# Throughput of Networks With Large Propagation Delays

Mandar Chitre, *Senior Member, IEEE*, Mehul Motani, *Member, IEEE*, and Shiraz Shahabudeen, *Member, IEEE*

***Abstract*—Propagation delays in underwater acoustic networks can be large as compared to the packet size. Conventional medium-access control (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. In this paper, we systematically explore the possibility that propagation delays can be exploited to make throughput far exceed that of networks without propagation delay. Under the assumptions of the protocol model in a single collision domain for a half-duplex unicast network, we show that the upper bound of throughput in an $N$-node wireless network with propagation delay is $N/2$. We illustrate network geometries where this bound can be achieved and study transmission schedules that help achieve it. We show that for any network, the optimal schedule is periodic and present a computationally efficient algorithm to find good schedules. Finally, we show that $N$-node network geometries that achieve throughput close to the $N/2$ bound exist for any $N$ and present a lower bound on achievable maximum throughput for bounded geometries. This paper chiefly endeavors to explore the impact and potential of nonzero propagation delays on network throughput. We believe that the novel observations in this paper could motivate further research into this area, especially random access networks with large propagation delay, with a fundamentally changed outlook on maximum achievable throughput. This could lead to novel scheduling and network configuration approaches with applications in underwater and satellite networks.**

***Index Terms*—Interference alignment by delay, large propagation delays, network geometry, throughput bounds, transmission schedules, underwater networks.**

## I. Introduction

PROPAGATION delay is the amount of time it takes a communication signal to travel from the source to the destination over a given transmission medium, i.e., $D_p = d/c$, where $D_p$ is the propagation delay, $d$ is the distance between the source and the destination, and $c$ is the speed of the signal. Since information cannot be transmitted instantaneously, the speed of any signal in any medium is finite, leading to nonzero propagation delay. In most terrestrial wireless systems, such as mobile cellular and WiFi networks, the propagation delay is small compared to the packet size, allowing us to effectively deal with the effects of propagation delay using techniques such as guard periods [1]. As a result of the slow speed of sound in water, the propagation delays in underwater networks are typically large, i.e., comparable to or larger than the packet size. For example, consider two underwater vehicles located 2000 m apart using acoustic communications. Noting the speed of sound in water is about 1500 m/s, the one-way trip takes over 1300 ms. These propagation delays are comparable to typical packet durations in these networks. The ill effects of large propagation delay have been extensively studied. The performance of handshaking protocols and acknowledgment-based retransmission schemes is known to suffer in the domain of large propagation delays [2]. That large propagation delays can adversely affect the performance of transport layer protocols like transport control protocol (TCP) has been studied in [3]. The effect of large propagation delays on medium-access control (MAC) layer protocols which prevent all data collisions has been discussed in [4].

Much effort has been spent to mitigate the ill effects of nonnegligible propagation delay (see related work in Section I-A). In this paper, we take a different approach. Rather than fighting what is a natural phenomenon (which is arguably out of our sphere of influence), we should perhaps explore how we can use propagation delay to our advantage. We draw a parallel to the opportunistic exploitation of another natural and equally troublesome phenomenon, i.e., the wireless fading channel, through the use of multiuser diversity [5]. Some authors have taken advantage of nonnegligible propagation delays in certain applications, e.g., underwater MAC [6]–[9]. However, the gains these techniques achieve are limited and the resulting performance is, in fact, no better than that with zero propagation delays. In this paper, we demonstrate the remarkable fact that, in a wireless network with nonnegligible propagation delays, the throughput performance has the potential to be significantly better than networks with negligible propagation delays.

Our aim in this paper is to develop a fundamental understanding of the impact and potential of nonzero propagation delays in half-duplex unicast networks. To illustrate how one might exploit nonzero propagation delays, we start with the simple two-node (one source–destination pair) network and see what we can learn from it.

*Example 1:* Consider a network with two nodes separated by a distance corresponding to a propagation delay $D_{12}$, as shown in Fig. 1. We assume half-duplex nodes, meaning that nodes may either transmit or receive, but not do both simultaneously. In a zero propagation delay environment, only one node may transmit at any given time. Transmission schedules are parameterized by the fraction of time you allow each node to transmit. If we want fair schedules then each node must be allowed to

Manuscript received February 10, 2012; accepted May 30, 2012. Date of publication July 18, 2012; date of current version October 09, 2012.

**Associate Editor: M. Stojanovic**.

M. Chitre is with the Electrical and Computer Engineering Department and the Acoustic Research Laboratory (ARL), National University of Singapore, Singapore 119227, Singapore (e-mail: mandar@arl.nus.edu.sg).

M. Motani and S. Shahabudeen are with the Electrical and Computer Engineering Department, National University of Singapore, Singapore 117576, Singapore (e-mail: motani@nus.edu.sg; shiraz@nus.edu.sg).

Digital Object Identifier 10.1109/JOE.2012.2203060

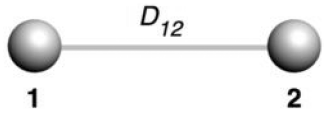

Fig. 1. A two-node network.

transmit for half of the time. An example of a fair schedule is one in which the nodes take turns transmitting to each other, i.e., first node 1 transmits and node 2 receives the packet, then node 2 transmits and node 1 receives the packet. Not surprisingly, this is the best you can do. Now let us consider the same network in a large propagation delay environment, i.e., when the propagation delay is large or comparable to the packet duration, and adopt the same schedule as before. When one node transmits, the other node must wait for a time equal to the propagation delay $D_{12}$ to receive the packet. Since both nodes are idle for part of the time that the packet is in flight, we see that a fair schedule that allows the nodes to alternately transmit is inefficient. However, in a nonzero propagation delay environment, both nodes may transmit at the same time without interfering with each other. For example, when node 1 transmits, the packet will reach node 2 after a time equal to the propagation delay $D_{12}$. During that time, node 2 may also transmit, but its transmission must complete before the packet from node 1 arrives to avoid being interfered with. This observation suggests a schedule in which both nodes transmit and receive at the same time. Setting the packet duration equal to the propagation delay leads to a fair and optimal schedule.

The idea of allowing nodes to transmit simultaneously and letting their packets "cross in flight" has been considered before [9]–[14]. Our contribution is to systematically generalize this observation to understand at a fundamental level the impact of nonzero propagation delays on the throughput of networks. Along the way, we answer the question of what are achievable throughputs of networks with nonzero propagation delays and discuss how to find optimal or near-optimal schedules for given networks. Specifically, we address the following questions.

