A scheduling algorithm for wireless networks with large propagation delays

Mandar Chitre
Acoustic Research Laboratory,
Tropical Marine Science Institute,
National University of Singapore,
Singapore 119227.
mandar@arl.nus.edu.sg

Mehul Motani
Department of Electrical
and Computer Engineering,
National University of Singapore,
Singapore 117576.
motani@nus.edu.sg

Shiraz Shahabudeen
Acoustic Research Laboratory,
Tropical Marine Science Institute,
National University of Singapore,
Singapore 119227.
shiraz@nus.edu.sg

***Abstract*—Underwater acoustic networks can have large propagation delays as compared to typical packet durations, as a result of the low speed of sound in water. The ill effects of large propagation delay on medium access control (MAC) are well known. Conventional MAC protocol design for such networks focuses on mitigation of the impact of propagation delay. Most proposed protocols to date achieve, at best, a throughput similar to that of the zero propagation delay scenario. We have explored the possibility that propagation delays can be exploited to make throughput far exceed that of networks without propagation delay and shown that the throughput of a N-node wireless network with propagation delay is upper bounded by $N/2$. In a small set of illustrative network geometries, we can manually determine transmission schedules that allow us to achieve this $N/2$ bound. However, for a given network, the problem of determining transmission schedules that maximize throughput is as yet unsolved. In this paper, we put forward an algorithm that generates transmission schedules with high throughput for arbitrary network geometries.**

***Index Terms*—Wireless networks, large propagation delays, interference overlap, transmission schedules.**

I. Introduction

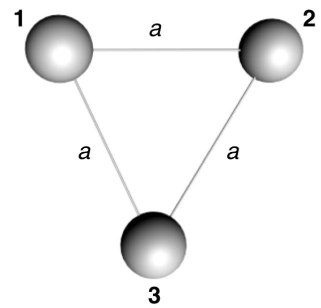
Propagation delays in underwater acoustic networks can be large compared to typical packet durations, as a result of the low speed of sound in water. The ill effects of large propagation delay have been studied extensively by researchers. The performance of handshaking protocols and acknowledgment based retransmission schemes is known to suffer in the domain of large propagation delays [1]. The effect of large propagation delays on medium access control (MAC) layer protocols which prevent all data collisions has been discussed in [2]. Conventional MAC protocol design for such networks focuses on mitigation of the impact of propagation delay [3]–[6]. Conventional MAC protocol design for such networks focuses on mitigation of the impact of propagation delay. Most proposed protocols to date achieve, at best, a throughput similar to that of the zero propagation delay scenario. We have explored the possibility that propagation delays can be exploited to make throughput far exceed that of networks without propagation delay and shown that the throughput of a N-node wireless network with propagation delay is upper bounded by $N/2$ [7]. In a small set of illustrative network geometries, we can manually determine transmission schedules that allow us to achieve this $N/2$ bound. However, for a given network, the problem of determining transmission schedules that maximize throughput is as yet unsolved. In this paper, we put forward an algorithm that generates transmission schedules with high throughput for arbitrary network geometries.

Example 1: In order to illustrate the idea of how a network with a large propagation delay may achieve a throughput larger than a network with negligible propagation delay, we present a simple 3-node network example. Consider a 3-node network with the nodes located at the vertices of an equilateral triangle as shown in Fig. 1(a). Let the length of each side of the triangle be such that the propagation delays $D_{12} = D_{13} = D_{23} = a$ seconds. We let each node transmit messages of duration a seconds as per the periodic schedule shown in Fig. 1(b). For each node, the schedule ensures that the interference from other nodes only arrives when the node is transmitting. Given a link data rate of β bits/second, the nodes can successfully transmit 6 messages with $a\beta$ bits each during each period $T = 4a$ seconds. Using the schedule shown, a normalized throughput $S = (6a\beta/4a)/\beta = 1.5$ can be achieved. This is 50% higher than the maximum throughput for a 3-node network without propagation delay! Even larger improvements in throughput are possible for networks with more nodes.

II. Assumptions and Mathematical Preliminaries

A. System Model and Assumptions

We consider an N-node network with nonzero propagation delay between every pair of nodes. The nodes in the network are *half-duplex* and the network carries only unicast messages, i.e., each message has a single destination node. The network is a single *collision domain* i.e. a message transmitted by a node reaches every node (other than the transmitting node) after the appropriate propagation delay. If two or more messages overlap at the destination node for any of the messages, a *collision* is said to occur and the message is lost – this is clearly undesirable if we wish to maximize throughput, and therefore we wish to generate collision-free schedules with our algorithm.



