ISSN: 0975-766X
CODEN: IJPTFI
Available Online through ***Research Article***
www.ijptonline.com

IMPROVING ENERGY EFFICIENCY OF UNDERWATER SENSOR USING DEPTH BASED ROUTING PROTOCOL

V.Parthipan[1], Dr.D.Dhanasekaran[2], S.A.Rathiya[3], Arun Prakaash.L.V[4]
Department of Computer Science and Engineering, Saveetha School of Engineering,
Saveetha University, Chennai, India.
Email: parthipan@saveetha.com

Received on 10-08-2016 *Accepted on 06-09-2016*

Abstract

Providing scalable and efficient routing services in underwater sensor networks (UWSNs) is very challenging due to the unique characteristics of UWSNs. Firstly, UWSNs often employ acoustic channels for communications because radio signals do not work well in water. Compared with radio-frequency channels, acoustic channels feature much lower bandwidths and several orders of magnitudes longer propagation delays. Secondly, UWSNs usually have very dynamic topology as sensors move passively with water currents. Some routing protocols have been proposed to address the challenging problem in UWSNs. In this paper, we propose a depth-based routing (DBR) protocol. DBR does not require full-dimensional location information of sensor nodes. Instead, it needs only local depth information, which can be easily obtained with an inexpensive depth sensor that can be equipped in every underwater sensor node.

Keywords: Underwater sensor networks (UWSNs), depth-based routing (DBR), sink node, sensor node, (GPSR) Greedy Perimeter Stateless Routing protocol, (NS) Network simulator.

I. Introduction

A Underwater Wireless Sensor Networks (UWSNs) consists of significant number of sensor nodes deployed at different depths throughout the area of interest. Sensor nodes are located at the sea bed, they cannot communicate directly with the surface nodes. Sensor networks used for underwater communications are different in many aspects Energy consumptions are different due to some applications require large amount of data. These networks are usually work on a common task instead of representing independent users.

II. Related Work

The existing system of our project uses two types of routing protocols one is for land based sensor networks another one is geographic based routing protocols. In the land based sensor networks the routing may be initiated by sensor node or sink node. The A sink first floods its interest for data across the network. Upon receiving an interest, a sensor node responds with data along all possible paths.

Then an optimal path is chosen from the sensor node to the sink. But it requires periodic update of routing information which leads to significant communication overhead, low energy efficiency, and high end-to-end delays. In the geographic based routing uses the GPSR. A GPSR protocol has two components:greedyalgorithm,and a recovery algorithm. In the greedy algorithm, a forwarding node tries to forwardpackets toone of its one-hop neighboring nodes which has the shortest distance to the destination node. It is possible that a forwarding node itself has the shortest distance to the destination, resulting in a "void" zone between the forwarder and the destination in which none of the forwarders' neighborsresides. In such cases, a recovery algorithm is performed to route the packet around the "void" zone until the routing can continue in the greedy mode.

Limitation of existing system:

- Communication overhead is high.
- Requires full dimensional location information.
- Energy consumption of the network become increased.
- Time consuming process.

III. Proposed System:

The proposed system of our project uses the two types of nodes. One is sink node another one is sensor node. The sink node equipped with both radio-frequency (RF) and acoustic modems which is deployed at water surface. The sensor nodes are also equipped with acoustic modems. The sink nodes can communicate with each other using radio frequency. Here we consider communication between sink nodes and sensor nodes. Furthermore, assume that each underwater node knows its depth information, namely the vertical distance from itself to the water surface.

In practice, depth information can be obtained easily with a depth sensor. In DBR, upon receiving a packet, a node first retrieves the depth of the packet's previous hop, which is embedded in the packet. The receiving node then compares its own depth with depth of the packet's previous hop.

If the node is closer to the water surface it will consider itself as a qualified candidate to forward the packet. Otherwise, it just simply drops the packet. Same packet can be received by a node multiple times because of availability of multiple paths. This leads to collision and energy consumption.

This is avoided by using priority queue. This reduces the number of forwarding nodes, and thus controls the number of forwarding paths. And then packet history buffer is used in DBR to ensures that a node forwards the same packet only once in a certain time interval.

Advantages

- Communication overhead is reduced.
- Does not require full dimension al location information.
- Energy consumption of the network becomes decreased.
- Time consumption is reduced.
- Packet drop is low.

Ns 2- Network Simulator:

Network simulator, a package of tools that simulates behaviour of networks. Ns are a discrete event simulator targeted at networking research. Ns provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. Ns began as a variant of the REAL network simulator in 1989 and have evolved substantially over the past few years.

