Peer-Reviewed Technical Communication

CAPTURE: A Communications Architecture for Progressive Transmission via Underwater Relays With Eavesdropping

Chris Murphy, *Member, IEEE*, Jeffrey M. Walls, *Student Member, IEEE*, Toby Schneider, *Member, IEEE*, Ryan M. Eustice, *Senior Member, IEEE*, Milica Stojanovic, *Fellow, IEEE*, and Hanumant Singh, *Member, IEEE*

***Abstract*—As analysis of imagery and other science data plays a greater role in mission execution, there is an increasing need for autonomous marine vehicles to transmit these data to the surface. Communicating imagery and full-resolution sensor readings to surface observers remains a significant challenge. Yet, without access to the data acquired by an unmanned underwater vehicle (UUV), surface operators cannot fully understand the mission state of a vehicle. This paper presents an architecture capable of multihop communication across a network of underwater acoustic relays. In concert with an abstracted physical layer, CAPTURE provides an end-to-end networking solution for communicating science data from autonomous marine vehicles. Automatically selected imagery, SONAR, and time-series sensor data are progressively transmitted across multiple hops to surface operators. To incorporate human feedback, data are transmitted as a sequence of gradually improving data "previews." Operators can request arbitrarily high-quality refinement of any resource, up to an error-free reconstruction. The results of three diverse field trials on SeaBED, OceanServer, and Bluefin AUVs, with drastically different software architectures, are also presented.**

***Index Terms*—Aquatic robots, disruption tolerant networking, underwater communication, unmanned underwater vehicles (UUVs).**

Manuscript received November 17, 2011; revised February 08, 2013; accepted April 26, 2013. Date of publication July 26, 2013; date of current version January 09, 2014. This work was supported by the National Science Foundation (NSF) Office of Polar Programs under Grant ANT-1039951, by the U.S. Office of Naval Research under Grant N00014-09-1-0700, by the NSF Gordon-CENSSIS ERC under Grant EEC-9986821, and by the NOAA Pacific Islands and Northwest Fisheries Science Centers. The work of R. Eustice was also supported by the NSF CAREER Award under Grant IIS-0746455. This work was presented in part at the Fifth ACM International Workshop on UnderWater Networks (WUWNet), Woods Hole, MA, USA, September 30–October 1, 2010.

Guest Editor: M. Seto.

C. Murphy was with the Deep Submergence Laboratory (DSL), Woods Hole Oceanographic Institution, Woods Hole, MA 02543 USA. He is now with Bluefin Robotics Corporation, Quincy, MA 02169 USA (e-mail: cmurphy@whoi.edu).

J. M. Walls and R. M. Eustice are with the Perceptual Robotics Laboratory (PeRL), University of Michigan, Ann Arbor, MI 48109 USA (e-mail: jmwalls@umich.edu; eustice@umich.edu).

T. Schneider is with the Laboratory for Autonomous Marine Sensing Systems (LAMSS), Massachusetts Institute of Technology (MIT), Cambridge, MA 02139 USA (e-mail: tes@mit.edu).

M. Stojanovic is with the Electrical and Computer Engineering Department, Northeastern University, Boston, MA 02115 USA (e-mail: millitsa@ece.neu.edu).

H. Singh is with the Deep Submergence Laboratory (DSL), Woods Hole Oceanographic Institution, Woods Hole, MA 02543 USA (e-mail: hsingh@whoi.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JOE.2013.2261331

I. Introduction

TELEMETRY from unmanned underwater vehicles (UUVs) historically has been limited to basic vehicle state information interspersed with the occasional measurement from one or two simple sensors. This level of communication has proven adequate (if unsatisfying) when the only decision facing an operator is whether to abort the mission of a single vehicle. Missions, however, may now involve multiple vehicles working toward a loosely defined set of goals, often in dangerous and unconstrained environments such as under ice. These goals can be defined during the mission and may be based on complex analysis of science data, including imagery. Architectural advances in autonomy, evident in MOOS-IvP [1], T-REX [2], and DAMN [3], [4], have not been met with similar advances in UUV telemetry handling.

Most UUVs transmit only a small and predefined set of state data to the surface. Typically, this would include the vehicle position, depth, battery life, heading, or similar status information. Photographic and SONAR imagery are only transmitted by a few special-purpose communication systems. These systems rely on specific vehicle geometries, and do not scale to support multiple vehicles. Missions involving multiple vehicles may extend beyond the effective range of a single acoustic link, yet existing systems do not support relaying large data across multiple UUV "hops."

This paper presents CAPTURE—a **C**ommunications **A**rchitecture that delivers arbitrary science data using **P**rogressive **T**ransmission, via multiple **U**nderwater **R**elays and **E**avesdropping. In concert with an abstracted physical layer, CAPTURE provides an end-to-end communication architecture for interactively obtaining data across an acoustic network of marine vehicles. CAPTURE employs progressively encoded compression to telemeter imagery, SONAR, and time-series sensor data from underwater vehicles. These resources are automatically selected by the vehicle, and transmitted as a sequence of gradually improving data "previews." High-quality versions of these previews, up to an error-free reconstruction, can be requested by operators immediately, or at any later time over the course of a mission. CAPTURE has been designed to operate on multiple vehicle architectures and supports multihop relay communication across several vehicles. CAPTURE has been deployed in multiple field operations on diverse vehicle

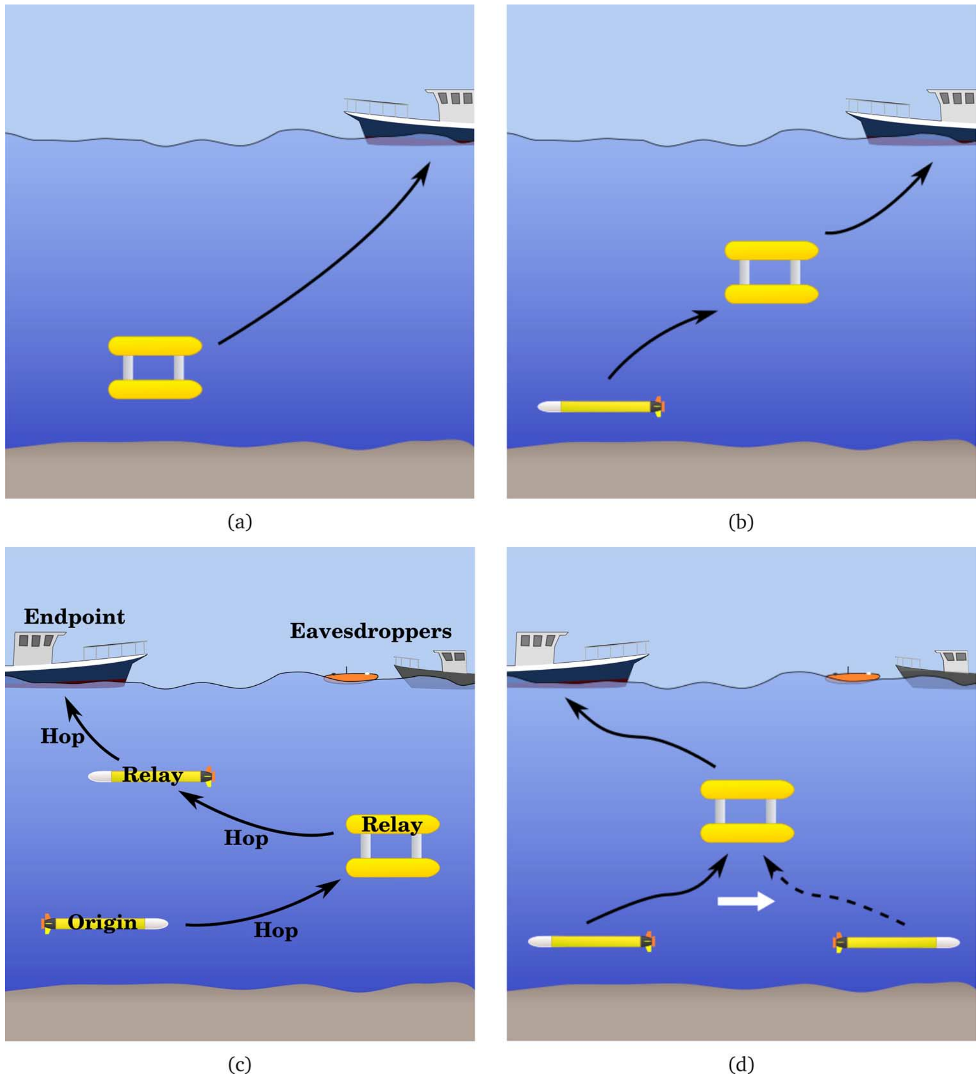


Fig. 1. Network configurations which have successfully been used in the field with CAPTURE. In the fourth example, the vehicle responsible for initiating transmissions was changed mid-dive, in response to a request from the surface. (a) One hop. (b) Two hop. (c) Three hop. (d) Route switch.

platforms, in each of the network configurations shown in Fig. 1.