(a) Geometry for a 3-node equilateral triangle network

Period T = 4τ

Time	-τ	0	τ	2τ	3τ	4τ	5τ	6τ
Node 1	... RX from 2	TX to 2	TX to 3	RX from 3	RX from 2	TX to 2	TX to 3	...
Node 2	... TX to 3	RX from 3	RX from 1	TX to 1	TX to 3	RX from 3	RX from 1	...
Node 3	... TX to 2	RX from 2	TX to 1	RX from 1	TX to 2	RX from 2	TX to 1	...

(b) Periodic schedule

Fig. 1. A 3-node equilateral triangle network and its transmission schedule. An entry "TX to j" means a transmit to node j in that time slot, while an entry "RX from i" means receive from node i in that time slot.

B. Delay matrices

We represent the network geometry in terms of a delay matrix $\mathbf{D}$ with time in units of message duration μ. For a network with nodes located at $\mathbf{x}_j$, $j \in \{1..N\}$:

$$D_{ij} = \frac{|\mathbf{x_i} - \mathbf{x_j}|}{c\mu} \tag{1}$$

where c is the signal propagation speed. For example, the delay matrix for the network shown in Fig. 1(a) is

$$\mathbf{D} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \tag{2}$$

C. Schedules

A schedule $\mathbf{Q}$ determines when each node transmits and receives messages. We let the message duration μ be equal to the time slot duration τ for the schedule. For a network with an integer delay matrix, messages transmitted on time slot boundaries are received at time slot boundaries at all nodes. If $Q_{jt} = i > 0$ then node j transmits a message to node i during time slot t. If $Q_{jt} = -i < 0$, then node j receives a message from node i during the time slot t. In all other cases, node j is defined to be idle during time slot t and we set $Q_{jt} = 0$.

A message received by node j during time slot t must have been transmitted by some other node i during time slot $t - D_{ij}$. Node i transmits a message to node j during time slot $t - D_{ij}$ only if node j is able to successfully receive the message during time slot t. Hence,

$$Q_{jt} = -i \;\Leftrightarrow\; Q_{i,t-D_{ij}} = j \tag{3}$$

To ensure that the message is successfully received (without collision), we require that no other nodes transmit messages that arrive at node j during time slot t:

$$Q_{jt} = -i \;\Rightarrow\; Q_{k,t-D_{jk}} \leq 0 \;\forall\, k \neq i \tag{4}$$

A schedule that repeats with a period T, i.e., $Q_{j,t+T} = Q_{jt} \;\forall\, j,t$, it is said to be *periodic*. A periodic schedule can be represented by an $N \times T$ matrix $\mathbf{Q}^{(T)}$ such that[1]:

$$Q_{jt} = Q^{(T)}_{j,\, t(\mathrm{mod}\ T)} \tag{5}$$

For example, the periodic schedule shown in Fig. 1(b) is represented by

$$\mathbf{Q}^{(4)} = \begin{bmatrix} 2 & 3 & -3 & -2 \\ -3 & -1 & 1 & 3 \\ -2 & 1 & -1 & 2 \end{bmatrix} \tag{6}$$

The average throughput S of a schedule with period T can be computed from the number of receptions (or equivalently transmissions) in schedule $\mathbf{Q}^{(T)}$:

$$\begin{aligned} S &= \frac{1}{T} \sum_t \sum_j \mathbb{I}(Q^{(T)}_{jt} < 0) \\ &= \frac{1}{T} \sum_t \sum_j \mathbb{I}(Q^{(T)}_{jt} > 0) \end{aligned} \tag{7}$$

where $\mathbb{I}(A)$ is the indicator function with value 1 if A is true and 0 otherwise.