- What is the maximum throughput of a network with nonzero propagation delays?
- What geometries and schedules achieve this maximum throughput?
- Given a network geometry, how do we determine optimal or near-optimal schedules?

To answer these questions, we adopt a system model for a network with $N$ nodes and large propagation delays. Under a mild set of assumptions, we proceed to develop a theoretical understanding of the throughput performance of such networks. The specific contributions of this paper are as follows.

- In Section III, we formulate a scheduling problem for these networks, with throughput as the metric of interest, that allows for different notions of fairness, i.e., per-node and per-link fairness.
- In Section IV-A, we prove that $N/2$ is an upper bound on the maximum throughput of such a network.
- Schedules that achieve the $N/2$ upper bound are called perfect schedules. In Section IV-B, we prove several useful results about the existence and properties of perfect schedules.
- In Sections V and VII-D, we demonstrate that there exist node topologies (for both small and large networks) for which the $N/2$ bound is achievable to any desired accuracy, assuming no constraints on the network size. For a specific class of networks with constrained size, we also explore achievable throughput.
- In Section VI, we prove that every network has an optimal (throughput maximizing) schedule, which is periodic. Furthermore, for a given arbitrary network, we present a computationally efficient algorithm to find schedules with high throughput.
- In Sections III-C, VII-C, and VII-D, we study and design fractional time-slot schedules, in which nodes are allowed to transmit in only a fraction of their assigned time slot, and show that this technique, which allows interference to be limited, can increase the throughput.

### A. Literature Survey

There is a large body of literature studying the throughput of terrestrial wireless networks which rely on the propagation of electromagnetic waves. In most of these types of systems, the propagation delay is small compared to the packet size and thus, most protocols in terrestrial wireless networks mitigate the effects of propagation delays with techniques such as guard periods. Our techniques explicitly account for and exploit propagation delay and are applicable to wireless networks with large propagation delay, such as underwater networks and satellite systems.

Propagation delays are an important issue in underwater networks and have been dealt with in a variety of ways. A time-division multiple-access (TDMA)-based scheduling algorithm that attempts to overlap communication packets between nodes and increase overall efficiency in a random network, is proposed in [7]. The performance is seen to be better than conventional TDMA-based schemes. The protocol uses features such as allowing multiple packets to arrive simultaneously at a node if none of them are meant for it, using propagation delay information. A similar concept was explored much earlier in a TDMA-based algorithm that considered interleaving packets in the underwater channel [10]. In this paper, only special cases of two- and three-node networks are considered. The internode one-way delays are assumed to be an integral multiple of the packet length. In the two-node case, the nodes exchange information simultaneously using a schedule, with a period equal to twice the one-way propagation delay. It allows for the packet duration to be equal to one-way latency. However, no performance measures are shown, and as noted by the authors, their protocol is only suitable for small networks with specific geometries. Moreover, the authors stated that with increasing number of nodes, the network throughput would be "decreased substantially." In contrast, we show that the maximum throughput scales linearly with the number of nodes in the network. In [15], Badia *et al.* propose an integer-linear programming-based scheduling algorithm that takes propagation delay and interference into account. However, they do not explicitly present any results on how the network performs with increasing propagation delay. In [8], the objective is to improve *ad hoc* request-to-send (RTS)/clear-to-send (CTS)-based

protocol performance in long propagation delay scenarios. It requires precise time synchronization and one-way range estimation ability, i.e., control packets carry departure times so that receiver can estimate range. This delay information is used in a scheduling algorithm for data transmission. The essential idea is to schedule data transmissions in the future using the RTS/CTS exchange, taking into account each node's current schedule. The results demonstrate better performance over most other published multiple-access collision avoidance (MACA)-based protocols. Though a maximum interference range is considered for each node, detection or decoding losses within the interference range are not considered and losses are only due to collisions. In [6], a distance-aware *ad hoc* protocol using RTS/CTS control packet exchange allows a node to use different control packet sizes for different receivers, leading to a decrease in the average control packet size and increased overall system efficiency. All the reviewed protocols and algorithms proposed for underwater networks that aim to counter the ill effects of propagation delay seem to have poorer performance as propagation delay increases, with the best throughput when propagation delay becomes zero. This notion is illustrated in [16], where the best case normalized throughput of ALOHA-based protocols is shown to occur at zero propagation delay. None of the reviewed protocols indicate how propagation delay could be exploited to increase throughput to exceed that of a zero propagation delay scenario.

Interference alignment [11] refers to the idea that signals can be designed to overlap at receivers where they cause interference but be interference free at the desired receiver. This leads to the interesting result that a $K$ user interference network has $K/2$ degrees of freedom. This idea is related to the approach in our paper but there are significant differences. In [11], a $K$ user interference network is defined as $2K$ nodes with $K$ arbitrary predetermined, source–destination pairs. Interference alignment relies on the careful design of transmit signals and achieves the precise alignment in the signal space domain, effectively exploiting phase differences at different receivers. We exploit large propagation delays to increase performance by overlapping interference in the time domain at certain nodes and further by opportunistically selecting these nodes to transmit at the same time. This idea of *interference alignment by delay* or *time interference alignment* has been explored in [12], [13], and [17], and more recently in [14]. In [12], Cadambe and Jafar show that a $K$-user network with uniformly distributed random propagation delays can almost surely achieve $K/2$ degrees of freedom. In [14], Blasco *et al.* further explore the degrees of freedom for interference alignment by delay for randomly placed nodes in $n$-dimensional Euclidean space. However, practical constraints on symbol durations and propagation delay measurement may severely limit the application of this idea in real networks. In [13] and [17], Mathar *et al.* explore the placement of transmitter–receiver pairs to achieve high throughput through the use of interference alignment by delay. In this paper, we take this idea a step further by systematically studying geometries and schedules that allow high throughput under different fairness constraints. We also present a practical algorithm that can be used to generate a high-throughput schedule for any given network geometry.