In 1995 ns development was supported by DARPA through the VINT project at LBL, Xerox PARC, UCB, and USC/ISI. Currently ns development is support through DARPA with SAMAN and through NSF with CONSER, both in collaboration with other researchers including ACIRI. Ns have always included substantial contributions from other researchers, including wireless code from the UCB Daedelus and CMU Monarch projects and Sun Microsystems. For documentation on recent changes, see the version change log. Ns (from network simulator) is a name for series of discrete event network simulators, specifically ns-1, ns-2 and ns-3. All of them are discrete-event network simulator, and teaching. ns-3 is free software, publicly available under the GNU GPLv2 license for research, development, and use.

The goal of the ns-3 project is to create an open simulation environment for networking research that will be preferred inside the research community The core of the simulator was written in C++, with Tcl-based scripting of simulation scenarios.[5]Long-running contributions have also come from Sun Microsystems, the UC Berkeley Daedelus, and Carnegie Mellon Monarch projects. In 1996-97, ns version 2 (ns-2) was initiated based on a refactoring by Steve Mc Canne. Use of Tcl was replaced by MIT's Object Tcl (OTcl), an object-oriented dialect of Tcl.

The core of ns-2 is also written in C++, but the C++ simulation objects are linked to shadow objects in OTcl and variables can be linked between both language realms. Simulation scripts are written in the OTcl language, an extension of the Tcl scripting language. Presently, ns-2 consists of over 300,000 lines of source code, and there is probably a comparable amount of contributed code that is not integrated directly into the main distribution (many forks of ns-2 exist, both maintained and unmaintained). It runs on GNU/Linux, FreeBSD, Solaris, Mac OS X and Windows versions that support Cygwin. It is licensed for use under version 2 of the GNU General Public License.

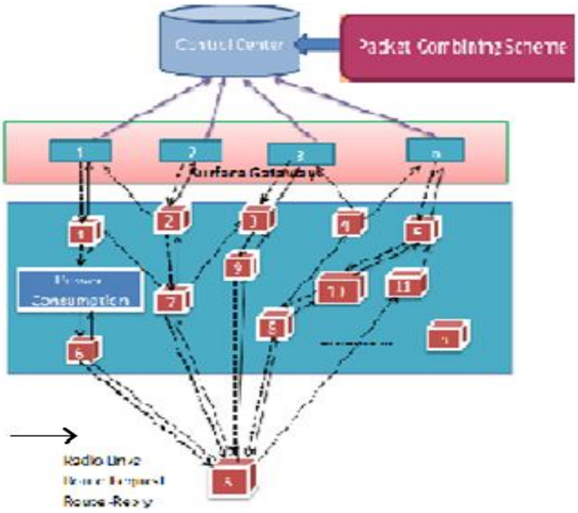
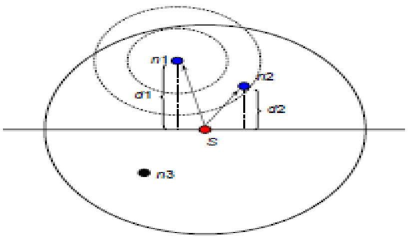


Fig.1 Architecture Diagram.

Ns2- Network Simulator

Network simulator, A package of tools that simulates behavior of networks. Ns are a discrete event simulator targeted at networking research. Ns provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. Ns began as a variant of the REAL network simulator in 1989 and have evolved substantially over the past few years. In 1995 ns development was supported by DARPA through the VINT project at LBL, Xerox PARC, UCB, and USC/ISI.



Currently ns development is support through DARPA with SAMAN and through NSF with CONSER, both in collaboration with other researchers including ACIRI. Ns have always included substantial contributions from other researchers, including wireless code from the UCB Daedelus and CMU Monarch projects and Sun Microsystems. For documentation on recent changes, see the version change log. Ns (from network simulator) is a name for series of discrete event network simulators, specifically ns-1, ns-2 and ns-3. All of them are discrete-event network simulator, and teaching. ns-3 is free software, publicly available under the GNU GPLv2 license for research, development, and use. The goal of the ns-3 project isto create an open simulation environment for networking research that will be preferred inside the research community. The core of the simulator was written in C++, with Tcl-based scripting of simulation scenarios. [5]Long-running contributions have also come from Sun Microsystems, the UC Berkeley Daedelus, and Carnegie Mellon Monarch projects. In 1996-97, ns version 2 (ns-2) was initiated based on a refactoring by Steve McCanne. Use of Tcl was replaced by MIT's Object Tcl (OTcl), an object-oriented dialect of Tcl.