II. Related Work

A. Underwater Telemetry Compression

The diversity of UUV missions, ranging from shallow-water mine hunting to under-ice exploration, has led to most users developing custom software for encoding and decoding messages. Many of these solutions are based on the early compact control language (CCL) [5] standard for acoustic communication, which provides a number of standard algorithms for encoding 256-b messages containing depth, latitude, bathymetry, altitude, salinity, and other data. CCL relies only upon quantization to provide compression, and makes no use of the inherent correlation between successive samples from most instruments. Eastwood *et al.* proposed predictive coding methods that could be used with these methods to improve performance [6]. Schneider and Schmidt have incorporated predictive coding into their recent work with dynamic CCL (DCCL) [7], sending up a mean value followed by smaller, quantized, difference values. For data that are highly correlated, transform codes allow much higher efficiency.

Transform compression methods typically follow a standard pattern. First, a source coder such as the discrete cosine transform (DCT) or discrete wavelet transform (DWT) exploits the inherent correlation within the data, and concentrates the energy of the signal into a sparse set of coefficients. Next, these coefficients are quantized and entropy encoded [8]. Wavelet compression is described by Donoho *et al.* as being especially appropriate for functions that are "piecewise-smooth away from discontinuities" [9]. There has been additional study suggesting that wavelet transform compression techniques are particularly applicable to underwater images, video, and acoustic imagery [10]–[13]. Eastwood *et al.* evaluated the performance of an early wavelet-based image compressor [6] as early as 1996.

The underwater community has investigated transmission of imagery and video data using other transform compression methods, such as JPEG, as well [14]–[16], yet these solutions rely on artificially high throughput by positioning a surface ship directly above the UUV. This "acoustic tethering" is impractical in many autonomous operations, nearly impossible for those in polar environments, and does not scale easily to handle multiple vehicles.

B. Multihop Networking

UUVs are confronted by an extremely challenging acoustic environment as their primary medium for communication with surface operators. The ocean imposes severe limitations on acoustic communication, including low available bandwidth and long propagation delays [17], [18], which lead to frequent data loss and high latency. Robust physical communication layers exist off-the-shelf from several manufacturers [19]–[21]. Each modem manufacturer provides a different software interface to the end user—yet there does not exist a common cross-manufacturer interface such as the Hayes/AT Command Set that dictated the course of terrestrial telephone modems. Webster *et al.* previously developed a modem abstraction layer for the Woods Hole Oceanographic Institution (WHOI) MicroModem [22]. More recently, the Goby Autonomy Project [23] has made advances in developing a generic abstraction for acoustic modems and implementing drivers for physical modem hardware. These drivers allow software to operate independent of the modems' underlying proprietary languages.

Numerous medium access control (MAC) protocols such as the multiple access with collision avoidance (MACA) [24] and the multiple access with collision avoidance for wireless (MACAW) [25] have been developed to mediate between multiple communicating nodes at the data link layer. Research at higher networking layers exists as well [26], yet few field experiments have involved multiple autonomous vehicles communicating high-bandwidth data across multiple hops. Perhaps the best known such experiment, Seaweb [27], was performed by Benthos in concert with the U.S. Navy, utilizing fixed nodes. There currently exist no transport or application layer protocols in widespread use for underwater vehicles.

Common terrestrial protocols are largely unfit for underwater use without adaptation on a number of fronts. The header for a typical user datagram protocol (UDP) packet (the more minimalist of the two primary transport protocols used by internet traffic) would consume three quarters of a standard 256-b CCL compatible message. Terrestrial networking commonly relies on the capability to rapidly forward messages from one network node to another node, dropping those messages which cannot be immediately forwarded. Recent research in delay-tolerant networking [28] suggests that "store and forward" approaches can significantly improve the performance of high latency, occasionally disconnected, networks. One implementation of this strategy, the Bundle Protocol, is now being pursued under the auspices of the Internet Society's Delay Tolerant Networking Research Group [29]. Relative to the bandwidth available with modern acoustic modems, underwater vehicles can be considered to have nearly infinite storage.[1] CAPTURE exploits that storage by having every node in the network permanently store each piece of data that is transmitted.

[1]Ten nodes communicating constantly at a generous throughput of 10 kb/s for one month still would have exchanged only about 30 GB, easily capable of fitting on a small and cheap USB storage key.

III. CAPTURE

CAPTURE consists of four distinct components, shown in Fig. 2. First, a set of data is acquired by the UUV and registered as a transmittable resource with the telemetry system, via a platform-specific driver. Examples of possible resources include a single image, or a time series of measurements from a single sensor. The platform-specific drivers isolate the telemetry system from the specific capabilities or limitations of each host vehicle. Second, new resources are automatically selected for compression and transmission to the surface, or existing resources are selected for further transmission based on requests from the surface. Automatic selection provides an avenue for high-level algorithms, such as mine identification or interest operators, to guide the selection of interesting telemetry. Third, selected resources are compressed using progressive coding methods. Progressive coding methods, specifically those that are fully embedded, ensure that an approximation to the data can be reconstructed with each newly received bit of data. Finally, the transmission of the resource to the surface is managed to ensure end-to-end delivery. When multiple underwater vehicles are available, intermediate vehicles can relay data to the surface as hops in the route, or help through "eavesdropping." The flow of data between the four subsystems is shown in detail in Fig. 3, and each subsystem is described in detail in the following subsections.

A. Platform Drivers

The platform drivers provide an interface to the existing software on each different vehicle platform. Software architectures vary significantly from vehicle to vehicle, as do sensing and computation capabilities. Platform drivers smooth over these architectural differences by providing:

- an interface for data transmission and reception via the modem;
- configuration of resource registration and prioritization;
- handling of non-CAPTURE acoustic traffic, such as command and control messages;
- logging support via lightweight communications and marshalling (LCM) [30].

Physical connections to the vehicle's acoustic modem vary, but most are connected to an RS-232 serial port. Projects such as Goby [23] provide software abstraction between the actual modem hardware and CAPTURE, as well as lower level networking layers such as MAC. MAC may require some configuration, such as information about any acoustic range-based navigation systems that are in use, or specification of a fixed communication cycle. Each modem also requires a unique integer identifier, typically specified as part of the configuration.