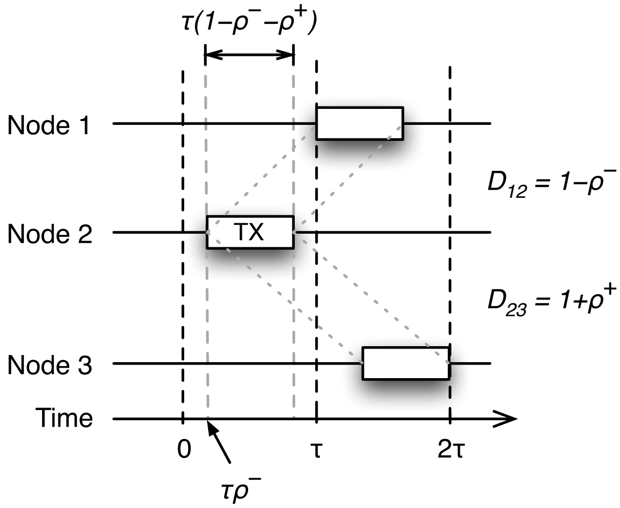


Fig. 2. An illustration of a ρ-schedule

D. ρ-Schedules

For a network with a non-integer delay matrix, messages transmitted on time slot boundaries may be received across time slot boundaries. If the duration of the message is equal to the time slot duration, the message reception will span multiple time slots. For a non-integer delay matrix $\mathbf{D}$, we can round off the entries in the delay matrix to yield an integer delay matrix $\mathbf{D}'$ and define ρ^+ and ρ^- such that

$$\rho^+ = \max_{ij}(D_{ij} - D'_{ij}) \tag{8}$$

$$\rho^- = -\min_{ij}(D_{ij} - D'_{ij}) \tag{9}$$

and $\rho^+, \rho^- \leq 0.5$. If we limit the duration of each transmitted message to $\tau(1-\rho^- - \rho^+)$ and transmit the message at time $\tau\rho^-$ after the start of the time slot, then we ensure messages are always received fully during a time slot as seen in Fig. 2. We call a schedule with shortened messages of duration $\mu = \tau(1-\rho^- - \rho^+)$ a fractional time-slot schedule or a *ρ-schedule*. The throughput S' of a ρ-schedule is related to the throughput S of the equivalent schedule by

$$S' = (1-\rho^- - \rho^+)S, \qquad \rho^- + \rho^+ \leq 1 \tag{10}$$

The throughput of a ρ-schedule is maximized by maximizing the throughput of the underlying schedule for an integer delay matrix. Hence, in the rest of this paper, we simply consider the problem of maximizing the throughput of a schedule with an integer delay matrix. The solution given by the proposed algorithm can be used with the corresponding ρ-schedule in case of a fractional delay matrix.

To obtain a good throughput with a ρ-schedule, we need to ensure that $\rho^+ \ll 0.5$ and $\rho^- \ll 0.5$. The values of ρ^+ and ρ^- depend on the choice of message duration μ and slot duration τ, which may be partially constrained by the available modems and operational requirements. In [7], we show that ρ^+ and ρ^- can be made arbitrarily small for a network with rational distances and no constraints, but the selection of μ and τ remains an open problem for a general network with irrational distances and constrained slot durations and message lengths.

[1]Matrix $\mathbf{Q}^{(T)}$ is indexed by (j,t) such that $j \in \{1..N\}, t \in \{0..T-1\}$.

III. Problem Formulation

A. Sequential decision problem

Given an N-node network geometry with a delay matrix $\mathbf{D}$, we denote the schedule that maximizes the average throughput S as $\mathbf{Q}^*$. We formulate the problem of finding the optimal schedule $\mathbf{Q}^*$ as a sequential decision problem. The state of the decision problem is represented by $\mathbf{Q}^{\{t\}}$ — the partial schedule given all transmissions before, and no transmission during or after time slot t are made.

Let the action to be taken at time t be $\mathbf{x}^{\{t\}}$. The action defines the the transmissions that occur on all nodes during time slot t such that $x_j^{\{t\}} = 0$ implies that node j does not transmit in time slot t, while $i = x_j^{\{t\}} > 0$ implies that node j transmits to node i during the time slot. The action $\mathbf{x}^{\{t\}}$ and the previous state $\mathbf{Q}^{\{t\}}$ fully define the new state $\mathbf{Q}^{\{t+1\}}$ as a result of the transmissions in time slot t:

$$\mathbf{Q}^{\{t+1\}} = \Gamma(\mathbf{Q}^{\{t\}}, \mathbf{x}^{\{t\}}) \tag{11}$$

The set of feasible actions $\mathcal{X}(\mathbf{Q}^{\{t\}})$ consists of all possible actions where $Q_{jt}^{\{t\}} = 0$ when $x^{\{t\}} > 0$ and the resulting state $\mathbf{Q}^{\{t+1\}}$ denotes a valid schedule in accordance with (3) and (4).

