Wireless Sensor Network For Highly Critical Applications

[1] S. Rakesh, [2] D. Jeevarani,[3] R. Manjupriya.

[1]Assistant Professor, Adithya Institute of Technology

[2]Pg Scholar , Adithya Institute of Technology

[3]Pg Scholar , Adithya Institute of Technology

Abstract- Wireless Sensor Networks(WSNs) are generally designed to support applications in long-term deployments, and thus WSN are primarily designed to be energy efficient and long range communication for critical applications. However, the research community has recently explored new WSN application such as Underwater Wireless Sensor Networks (UWSN). Ocean monitoring helps us to detect tectonic movements, incoming tsunamis, water pollution and global warming. we could spread a number of sensor nodes in the water, some on the sea floor, some floating at different depths, and others moving around, and these devices were able to communicate to each other, autonomously organize themselves into a network, exchange data, and eventually deliver it to a collection point where it can be easily and cheaply accessed or transmitted. The major outcome will be a complete solution for a communications and Networking architecture for an underwater sensing, monitoring and exploration system comprising intelligent robots that need to perform collaborative missions. The final solution will be based on very innovative communications and protocol solutions, as well as on a deep understanding of the application needs and features such as swarm behaviors, collaborative missions, cooperative navigation. The main area of research will include underwater channel models, efficient acoustic communications, Networking protocols (multiple access, routing, transport), and cross-layer protocol design.

KEYWORDS: Wireless Sensor Networks,acoustic communications, Networking protocols, cross-layer protocol design, autonomous vehicles,routing protocols.

I.INTRODUCTION

Initial time ocean life and the underwater world have fascinated humankind's imagination. Submarine explorations discover about new forms of life and animal and botanical species. Underwater communication systems today mostly use acoustic technology. Acoustic communications offer longer ranges, but are constrained by three factors: limited and distance-dependent bandwidth, time-varying multipath propagation, and low speed of sound. Together, these constraints result in a communication channel of poor quality and high latency, thus combining the worst aspects of terrestrial mobile and satellite radio channels into a communication medium of extreme difficulty.

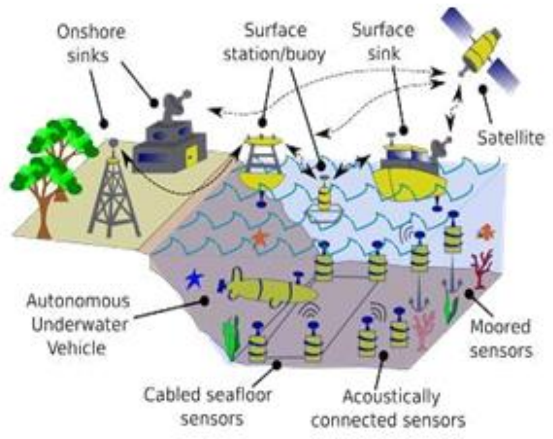

Fig.1 Underwater Sensor Network Communication

Underwater instruments (sensors, robots, modems, batteries) are neither cheap nor disposable. The underlying systems include fleets of cooperating autonomous vehicles and long-term deployable bottom-mounted sensor networks. Underwater networks are often static.

The topologies of these networks are static for long durations, allowing engineering of the network topology to promote connectivity.