The platform driver is responsible for registering existing sensors, such as cameras, SONARs, and conductivity–temperature–depth (CTD) sensors, as resource generators. The importance of different resources will vary by mission and vehicle, so their prioritization may require premission configuration by users. That configuration is again performed through the platform driver. Some vehicles may only register a single camera and transmit imagery. Other vehicles may switch between multiple sensors, such as a camera and a CTD, selecting between the resources during the prioritization phase. Command and control

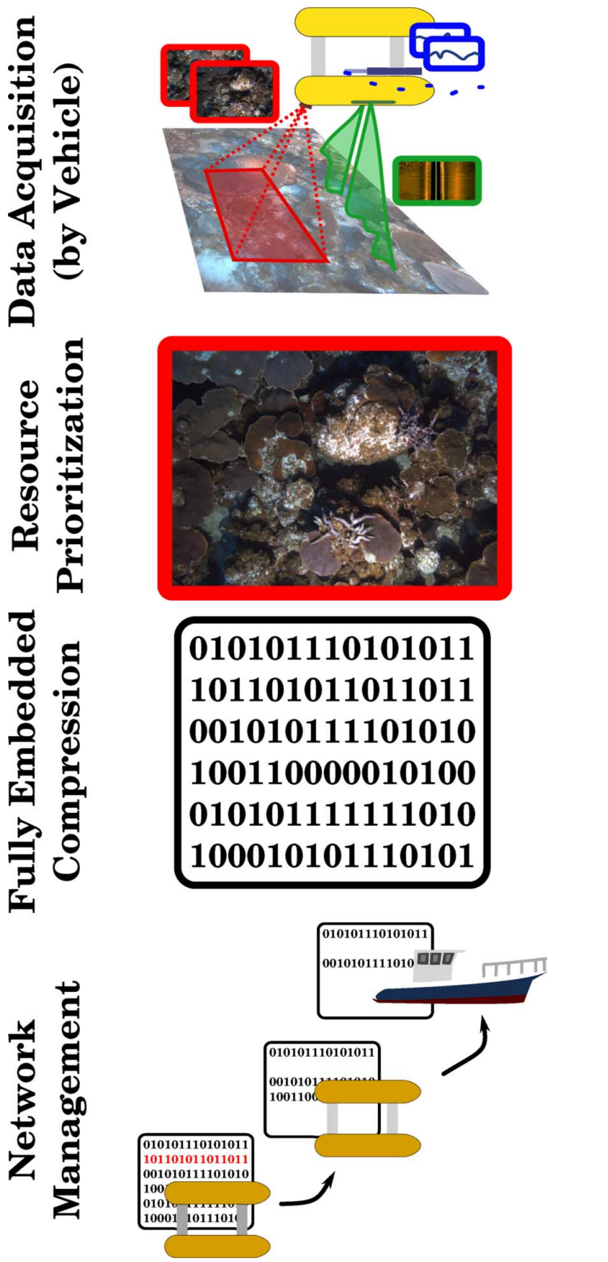


Fig. 2. High-level overview of data flow through the four main components of CAPTURE. Platform drivers connect acquired data into the CAPTURE system, which is then winnowed down, compressed, and eventually transmitted to the surface.

messages, such as vehicle aborts or mission changes, are also delivered by the driver to appropriate handlers.

B. Resource Prioritization

Over the course of a dive, a single UUV can easily collect millions of samples of scalar environmental data, ranging from temperature to salinity, from measured methane concentrations to vehicle depth. The same vehicle may easily capture tens of thousands of photos, and sonar imagery. Modern UUV platforms generate orders of magnitude more data than could possibly be transmitted to the surface—the first task facing any telemetry system is to prioritize which data should be transmitted.

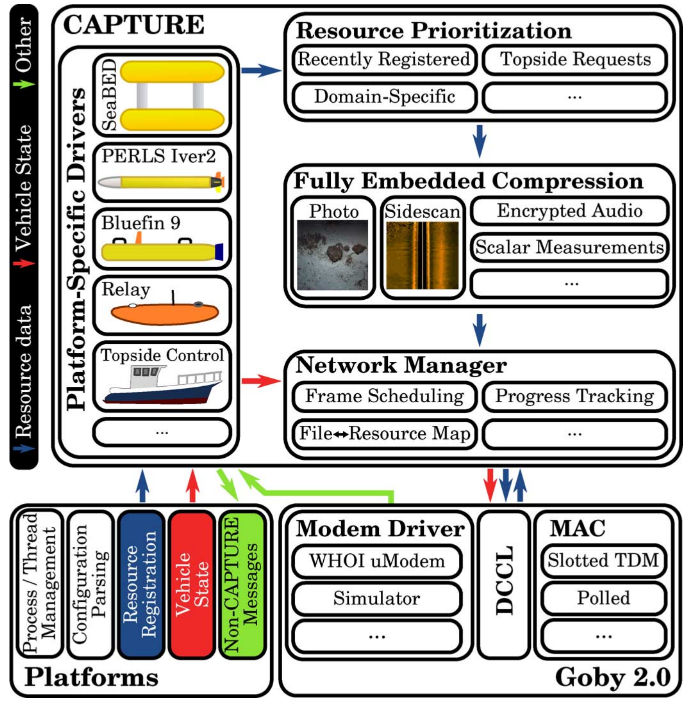


Fig. 3. A detailed view of the interactions between CAPTURE components. Arrow colors indicate the type of data, as shown in the black key on the left.

At any given time, surface operators can choose whether to request refinement of a specific resource or whether to allow the vehicle to automatically select new resources for transmission. For vehicles with multiple sensors of interest, it is also necessary to multiplex the transmissions between those sensors. These steps can be quite simple, such as always sending the most recent resource registered by a single sensor. More complex missions may involve significant computation in this step, such as identifying seafloor mines through image analysis. Multiplexing approaches could range from a round-robin scheduling-based approach, to priority queues, or computed metrics.

While a single image is easy to consider as a distinct "resource," transmitting environmental sensor data requires identifying a section of data to transmit. This is best done by breaking a time series into large chunks of data; for correlated time-series data, compressing a few samples at a time is much less efficient than compressing long sequences simultaneously. Fig. 4 shows this result while piecewise compressing a long series of temperature data.

C. Progressively Encoded Compression

After identifying a resource for transmission to the surface, that resource must be compressed to maximize the throughput of the channel. CAPTURE relies on progressively coded compression methods—preferably fully embedded ones.[2] CAPTURE transmits enough data to the surface to reconstruct a low-quality "preview" of each automatically selected resource before moving onto a new resource. Due to the progressive

[2]*Progressive* coding methods allow reconstruction of an intermediate data representation at one or more stopping points within an encoded data stream. *Fully embedded* coding methods allow reconstruction to occur at any point in the encoded data stream.

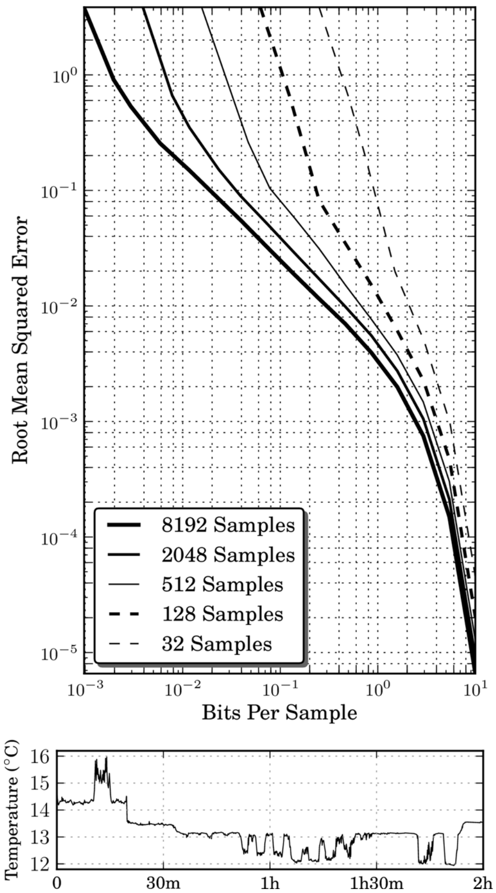


Fig. 4. The reconstruction error [Y-axis, in root mean square error (RMSE)] versus the compression level (X-axis, in bits) for a 2-h sequence of temperature data [shown at the bottom]. Each plotted line shows the result of compressing the full data set, but doing so by different length subsets of the data at a time. Since the original temperature data were collected at 4 Hz, compressing 8192 samples at a time would be equivalent to transmitting updated data every 34 min, versus every 30 s when data are compressed 128 samples at a time. Encoding more samples in each transmission lowers the reconstruction error for any given compression level.