The number of transmissions that occur as a result of each action constitute the reward for that decision:

$$C(\mathbf{x}^{\{t\}}) = \sum_{j=1}^{N} \mathbb{I}(x_j^{\{t\}} > 0) \tag{12}$$

The throughput S is the average reward:

$$S = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} C(\mathbf{x}^{\{t\}}) \tag{13}$$

An optimal policy X^* makes decisions $\mathbf{x}^{\{t\}} = X^*(\mathbf{Q}^{\{t\}})$ to yield a maximum throughput S^* and the corresponding schedule $\mathbf{Q}^*$. Although the action taken depends only on the current state, the optimal policy is not, in general, a greedy policy as it must take into account the expected evolution of the state in the future.

B. Reduced sequential decision problem

For a network of girth $G = \max_{i,j} D_{ij}$, only transmissions occurring between time slot $t-G$ and $t-1$ may affect the optimal decision at time slot t. Hence the optimal policy X^* only depends on a reduced state $\hat{\mathbf{Q}}^{\{t\}}$ which contains the transmissions made during time slots $t-G$ to $t-1$. $\hat{\mathbf{Q}}^{\{t\}}$ can be represented as an $N \times G$ matrix with $\hat{Q}_{jt}^{\{t\}} = i$ for transmissions made by node j to node i at time slot t.

The new reduced state generated as a result of the decision is also fully determined by the reduced state and the decision. Hence:

$$\mathbf{x}^{\{t\}} = X^*(\hat{\mathbf{Q}}^{\{t\}}) \quad (14)$$

$$\hat{\mathbf{Q}}^{\{t+1\}} = \hat{\Gamma}(\hat{\mathbf{Q}}^{\{t\}}, \mathbf{x}^{\{t\}}) \quad (15)$$

Let $\hat{\mathcal{Q}}$ be the state space of $\hat{\mathbf{Q}}^{\{t\}}$. $\hat{\mathcal{Q}}$ is finite with a cardinality $|\hat{\mathcal{Q}}| \leq N^{NG}$. The decision space $\hat{\mathcal{X}}(\hat{\mathbf{Q}}^{\{t\}})$ is also finite with a cardinality $|\hat{\mathcal{X}}_t| \leq N^N$.

Since the state space is finite with no termination state, the system must return to some previously visited state after a finite number of time slots. Since the decision at each time slot only depends on the state at that time, the schedule generated by an optimal policy beginning at the previous visit to the state must be repeated after the current visit to the same state. In other words, the optimal schedule must be periodic with some period T. We can therefore compute the throughput over a single period:

$$S = \frac{1}{T} \sum_{t=0}^{T} C(\mathbf{x}^{\{t\}}) \quad (16)$$

IV. Scheduling Algorithm

A. Dynamic programming

The above deterministic sequential decision problem can be solved using techniques in dynamic programming [8]. We denote the value function, which is the maximum reward of being in a given state, by $V(\hat{\mathbf{Q}}) : \hat{\mathcal{Q}} \to \mathbb{R}$. We can write the optimal policy in terms of the value function:

$$X^*(\hat{\mathbf{Q}}) = \arg \max_{x \in \mathcal{X}(\hat{\mathbf{Q}})} \left(C(\mathbf{x}) + V\left(\hat{\mathbf{\Gamma}}(\hat{\mathbf{Q}}, \mathbf{x})\right)\right) \quad (17)$$

The value function must satisfy the Bellman's equation [8, section 3.1.1]:

$$V(\hat{\mathbf{Q}}) = \max_{x \in \mathcal{X}(\hat{\mathbf{Q}})} \left(C(\mathbf{x}) + V\left(\hat{\mathbf{\Gamma}}(\hat{\mathbf{Q}}, \mathbf{x})\right)\right) - V_0 \quad (18)$$

where V_0 is an appropriate constant required to keep the value function finite. A standard technique known as *relative value iteration* [8, section 3.4.2] is able to solve the dynamic programming problem to iteratively estimate the value function V. Value iteration algorithms are known to converge if the underlying state graph has no cycles. However with periodic schedules, the state graph has cycles and we have to introduce a "stepsize" to ensure that the values converge [8, section 4.2.5]. Although the resulting algorithm works in practice and yields optimal schedules for many small networks, it requires the state space and decision space to be enumerated. Since the cardinality of these spaces grows very rapidly with N and G, the solution is computationally infeasible for larger networks (in terms of nodes or girth).