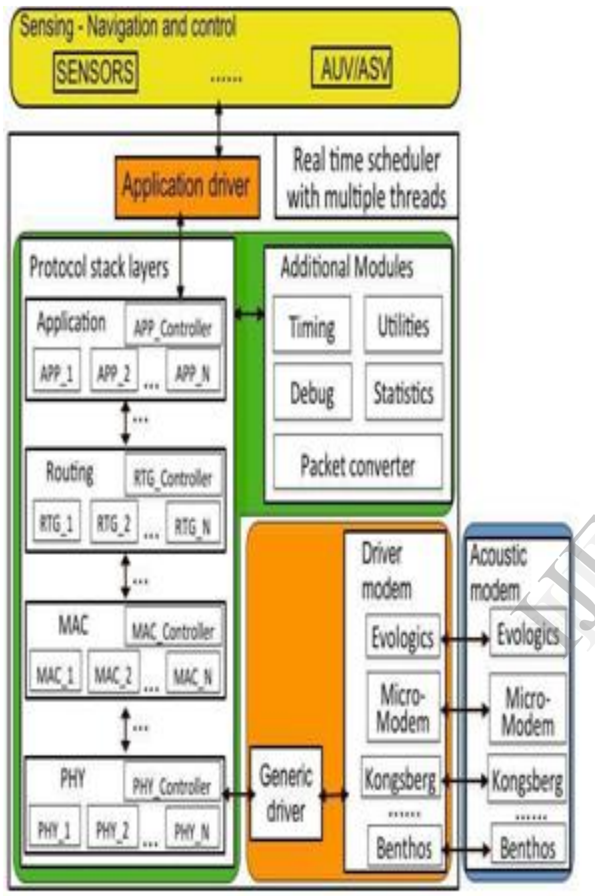

Fig. 2 Underwater wireless sensor network architecture

Networking Components include:

• MAC: CSMA, Aloha, CSMA-Aloha,Slotted-CSMA, TDMA, CSMA, Tone-Lohi, DACAP and PDAP

• Routing: Flooding, probabilistic flooding,static routing.

• Cross-layering: Channel-Aware RoutingProtocol (CARP)

II.NETWORKING TECHNOLOGIES

1. Physical Layer

In this layer, water absorbs and disperses almost all electro-magnetic frequencies, making acoustic waves a preferred choice for underwater communication beyond tens of meters. Propagation of acoustic waves in the frequency range of interest for communication can be described in several stages. The first, basic stage, takes into account this fundamental loss that occurs over a transmission distance d. The second stage takes into account the site specific loss due to surface-bottom reflections and refraction that occurs as sound speed changes with depth, and provides a more detailed prediction of the acoustic field around a given transmitter. The third stage addresses the apparently random changes in the large-scale received power which are caused by slow variations in the propagation medium (e.g., tides).
Drivers: Several external devices are: 1)Acoustic modems (WHOI Micro-Modem; Evologics modem, Kongsberg modem and Teledyne Benthos modem); 2) Sensing platforms (temperature, CO2 and methane concentrations, Acoustic Doppler Current Profiler - ADCP); 3) Mobile vehicles (MARES AUV and INESC/TEC ASV, eFolaga AUV).

2. Medium Access Control and Resource Sharing

Protocols specifically designed for underwater networks following the CSMA/CA approach are DACAP and T-Lohi. DACAP is based on an initial signaling exchange in order to reserve the channel, thereby decreasing the probability of collision. T-Lohi exploits collision avoidance tones, that transmits signals to their intention by sending narrowband signals, and proceed with data transmission. CSMA-based protocol that uses synchronization to reduce the probability of collision. These protocols are used in data link layer as in [8].

A. CSMA (Carrier Sensing MultipleAccess) is a well-known protocol for channel access. When a node has a data packet to transmit, it first checks whether the channel is idle or busy. In the first case, it starts the packet transmission. We consider two versions of this protocol. The first is the following: If the ACK is not received within a given time (set to 2Delay+ackTime), the data packet is re-transmitted either till successful reception, every time choosing the backoff time in an interval twice as long as the previous one, or till the maximum limit of retries has been reached. The backoff time is chosen randomly and uniformly in [0,T], where, T=2txRetry(2maxDelay),as in[2][7]. Retransmission of the same packet stops after a predefined number of times.

