This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2009 proceedings.

STUMP: Exploiting Position Diversity in the Staggered TDMA Underwater MAC Protocol

Kurtis Kredo II
Electrical and Computer Engineering Dept.
University of California, Davis
kbkredo@ucdavis.edu

Petar Djukic
Dept. of Systems and Computer Engineering
Carleton University
djukic@sce.carleton.ca

Prasant Mohapatra
Computer Science Dept.
University of California, Davis
pmohapatra@ucdavis.edu

***Abstract*—In this paper, we propose the Staggered TDMA Underwater MAC Protocol (STUMP), a scheduled, collision free TDMA-based MAC protocol that leverages node position diversity and the low propagation speed of the underwater channel. STUMP uses propagation delay information to overlap node communication and increase channel utilization.**

Our work yields several important conclusions. First, leveraging node position diversity through scheduling yields large improvements in channel utilization. Second, STUMP does not require tight node synchronization to achieve high channel utilization, allowing nodes to use simple or more energy efficient synchronization protocols. Finally, we briefly present and evaluate algorithms that derive STUMP schedules.

I. Introduction

Wireless communication in underwater networks overlaps with terrestrial communication on several topics, but many important characteristics only appear in the underwater environment. One of the main differences between underwater and terrestrial networks comes from using acoustic communication underwater. Extreme multipath [1], long propagation delay, and a channel capacity dependent on distance [2] differentiate communications in underwater and terrestrial networks. Additionally, changing water conditions contribute a time varying aspect to these characteristics, limiting underwater devices to low data rates (kilobits per second) and complicating network protocol design.

Acoustic communication forces protocol designers to adapt common terrestrial techniques for underwater networks. Avoiding or reducing collisions becomes vitally important because communication typically consumes the largest fraction of a node's energy and collisions waste the limited available throughput. Long propagation delays, up to several seconds, invalidate or decrease the usefulness of many feedback techniques commonly used in terrestrial networks, such as carrier sensing and control packet exchanges. Additionally, the long and varying propagation delays complicate time synchronization protocols.

Previous work in underwater network MAC protocols has focused on overcoming the challenges of the acoustic channel, but we exploit these characteristics through scheduling to improve network performance. Our Staggered TDMA Underwater MAC Protocol (STUMP) uses node position diversity, through propagation delay estimates, to schedule overlapping transmissions without collisions. We show that: STUMP, which also provides an upper bound on many previously proposed underwater ad hoc MAC protocols, performs better than Aloha and optimized TDMA in underwater networks; STUMP handles significant synchronization error; and distributed and centralized algorithms exist for implementing STUMP in underwater networks.

II. Staggered TDMA Underwater MAC Protocol

Unlike terrestrial networks, underwater nodes do not need to reserve the communication channel for long periods to prevent collisions. Node position diversity can cause packets from two different nodes to arrive successfully, even if the packets were transmitted at the same time [3]. Likewise, packets transmitted at different times can collide due to propagation delays. To prevent collisions, nodes only need ensure that packets arrive during different times at the intended receiver.

STUMP leverages this knowledge to overlap communications without collisions and increase channel utilization. To accomplish this, STUMP nodes develop schedule constraints, presented in Section IV, based on local information and find schedules that satisfy those constraints. Using propagation delay estimates enables STUMP nodes to construct constraints that overlap communications and decrease the time the channel remains idle, while preventing collisions. Furthermore, as presented in Section V, distributed algorithms exist for finding STUMP schedules.

The schedule constraints used by STUMP only require nodes to share propagation delay estimates and time slot requirements among their two-hop neighbors. Current synchronization protocols [4], [5] provide propagation delay estimates and could be adapted to distribute estimates during operation. We assume nodes remain nearly stationary, so the delay estimates only vary slightly as nodes drift on their tethers and ocean conditions change. Time slot requirements can come from application specifications know before deployment or dynamically during network operation.

STUMP organizes transmissions into time slots of duration T seconds using a repeating frame of m slots. During a frame, each node transmits in contiguous slots as required by the traffic and routing conditions. This differs from the traditional TDMA structure where each node transmits entirely in one time slot. Depending on the schedule constraints and network conditions,

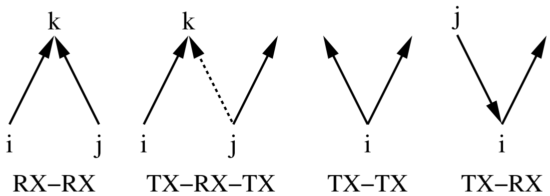

Fig. 1. Scheduling Conflicts

several time slots may be scheduled between transmissions to prevent collisions.