## II. System Model and Assumptions

We consider an $N$-node network with nonzero propagation delay $D_{ij}$ between every pair of nodes $(i, j)$, $\forall\ i \neq j$ and $i, j \in \mathbb{Z}^+$. 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*. A message transmitted by a node reaches every node (other than the transmitting node) after the appropriate propagation delay. In compliance with the *protocol model* [18], we assume that if two messages overlap in time at the receiver node, that node is unable to receive either message successfully. By setting the transmission range of the protocol model to the size of our network, we get an interference range larger than the network and therefore a single collision domain. This model allows us to study the effects of propagation delay independently of physical layer considerations such as transmit power and propagation loss.

Moreover, the single collision domain model is directly applicable to many underwater sensor networks. Due to the complex time-varying channel experienced by underwater acoustic communication links [19], small underwater sensor networks are often configured to transmit with sufficient power for all nodes in the network to successfully receive packets (e.g., [20]). It is also fairly common in the analysis of MAC protocols to assume a fully connected network, or equivalently, a single collision domain (e.g., [8], [10], [21], and [22]). Networks that are not fully connected (i.e., partially connected) have to deal with less interference than fully connected networks, as nodes far enough from each other do not interfere with each other. Since the number and timings of the allowed transmissions are limited by interference constraints, the single collision domain throughput analysis provides a lower bound for more general arbitrarily connected networks.

When the node at which the overlap occurs is the destination node for any of the overlapping messages, a *collision* is said to occur and the message is lost. A message is considered to be an *interference* at all nodes other than the source and destination nodes. We assume a network with links with constant rate $\beta$ and no message loss (except for loss as a result of collision); the number of bits of information carried by a message of duration $\mu$ is thus $\beta\mu$. The *normalized throughput* (or simply *throughput*) $S$ of the network is measured as the total number of bits of information successfully received by all nodes in the network per unit time, normalized by the link rate $\beta$. We define a *successful transmission* as a transmission that results in a successful reception of the message at the destination node. The throughput can also be measured in terms of the total number of bits of information successfully transmitted by all nodes in the network per unit time, normalized by the link rate.

A collision at the destination node of a message results in loss of the message. This is clearly undesirable if we wish to maximize the throughput in the network. In a single collision domain network without propagation delay, at most one node may transmit a message at a given time to ensure successful reception thus constraining the maximum throughput to 1. This maximum can be easily achieved using TDMA. In a network with nonzero propagation delay, more than one node may be

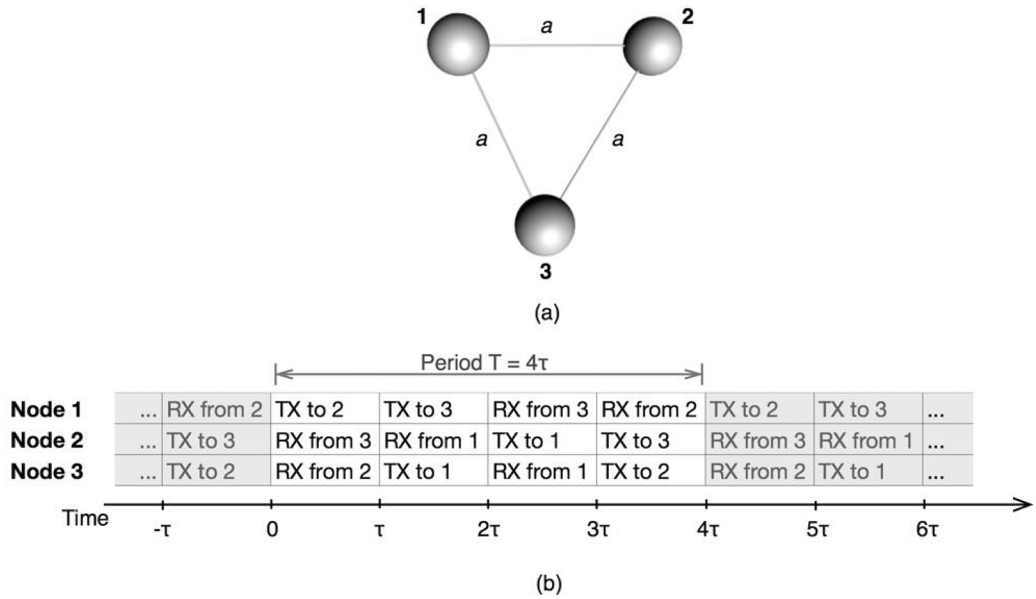


Fig. 2. A three-node equilateral triangle network and its transmission schedule. (a) Geometry for a three-node equilateral triangle network, (b) Periodic schedule.

allowed to transmit messages at one time as long as the messages do not collide at the destination nodes. In this paper, we seek to understand the maximum throughput of an $N$-node network with propagation delays. We also develop methods to construct the transmission schedules that enable us to achieve high throughput.

*Example 2:* The concept is best illustrated through an example. Consider a three-node network with the nodes located at the vertices of an equilateral triangle, as shown in Fig. 2(a). Let the length of each side of the triangle be such that the propagation delays $D_{12} = D_{13} = D_{23} = a$. We let each node transmit messages of duration $\mu = a$ as per the periodic schedule shown in Fig. 2(b). For each node, the schedule ensures that the interference from other nodes only arrives when the node is transmitting. The nodes can successfully transmit six messages with $\beta\mu$ bits each during each period $T = 4\ \mu$. Using the schedule shown, a throughput $S = (6\beta\mu/4\ \mu)/\beta = 1.5$ can be achieved. This is 50% higher than the maximum throughput for a three-node network without propagation delay. We will show than even larger improvements in throughput are possible for networks with more nodes.

## III. Mathematical Preliminaries

### A. Delay Matrices

Let $\mathbf{x}_j$ be the position vector of the node $j$ in a wireless network with $N$ nodes. The propagation delays between every pair of nodes can be written as a *delay matrix*. The entries in the delay matrix are nonnegative real numbers. In this paper, we assume a slotted model for time, where the length of a time slot is $\tau$. If needed, we can let $\tau \to 0$ to derive results for a continuous time model. Throughout this paper, we represent the network geometry in terms of a delay matrix $\mathbf{D}$ with time in units of slot length

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

where $c$ is the signal propagation speed. The largest propagation delay $G = \max_{i,j} D_{ij}$ characterizes the physical size of the network with respect to $c\tau$ and is termed as the *size* of the network.