B. ALOHA is the well-known protocol forchannel access. We here consider carrier-sensing ALOHA. When a node has a data packet to transmit, it first checks whether the channel is idle or busy,as in [2][3]. We consider two versions of this protocol. The first is the following: A node that transmits a data packet receives no feedback about whether the intended recipients has received it or not. The second adds robustness by having the destination node acknowledging the data reception to its source. If the ACK is not received within a given time (set to 2Delay + ackTime), the data packet is retransmitted either till successful reception, every time choosing the backoff time in an interval twice as long as the previous one, or till the maximum limit of retries has been reached. Here Delay is the transmission delay between source and destination. Its value is initially set to maxDelay and successively set by the nodes to a value computed according to the (estimated) distance between source and destination (which is based on the time difference between data packet transmission and ACK reception). The backoff time is chosen randomly and uniformly in [0, *T*], where *T* = 2txRetry (2maxDelay + data Time).

C. Slotted ALOHA. We consider theversion of slotted ALOHA, adding the feature of carrier sensing. Time is divided into slots that start at the same time for all nodes. As in we set the slot duration σ to βmaxDelay + data Time, where β is thefraction of the maximum propagation delay that a node waits after transmitting the packet. Choosing $\beta = 1$ ensures that no collision occurs at the receiver. The packet is transmitted in the first case. In the second, instead, the nodes use the exponential backoff mechanism of ALOHA protocols, now counted in number of slots. In this case, the slot duration σ is set to 2βmaxDelay + data Time + ackTime. More precisely, the number of slots that a node waits before trying to retransmit a packet is chosen randomly and uniformly in [0, *T*], where $T = \sigma$2txRetry,as in[2].

D. APCAP, The Adaptive Propagation-Delay Collision Avoidance Protocol (APCAP), uses the RTS/CTS scheme to reserve channel and send data. All nodes are synchronized. When a node has a data packet to transmit, it sends an RTS packet (13Bytes long). The APCAP mechanism
adopts an aggressive approach to channel access, in the sense that a node that has scheduled a given transmission does not delay or cancel,as in[2] it if it is made aware of other communications going on in the channel. When a node that has scheduled a transmission senses that another transmission is ongoing, that transmission could be finished already and therefore there is no reason to delay its own sending.

E. DACAP stands for Distance AwareCollision Avoidance Protocol. This protocol, uses the RTS/CTS scheme to reserve the channel and for transmitting data packets. When a node has a data packet to send it transmits an RTS (6Bytes). It then waits for the data packet. If during this time it hears a control packet for some other node, it sends a very short (3Bytes) WARNING packet to its sender,as in [2][7]. If it hears another control packet or receives a WARNING from the destination during this time, the node aborts the packet transmission. The length of the WARNING time depends on the distance between the source and destination. The sender can compute this distance by measuring the RTS/CTS round-trip time.

F. T-Lohi. T-Lohi is a protocol for single-hop underwater networks. When a node has a data packet to transmit, before the actual transmission it starts a reservation period (RP). An RP is made up of a certain number of slots called contention rounds (CRs). During a CR, the sender transmits a short 3Bytes control packet (tone packet) to inform other nodes about its need to access the channel.If contention occurs, the contenders back off for a number of CR chosen randomly and uniformly between 0 and the number of competitors,as in[2]. The duration of a CR is appropriately set so that a node has enough time to detect as many contenders as possible. In the aggressive T-Lohi the CR lasts for the time needed totransmit a tone packet plus the maximum propagation delay.

G. PDAP. The Propagation Delay AwareProtocol (PDAP) uses the RTS/CTS mechanism for channel reservation and transmission. All nodes are synchronized. The time taken for receiving the CTS must start after the time needed for the RTS to reach the destination plus the time for the CTS to get back,as in[2]. Before sending the RTS, the sender waits for a random time to avoid synchronization with other potential senders. This waiting time is randomly and uniformly chosen in [0, *T*], where *T* = 2txRetry (2maxDelay + rtsTime). Both RTS and CTS control packets contain information about the distance between the source and destination. When the interferer receives an RTS packet, it sets its NAV so that it will not be transmitting control information that would arrive at the sender at the scheduled CTS reception time.