Two options exist for determining the frame size. First, a fixed frame size may be used to simplify protocol operation. In this case nodes have a fixed throughput and "better" schedules are those that utilize less of the frame, possibly leaving room for ad hoc communications or network maintenance. Second, nodes may use a variable sized frame that changes according to network demand. Changing the frame size rarely, for example, only when it is a certain percent larger or smaller than needed, decreases the frequency of distributing the new frame size throughout the network.

III. Network Model

We model a typical, non-mobile underwater network designed to gather data and forward it to a remote user through a single gateway node, called the sink. Each underwater node has a single, half-duplex radio capable of single packet reception. Links exist between nodes whenever the source has the ability to successfully communicate to the destination, but links need not be symmetric. Node interference follows the protocol model. These assumptions yield four possible conflicts in the network [6], as illustrated in Fig. 1.

Nodes communicate over multi-hop paths determined by a routing protocol, where paths remain stable for long periods of time. We consider uplink (to sink) and downlink (from sink) traffic, but our work allows other traffic patterns as well. To prevent congestion at any node, the routing layer provides the MAC protocol with the number of slots required to transmit all packets the node will receive and generate during that frame.

Scheduled protocols require nodes maintain time synchronization among themselves to prevent schedules from drifting and causing collisions. To model the synchronization error present in any protocol, we define σ as the maximum synchronization error at any node from a global time. Thus, the local time between any two nodes differs by at most 2σ. While synchronization in underwater networks is more complicated than terrestrial networks, protocols exist for underwater networks [4], [5].

Additionally, variation in the water characteristics and node position results in changing propagation delays between any pair of nodes. We define π as the maximum error experienced in estimating the one way propagation delay between any two nodes. Hence, a packet sent at time t can arrive anywhere during the interval $(t - \pi, t + \pi)$. Propagation delay variances also affect synchronization protocols, but we model those effects within the parameter σ.

IV. Conflict-Free Scheduling

We now define the characteristics of valid STUMP and TDMA schedules. In both cases, a valid schedule is a set of time slots assigned to each node for transmission, $\mathbf{S} = \{s_i\}$, that prevents all the conflicts in Fig. 1 and satisfies the demand, $\mathbf{\Delta} = \{\Delta_i\}$, of all nodes. We allow multiple packets to arrive at a node if none of the packets were destined for that node. In each frame, node i is assigned Δ_i continuous slots, starting at slot s_i. Each frame contains m time slots.

A valid schedule must ensure all conflicts are resolved to prevent collisions. To accomplish this, we define constraints, $\mathcal{C}$, on the schedules of each protocol that ensure node transmission times are sufficiently separated. Each conflict in the network requires a schedule constraint to resolve it. TDMA constraints, which simply prevent collisions through large guard periods, are the same for each node, but STUMP constraints, which include propagation delay estimates, optimize the constraints for each possible conflict.

Within each frame there exists an ordering between event pairs, such as transmissions and receptions. When formulating the schedule constraints, we use binary ordering variables, $\mathcal{O} = \{o_{ij}\}$, to determine in what order events between nodes i and j occur within a frame. Each constraint contains an ordering variable used to resolve the associated conflict. For example, a TX-RX conflict will have $o_{ij} = 1$ when node i transmits in the frame before receiving the the packet from node j.

Solving for a valid schedule involves finding sets $\mathcal{S}$ and $\mathcal{O}$ that satisfy the schedule constraints $\mathcal{C}$ and node demands Δ. We now develop the constraints for TDMA and STUMP schedules and discuss how to solve for valid schedules in Section V.

A. STUMP Schedule Constraints

Initially, each node is temporarily assigned a starting slot for every neighbor it will transmit to in the frame. Schedule constraints then ensure these groups of slots are contiguous, so the final schedule has a single starting slot for each transmitting node. Node i transmitting to neighbor j transmits for Δ_{ij} time slots starting in slot s_{ij}. Define p_{ij} as the propagation delay from node i to node j. p_{ij} may not equal p_{ji} due to different propagation paths between the two nodes.