Since $|\mathbf{x_i} - \mathbf{x_j}| = |\mathbf{x_j} - \mathbf{x_i}|$, delay matrices are symmetric, i.e., $D_{ij} = D_{ji}$. Furthermore, since $|\mathbf{x_i} - \mathbf{x_i}| = 0$, delay matrices have an all-zero diagonal, i.e., $D_{jj} = 0$. For a network with no two nodes in the same location, $D_{ij} > 0,\ \forall\ i \neq j$. These conditions are automatically satisfied by all delay matrices representing a physical network geometry. However, we will also encounter delay matrices that satisfy these conditions but do not represent any physical network geometry, i.e., no set $\{\mathbf{x}_j\}$ of node positions in 3-D Euclidean space can be found such that the delay constraints imposed by the delay matrix are met. We term a delay matrix that has a corresponding physical network geometry in $n$-dimensional space Euclidean space as an $n$-dimensional Euclidean delay matrix (EDM).[1] It can be shown that every EDM satisfies the Schoenberg criterion [23, p. 231]

$$-\mathbf{V}_N^T \dot{\mathbf{D}} \mathbf{V}_N \in \mathbb{S}_+^N \tag{2}$$

where $\dot{D}_{ij} = (c\tau D_{ij})^2$ and $\mathbf{V}_N$ is the Schoenberg auxiliary matrix [23, p. 228]

$$\mathbf{V}_N = \frac{1}{\sqrt{2}} \begin{bmatrix} -\mathbf{1}^T \\ \mathbf{I} \end{bmatrix} \in \mathbb{R}^{N \times N-1} \tag{3}$$

and $\mathbb{S}_+^N$ is the subspace of all symmetric matrices with positive entries. Furthermore, the rank of $-\mathbf{V}_N^T \dot{\mathbf{D}} \mathbf{V}_N$ is equal to the dimension $n$ of the Euclidean space.

Consider a network with all rational internode delays $D_{ij} = p_{ij}/q_{ij}$ with $p_{ij} \in \mathbb{Z}, q_{ij} \in \mathbb{Z}^+$. We choose a new slot length

[1]Note that we use the term EDM for a matrix of delays between nodes, whereas Dattorro [23] uses the term EDM for a matrix containing the square of distances.

$\tau' = \tau/\text{LCM}\{q_{ij}\}$, where $\text{LCM}\{q_{ij}\}$ is the least common multiple of all the denominators in the delay matrix. Using this slot length, the new delay matrix is given by

$$D'_{ij} = D_{ij}\frac{\tau}{\tau'} \tag{4}$$

$$= \frac{p_{ij}}{q_{ij}}\text{LCM}\{q_{ij}\}. \tag{5}$$

Since $\text{LCM}\{q_{ij}\}$ is divisible by all $q_{ij}$, all entries in the delay matrix $\mathbf{D}'$ are integers. Hence, any network with rational internode delays can be represented by an *integer delay matrix* under the appropriate choice of slot length. In general, internode Euclidean distances may be irrational and hence the corresponding entry in an EDM may also be irrational. Since any irrational number can be approximated arbitrarily closely by a rational number, we can represent any network by an integer delay matrix which is arbitrarily close to the equivalent EDM. Without loss of generality, we therefore consider integer delay matrices in some of our analysis.

### B. Schedules

A schedule $\mathbf{Q}$ determines when each node transmits and receives messages. 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 $Q_{jt} = 0$.

If the schedule 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[2]

$$Q_{jt} = Q^{(T)}_{j,t \pmod T}. \tag{6}$$

For example, the periodic schedule shown in Fig. 2(b) is represented by

$$\mathbf{Q}^{(4)} = \begin{bmatrix} 2 & 3 & -3 & -2 \\ -3 & -1 & 1 & 3 \\ -2 & 1 & -1 & 2 \end{bmatrix}. \tag{7}$$

Any matrix resulting from the cyclic shift of all rows of $\mathbf{Q}^{(T)}$ to the left or right represents the same periodic schedule.

The length of a message $\mu$ is equal to the time-slot length $\tau$. For a network with an integer delay matrix, messages transmitted on time-slot boundaries are received at time-slot boundaries on all nodes. A message received by node $j$ during time slot $t$ must be 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

$$Q_{jt} = -i \Leftrightarrow Q_{i,t-D_{ij}} = j. \tag{8}$$

[2]Matrix $\mathbf{Q}^{(T)}$ is indexed by $(j,t)$ such that $j \in \{1,\ldots,N\}, t \in \{0,\ldots,T-1\}$. We use the notation $Q_{jt}$ and $Q_{j,t}$ to mean the row $j$ and column $t$ of matrix $Q$, with the former being used for brevity and the latter being used for clarity when the indices are more complicated mathematical expressions.

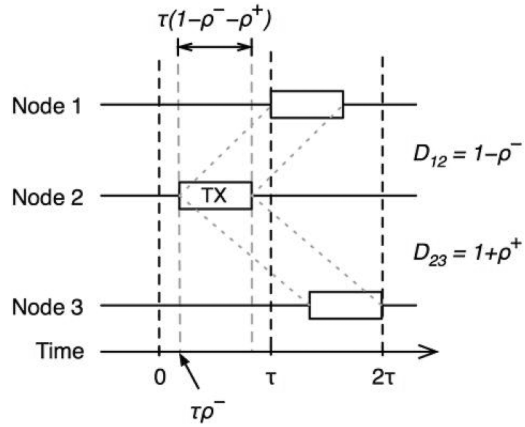


Fig. 3. An illustration of a $\rho$-schedule.

Furthermore, this implies that a schedule has equal number of transmit and receive entries, i.e.,

$$\sum_t \sum_j \mathbb{1}\left(Q^{(T)}_{jt} < 0\right) = \sum_t \sum_j \mathbb{1}\left(Q^{(T)}_{jt} > 0\right) \tag{9}$$

where $\mathbb{1}(A)$ is the indicator function with value 1 if $A$ is true and 0 otherwise. To ensure that the message is successfully received, we require that no other nodes transmit messages that arrive at node $j$ during time slot $t$

$$Q_{jt} = -i \Rightarrow Q_{k,t-D_{jk}} \leq 0 \qquad \forall\, k \neq i. \tag{10}$$

Any $N \times T$ matrix $\mathbf{Q}^{(T)}$ that satisfies (8) and (10) can be used as a schedule with period $T$ for an $N$-node network.

### C. $\rho$-Schedules

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

$$\rho^+ = \max_{ij}(D_{ij} - D'_{ij}) \tag{11}$$

$$\rho^- = -\min_{ij}(D_{ij} - D'_{ij}) \tag{12}$$

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. 3. We can then apply the constraints (8) and (10) to these networks as well. We call a schedule with shortened messages of length $\mu = \tau(1-\rho^- - \rho^+)$ a fraction time-slot schedule or a *$\rho$-schedule*.