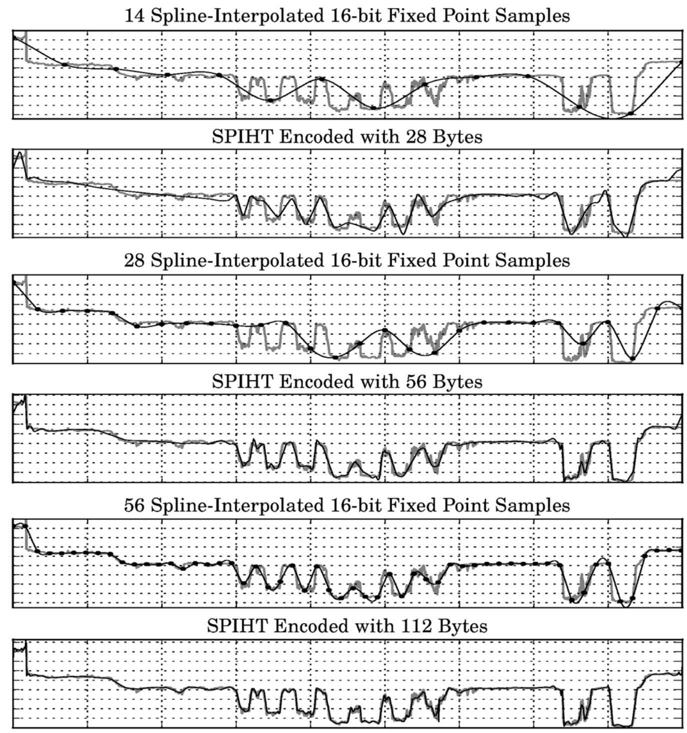


Fig. 5. SPIHT-encoded scalar temperature data at different levels of compression, compared to interpolating quantized samples. Note that each SPIHT/fixed-point comparison pair is encoded with the same number of bytes.

Fig. 6. The same image encoded at four different sizes using SPIHT. 81% of the transmitted data is used to reconstruct the luminance; the rest describes the image color channels.

nature of the encoding, each new piece of data received on the surface will allow an increasingly higher quality representation of the resource to be reconstructed. This serves two equally important purposes. If the "preview" piques the operator's interest, the operator can request more encoded data from that resource to refine the already transmitted data with no wasted transmissions. Every byte sent up for the preview will be used as the basis for the higher quality version. If, on the other hand, the resource is uninteresting, the operator may be able to determine that after only a few transmissions and avoid wasting further bandwidth to deliver a full preview.

The success of wavelet-based analysis techniques in the underwater domain suggests the use of fully embedded coding methods that support wavelet compression. The embedded zerotree eavelet (EZW) [31] algorithm is one noted early example, which led to the more efficient set partitioning in hierarchical trees (SPIHT) [32] coding method and others [33], [34]. These quantization and entropy coding methods can be coupled with the 1-D, 2-D, or multidimensional DWT, allowing compression of both time-series scalar data and imagery using the same algorithm. While CAPTURE will work with any progressively encoded compression method, SPIHT [32] is discussed here as it has been used in our field experiments, and provides some insight into how all these similar methods operate.

Data compression with SPIHT consists of three discrete steps. First, data are transformed into the wavelet domain using the DWT. Next, these (typically floating-point) coefficients are requantized as signed fixed-point numbers. Finally, this fixed-point representation is encoded using the SPIHT coding algorithm, which results in a sequence of bits. Any truncated portion of this bitstream can be decoded into a signed fixed-point approximation of the wavelet coefficients. The inverse DWT on these coefficients then results in an approximation of the original data. Fig. 5 shows increasingly accurate reconstructions of a sequence of temperature data, and Fig. 6 shows four different

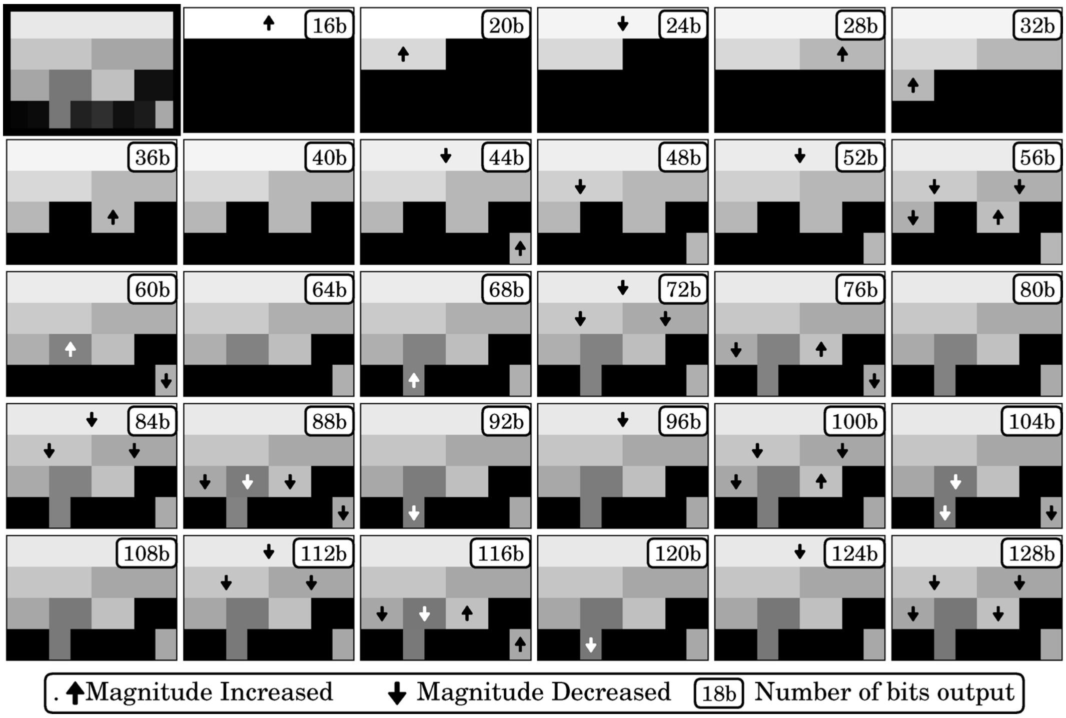


Fig. 7. A wavelet decomposition at upper left, followed by the reconstruction from increasing lengthy SPIHT bitstreams. As the number of bits grows, the reconstruction is closer to the original coefficients. For clarity, only coefficient magnitudes have been visualized, not signs.

reconstructions of a 16-b 1024 × 1024 pixel (2 MB) color photograph with increasing length bitstreams.

SPIHT and similar algorithms treat the wavelet decomposition as a tree of coefficients, rooted at the lowest level detail coefficients. Many real signals that have large magnitude coefficients at high levels also have higher magnitude coefficients at lower levels. SPIHT exploits this cross-level correlation through a clever sorting algorithm. As the SPIHT authors write in their tutorial [35] set partition coding "recursively splits groups of [coefficients] guided by a sequence of threshold tests, producing groups of elements whose magnitudes are between two known thresholds."

A SPIHT-encoded bitstream consists of a sequence of sorting bits and refinement bits, interlaced in a data-dependent order. Sorting bits provide an efficient way to identify high magnitude, and therefore important, wavelet coefficients. Refinement bits provide a continually improving estimate for the magnitude of a wavelet coefficient. In particular, sorting bits indicate:

- whether a coefficient is greater in magnitude than the current threshold, or "significant";
- whether any descendant in the wavelet tree of the currently considered coefficient is "significant";
- whether any grand-descendant is significant.

Refinement bits indicate either the sign of a coefficient or a single bit of a coefficient's magnitude. Fig. 7 shows the progressive reconstruction of a small set of coefficients using an increasing number of (indicated) bits.