We develop the schedule constraint for the RX-RX conflict in detail, but introduce the remaining constraints more briefly since they are developed similarly.

1) RX-RX Conflicts: The schedule resolves RX-RX conflicts by preventing the transmissions from node i and node j from colliding at node k. If the transmission from node i arrives at node k first, $o_{ij} = 1$, then the schedule must ensure the packet from node j arrives after node k finishes receiving node i's packet. Node k finishes receiving node i's packet at time $s_{ik} + \Delta_{ik} + \frac{p_{ik}}{T}$. The packet from node j arrives at $s_{jk} + \frac{p_{jk}}{T}$. This yields the inequality $s_{ik} + \Delta_{ik} + \frac{p_{ik}}{T} \leq s_{jk} + \frac{p_{jk}}{T}$. A valid schedule must also ensure that node j's transmission does not cause a collision with the transmission of node i in the next frame, which yields the inequality $s_{jk} + \Delta_{jk} + \frac{p_{jk}}{T} \leq s_{ik} + \frac{p_{ik}}{T} + m$. A similar pair of inequalities arise under the assumption

$o_{ij} = 0$. Combining the four inequalities and adding buffer time slots for synchronization and propagation delay estimate errors yields the STUMP constraint for RX-RX conflicts in (1).

$$\Delta_{ik} + \frac{p_{ik} - p_{jk} + 2\sigma + 2\pi}{T} - m\left(1 - o_{ij}\right) \leq s_{jk} - s_{ik} \leq -\Delta_{jk} + \frac{p_{ik} - p_{jk} - 2\sigma - 2\pi}{T} + m o_{ij} \quad (1)$$

2) TX-RX-TX Conflicts: For TX-RX-TX conflicts, the schedule must ensure an interference packet does not arrive at a node while it is receiving a valid packet. This condition is nearly identical to the RX-RX conflict. Simply use node j as the interfering node to add conflicts between each pair of nodes i and j when node j could interfere with a transmission from node i. The resulting constraint is identical to (1).

3) TX-TX Conflicts: Each node can only transmit to a single neighbor at a time, so a valid schedule must include constraints to ensure this. For simplicity, we assume nodes transmit all packets to the farthest neighbor first, then sequentially to closer neighbors until they have transmitted all their packets. Thus, $s_{i0} = s_{i1} + \Delta_{i1}, s_{i1} = s_{i2} + \Delta_{i2}, \ldots$, where neighbor 1 is farther than neighbor 0, neighbor 2 is farther than neighbor 1, and so on. Formally,

$$s_{ik} - s_{ij} = -\Delta_{ik} \quad (2)$$

where node k is the next farther neighbor than node j from node i. This schedule constraint only exists between transmit times of the same node to different neighbors, never between two different nodes.

4) TX-RX Conflicts: TX-RX conflicts ensure nodes do not transmit while receiving a packet. There may be multiple TX-RX conflicts between a sender and receiver, with one for each neighbor of the receiver. A TX-RX conflict yields the schedule constraint (3).

$$\Delta_{in} + \frac{-p_{ji} + 2\sigma + \pi}{T} - m\left(1 - o_{ij}\right) \leq s_{ji} - s_{in} \leq -\Delta_{ji} - \frac{-p_{ji} + 2\sigma + \pi}{T} + m o_{ij} \quad (3)$$

for some neighbor node n of node i.

The STUMP scheduling problem, thus, involves finding the set of transmission times, $\mathcal{S}$, that satisfies the schedule constraints (1), (2), and (3) for the conflicts in the network, given the node demand Δ.

B. TDMA Schedule Constraints

Unlike traditional TDMA protocols, which add guard slots long enough to accommodate the full propagation range of a node, the TDMA used here only adds guard slots to reach the farthest node in interference range of a transmitter. Define G_i as the guard slots required after the transmission of node i using TDMA. With a maximum propagation delay of p_i to the farthest neighbor of node i in its interference range, we calculate the guard slots as:

$$G_i = \frac{p_i + 2\sigma + 2\pi}{T}$$

Since the guard slots prevent collisions between nodes that transmit at different times, the schedule only needs to ensure that nodes which cause interference to each other are assigned non-overlapping time slots. This yields the TDMA schedule constraint in (4).

$$\Delta_i + G_i + m\left(1 - o_{ij}\right) \leq s_j - s_i \leq -\Delta_j - G_j + m o_{ij} \quad (4)$$

A valid TDMA schedule is the set of all transmit times, $\mathcal{S}$, such that (4) holds for each pair of conflicting nodes, given the node demand Δ.