### D. Throughput

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

$$S = \frac{1}{T}\sum_t \sum_j \mathbb{1}\left(Q^{(T)}_{jt} < 0\right)$$

$$= \frac{1}{T}\sum_t \sum_j \mathbb{1}\left(Q^{(T)}_{jt} > 0\right). \tag{13}$$

For an aperiodic schedule $\mathbf{Q}$, the average throughput $S$ is evaluated over an infinite horizon

$$S = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \sum_{j} \mathbb{1}(Q_{jt} < 0). \tag{14}$$

The throughput $S'$ of a $\rho$-schedule is related to the throughput $S$ of the equivalent schedule by

$$S' = (1 - \rho^- - \rho^+)S, \qquad \rho^- + \rho^+ \leq 1. \tag{15}$$

### E. Fairness

Different schedules offer different opportunities for nodes to transmit messages to other nodes. We define several levels of *fairness* based on whether the schedule offers equal opportunities to all nodes. A schedule is said to be *per-link fair* if all nodes have equal opportunity to transmit to all other nodes. For a schedule with period $T$, this implies

$$\sum_{t=0}^{T-1} \mathbb{1}\left(Q_{jt}^{(T)} = i\right) = \text{constant} > 0 \qquad \forall\, j, i \neq j. \tag{16}$$

A schedule is said to be *per-node fair* if every node has an equal opportunity to transmit

$$\sum_{t=0}^{T-1} \mathbb{1}\left(Q_{jt}^{(T)} > 0\right) = \text{constant} > 0 \qquad \forall\, j. \tag{17}$$

Schedules are considered to be *weakly fair* if they offer some opportunity for every node to transmit, but not necessarily equal opportunity. For example, a *weakly per-node fair* schedule would imply

$$\sum_{t} \mathbb{1}\left(Q_{jt}^{(T)} > 0\right) > 0 \qquad \forall\, j. \tag{18}$$

## IV. Properties of Optimal Schedules

### A. An Upper Bound on Throughput of an $N$-Node Network

We now derive a fundamental limit on the throughput of an $N$-node network. We also show that this limit is achievable at least for some networks.

*Theorem 1 ($N/2$ Upper Bound):* The average normalized throughput of an $N$-node network cannot exceed $N/2$.

*Proof:* Consider an $N$-node network represented by an integer delay matrix and a periodic schedule with slot length $\tau$ and period $T$. During any given time slot, a node may transmit a message, receive a message, or remain idle. Since we have $N$ nodes and $T$ time slots, there are $NT$ entries in $\mathbf{Q}^{(T)}$. Hence

$$\sum_{t}\sum_{j} \mathbb{1}\left(Q_{jt}^{(T)} < 0\right) + \sum_{t}\sum_{j} \mathbb{1}\left(Q_{jt}^{(T)} > 0\right) \leq NT. \tag{19}$$

Substituting (9), we have

$$2\sum_{t}\sum_{j} \mathbb{1}\left(Q_{jt}^{(T)} < 0\right) \leq NT. \tag{20}$$

Using (13), we get

$$S \leq \frac{N}{2}. \tag{21}$$

This result is independent of $T$ and hence valid for any $T$. By letting $T \to \infty$, we can generalize this result to aperiodic schedules. Therefore, we have an upper bound of $N/2$ for all networks. ∎

### B. Perfect Schedules

A *perfect schedule* is a schedule $\mathbf{Q}$ that satisfies constraints (8) and (10) and has no zero entries

$$\sum_{t}\sum_{j} \mathbb{1}(Q_{jt} = 0) = 0. \tag{22}$$

A perfect schedule achieves the $N/2$ upper bound. The schedule we encountered in Fig. 2(b) is a periodic perfect schedule with $N = 3$, $T = 4$, and $S = 3/2$.

A perfect $\rho$-schedule uses messages that are shorter than the time slot, and therefore, does not achieve the $N/2$ upper bound. The throughput $S$ for a perfect $\rho$-schedule is

$$S = (1 - \rho^- - \rho^+)N/2. \tag{23}$$

Periodic perfect schedules play an important part in our understanding of maximum possible throughput of a network with propagation delays. Next, we derive a few important results related to perfect schedules of period $T$ for $N$-node networks.

*Theorem 2:* For networks with odd number of nodes, perfect schedules with odd period do not exist.

*Proof:* Consider an $N$-node network with a periodic schedule $\mathbf{Q}^{(T)}$. Let $N$ and $T$ be odd. The total number of entries in the schedule $NT$ is therefore also odd and is given by

$$NT = \sum_{t}\sum_{j} \mathbb{1}\left(Q_{jt}^{(T)} < 0\right) + \sum_{t}\sum_{j} \mathbb{1}\left(Q_{jt}^{(T)} > 0\right) + \sum_{t}\sum_{j} \mathbb{1}\left(Q_{jt}^{(T)} = 0\right). \tag{24}$$

Let us assume that $\mathbf{Q}^{(T)}$ is a perfect schedule. Using (9) and (22), we have

$$NT = 2\sum_{t}\sum_{j} \mathbb{1}\left(Q_{jt}^{(T)} < 0\right). \tag{25}$$

The term on the left-hand side of the above equation is odd, but the term on the right-hand side is even. Hence, we have a contradiction, and therefore conclude that $\mathbf{Q}^{(T)}$ cannot be a perfect schedule. ∎

*Corollary 3:* For a network with odd number of nodes $N$ and a periodic schedule with an odd period $T$, the throughput is upper bounded by $(NT - 1)/2T$.

*Proof:* Let the periodic schedule $\mathbf{Q}^{(T)}$ contain $a \in \mathbb{Z}$ zeros. From (24) and (9), we have

$$NT = 2\sum_{t}\sum_{j} \mathbb{1}\left(Q_{jt}^{(T)} < 0\right) + a. \tag{26}$$

Since $NT$ is odd and $2\sum_t \sum_j \mathbb{1}(Q_{jt}^{(T)} < 0)$ is even, $a$ must be odd. $a$ can then be written as $2b+1$ for some $b \in \mathbb{Z}$. Hence

$$NT = 2\sum_t \sum_j \mathbb{1}\left(Q_{jt}^{(T)} < 0\right) + 2b + 1 \tag{27}$$

$$\therefore \frac{1}{T}\sum_t \sum_j \mathbb{1}\left(Q_{jt}^{(T)} < 0\right) = \frac{NT - 2b - 1}{2T}. \tag{28}$$

Noting that the right-hand size is largest when $b = 0$ and substituting (13), we get

$$S \leq \frac{NT-1}{2T}. \tag{29}$$

■

*Theorem 4:* For an $N$-node network, periodic per-link fair schedules can only exist for period $T = 2k(N-1), k \in \mathbb{Z}^+$.