B. Approximate value function

If we knew the value function, the problem simplifies to enumerating the decision space and finding the optimal decision from (17). Rather than estimate the value function iteratively, it may be possible to develop an approximate value function based on the structure of the problem [8]. We develop an approximate value function guided by our intuition and understanding of the problem. The computational complexity of this solution still grows rapidly with the number of nodes, but not with girth.

The value function represents the reward potential of being in a state. A state with an empty schedule has a lot of potential. A state with a schedule that allows no transmissions in the near future has no potential for rewards in the near future. Since transmissions in the past cannot affect schedule slots beyond a horizon given by the girth of the network, we only need to consider the potential transmissions within the next G time slots of the schedule. The value function must generally increase with the number of slots available for transmission within the next G time slots. A slot $Q_{jt}^{\{t'\}}$ can be considered to be available for transmission to node i provided the following criteria are satisfied:

1) The slot must not already be allocated for receiving a message transmitted by some other node in the past i.e. $Q_{jt}^{\{t'\}} = 0$.
2) A transmission in that slot must not cause interference at any node that is scheduled to receive a message transmitted by some node in the past i.e. $Q_{k,t+D_{jk}}^{\{t'\}} = 0 \; \forall \; k$.
3) The receiving node of a transmission in that slot must not be interfered with by a transmission that has occurred in the past i.e. $Q_{k,t+D_{ij}-D_{ik}}^{\{t'\}} \leq 0 \; \forall \; k$.

Although these criteria capture the effect of decisions made in the past, they do not capture that we may make transmission decisions in the future that may make the slot $Q_{jt}^{\{t'\}}$ unavailable for transmission to node i. However, it is possible that the slot could be used for transmission to nodes other than i. To capture this, we define the *degrees of freedom* of a slot as the number of possible nodes that the given slot could be used for transmission to. The higher the degrees of freedom of a slot, the lower the chance that the node becomes completely unavailable for transmission by decisions made in the future. The potential of a partial schedule $\mathbf{Q}^{t'}$ is then the average degrees of freedom of all slots in the schedule from time t' to $t'+G-1$. We use this potential to estimate the value function:

$$\hat{V}(\mathbf{Q}^{\{t'\}}) = \frac{\alpha}{NG} \sum_{t=t'}^{t'+G-1} \sum_{j=1}^{N} \Theta(Q_{jt}^{\{t'\}}) \quad (19)$$

where α is some constant (we found $\alpha = 1$ to give us good results in most cases) and $\Theta(Q_{jt}^{\{t'\}})$ is the degrees of freedom for slot $Q_{jt}^{\{t'\}}$:

$$\Theta(Q_{jt}^{\{t'\}}) = \prod_{k=1}^{N} \mathbb{1}(Q_{k,t+D_{jk}}^{\{t'\}} = 0) \times \left(\sum_{i=1}^{N} \prod_{k=1}^{N} \mathbb{1}(Q_{k,t+D_{ij}-D_{ik}}^{\{t'\}} \leq 0) \right) \quad (20)$$

We use this approximate value function $\hat{V}$ in place of V in (17).

V. Results

The algorithm described in the previous section is computationally feasible for small N. We tested this algorithm on several networks for which high throughput solutions are known to exist [7], and some networks for which we had no previous solutions.

When applied to the 3-node equilateral triangle network of example 1, the algorithm yields the known solution given by (6) with $S = 1.5$. When applied to an isosceles triangle with sides of length (1,2,2), it also gives a solution with $S = 1.5$. This period 12 solution is not the same as the previously known period 8 solution:

$$\begin{bmatrix} -3 & 2 & 2 & 2 & 3 & -3 & 3 & -3 & -2 & -2 & -2 & 3 \\ 3 & -3 & -1 & -1 & -1 & 3 & -3 & 1 & 1 & 1 & 3 & -3 \\ -2 & -1 & -2 & 1 & 2 & 1 & -1 & -2 & -1 & 2 & 1 & 2 \end{bmatrix}$$

The 3-node isosceles triangles with sides of length (2,3,3) and (1,3,3) had not previously been analyzed. The algorithm was able to find a $S = 1.5$ solution with period 8 for the first network. The second network yielded a $S = 1.4762$ solution with period 21. It is not known whether a $S = 1.5$ solution exists for this geometry. It is known that a linear 3-node network with distance 1 between adjacent nodes has an optimal solution with $S = 4/3$. This solution was also found by the algorithm.

