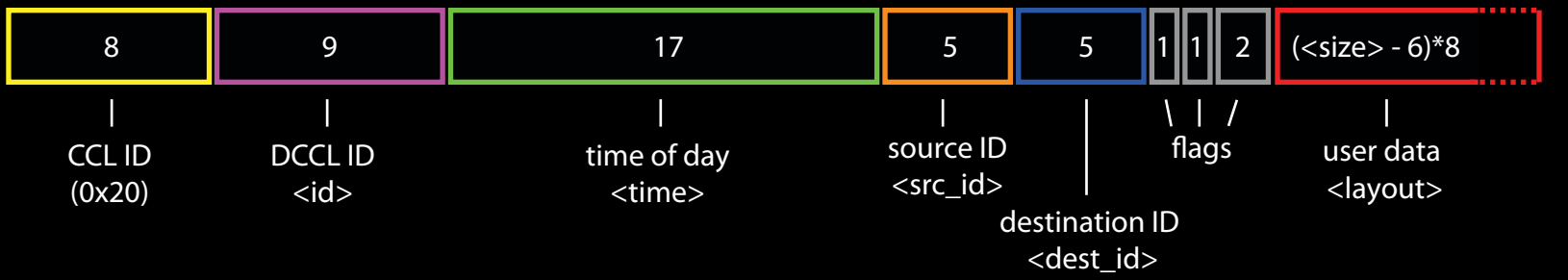
- Custom encoders/decoders can be defined for any primitive type or user-defined Protobuf message
- Defaults for primitive types (int, double, string, etc.) are same as DCCL v1.

Examples:

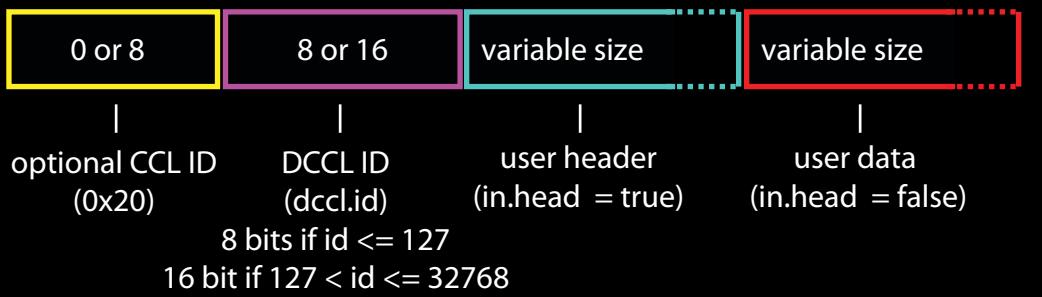
- Time-correlated physical sample (e.g. temperature) compression
- Shape function encoding (lossy), e.g. CTD profile
- Speciality image compression

DCCL v2: Smaller header

DCCL v1 header is a fixed 6 bytes:



DCCL v2 header is 1-3 bytes + user additions:



DCCL v2: Variable fields

Fields in DCCL v1 are fixed (always same bit-size).

DCCL v2 fields can be variable width but must have:

- fixed minimum size
- fixed maximum size

These bounds allow the message designer to still mandate a fixed upper bound on the DCCL message size.

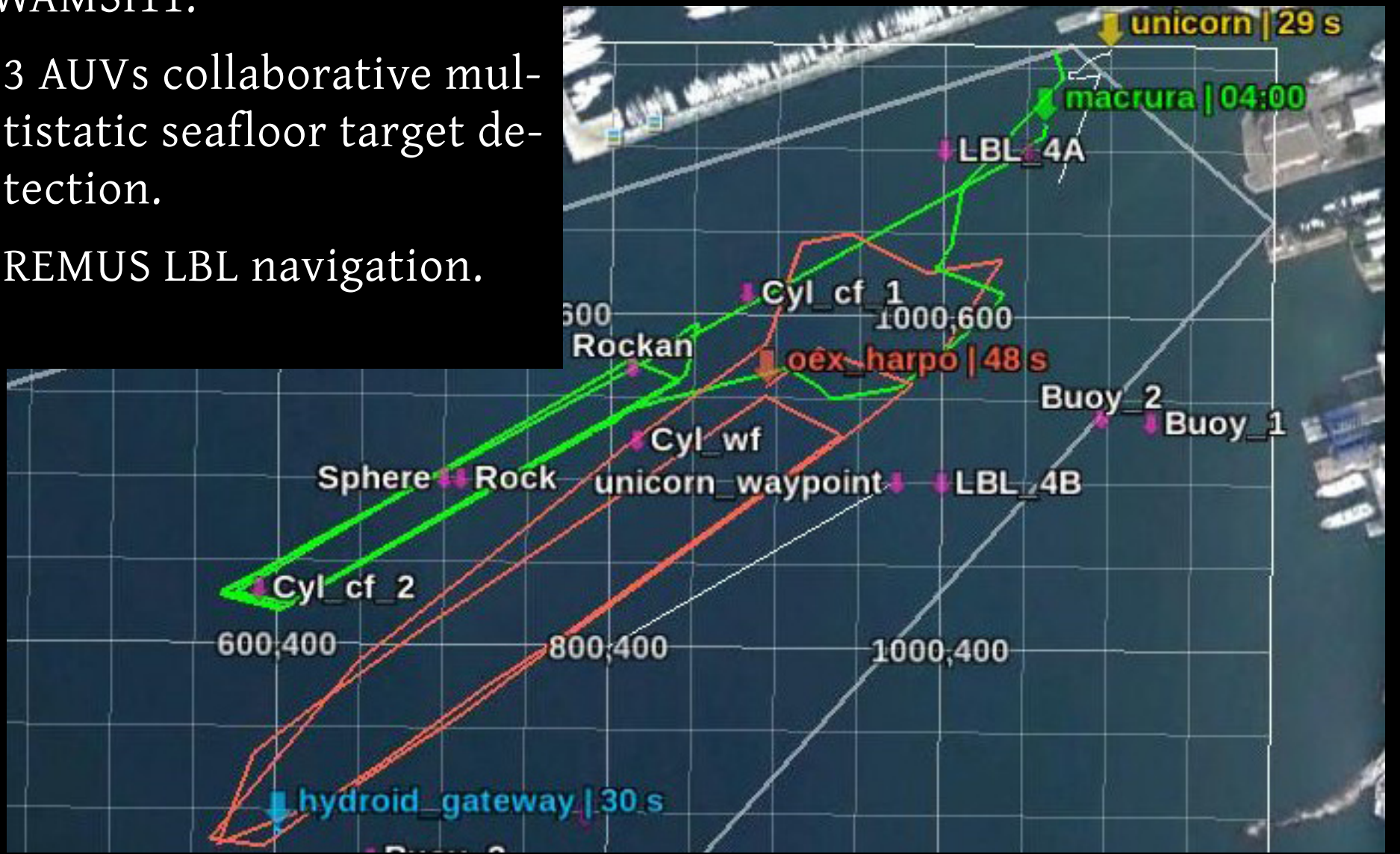
Example:

maximum size: N+1
minimum size: 1

Experiments: Case Studies

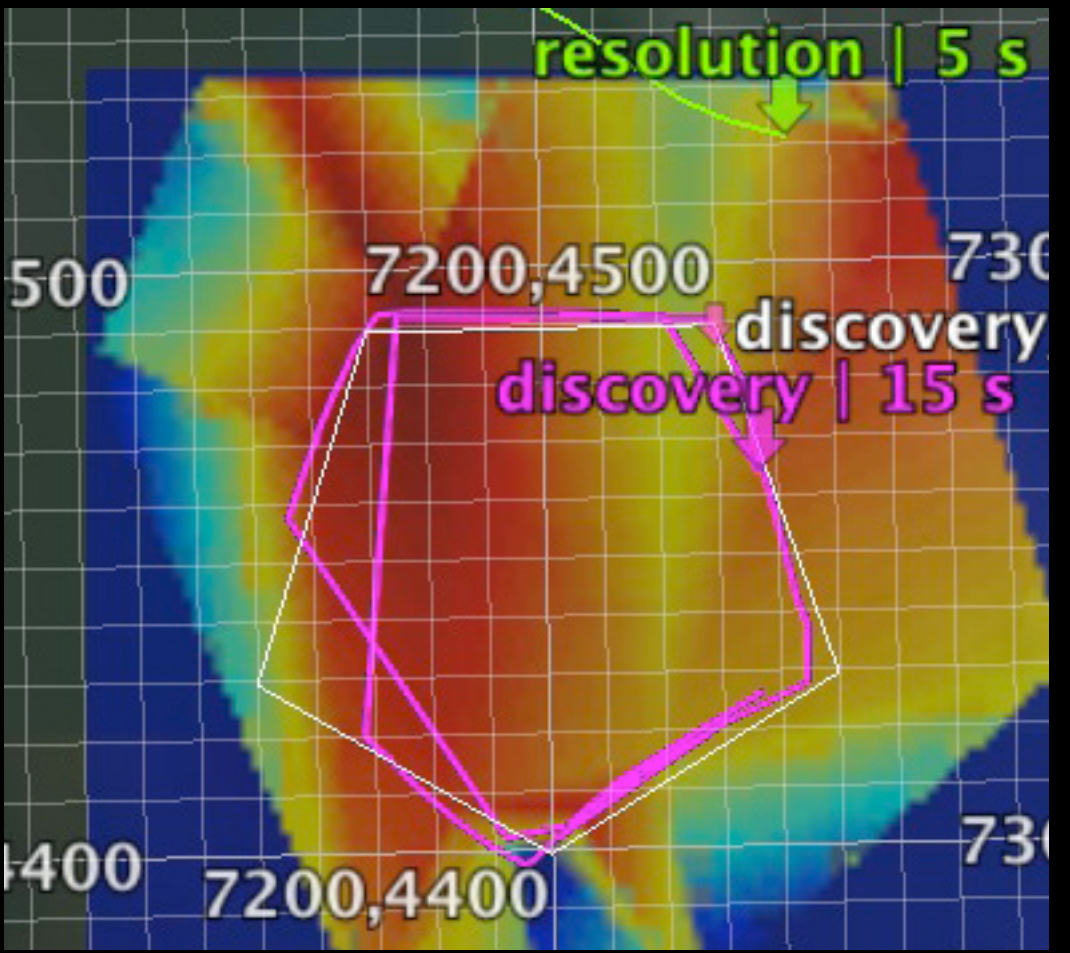
SWAMSI11:

- 3 AUVs collaborative multistatic seafloor target detection.
- REMUS LBL navigation.



Experiments: Case Studies

- BF9_MAY11:
  Backend for C. Murphy's sidescan image sending
  CTD samples for real-time lat / lon slice display:



Acknowledgments

- Goby-Developers group (MIT / WHOI / UMichigan)
- Office of Naval Research
- L. Freitag and the WHOI Micro-Modem group
- P. Newman / M. Benjamin (MOOS-IvP)
- Field trial support: NATO Undersea Research Centre, NUWC (Newport), NAVSEA (Panama City), Bluefin Robotics, Robotic Marine Systems
- Open source projects used by pAcommsHandler & Goby: Boost, Crypto++, NCurses, ASIO, Google Protobuf
- LAMSS: S. Petillo, I. Katz, A. Balasuriya, S. Danesh, E. Fischell