D. Multihop Networking

A CAPTURE network consists of multiple nodes, including an origin, endpoint, zero, or more ordered hops, and possibly some eavesdroppers, as shown in Fig. 1(c). Resources, such as photographs, are captured by the origin and relayed by hops to the endpoint. The network can operate in either an automatic selection mode where transmitted resources are automatically selected by the origin, or in a refinement mode. When in automatic selection mode, enough data are relayed for the endpoint to reconstruct a low-quality preview of a resource. When the origin learns that the endpoint has received enough segments to reconstruct a preview, it will automatically select a new resource for transmission. The endpoint, typically a manned surface ship, can request that the network instead operate in refinement mode. In refinement mode, data continue to be transmitted for a specific, previously transmitted resource, selected by the endpoint. The origin and hops will relay additional data from that resource until the network is put back into automatic selection mode by the endpoint.

1) Message Types: CAPTURE uses two types of network messages to communicate information between nodes: chunk and control messages. The bulk of traffic in a CAPTURE network consists of chunk messages. In plain English, an example chunk message could be:

> The 4th segment of SeaBED's 33rd resource consists of the following data. . .

Control messages are significantly more complicated, though one possible example might be:

> The route consists of SeaBED, hop A, hop B, and the endpoint. SeaBED's 33rd resource is being refined. The endpoint has received the first 9 contiguous segments. Beyond the 9th segment, the hops and endpoint are known to have received the following segments:. . .

Even after compression, resources will likely be too large for transmission by today's acoustic modems, and thus must be broken into segments. Chunk messages consist of a single segment of data, along with the identifier for the segment's position within the resource, and a unique identifier for the resource itself. Chunk messages are designed to stand alone—any vehicle receiving a message can uniquely identify the resource the segment belongs to, and the segment's position within the resource, without any additional knowledge. Segments are of a fixed size, which must be agreed upon within the network before deployment. The segment size should be based on the maximum transmission unit (MTU) supported by the modem hardware. For the WHOI MicroModem, this could be 256, 512, or 2048 b, depending on the level of error correction that is applied.

The second type of message used in a CAPTURE network is a control message. Control messages contain a variety of data used to synchronize the state of the network between nodes, including acknowledgement and routing information. Control messages include the current resource identifier being transmitted by the network, just like chunk messages, but otherwise serve a different purpose. Each network node tracks the segments, for each resource, that it knows each other node to possess. The primary purpose of control messages is to convey partial estimates of these "segment masks" between CAPTURE nodes, acting as a selective acknowledgement. In particular, the message indicates the highest known index of the endpoint's contiguously received segments, and lists of the segments beyond that which are known to be possessed by network hops or the endpoint. Control messages also identify whether the network is operating in refinement mode.

Control messages also include the current route from the origin to the endpoint, and a revision ID. The endpoint can alter this route or select a different vehicle as the origin by incrementing the revision ID. The route consists of the hardware IDs, in order, for the nodes currently in the network: $\langle \text{origin}, \text{hop}_A, \ldots, \text{endpoint} \rangle$. The overhead of this routing information would be substantial in traditional networks, but adds minimal overhead for small numbers of vehicles. For networks with small numbers of vehicles, a single network node can be identified by a few bits, and routes can be expressed in a byte or two.

2) Message Handling: Since the ocean is a broadcast medium, messages may "skip" any individual hop in a network, or even be communicated directly from origin to endpoint. There is no guarantee or requirement that each message be communicated along every node in the route. When any message is received, some components of a message may be ignored depending on the source of the message. In particular, some data are not assumed to be valid unless they come from *upstream*, closer to the origin, or *downstream*, closer to the endpoint. For example, both chunk and control messages contain a resource ID. If the network is believed to be in automatic selection mode, that resource ID is taken to be the currently active resource only if it came from upstream. On the other hand, if the network is in refinement mode, the resource ID will be taken as the active ID only if it came from downstream. This allows the origin to control the transmission of automatically selected resources, yet also propagates resources requested from the surface toward the origin when operating in refinement mode.

TABLE I
WHICH MESSAGES ARE TRANSMITTED BY A NODE DEPENDS UPON ITS ROLE IN THE NETWORK; AS SHOWN HERE

	Chunk	Control
Origin	×	
Hop	×	×
Endpoint		×
Eavesdropper	×	

When a chunk message is received, the data segment it contains is stored at the appropriate offset in the local copy of the resource. The receiving node also stores that the transmitter has the segment.

Any node receiving a control message first incorporates the included segment masks into their own segment mask. If the message was transmitted by the immediate downstream neighbor, the current autonomy mode is also stored from the message. Finally, if the route revision in the message is higher than that of the currently stored route, the local copy of the routing information is updated.

3) Transmitting Messages: Which messages are transmitted by a network node depends upon the node's role in the CAPTURE network, as shown in Table I.

When transmitting a chunk message, the segment masks for downstream nodes should be used to select what is transmitted. Early resource segments that have not been received by any nodes closer to the endpoint are the highest priority. In particular, nodes should start by transmitting the earliest segments for the active resource that a downstream node is not believed to possess, and continue in-order transmission of any later segments not held by downstream nodes. When a control message is received from a downstream node, this process starts over by transmitting the earliest segment now known to not be received.

IV. Field Experiments

CAPTURE has been field tested in multiple experiments. All told, these experiments involved six distinct autonomous platforms, including two different SeaBED AUVs, two different OceanServer Iver AUVs, and a Bluefin 9 AUV. In addition, four manned surface ship platforms have been used, involving researchers from the National Oceanic and Atmospheric Administration (NOAA, Washington, DC, USA), the Massachusetts Institute of Technology (MIT, Cambridge, MA, USA), WHOI (Woods Hole, MA, USA), Northeastern University (Boston, MA, USA), University of Michigan (Ann Arbor, MI, USA), and Bluefin Robotics Corporation (Quincy, MA, USA).

In February 2010, an early version of the CAPTURE architecture was tested on *Lucille*, a SeaBED-class [36] AUV owned by NOAA, during a research expedition aboard the NOAA Ship *Oscar Elton Sette*. A single dive was performed near Rota, an island in the Northern Marianas Archipelago [37], ranging in depth between 100 and 350 m. No specific constraints were put on the surface ship, which remained within 600 m of the vehicle throughout the dive.

Fig. 8. Vehicles used during the 2011 CAPTURE Experiment. The two Iver AUVs are visible on top, with the SeaBED AUV to the left.

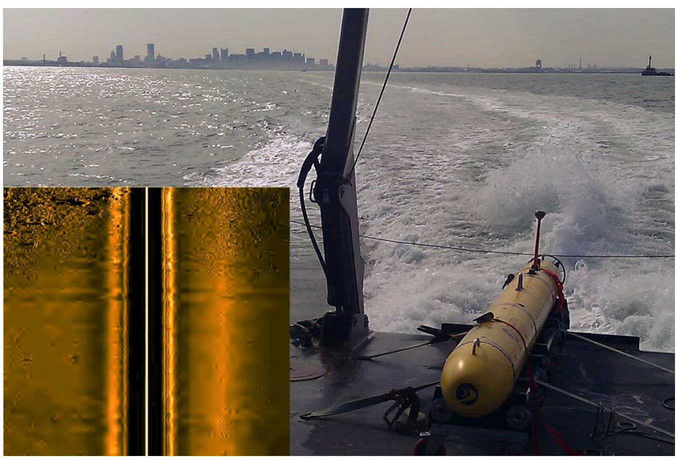


Fig. 9. Bluefin 9 AUV before deployment, and the SONAR imagery transmitted during the dive.

In late May 2011, CAPTURE was extended to operate on a Bluefin 9 AUV equipped with a "backseat driver" computation stack running the mission-oriented operating suite (MOOS) software suite. As part of ongoing mine countermeasures development, we seek to identify seafloor mine-like objects and transmit their SONAR signatures to the surface for confirmation [38].

In August 2011, the CAPTURE networking stack was tested on three autonomous platforms and one manned platform operating simultaneously. Two OceanServer Iver AUVs with payload and navigation suites custom-developed by the University of Michigan [39] provided long-range midwater-column survey capability, while a SeaBED AUV [36] provided the ability to capture detailed low-altitude photographic surveys. These platforms were coupled with a manned surface ship, the *R/V Tioga*, and a number of dives were performed in Buzzards Bay, MA, USA.