V. Scheduling Algorithms

We now present several algorithms, both distributed and centralized, to solve the scheduling problems. We use two metrics to compare the algorithms and protocols under study: the normalized network throughput and the maximum uplink delay. The network throughput is calculated as the number of slots used by the sink for transmission or reception divided by the number of time slots in the frame, m. We evaluate the uplink delay as the maximum delay experienced by any uplink traffic in the network.

A. Centralized Algorithm

Combining a scheduling problem detailed in the previous section with an appropriate objective function, either for minimum frame size or minimum uplink delay, yields an integer linear programming problem. These problems could be solved centrally at a node with enough computational resources if given the network state, but would require significant overhead to collect the network information. Therefore, we evaluate distributed scheduling algorithms, but compare the results with the optimal, centralized algorithm.

B. Distributed Algorithms

The TDMA and STUMP scheduling problems are NP-Complete [7], but we simplify the problem by solving it in two steps [8]. First, we determine the ordering variables, o_{ij}, by prioritizing nodes. If node i has a higher priority than node j, then $o_{ij} = 1$, otherwise $o_{ij} = 0$. Only conflicting nodes need to determine relative priority since ordering variables do not exist between nodes without schedule conflicts. With fixed ordering variables, the scheduling constraints become a set of difference equations, which nodes can solve using the Bellman-Ford algorithm [9]. The problem difficulty now lies in finding a good set of ordering variables. We present two distributed algorithms that determine the ordering variables for the nodes.

A simple way to find node priorities is to select them at random, such as from node ids, random numbers generated locally, or graph coloring algorithms. However, each pair of conflicting nodes must select unique priorities so the ordering variables are well defined. Selecting priorities at random requires little effort from the nodes, but it does not guarantee any level of performance. However, we show that random schedules yield characteristics useful in some applications. Nodes could improve random selection by computing several schedules in parallel and using the schedule with the best characteristics.

Another way to select node priorities is to organize the transmissions so packets arrive at the sink within a single frame. To do this, a node simply selects a lower priority than any neighbor that relays traffic to it. Leaf nodes have the highest priority since they must transmit earliest in the frame and the last relay before the sink has the lowest priority. While this ordering bounds uplink delay to a single frame, it does not guarantee the minimum delay since nodes that forward traffic to a common relay (an RX-RX conflict) may get the same priority. Thus, nodes must have a secondary way to select ordering variables, such as with node id, to break ties. The secondary comparison may choose incorrectly, resulting in non-optimal uplink delay.

VI. Numerical Results

We now compare STUMP with the TDMA and Aloha protocols by evaluating their schedule's average throughput and delay performance over 100 random topologies. We do not simulate the protocols; we derive the schedules the protocols would use and evaluate them. We use Aloha's theoretical optimal throughput [3]. We show that STUMP achieves higher throughput than TDMA and (optimal) Aloha, lower latency than TDMA, and tolerates large synchronization error.

Energy is an important metric for underwater acoustic networks, but we do not explicitly measure it in this work for two reasons. First, TDMA and STUMP consume the same energy under identical traffic conditions, so their results would be similar. Second, random protocols waste significant energy through collisions in high load situations, so their efficiency would be low. We leave to future work the energy performance comparison of STUMP to random protocols under light to moderate load.

Each topology consists of nodes deployed in a grid pattern made of square cells 3500m on a side. Cells with a center within 7500m of the sink, except for the cell containing the sink, receive a node, resulting in a network of 12 nodes plus the sink. To simulate inaccurate deployment, we randomly select a position from the center third of a cell for each node. Nodes remain stationary during operation, but small movements caused by ocean currents are modeled by the synchronization and propagation delay estimate error parameters σ and π. This paper models a two dimensional network, but our work applies just as readily to networks with nodes at different depths.

Nodes have a communication range of 4000m and an interference range of 8000m. Each frame consists of time slots with a duration of 0.4s. Unless stated otherwise, each node generates 10 time slots worth of data destined for the sink and the sink sends one slot worth of data to each node in each frame. A lifetime maximizing protocol [10] determines the routing paths.