TCP/IP model:

a. Application Layer

Application-a-gents are-Telnet, FTP, Rlogin, CBR

b. Transport Layer

The transport uses two protocols, UDP and TCP. UDP, which stands for User Datagram Protocol, does not guarantee packet delivery and applications, which use this, must provide their own means of verifying deliver. TCP does guarantee delivery of packets to the applications, which use it.

c. Network Layer

The network layer is concerned with packet routing and used low-level protocols such as ICMP, IP, and IGMP. Here Network protocols are omitted for simple examples

d. Link Layer

The link layer is concerned with the actual transmittal of packets as well as IP to Ethernet address translation. Data transfer in two modes i.e. simplex mode and in duplex mode. All the links are created at data link layer. Network layer next is to data link layer.

e. Physical Layer

The physical layer consists of the basic networking hardware transmission technologies of a network. It is a fundamental layer 41 underlying the logical data structures of the higher level functions in a network. Due to the plethora of available hardware technologies with widely varying characteristics, this is perhaps the most complex layer in the OSI architecture

Features of NS2

- Protocols: TCP, UDP, HTTP, Routing algorithms, MAC etc
- Traffic Models: CBR, VBR, Web etc
- Error Models: Uniform, burstyetc
- Misc: Radio propagation, Mobility models, Energy Models
- Topology Generation tools
- Visualization tools (NAM), Tracing.

IV. Implementation

- Node formation
- Depth based routing
- Packet transmission
- Graph construction

a. **Node formation:**

In node formation nodes are added to the network and they are in mobile nature. Nodes are free to move and node movements are updated by each node. Which forms the topology .topology is nothing but how the nodes interconnect with each other. First select source and destination node in order to make packet transmission. Then nearest neighbour nodes are chosen for packet forwarding. Consider two types of nodes one is sink node another one is sensor node. Sensor nodes are equipped with acoustic modem. Sink nodes are equipped with acoustic and RF modem. Each sink node can communicate with each other through RF signals.

b. **Depth Based Routing**

Source sensor node forwards the packet to the near bynode.In DBR each node receives the packet from other node and it will calculate its depth. Nodes closer to the water surface will forward the packet. Otherwise it just simply drops that packet. Availability of multiple paths makes the packet collision or same packet may be received by the node multiple times.

To avoid these disadvantages the DBR method uses the priority queue and then packet history buffer.These two scenarios are maintained by each node in the network. Priority queue chooses the node with least depth & reduced traffic. Packet history buffer is used to avoid multiple receptions of same packet.

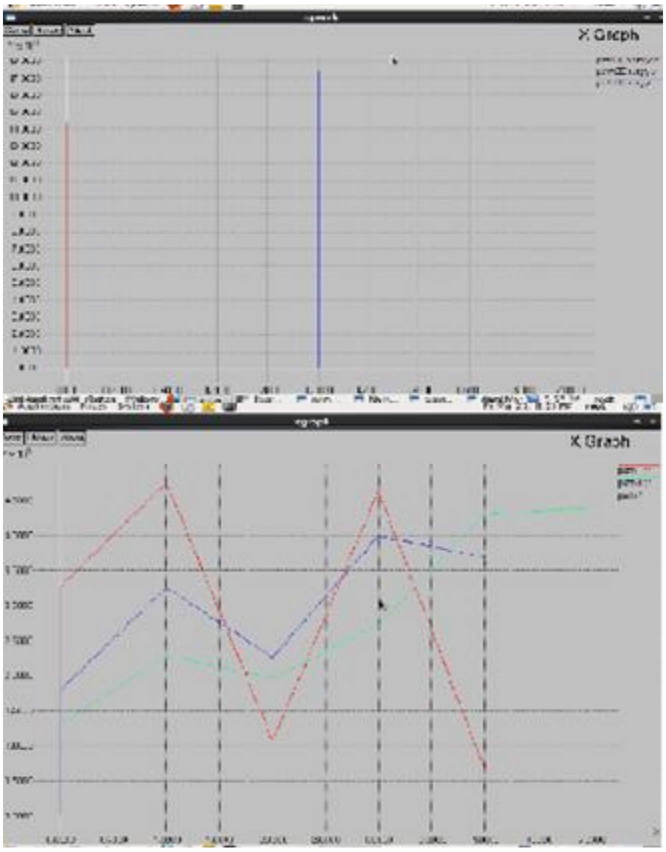
c. **Packet Transmission**

Packet transmission is done by choosing the sender and receiver node. Forwarder node is chosen by DBR method. Each node calculates its depth with the previous node. Then forwards the packet to the next node. Likewise the packet is forwarded to the destination. Packet format consists of sender ID, packet sequence number, depth and then data.

d. **Graph Construction**

Performance evaluation of this project is done by drawing graph for Throughput, Packet delivery ratio, Energy and Time synchronization. Simulation result shows that our proposed system has increased throughput and packet delivery ratio. Energy consumption and time consumption is decreased.