*Proof:* Let $\mathbf{Q}^{(T)}$ be the periodic per-link fair schedule for an $N$-node network. From (16), we have

$$\sum_t \mathbb{1}(Q_{jt}^{(T)} = i) = k \qquad \forall\, i \neq j \tag{30}$$

for some constant $k \in \mathbb{Z}^+$. Exchanging labels $i$ and $j$ and using (8), we get

$$\sum_t \mathbb{1}\left(Q_{jt}^{(T)} = -i\right) = k \qquad \forall\, i \neq j. \tag{31}$$

Row $j$ of the $N \times T$ schedule matrix $\mathbf{Q}^{(T)}$ thus contains $k(N-1)$ positive entries and $k(N-1)$ negative entries. Since $\mathbf{Q}^{(T)}$ is a perfect schedule, there are no zero entries. The total number of entries in row $j$ is therefore

$$T = 2k(N-1). \tag{32}$$

■

*Theorem 5:* Perfect schedules do not exist for $N$-node linear networks for $N > 2$.

*Proof:* Consider an $N$-node linear network. Let some node $i$ transmit a message to some node $k$ at time 0, i.e.,

$$Q_{i,0} = k. \tag{33}$$

From (8) and (10), we have

$$Q_{k,D_{ik}} = -i \tag{34}$$

$$Q_{j,D_{ik}-D_{jk}} \leq 0 \qquad \forall\, j \neq i. \tag{35}$$

Assume that there exists an intermediate node $j$ in the linear network such that $D_{ik} = D_{ij} + D_{jk}$. The message arrives at node $j$ during time slot $D_{ij}$. Since $Q_{i,D_{ij}-D_{ji}} = Q_{i,0} > 0$, constraint (10) gives us

$$Q_{j,D_{ij}} = Q_{j,D_{ik}-D_{jk}} \geq 0. \tag{36}$$

Combining with (35), we have $Q_{j,D_{ij}} = 0$. Since we desire a perfect schedule, we cannot leave any entry in the schedule idle. Therefore, we conclude that the intermediate node $j$ cannot exist and that messages can only be transmitted between adjacent nodes in a linear network with a perfect schedule.

Now consider nodes 1 and 2. Node 1 has only node 2 as a neighbor and we desire a perfect schedule. Therefore, $Q_{1,t} = \pm 2,\ \forall\, t$. From (8), we then have $Q_{2,t} = \pm 1,\ \forall\, t$. Consider another node $k, 3 \leq k \leq N$ that transmits a message to node $f$ during time slot 0, i.e.,

$$Q_{k,0} = f \geq 3. \tag{37}$$

The message reaches node $h \neq f$ during time slot $D_{kh}$. If $Q_{h,D_{kh}} < 0$, then from (10), we have

$$Q_{k,D_{kh}-D_{hk}} = Q_{k,0} \leq 0. \tag{38}$$

Since this is in contradiction with (37), we conclude that

$$Q_{h,D_{kh}} \geq 0 \qquad \forall\, h \neq f. \tag{39}$$

Setting $h = 2$ and recalling that $Q_{2,t} = \pm 1,\ \forall\, t$, we have $Q_{2,D_{k,2}} = 1$. Using (8) and recalling that we have a linear network with $D_{k,2} + D_{21} = D_{k,1},\ \forall\, k \geq 3$, we get

$$Q_{1,D_{k,2}+D_{21}} = Q_{1,D_{k,1}} = -2. \tag{40}$$

Setting $h = 1$ in (39) and recalling that $Q_{1,t} = \pm 2,\ \forall\, t$, we get $Q_{1,D_{k,1}} = 2$. But this is in contradiction with (40). Hence, node $k \geq 3$ cannot be transmitted during any time slot. Since nodes 1 and 2 can only transmit to each other and all other nodes are unable to transmit, the throughput is equivalent to that of a two-node network. For a perfect schedule, an $N$-node network achieves the maximum possible throughput of $N/2$. Therefore, we conclude that a perfect schedule does not exist for a linear $N$-node network with $N > 2$. ■

## V. Illustrative Network Geometries

In this section, we study some special geometries of networks with small number of nodes, most of them achieving the $N/2$ upper bound. This helps us develop some of the intuition which will become important in later sections for the understanding of networks with large number of nodes.

### A. Two-Node Network

We have already encountered a two-node network in Example 1. Setting the time-slot duration equal to the propagation delay between the nodes, we have $\mathbf{D}$ as the delay matrix and $\mathbf{Q}^{(2)}$ as the perfect per-link fair schedule for the network

$$\mathbf{D} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{Q}^{(2)} = \begin{bmatrix} 2 & -2 \\ 1 & -1 \end{bmatrix}. \tag{41}$$

This schedule achieves the $N/2$ upper bound from Theorem 1 and has the minimum period given by Theorem 4.

### B. Three-Node Equilateral Triangle Network

We have also already encountered the three-node equilateral triangle network in Example 2. Setting the time-slot duration

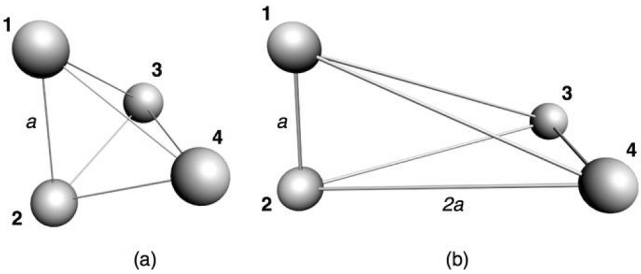


Fig. 4. (a) Regular tetrahedron and (b) stretched tetrahedron networks.

equal to the propagation delay between the nodes, we have $\mathbf{D}$ as the delay matrix and $\mathbf{Q}^{(4)}$ as the perfect per-link fair schedule for the network

$$\mathbf{D} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \quad \mathbf{Q}^{(4)} = \begin{bmatrix} 2 & 3 & -3 & -2 \\ -3 & -1 & 1 & 3 \\ -2 & 1 & -1 & 2 \end{bmatrix}. \quad (42)$$

This schedule achieves the $N/2$ upper bound and has the minimum period given by Theorem 4.