3. The Network Layer

Early work on underwater routing includes where distributed protocols are proposed for both delay-sensitive and delay insensitive applications and allow nodes to select the next hop with the objective of minimizing the energy consumption. Protocols such as TCP are designed for low to moderate latencies, not the large fractions of a second commonly encountered in underwater networks, and limited bandwidth and high loss suggest that end-to-end retransmission will perform poorly.Network coding and forward-error correction can also be employed to cope with losses given long delays; coding benefits from optimizing coding and feedback. Different approaches such as Delay Tolerant Networking may be a better match to many underwater networks, by avoiding end-to-end retransmission and supporting very sparse and often disconnected networks.

Network Services:

Of the many network services that are possible, localization and time synchronization have seen significant research because of their applicability to many scenarios. localization often estimates communication time-of-flight, assuming accurate clocks; time synchronization estimates clock skew, modeling slowly varying communication delays. Time-Synchronization for High Latency networks (TSHL) showed that clock drift during message propagation dominates the error for acoustic channels longer than 500m. Sinks may then forward the received information to onshore control stations. Cross layer techniques can impact protocol performance positively, using Channel-aware Routing Protocol (CARP),as in [3].

Channel-aware Routing Protocol (CARP):

CARP is an effective solution for transmitting packets through time-varying channels, capable of routing around connectivity holes and shadow zones, and to maintain a high packet delivery ratio for increasing traffic as in[8].

CARP Protocol Description:

At network set up, HELLO packets are flooded from the sink through the network. In this way,as in [3][8] every node x acquires its hop count HC(x).When a node x has one or more data packets to forward it chooses a suitable relay among its neighbors. The search for the relay is initiated by x by broadcasting a control packet, called PING, which carries the following information.

< src, numpkt >.

The field src is x unique identifier, and numpkt is the number of packets that x has to transmit. If numpkt > 1 a train of packets is transmitted back to back.

A node y that receives the PING packet immediately replies with a PONG packet, directly transmitting it to the PING source x

(unicast communication). The PONG packet carries the following information.

< src, dst, hop, queue, energy, lq >.

The goodness value of each node being calculated as for each responding y, node x computes:

goodnessy = lqylqx, y.

Metrics of interest

The metrics of using Channel-awareRouting Protocol (CARP) are

a. Throughput Efficiency

It is of the three considered protocols for increasing traffic λ. As expected, when the traffic increases the packet delivery ratio decreases. The increased traffic results in higher probability of packet collisions and re-transmissions as well as in[6] a higher number of times the nodes find the channel busy. CARP creates link quality-based relay selection and also because data are forwarded on links that are robust for both control and data packets.

b. End to End Latency Per Meter

CARP shows the best latency per meter performance, due to its relay selection based on link quality and the fact it considers robust links for both control and data packets, as in[6], which keeps interference and retransmission at bay. c.Energy Consumption Per Bit. The energy spent for delivering one bit to the sink correctly. When traffic increases more bits are correctly delivered to the sink, and although energy consumption increases and packet delivery ratio decreases, as in[6] the consumption per bit decreases. since it correctly delivers to the sink a lower number of bits and since each bit travels longer routes than those of CARP; its energy demands are higher.

4) Application Layer

Metrics used in application layer are

a. Timing module: It is responsible tocompute the delays related to the use of real devices. When in simulation mode, the Timing module returns the simulated values defined by the user,as in [4][1]. When running in emulation mode instead it collects these delays from the real hardware inquiring the corresponding drivers. When running simulations, the delays related to the use of real devices are usually ignored. These delays include: 1) Computational delay; 2) Operational delay related to internal operations of the considered device 3) Delays related to the communication between different hardware components on the same node. Moreover, it is not always true that the message size sent by the external device exactly corresponds to the original message size, because of coding performed by the device.