Path Energy & Energy Consumed



Simulation setting

All simulations are performed using the Network Simulator (ns2) with an underwater sensor network simulation package (called Aqua-Sim) extension. In our simulations, sensor nodes are randomly deployed in a 500m×500m×500m 3-D area. Multiple sinks are randomly deployed at the water surface. Assume that the sinks are stationary once deployed, the sensor nodes follow the random-walk mobility pattern. Each sensor node randomly is selected.

Algorithm for packet forwarding in DBR

```
1:Get previous
depth dp from p
2:Get node’s
current depth dc
3: Compute Δd = (dp – dc)
4: IF Δd< Depth Threshold dth THEN
5: IF p is in Q1 THEN
6: Remove <p, Sending
Time> from Q1 7:
ENDIF
8: Drop p
9:    return
10: ENDIF
11: IF p is in Q2 THEN
12: Drop p
13:    return
14: ENDIF
15: Update p with
current depth dc
16: Compute
Holding Time HT
17: Compute
Sending Time ST
18: IF p is in Q1
THEN
19: Get previous sending time of p STp
20: Update p’s Sending
Time with min(ST, STp) 21:
ELSE
22: Add the item <p,
ST> into Q1 23:
ENDIF
```

DBR packet forwarding algorithm

a. Packet Delivery Ratio: The ratio of the number of distinct packets received successfully at the sinks to the total number of packets generated at the source node. Although a packet may reach the sinks multiple times, these redundant packets are considered as only one distinct packet.

b. Average End-to-end Delay: Itrepresents the average time taken by a packet to travel from the source node to any of the sinks.

c. Depth Threshold: When the depth threshold increases, both the packet delivery ratio and the total energy consumption decrease. This is because increasing the depth threshold has a similar effect to reducing the number of available nodes in the network. Therefore, the number of forwarding nodes will decrease. Consequently, the packet delivery ratio decreases and less energy is consumed.

V. Conclusion:

Depth Based Routing (DBR), a routing protocol based on the depth information of nodes, for underwater sensor networks. DBR uses a greedy approach to deliver packets to the sinks at the water surface. Different from other geographical-based routing protocols that require full-dimensional location information of nodes, DBR only needs the depth information, which can be easily obtained locally at each sensor node. Further, DBR can naturally takes advantages of the multiple-sink underwater sensor network architecture without introducing extra cost. Our simulation results have shown that DBR can achieve high packet delivery ratios (at least 95%) for dense networks, with reasonable energy consumption.

VI. References:

1. F.Akyildiz, D.Pompili, and T.Melodia. Challenges for efficient communication in under water acoustic sensor networks. *ACMSIGBED Review*,1(1), July2004.
2. S.Basagni, I.Chlamtac, V.R.Syrotiuk, and B.A.Wood ward. Adistancerouting effect algorithm form obility(dream). *MOBICOM98*, 1998.
3. D.Braginsky and D.Estrin. Rumorrouting algorithm for sensor networks. *WSNA02*, September 2002.
4. J.H.Cui, J.Kong, M.Gerla, and S.Zhou.Challenges: Building scalable mobile under water wireless sensor networks for aquatic applications. *Special Issue of IEEE Networkon Wireless Sensor Networking*, May 2006.
5. M.Grossglauser and M. Vetterli. Locating nodeswithease: Mobility diffusion of last encountersinadhoc networks. *IEEEINFOCOM03*, March2003.
6. J.Heidemann, W.Ye, J.Wills, A.Syed, and Y.Li.Research challenges and applications for under water sensor networking. *IEEE Wireless CommunicationsandNetworkingConference*, April2006.
7. W.R.Heinzelman, J.Kulik, and H.Balakrishnan. Adaptive protocols for information dissem-inationin wireless sensor networks.*Mobicom99*, August1999.
8. C.Intanagonwiwat, R.Govindan, and D.Estrin. Directed diffusion: A scalable androust communication paradigm for sensor networks. *MOBICOM00*, August 2000.
9. B.Karpand H.T.Kung. Gpsr: Greedy perimeter stateless routing for wireless networks.*Proc.Mobicom*,2000.
10. Linkquest. http://www.link-quest.com/.
11. N.Nicolaou, A.See, P.Xie, J.-H.Cui, and D.Maggiorini. Improving the robustness of location-based routing for under water sensor networks. *Proceedings of IEEE OCEANS'07*, June 2007.