### C. Three-Node Isosceles Triangle Network

We next consider a three-node network with the nodes placed on the vertices of an isosceles triangle with the length of the long edges equal to twice the length of the short edge. Setting the time-slot duration equal to the propagation delay between the nodes at the vertices of the short edge, we have $\mathbf{D}$ as the delay matrix and $\mathbf{Q}^{(8)}$ as the perfect per-link fair schedule for the network

$$\mathbf{D} = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 2 \\ 2 & 2 & 0 \end{bmatrix} \quad (43)$$

$$\mathbf{Q}^{(8)} = \begin{bmatrix} 2 & -2 & -3 & 3 & -3 & 3 & 2 & -2 \\ 1 & -1 & 3 & -3 & 3 & -3 & 1 & -1 \\ 1 & 2 & 1 & 2 & -2 & -1 & -2 & -1 \end{bmatrix}.$$

This schedule achieves the $N/2$ upper bound and has a period given by Theorem 4 with $k = 2$.

### D. Three-Node Linear Network

Consider a three-node network where the nodes are equidistantly placed along a line. From Theorem 5, we know that linear networks with more than two nodes do not have perfect schedules. We set the time-slot duration to be equal to the propagation delay between adjacent nodes. The delay matrix $\mathbf{D}$ and a weakly fair schedule $\mathbf{Q}^{(3)}$ that achieves the upper bound $S = 4/3$ from Corollary 3 are shown as follows:

$$\mathbf{D} = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{bmatrix}, \quad \mathbf{Q}^{(3)} = \begin{bmatrix} 3 & 2 & -3 \\ 3 & 0 & -1 \\ 1 & -2 & -1 \end{bmatrix}. \quad (44)$$

### E. Four-Node Regular Tetrahedron Network

We next look at a 3-D network with four nodes placed at the vertices of a regular tetrahedron, as shown in Fig. 4(a). Setting the time-slot duration equal to the propagation delay between the nodes, we have $\mathbf{D}$ as the delay matrix and $\mathbf{Q}^{(2)}$ as the perfect per-node fair schedule for the network

$$\mathbf{D} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}, \quad \mathbf{Q}^{(2)} = \begin{bmatrix} 2 & -2 \\ 1 & -1 \\ -4 & 4 \\ -3 & 3 \end{bmatrix}. \quad (45)$$

### F. Four-Node Stretched Tetrahedron Network

If we double the length of four edges of the regular tetrahedron, as shown in Fig. 4(b), we get a four-node "stretched" tetrahedron network. Setting the time-slot duration equal to the propagation delay along the unstretched edge, we have $\mathbf{D}$ as the delay matrix and $\mathbf{Q}^{(2)}$ as the perfect per-node fair schedule for the network

$$\mathbf{D} = \begin{bmatrix} 0 & 1 & 2 & 2 \\ 1 & 0 & 2 & 2 \\ 2 & 2 & 0 & 1 \\ 2 & 2 & 1 & 0 \end{bmatrix}, \quad \mathbf{Q}^{(2)} = \begin{bmatrix} 2 & -2 \\ 1 & -1 \\ 4 & -4 \\ 3 & -3 \end{bmatrix}. \quad (46)$$

## VI. Scheduling for Arbitrary Networks

In the previous sections, we have seen that many networks can achieve high throughput by adopting optimally designed schedules that exploit propagation delays. It is natural to ask how one would go about determining the optimal schedule given the locations of the nodes in a network. In this section, we formulate this optimization problem as a sequential decision problem and solve it using dynamic programming. The resulting solution is optimal, but computationally infeasible for networks with large size and many nodes. Therefore, we also find an approximate solution that reduces the computational complexity, yet works well in practice.

### 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. Although $\mathbf{Q}^{\{t\}}$ only contains transmissions made until time slot $t$, it captures the resulting receptions and interference at later time slots.

Let the action to be taken at time $t$ be $\mathbf{x}^{\{t\}}$. The action defines 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 $t$

$$\mathbf{Q}^{\{t+1\}} = \Gamma\left(\mathbf{Q}^{\{t\}}, \mathbf{x}^{\{t\}}\right) \quad (47)$$

where the state transition function $\Gamma(\cdot)$ updates $\mathbf{Q}^{\{t\}} \rightarrow \mathbf{Q}^{\{t+1\}}$ to include transmissions in $\mathbf{x}^{\{t\}}$ and the corresponding receptions in accordance with (8). The set of feasible actions $\mathcal{X}(\mathbf{Q}^{\{t\}})$ consists of all possible actions where $Q_{jt}^{\{t\}} = 0$

when $x_j^{\{t\}} > 0$ and the resulting state $\mathbf{Q}^{\{t+1\}}$ denotes a valid schedule in accordance with (8) and (10).

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

$$C\left(\mathbf{x}^{\{t\}}\right) = \sum_{j=1}^{N} \mathbb{1}\left(x_j^{\{t\}} > 0\right) \qquad \forall\, \mathbf{x}^{\{t\}} \in \mathcal{X}\left(\mathbf{Q}^{\{t\}}\right). \quad (48)$$

The throughput $S$ is the average reward

$$S = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} C\left(\mathbf{x}^{\{t\}}\right). \quad (49)$$

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 size $G = \max_{i,j} D_{ij}$, 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 only 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 $t'$. Matrix $\hat{\mathbf{Q}}^{\{t\}}$ is sufficient to reconstruct the full schedule (including receptions) when necessary.

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

$$\mathbf{x}^{\{t\}} = X^*\left(\hat{\mathbf{Q}}^{\{t\}}\right) \quad (50)$$

$$\hat{\mathbf{Q}}^{\{t+1\}} = \Gamma(\hat{\mathbf{Q}}^{\{t\}}, \mathbf{x}^{\{t\}}). \quad (51)$$

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 $|\hat{\mathcal{X}}_t| \leq N^N$. Using this finite state–space formulation of the dynamic decision problem described above, we can show that periodic optimal schedules must exist for all networks.

*Theorem 6:* Every network has an optimal schedule that is periodic.