b. Utilities module: It is responsible toschedule events and update timing information according to the used scheduler, when running in simulation mode, and the real-time scheduler, when running in emulation mode,as in [4][1]. The utilities module is also responsible to properly free the memory allocated for an ns-2 packet and its payload, if any. Timing and Utilities modules are used to make transparent to the users the use of the framework in simulation and emulation mode. The utilities module is also responsible to properly deallocate the memory allocated for an ns-2 packet, schedule events and update timing information

c. Information dispatcher module: Itimplements a publish subscribe pattern, allowing each module to request or provide information to the other modules without the need to send cross-layer messages across the entire protocol stack, as it is done by ns2-Miracle,as in [4][1]. Each information is timestamped in order to check its freshness and it can work together with the ns2-Miracle cross-layer messaging system.

d. Debug module: It implements a moreflexible system to log the debugging information according to their priorities. We have developed and implemented a debug module which allows to assign to each information to log a debug level. When the user starts the experiment it can define the debug level of the information it wants to log. This module also allows to change the debug level on demand thus increasing and reducing the logged information according to the user needs

e. Statistics module: This module providesbasic functionalities to evaluate the protocol performance. This module allows the user to collect all the information about packet transmissions and receptions, energy consumption, additional delays (back off period, device delays, etc.) and others. When running in simulation mode, the statistics module collects the information from all the nodes in the network and all the required metrics are available at the end of the simulation. When running in emulation mode, however, each node is an independent unit, storing its own information.

f. Energy module: It implements a moreaccurate energy model to trace the energy consumption of underwater devices, supporting also the use of different transmission powers.

g. Packet converter module: Each ns-2packet header is a data structure containing all the information used by the specific protocol layer. When simulating packet transmissions the exchanged packet header size does not need to match the actual size of the header according to the packet data structure,as in [4][1]. Our approach is flexible enough to allow the user to decide through the Tcl script what are the packet header information to be considered or discarded in the conversion process and what size has to be assumed for the different values.

h. Application drivers: The UWSNarchitecture is flexible enough to allow fast integration not only with real acoustic modems but also with different types of devices, such as sensors for underwater measurements and AUV (Autonomous Underwater Vehicle) or ASV (Autonomous Surface Vehicle) navigation control systems. The application driver is a generic interface which provides the methods to set and get parameters from the connected device and to execute the desired actions,as in[4][1]. Using the acoustic communications and networking capabilities requests and commands can be delivered to a remote node thus allowing remote control of the device using acoustic links in a real-time and on-line way.

The ns-2 and ns2-Miracle simulators

Ns-2 is a well established and reliable discrete event network simulator, which is widely used by the academic networking research community for its extensibility (due to its open source model) and rich online documentation.

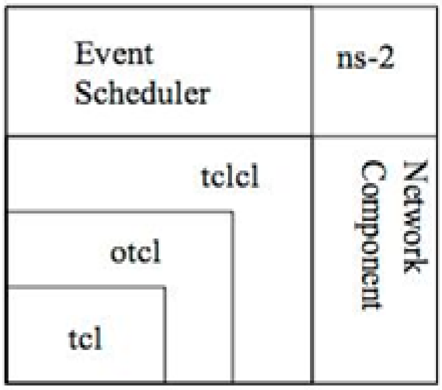

Figure 3: Network simulator architecture

Figure 3 presents the main ns-2 components. It uses a discrete event scheduler and *Tcl* as its scripting language. *OTcl*, an object-oriented dialect of Tcl, is used to execute command scripts, while *Tclcl* is used to link C++ and OTcl classes. The user describes a network topology by writing Tcl scripts. If two events are in the scheduler, one scheduled for execution at time T and the other at time T0 > T,as in [4] and there are no events to be scheduled between T and T0, after the scheduler executes the event at T, it instantaneously changes the simulator time to T0 and executes the event at T0. Ns-2 supports also the use of a basic real-time scheduler which forces the system to wait for the actual T0−T seconds before scheduling the next event and supports cross-layer messages communication between different layers.