A. Platform Driver/Resource Acquisition

The *Lucille* AUV used during the 2010 field experiment is equipped with a 5-Mpixel Prosilica color camera, featuring a charge-coupled device (CCD) with high dynamic range.

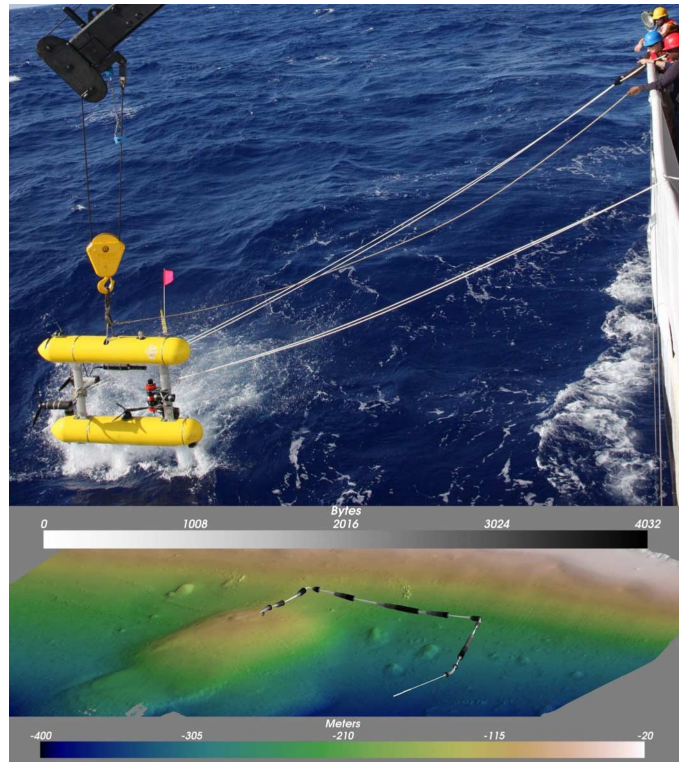


Fig. 10. Top: *Lucille*, a SeaBED AUV, prior to launch near Rota, 2500 km south of Tokyo, Japan. Bottom: Transmission progress for each image, overlaid on bathymetry.

During the 2010 field experiment, this camera captured one color image every 5 s at a resolution of 2048×2048 pixels. Those raw images were processed and converted to the AUV colorspace onboard the AUV's main control computer, resulting in 1024×1024 pixel square full color images.

The Bluefin 9 AUV used during the brief mine countermeasure experiment is equipped with a MarineSonic sidescan SONAR system, which generates 2-D imagery in a proprietary TIFF-like format after a fixed number of scanlines. A platform driver was developed to support reading the imagery from the SONAR, and to interface with the onboard MOOS autonomy software. Goby software was used to abstract the interface with the onboard WHOI MicroModem. During a very short mission, there was time to transmit a single SONAR image to the surface from the UUV, shown in Fig. 9.

B. Resource Prioritization

To date, our field experiments have relied on a single-resource queuing model to identify the next resource for transmission. The *Lucille* AUV used during the 2010 field experiment, and shown in Fig. 10, has a single onboard central processing unit (CPU) used for both CAPTURE and vehicle control. To minimize the risk of overloading the onboard CPU's limited resources, the most recently captured photograph was compressed every 3 min. This led to several images being compressed but not transmitted, but ensured that new data were always available for transmission. When CAPTURE was ready to transmit a new resource to the surface, the most recently compressed new image was selected for transmission.

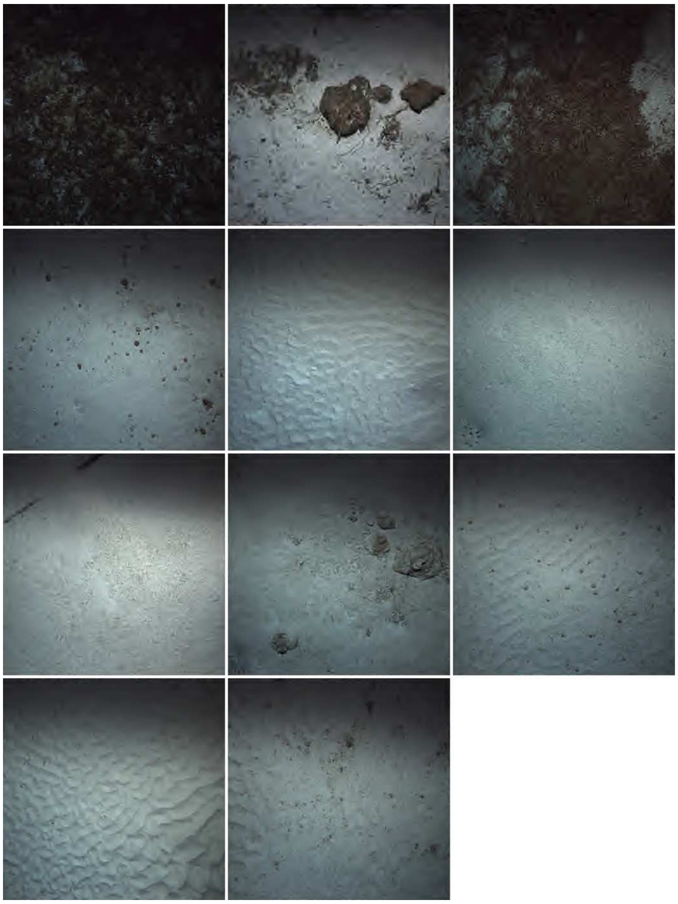


Fig. 11. Color imagery captured by a SeaBED-class AUV, compressed *in situ*, and transmitted to surface operators.

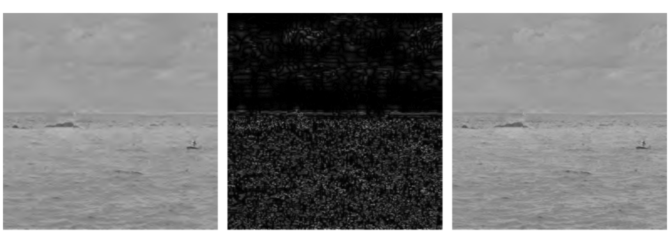


Fig. 12. A transmitted grayscale photo before and after requesting additional refinement. The difference in magnitude is shown between the two versions on a logarithmic scale to highlight changes: (a) 41 segments; (b) log difference; and (c) 97 segments.

C. Progressive Encoding

The photographic and SONAR imagery were compressed using the same SPIHT compression, in conjunction with the Cohen-Daubechies-Feauveau 9/7 wavelet [40]. The MarineSonic SONAR source imagery was a grayscale image of 1024×960 pixels in a proprietary format. For the color photographic imagery captured by the *Lucille* AUV, 50% of the encoded data stream was allocated to luminance data, and 50% to chrominance data. In retrospect, allocating a higher proportion to luminance data would have resulted in more visually pleasing imagery.

During the 2010 field experiment, in total, 15 color photographs were received over the course of a 3.75-h period. Of the 15 successfully received images, four were captured during descent or ascent and were completely black as a result. The 11 nonblack images received are shown in Fig. 11. The 15 images were transmitted over a 3.75-h period, resulting in about 15 min per image, or approximately 35 b/s achieved. While this low number is largely due to packet loss and scheduling in real-world conditions, the modem also varied the level of forward error correction, between encodings with maximum burst rates of 520 and 5400 b/s, to obtain richer statistics on transmission success.

During the most recent CAPTURE field experiment, extremely murky water conditions prevented capturing photographs, and precaptured imagery was used instead. In addition, one test was performed with a nonprogressively encoded data set to allow analysis of the architecture performance with nonprogressive data sets. A short segment of audio, Neil Armstrong's first words on the surface of the moon, was compressed with the Speex voice codec to 4368 B. The audio was then encrypted using AES with a 256-b key.

