TDMA-based Protocol for Multihop Underwater Wireless Networks

1 Proposed Protocol

There are n nodes in the network, denoted as $N = \{n_i \mid i = 0..n-1\}$. Note that i represents the ID of node n_i. We assume n_0 is the *master* node. The *master* node is wired to the cable, and hence has plenty of power and bandwidth resources. The remaining nodes are wireless nodes, and use light waves to communicate with each other. We assume that communication is half-duplex, i.e., nodes can either send a packet or listen to the channel, but cannot do both at the same time.

The rough idea behind our algorithm is sketched as follows, while the detailed protocol is illustrated in Algorithm 1:

1. Construct a tree with n_0 as the root in a level by level, as well as TDMA fashion. Each node acquires its parent and children information.
2. Starting from the leaf nodes, the data is propagated back up to the root node, so that the *master* node has the entire information about the network.
3. (OPTIONAL) In case we want to make the network more robust, e.g. by providing each node with a copy of entire network data, we do another pass top-down propagating all info from the root node.

Based on the idea above, besides ID, the nodes also maintain the *level* information ($n_i.level$), which is defined with respect to its distance from the *master* node, i.e., how many communication hops one needed from n_0 to the node itself (Initially, $n_i.level = n_0.level = 0$). Also, each node keeps track of its parent and children nodes, such as $n_i.parent$ and $n_i.child$, where $n_i.child$ is a linked list of n_i's children. We assume nodes are well synchronized and the protocol proceeds in unit time slots. Since the protocol is TDMA-based, it means that each time frame is divided into n time slots, and in each time slot i, only n_i can transmit. This assumption guarantees that there will never be a collision because of concurrent transmissions.

Several assumptions:

1. number of nodes in the network: 1 master node and 5 normal nodes
2. size of a packet: 256 bits, and the format is there already, and can be changed if needed.
3. the range between any two nodes is targeted to be over $10m$, but in the initial experiments it is $\leq 2m$. The transmission range of the sensors is up to $50m$.
4. speed of transmission: 9.6kHz for now, and would like to go up to 300kHz. (To clarify: 10Hz means $10bit/sec$)
5. data collection cycle may be set to every 12 minutes. Sensors can sense anytime they want.
6. nodes broadcast its clock time, starting from the master node.
7. master node will be protected against blinking from robots because of their LED's light is too bright. There could be faulty sensors, sensors may join and/or leave. Also, error correction code, such as CRC could be introduced to improve robustness.

Algorithm 1 Underwater Communication Protocol for node n_i

$n_i.t \leftarrow 0$
$n_i.sendFirstTime \leftarrow -1$
if $n_i == masterNode||$ successfully received a packet P **then**
  while $n_i.t < 2n^2$ or $n_i.parent == null$ **do**
    while interval % 10!=0 **do**
      hold
    end while
    if $n_i.t\%n \neq i$ or $n_i.parent == null$ **then**
      if n_i successfully received a packet P **then**
        if $n_i.parent == null$ **then**
          $n_i.parent \leftarrow P.src$
          $n_i.t \leftarrow P.t$
        else if $P.src == n_i.botId$ **then**
          $n_i.child \leftarrow P.botId$
        end if
        $n_i.t = P.t$
      end if
    else
      if $n_i.parent \neq null$ **then**
        n_i sends a packet P(=$P.n_i, P.src, n_i.t, timeStamp(hh, mm, ss)$, ”I’m alive”) containing the information of the n_i:{ID, timestamp, level, ”I’m alive”}
        if $n_i.sendFirstTime == -1$ **then**
          $n_i.sendFirstTime \leftarrow n_i.t$
        end if
      end if
    end if
    if $n_i.t == n_i.sendFirstTime + n$ and $n_i.child == \emptyset$ **then**
      $n_i.isLeaf \leftarrow true$
    end if
    {Entering data transmission phase.}
    if ($n_i.isLeaf == true$ or $n_i.hasReceivedFromAllChildren = true$) and $n_i.t\%n == i$ **then**
      n_i sends a DATA packet P containing all the data gathered by n_i and its children, i.e., $n_i.DATA[j]$ for all n_i' child n_j.
    else
      if n_i successfully received a packet P **then**
        $n_i.DATA[P.src] \leftarrow P.DATA$
        if n_i has successfully received all packets from all children **then**
          $n_i.hasReceivedFromAllChildren \leftarrow$ true.
        end if
      end if
    end if
    $n_i.t \leftarrow n_i.t + 1$
  end while
end if