III.APPLICATIONS

1) Scientific applications observe theenvironment: from geological processes on the ocean floor, to water characteristics (temperature, salinity, oxygen levels, bacterial and other pollutant content, dissolved matter, etc.) to counting or imaging animal life (microorganisms, fish, or mammals).

2) Industrial applications monitor andcontrol commercial activities, such as underwater equipment related tool or mineral extraction, underwater pipelines, or commercial fisheries. Industrial applications often involve control and actuation components as well.

3) Military and homeland security applications involve securing or monitoringport facilities or ships in foreign harbors, de-mining,and communication with submarines and divers.

IV.CONCLUSION

Thus this paper presented a detailed explanation of underwater Wireless Sensor Network (UWSN) architecture and different modules occurred in UWSN. The main area of research in this paper will include underwater channel models, efficient acoustic communications, networking protocols(multiple access, routing, transport), and cross-layer protocol design. The major outcome will be a complete solution for communication and networking architecture for an underwater sensing, monitoring and exploration system comprising intelligent robots that need to perform collaborative missions.

REFERENCES

[1] SUNSET: Simulation, Emulation and Real-life Testing of Underwater Wireless Sensor Networks, Chiara Petrioli, Roberto Petroccia, Dipartimento di Informatica Universit`a di Roma “La Sapienza,” Roma, Italy {petrioli,petroccia}@di.uniroma1.it Consorzio Interuniversitario Nazionale per l’Informatica (CINI), Roma, Italy

[2] A Comparative Performance Evaluation of MAC Protocols for Underwater Sensor Networks. Chiara Petrioli and Roberto Petroccia Dipartimento di Informatica Universit`a di Roma “La Sapienza” Roma, Italy, Milica Stojanovic Massachusetts Institute of Technology Boston, MA, U.S.A.

[3] Channel-aware Routing for Underwater Wireless Networks. Stefano Basagni_, Chiara Petrioli, Roberto Petroccia and Daniele Spaccini, Dept. of Electrical and Computer Engineering Northeastern University, Boston, MA, U.S.A.

[4] Comparing the SUNSET and DESERT Frameworks for In Field Experiments in Underwater Acoustic Networks. Roberto Petroccia and Daniele Spaccini, Dipartimento di Informatica Universit`a di Roma “La Sapienza,” Rome, Italy {petroccia, spaccini}@di.uniroma1.it WSENSE s.r.l., Rome, Italy

[5] Optimized Packet Size Selection in Underwater Wireless Sensor Network Communications. Stefano Basagni, *SeniorMember*, *IEEE*, Chiara Petrioli, *Senior Member*, *IEEE*, Roberto Petroccia, *Member*, *IEEE*, and Milica Stojanovic, *Fellow*, *IEEE*

[6] Optimizing Network Performance through Packet Fragmentation in Multi-hop Underwater

Communications. Stefano Basagni, Chiara Petrioli, Roberto Petroccia and Milica Stojanovic ECEDepartment.NortheasternUniversity,{basagni,millitsa}@ece.neu.eduDipartimentodiInformatica Universit`a di Roma “La Sapienza”{petrioli, petroccia}@di.uniroma1.it

[7] Choosing the Packet Size in Multi-hop Underwater Networks. Stefano Basagni, Chiara Petrioli, Roberto Petroccia and Milica Stojanovic, ECE Department Northeastern University {basagni, millitsa}@ece.neu.eduDipartimentodi Informatica Universit`a di Roma “La Sapienza”{petrioli, petroccia}@di.uniroma1.it

[8] A Study on Channel Dynamics Representation and its Effects on the Performance of Routing in Underwater Networks. Paolo Casari, Daniele Spaccini, Giovanni Toso, Beatrice Tomasi,Roberto Petroccia, Chiara Petrioli, Michele Zorzi,Department of Information Engineering, University of Padova, via Gradenigo 6/B, I-35131 Padova, Italy Department of Computer Science, University of Rome “