D. Networking

Three separate successful CAPTURE dives were performed during the most recent field experiment, each testing different capabilities of the networking protocol. During one trial, data were communicated across a two-hop network, as shown in Fig. 1(b). After six preview images were sequentially transmitted as 2047-B previews, the fourth transmitted image was identified by the surface operator as warranting further refinement. Upon request, the transmitting vehicle went back and provided additional data to refine the image, as shown in Fig. 12. The origin eventually transmitted 5529 contiguous bytes of the image before being instructed to return to automatic selection.

In total, seven images were eventually transmitted, each decoded progressively, with gradually improving reconstructions over the course of the transmission. CAPTURE was also tested in the three-hop linear network, depicted in Fig. 1(c), successfully relaying four images across the heterogeneous network.

In the final experiment, a two-hop network was employed, as shown in Fig. 1(d). Four grayscale images were transmitted from an Iver AUV, via a SeaBED AUV, to the surface. The surface operator then requested a route change, granting the other Iver AUV the responsibility for transmitting resources. That vehicle transmitted the preloaded encrypted speech, followed by another two grayscale photos.

E. Network Protocol Implementation

Each autonomous platform had a platform driver developed to fit the needs of their specific software environment. A number of revisions to Goby v2.0 were made as part of this work, which allowed it to be used as a software abstraction layer for the acoustic modem on each vehicle. The implementation of the CAPTURE protocol relied on the two message types described in Section III-D. These messages were constructed as 512-b messages, to fit the requirements of the physical layer. The specific message definitions used are shown in Figs. 13 and 14.

Since the entire route is encoded in the control packet, which currently is 12 b long (plus three to allow changing the route), this implementation supports routes containing up to four vehicles, and networks containing seven vehicles in total. This could

0	8	16	24	31
Message ID	Time of Launch	Resource Origin / ID		
…	Segment ID	56 Byte Segment…		

Fig. 13. Definition for chunk messages used during 2011 field experiments in Buzzards Bay.

0	8	16	24		31
Message ID	Time of Launch	RID	Route	Mode	…
…Resource Origin / ID			End-to-end Reception Count		
54 Byte Acknowledgement Masks …					

Fig. 14. Definition for control messages used during 2011 field experiments in Buzzards Bay.

easily be expanded for longer routes, consuming only a few additional bits.

The chunk and control messages both contain a time-of-launch field, allowing the second of transmission to be encoded in a message. All vehicles in the Buzzard's Bay experiment were equipped with a high-precision, low-drift clock [41]. By synchronizing each vehicle's clock at the surface, all nodes can passively measure the one-way travel time (OWTT) of each acoustic broadcast by simply comparing the encoded time of launch and the observed time of arrival. Since the sound-speed profile is well known in water, the intervehicle range can be easily computed. Over time, vehicles within the network can augment each other's navigation estimates using these additional range constraints.

V. Conclusion and Future Work

The focus of this work has been on building the networking and compression infrastructure to support transmission and interactive refinement of SONAR and photographic data, along with time-series measurements. We have presented progress in defining a communication architecture that provides this capability across small networks of autonomous vehicles. Finally, we discussed summary results for a number of field trials. Areas of future research include refinement of the network protocol, and a detailed analysis of transmission statistics. Multiple approaches have been considered for online selection of data from multiple sensors, both automatic and with user guidance. We also see clear opportunities to incorporate more advanced data interest detectors on a per-sensor basis, such as image recognition algorithms.

References

[1] M. Benjamin, H. Schmidt, P. Newman, and J. Leonard, "Nested autonomy for unmanned marine vehicles with MOOS-IvP," *J. Field Robot.*, vol. 27, no. 6, pp. 834–875, Nov. 2010.

[2] K. Rajan, F. Py, C. McGann, J. Ryan, T. O'Reilly, T. Maughan, and B. Roman, "Onboard adaptive control of AUVs using automated planning and execution," in *Proc. Int. Symp. Unmanned Untethered Submersible Technol.*, Durham, NH, Aug. 2009, pp. 1–13.

[3] J. K. Rosenblatt, "DAMN: A distributed architecture for mobile navigation," Ph.D. dissertation, Robotics Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, 1997.

[4] J. K. Rosenblatt, S. B. Williams, and H. Durrant-Whyte, "Behavior-based control for autonomous underwater exploration," *Int. J. Inf. Sci.*, vol. 145, no. 1-2, pp. 69–87, 2002.

[5] R. P. Stokey, L. E. Freitag, and M. Grund, "A compact control language for AUV acoustic communication," in *Proc. OCEANS Eur. Conf.*, Jun. 2005, vol. 2, pp. 1133–1137.

[6] R. L. Eastwood, L. E. Freitag, and J. A. Catipovic, "Compression techniques for improving underwater acoustic transmission of images and data," in *Proc. MTS/IEEE OCEANS Conf., Prospects for the 21st Century*, Fort Lauderdale, FL, USA, Sep. 1996, DOI: 10.1109/OCEANS.1996.566719.

[7] T. Schneider and H. Schmidt, "The dynamic compact control language: A compact marshalling scheme for acoustic communications," in *Proc. MTS/IEEE OCEANS Conf.*, Sydney, Australia, 2010, DOI: 10.1109/OCEANSSYD.2010.5603520.

[8] S. Saha, "Image compression—From DCT to wavelets : A review," *ACM Crossroads: Data Compress.*, vol. 6, no. 3, pp. 12–21, Mar. 2000.

[9] D. L. Donoho, M. Vetterli, R. DeVore, and I. Daubechies, "Data compression and harmonic analysis," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2435–2476, Oct. 1998.

[10] D. F. Hoag and V. K. Ingle, "Underwater image compression using the wavelet transform," in *Proc. OCEANS Conf.*, Sep. 1994, vol. 2, pp. 533–537.

[11] D. F. Hoag, V. K. Ingle, and R. J. Gaudette, "Low-bit-rate coding of underwater video using wavelet-based compression algorithms," *IEEE J. Ocean. Eng.*, vol. 22, no. 2, pp. 393–400, Apr. 1997.

[12] J. R. Goldschneider, "Lossy compression of scientific data via wavelets and vector quantization," Ph.D. dissertation, Dept. Electr. Eng., Univ. Washington, Seattle, WA, USA, 1997.

[13] C. Murphy, R. Y. Wang, and H. Singh, "Seafloor image compression with large tilesize vector quantization," in *Proc. IEEE Autonom. Underwater Veh. Conf.*, Monterey, CA, USA, 2010, DOI: 10.1109/AUV.2010.5779653.

[14] M. Suzuki, T. Sasaki, and T. Tsuchiya, "Digital acoustic image transmission system for deep-sea research submersible," in *Proc. MTS/IEEE OCEANS Conf.*, Oct. 1992, vol. 2, pp. 567–570.

[15] C. Pelekanakis, M. Stojanovic, and L. Freitag, "High rate acoustic link for underwater video transmission," in *Proc. OCEANS Conf.*, Sep. 2003, vol. 2, pp. 1091–1097.

[16] C. Sayers, A. Lai, and R. Paul, "Visual imagery for subsea teleprogramming," in *Proc. IEEE Robot. Autom. Conf.*, May 1995, vol. 2, pp. 1567–1572.

[17] M. Stojanović, "Recent advances in high speed underwater acoustic communications," *IEEE J. Ocean. Eng.*, vol. 21, no. 2, pp. 125–136, Apr. 1996.

[18] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: Research challenges," *Ad Hoc Netw.*, vol. 3, no. 3, pp. 257–279, Mar. 2005.

[19] I. LinkQuest, "Linkquest underwater acoustic modems, uwm3000h specifications," 2008 [Online]. Available: http://www.link-quest.com/html/uwm3000h.htm

[20] T. Benthos, "Telesonar underwater acoustic modems," 2008 [Online]. Available: http://www.benthos.com/pdf/telesonar2007.pdf

[21] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball, "The WHOI micro-modem: An acoustic communications and navigation system for multiple platforms," in *Proc. MTS/IEEE OCEANS Conf.*, 2005, vol. 2, pp. 1086–1092.

[22] S. E. Webster, R. M. Eustice, C. Murphy, H. Singh, and L. L. Whitcomb, "Toward a platform-independent acoustic communications and navigation system for underwater vehicles," in *Proc. IEEE/MTS OCEANS Conf. Exhib.*, Biloxi, MS, USA, Oct. 2009, pp. 1–7.