A 4-node tetrahedral network with sides of length (1,1,1,1) yielded the previously known solution with period 2 achieving $S = 2$. The 4-node tetrahedral network with sides of length (1,1,2,2) yielded the previously known solution with period 2 achieving $S = 2$ as well. Finally, we tested a 6-node network known to have a good ρ-schedule with $\rho^- = 0$ and $\rho^+ = 0.06$. The network is shown in Fig. 3 and the positions of the 6 nodes are given by the sound speed multiplied by the columns of the following matrix:

$$\begin{bmatrix} 0 & -4.83 & 0.84 & -3.20 & -1.23 & -1.08 \\ 0 & 1.28 & 0.61 & -2.23 & 0.19 & 1.15 \\ 0 & -0.15 & 1.76 & 0.90 & -1.64 & 1.24 \end{bmatrix}$$

The algorithm found the known solution with $S = 2.82$:

$$\begin{bmatrix} -2 & 2 \\ -1 & 1 \\ -4 & 4 \\ -3 & 3 \\ -6 & 6 \\ -5 & 5 \end{bmatrix}$$

We did not find any networks where the algorithm failed to find a schedule with throughput at least as good as the best known schedule.

VI. Conclusions

Large propagation delays in underwater networks can lead to significant performance gains as compared to networks with negligible propagation delays. In order to harness these gains, a schedule that overlaps interferences in time at unintended nodes, and uses interference laden slots for transmission has to be found. In this paper, we formulated the problem of finding such a schedule as a dynamic programming problem. This dynamic program is computationally infeasible. We found an approximate solution to this problem. This solution was tested against numerous known N-node network geometries (for small N) and found to yield good schedules with throughput close to $N/2$. The algorithm may be used in small static underwater networks to increase the throughput substantially as compared to traditional MAC schemes.

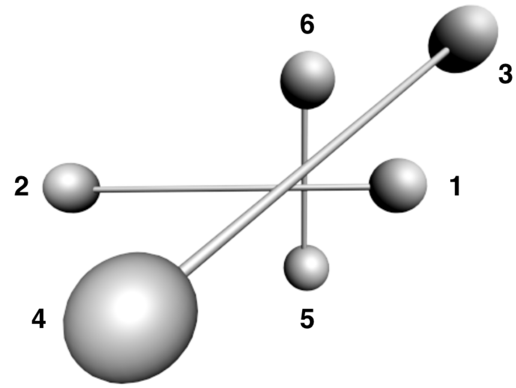


Fig. 3. A 6-node network that achieves a throughput $S = 2.82$

References

[1] A. Leon-Garcia and I. Widjaja, Communication networks: fundamental concepts and key architectures. McGraw-Hill Science Engineering, 2004.

[2] C. Fullmer and J. Garcia-Luna-Aceves, "Floor acquisition multiple access (FAMA) for packet-radio networks," ACM SIGCOMM Computer Communication Review, vol. 25, no. 4, pp. 262273, 1995.

[3] H. Peyravi, "Medium access control protocols performance in satellite communications," IEEE Communications Magazine, vol. 37, no. 3, pp. 6271, 1999.

[4] B. Peleato and M. Stojanovic, Distance aware collision avoidance protocol for ad-hoc underwater acoustic sensor networks, Communications Letters, IEEE, vol. 11, no. 12, pp. 10251027, December 2007.

[5] A. A. Syed, W. Ye, J. Heidemann, and B. Krishnamachari, "Understanding spatio-temporal uncertainty in medium access with ALOHA protocols," in WuWNet 07: Proceedings of the second workshop on Underwater networks. New York, NY, USA: ACM, pp. 4148, 2007.

[6] X. Guo, M. Frater, and M. Ryan, "Design of a propagation-delay-tolerant MAC protocol for underwater acoustic sensor networks," IEEE Journal of Oceanic Engineering, vol. 34, no. 2, pp. 170180, 2009.

[7] M. Chitre, M. Motani, and S. Shahabudeen, "Throughput of Wireless Networks with Large Propagation Delays," in *IEEE Transactions on Networking*, under review.

[8] W. B. Powell, "Approximate dynamic programming," New Jersey: Wiley-Interscience, ISBN: 0470171553, 2007.