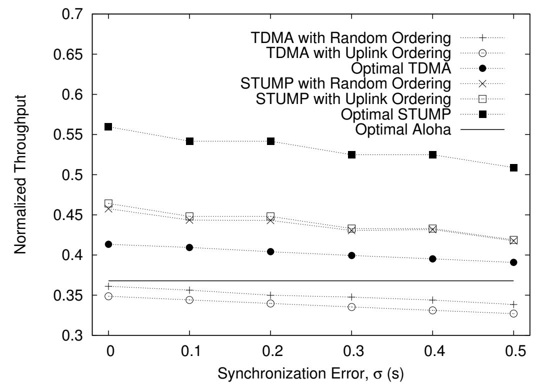

Fig. 2. Normalized Throughput as σ Varies

A. Synchronization Error Impact

We first examine how the protocols perform as the synchronization error, σ, varies. Since synchronization and propagation delay estimate errors affect the schedules in a similar way, we only vary σ and set the propagation delay error, π, equal to zero.

Fig. 2 shows how the throughput of each protocol varies with synchronization error. STUMP achieves a much higher throughput than TDMA and Aloha for all synchronization error values, indicating scheduling protocols can gain large benefits by using propagation delay information to overlap communications in underwater networks. The benefit of overlapping communication is so large that using STUMP with the non-optimal distributed scheduling algorithms yields better performance than TDMA or Aloha can ever achieve.

Additionally, STUMP adapts to synchronization error with marginal degradation and at a similar rate to TDMA, even though STUMP transmission are scheduled much closer together. When $\sigma = 0.5$s, meaning node clocks may differ by up to a second, STUMP decreases its throughput by less than 10%. Thus, nodes can use lower energy, but less precise, synchronization protocols without significantly affecting STUMP's performance.

Next, compare the maximum uplink delay for the protocols as synchronization error varies, as displayed in Fig. 3. STUMP again performs better than TDMA, achieving a 20% lower latency on average when using the Uplink ordering. However, unlike throughput, STUMP with Uplink ordering does not perform better than TDMA with an optimal ordering due to sub-optimal choices made by the distributed scheduling algorithm. Similar to the throughput results, even with significant synchronization error, STUMP's latency increases by less than 9% and at a similar rate to TDMA. Fig. 3 does not show the Random ordering results because those values depend on the frame size selected.

The delay results have shown that Uplink ordering provides a low maximum uplink delay when compared to Random ordering, but this comes at the cost of downlink delay. Fig. 4 shows both the uplink and downlink delay for both ordering algorithms. Uplink ordering results in significant downlink

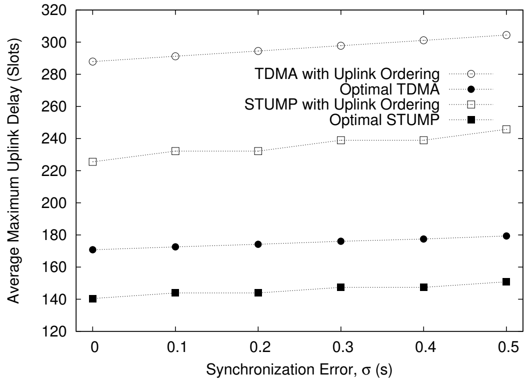

Fig. 3. Average Maximum Uplink Delay as σ Varies

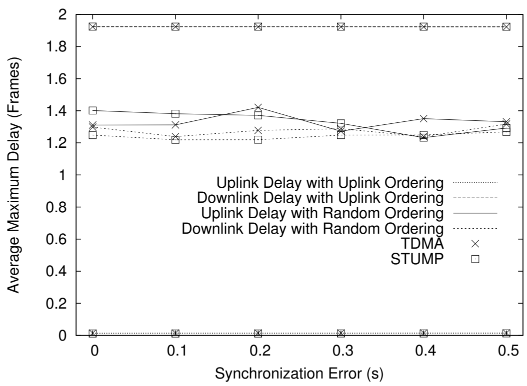

Fig. 4. Average Maximum Delay as σ Varies

latency since all the nodes transmit in exactly the wrong order, resulting in an average of nearly two frames of delay. However, Random ordering performs consistently for both uplink and downlink traffic. This indicates that networks which require fast response times, such as surveillance or monitoring networks, would benefit from Uplink ordering, but general purpose networks may desire Random ordering since it provides balanced and equal performance to all traffic.