[23] T. Schneider and H. Schmidt, "Unified command and control for heterogeneous marine sensing networks," *J. Field Robot.*, vol. 27, no. 6, pp. 876–889, 2010.

[24] P. K. Karn, "MACA—A new channel access method for packet radio," in *Proc. 9th ARRL Comput. Netw. Conf.*, London, ON, Canada, 1990, pp. 134–140.

[25] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A medium access protocol for wireless LAN's," in *Proc. ACM SIGCOMM Conf.*, Aug. 1994, pp. 212–225.

[26] M. Chitre, S. Shahabudeen, L. Freitag, and M. Stojanović, "Recent advances in underwater acoustic communications & networking," in *Proc. IEEE OCEANS Conf.*, Quebec City, QC, Canada, Sep. 2008, DOI: 10.1109/OCEANS.2008.5289428.

[27] J. A. Rice, "US Navy Seaweb development," in *Proc. MOBICOM, WUWNet*, Montréal, QC, Canada, 2007, pp. 3–4.

[28] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proc. SIGCOMM*, Karlsruhe, Germany, Aug. 2003, pp. 27–34.

[29] K. Scott and S. Burleigh, "Bundle protocol specification," Internet Engineering Task Force, RFC 5050 (Experimental), 2007.

[30] A. S. Huang, E. Olson, and D. Moore, “LCM: Lightweight communications and marshalling,” in *Proc. Int. Conf. Intell. Robots Syst.*, Taipei, Taiwan, Oct. 2010, pp. 4057–5062, DOI: 10.1109/IROS.2010.5649358.

[31] J. M. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.

[32] A. Said and W. A. Pearlman, “A new, fast, and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 243–250, Jun. 1996.

[33] J. Tian and R. Wells, “Embedded image coding using wavelet difference reduction,” in *Wavelet Image and Video Compression*, ser. Engineering and Computer Science, P. Topiwala, Ed. Amsterdam, The Netherlands: Springer-Verlag, 2002, vol. 450, pp. 289–301.

[34] J. S. Walker, “Lossy image codec based on adaptively scanned wavelet difference reduction,” *Opt. Eng.*, vol. 39, no. 7, pp. 1891–1897, 2000.

[35] W. A. Pearlman and A. Said, “Set partition coding: Part I of set partition coding and image wavelet coding systems,” in *Foundations and Trends in Signal Processing*. New York, NY, USA: Now Publishers, 2008, pp. 95–180.

[36] H. Singh, A. Can, R. Eustice, S. Lerner, N. McPhee, O. Pizarro, and C. Roman, “Seabed AUV offers new platform for high-resolution imaging,” *EOS, Trans. AGU*, vol. 85, no. 31, pp. 289,294–295, Aug. 2004.

[37] C. Murphy and H. Singh, “Wavelet compression with set partitioning for low bandwidth telemetry from AUVs,” in *Proc. 5th ACM Int. Workshop UnderWater Networks Conf.*, 2010, DOI: 10.1145/1868812.1868813, article 1.

[38] C. Murphy and H. Singh, “Fully embedded wavelet compression for low bandwidth image telemetry,” in *Proc. MOOS Develop. Appl. Working Group*, Cambridge, MA, USA, Jul. 2011, vol. 2 [Online]. Available: http://oceanai.mit.edu/moos-dawg11/pmwiki/pmwiki.php?n=Talk.ListingSorted

[39] R. M. Eustice, H. C. Brown, and A. Kim, “An overview of AUV algorithms research and testbed at the University of Michigan,” in *Proc. IEEE/OES Autonom. Underwater Veh. Conf.*, Woods Hole, MA, Oct. 2008, DOI: 10.1109/AUV.2008.5290531.

[40] A. Cohen, I. Daubechies, and J.-C. Feauveau, “Biorthogonal bases of compactly supported wavelets,” *Commun. Pure Appl. Math.*, vol. 45, no. 5, pp. 485–560, 1992.

[41] R. M. Eustice, H. Singh, and L. L. Whitcomb, “Synchronous-clock one-way-travel-time acoustic navigation for underwater vehicles,” *J. Field Robot.*, vol. 28, Special Issue on State of the Art in Maritime Autonomous Surface and Underwater Vehicles, no. 1, pp. 121–136, Jan./Feb. 2011.

Chris Murphy (S’05–M’12) received the B.S. degree in electrical and computer engineering from Olin College of Engineering, Needham, MA, USA, in 2006 and the M.S. and Ph.D. degrees in computer science and oceanographic engineering from the Joint Program in Oceanography between the Massachusetts Institute of Technology, Cambridge, MA, USA and Woods Hole Oceanographic Institution, Woods Hole, MA, USA, in 2008 and 2012, respectively.

He is now researching and developing software with Bluefin Robotics Corporation, Quincy, MA, at the intersection of seafloor imaging, subsea networking, and autonomy. He is also a Visiting Scholar at the Olin College of Engineering.

Jeffrey M. Walls (S’11) received the B.S. degree in mechanical engineering from the University of Virginia, Charlottesville, VA, USA, in 2009 and the M.S. degree in mechanical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2011, where currently, he is working toward the Ph.D. degree at the Department of Mechanical Engineering.

His research interests are in the areas of cooperative navigation and coordinated planning for autonomous underwater vehicles.

Toby Schneider (M’05) received the B.A. degree in physics from Williams College, Williamstown, MA, USA, in 2007 and the Ph.D. degree in oceanographic engineering from the Joint Program between the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA and the Woods Hole Oceanographic Institution (WHOI), Woods Hole, MA, USA, in 2013.

He is currently a Postdoctoral Associate in the Laboratory for Autonomous Marine Sensing Systems (LAMSS), MIT, working on problems at the interface of marine robotic autonomy and communications.

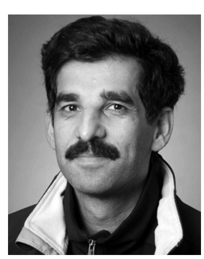
Ryan M. Eustice (S’00–M’05–SM’10) received the B.S. degree in mechanical engineering from Michigan State University, East Lansing, MI, USA, in 1998 and the Ph.D. degree in ocean engineering from the Joint Program between the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA and the Woods Hole Oceanographic Institution (WHOI), Woods Hole, MA, USA, in 2005.

Currently, he is an Assistant Professor with the Department of Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor, MI, USA, with joint appointments in the Department of Electrical Engineering and Computer Science and in the Department of Mechanical Engineering. His research interests include autonomous navigation and mapping, computer vision and image processing, mobile robotics, and autonomous underwater vehicles.

Milica Stojanovic (SM’08–F’10) graduated from the University of Belgrade, Belgrade, Serbia, in 1988 and received the M.S. and Ph.D. degrees in electrical engineering from Northeastern University, Boston, MA, USA, in 1991 and 1993, respectively.

After a number of years with the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, where she was a Principal Scientist, she joined the faculty of Electrical and Computer Engineering Department, Northeastern University, Boston, MA, USA, in 2008. She is also a Guest Investigator at the Woods Hole Oceanographic Institution (WHOI), Woods Hole, MA, USA, and a Visiting Scientist at MIT. Her research interests include digital communications theory, statistical signal processing and wireless networks, and their applications to underwater acoustic communication systems.

Dr. Stojanovic is an Associate Editor for the IEEE Journal of Oceanic Engineering and the IEEE Transactions on Signal Processing.



Hanumant Singh (S’87–M’95) received the Ph.D. degree from the Joint Program between the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA and the Woods Hole Oceanographic Institution (WHOI), Woods Hole, MA, USA, in 1995.

He is currently an Associate Scientist at the Woods Hole Oceanographic Institution (WHOI), Woods Hole, MA, USA. His research interests lie in the area of underwater imaging and underwater robotics. The Seabed class of autonomous underwater vehicles developed under his direction is in use in every one of the world’s oceans for a variety of oceanographic applications.