B. Varying Traffic Load

Some applications may not require ten times more uplink traffic than downlink traffic and these parameters have a large effect on protocol performance, so we investigate several different traffic loads. We varied the uplink traffic from each node to 1, 5, or 10 time slots and the downlink traffic as either 0 or 1 time slots to each node. Fig. 5 shows the results for various traffic loads. As before, STUMP outperforms TDMA for all traffic loads. However, at low traffic levels, STUMP cannot achieve the throughput possible with Aloha when using the distributed ordering algorithms. As expected, random protocols perform better at low data rates, but scheduled protocols, especially STUMP, perform better at moderate to high data rates. However, scheduled protocols achieve their throughput without causing collisions, so their energy consumption may be lower than Aloha at low data rates. Similar trends between TDMA and STUMP were found with the uplink delay, but are not included.

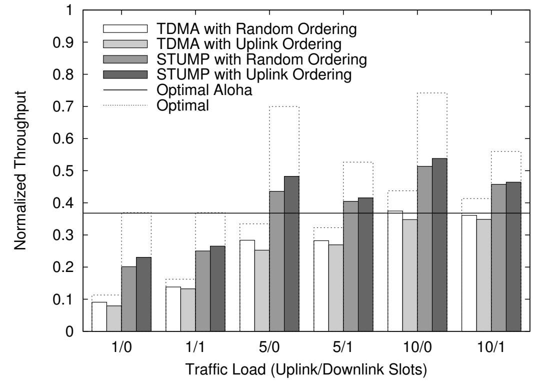

Fig. 5. Normalized Throughput as Traffic Load Varies

VII. Conclusions

We have shown that synchronized protocols do perform well in underwater environments if they use propagation delay estimates to improve channel scheduling. The Staggered TDMA Underwater MAC Protocol increases the performance of traditional TDMA by using propagation delay estimates to schedule overlapping transmissions. STUMP does this while degrading gracefully (by less than 10%) with synchronization and propagation delay estimate errors. Finally, we demonstrated how to develop STUMP schedules with both distributed and centralized algorithms and provide users the ability to develop scheduling algorithms suited to their application.

Acknowledgments

This research was supported in part by the National Science Foundation through the grants CNS-0831914 and CNS-0709264, and by the US Army Research Office through the MURI grant W911NF-07-1-0318.

References

[1] J. Preisig, "Acoustic propagation considerations for underwater acoustic communications network development," in *Proc. ACM WUWNet Conf.*, Sep. 2006, pp. 1–5.
[2] M. Stojanovic, "On the relationship between capacity and distance in an underwater acoustic communication channel," in *Proc. ACM WUWNet Conf.*, Sep. 2006, pp. 41–47.
[3] A. Syed, W. Ye, B. Krishnamachari, and J. Heidemann, "Understanding spatio-temporal uncertainty in medium access with ALOHA protocols," in *Proc. ACM WUWNet Conf.*, Sep. 2007, pp. 41–48.
[4] A. A. Syed and J. Heidemann, "Time synchronization for high latency acoustic networks," in *Proc. IEEE InfoCom Conf.*, Apr. 2006, pp. 892–903.
[5] N. Chirdchoo, W.-S. Soh, and K. C. Chua, "MU-Sync: A time synchronization protocol for underwater mobile networks," in *Proc. ACM WUWNet Conf.*, Sep. 2008, pp. 35–42.
[6] S. Ramanathan, "A unified framework and algorithm for channel assignment in wireless networks," *Wireless Networks*, vol. 5, no. 2, pp. 81–94, Mar. 1999.
[7] P. Serafini and W. Ukovich, "A mathematical model for periodic scheduling problems," *SIAM Journal on Discrete Mathematics*, vol. 2, no. 4, pp. 550–581, 1989.
[8] P. Djukic and S. Valaee, "Link scheduling for minimum delay in spatial re-use TDMA," in *Proc. IEEE InfoCom Conf.*, May 2007, pp. 28–36.
[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. The MIT Press, 2001.
[10] J.-H. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 609–619, Aug. 2004.