*Proof:* Consider an $N$-node network with size $G$. Under policy $X^*$, the dynamic program generates a sequence of states $(\hat{\mathbf{Q}}^{\{0\}}, \hat{\mathbf{Q}}^{\{1\}}, \hat{\mathbf{Q}}^{\{2\}}, \ldots \hat{\mathbf{Q}}^{\{N^{NG}\}})$ at time slots $(0, 1, 2, \ldots, N^{NG})$. Since the number of possible states $|\hat{\mathcal{Q}}| \leq N^{NG}$, at least two of the states in this sequence must be identical (pigeonhole principle). Let $t_1$ and $t_2$ be two time slots such that state $\hat{\mathbf{Q}}^{\{t_1\}} = \hat{\mathbf{Q}}^{\{t_2\}}, 0 \leq t_1 < t_2 \leq N^{NG}$ and there exists no $t_3, t_1 < t_3 < t_2$ with state $\hat{\mathbf{Q}}^{\{t_3\}} = \hat{\mathbf{Q}}^{\{t_1\}}$. The decision made by policy $X^*$ only depends on $\hat{\mathbf{Q}}^{\{t\}}$ and fully determines the next state $\hat{\mathbf{Q}}^{\{t+1\}}$. Hence, the sequence of states after $\hat{\mathbf{Q}}^{\{t_2\}}$ must be identical to the sequence of states after $\hat{\mathbf{Q}}^{\{t_1\}}$, i.e., the schedule generated must be periodic with a period $T = t_2 - t_1 \leq N^{NG}$. ∎

Since every network has an optimal schedule with some period $T$, we can compute the throughput over a single period

$$S = \frac{1}{T} \sum_{t=0}^{T} C\left(\mathbf{x}^{\{t\}}\right). \quad (52)$$

A perfect schedule is optimal, since its throughput satisfies the $N/2$ upper bound with an equality. A search for a perfect schedule for a given network therefore has to only consider periodic schedules with period $T \leq N^{NG}$. If we are interested in per-link fair schedules or $N$ is odd, Theorems 4 and 2 further limit the search space. Since the search space of periodic $N$-node schedules with finite $T$ is finite (although it could be very large), the question of existence of a periodic schedule for a given network is *computable*. In fact, many of the perfect schedules for small networks shown in this paper were computed using a recursive search algorithm exploring the search space of periodic schedules and guided by the intuition that all interference slots must be used for transmission (since they cannot be used for successful reception).

### C. Dynamic Programming

The deterministic sequential decision problem described in Section VI-B can be solved using techniques in dynamic programming [24]. We denote the value function by $V(\hat{\mathbf{Q}}) : \hat{\mathcal{Q}} \to \mathbb{R}$. We can write the optimal policy in terms of the value function

$$X^*(\hat{\mathbf{Q}}) = \arg \max_{\mathbf{x}\in\mathcal{X}(\hat{\mathbf{Q}})} (C(\mathbf{x}) + V(\Gamma(\hat{\mathbf{Q}}, \mathbf{x}))). \quad (53)$$

The value function must satisfy the Bellman equation

$$V(\hat{\mathbf{Q}}) = \max_{\mathbf{x}\in\mathcal{X}(\hat{\mathbf{Q}})} (C(\mathbf{x}) + V(\Gamma(\hat{\mathbf{Q}}, \mathbf{x}))) - V_0 \quad (54)$$

where $V_0$ is an appropriate constant required to keep the value function finite. A standard technique known as *relative value iteration* 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 [24, Sec. 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 size).

### D. Computationally Efficient Approximate Algorithm

If we know the value function, the problem simplifies to enumerating the decision space and finding the optimal decision. Rather than estimate the value function iteratively, it is possible to develop an approximate value function based on the structure of the problem. One such approximate value function based on an intuitive understanding of the problem is presented in [25]. The main idea is to make transmission decisions such that the interference they cause overlaps as much as possible, and then

to use the interfered slots for additional transmissions. The computational complexity of the algorithm still grows rapidly with $N$, since the decision space with a cardinality of $\mathcal{O}(N^N)$ has to be enumerated.

In this section, we present a two-step algorithm with significantly lower computational complexity. First, the large decision space is replaced with a number of smaller sequentially enumerated decision spaces. Second, the value function is estimated by an approximate value function, which is based on the potential of a given partial schedule to accommodate future transmissions. Details are given below.

*1) Factorization of the Decision Space:* The decision space $\hat{\mathcal{X}}_t$ consists of all feasible combinations of transmission decisions for each of the $N$ nodes during the time slot under consideration, and therefore has a cardinality of $\mathcal{O}(N^N)$. We can reduce this by making sequential transmission decisions, with each decision represented by a 2-tuple $(j,k)$ for a single transmission from node $j$ to node $k$ at time $t$. Since we have $N(N-1)$ possible 2-tuples and a maximum of $N$ transmissions during time slot $t$, the computational complexity of the enumeration of the decision spaces for a given time slot reduces from $\mathcal{O}(N^N)$ to $\mathcal{O}(N^3)$.

Let $\bar{\mathbf{Q}}^{\{t,u\}} \in \bar{\mathcal{Q}}$ be the partial schedule after $u-1$ transmission decisions have been made for time slot $t$, and $\bar{C}_t$ be the total number of transmissions during time slot $t$. Since we have $N$ nodes, $\bar{C}_t \leq N$. When the $u$th transmission decision $\bar{\mathbf{x}}^{\{t,u\}}$ for time slot $t$ is made

$$\bar{\mathbf{Q}}^{\{t,u+1\}} = \Gamma\left(\bar{\mathbf{Q}}^{\{t,u\}}, \bar{\mathbf{x}}^{\{t,u\}}\right) \qquad \forall\, u < \bar{C}_t \tag{55}$$

$$\bar{\mathbf{Q}}^{\{t+1,1\}} = \Gamma\left(\bar{\mathbf{Q}}^{\{t,\bar{C}_t\}}, \bar{\mathbf{x}}^{\{t,\bar{C}_t\}}\right). \tag{56}$$

The throughput $S$ is the average number of transmissions in a time slot

$$S = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \bar{C}_t. \tag{57}$$

*2) Approximate Value Function:* A value function $\bar{V}(\bar{\mathbf{Q}})$ on state space $\bar{\mathcal{Q}}$ must satisfy Bellman's equation

$$\begin{aligned}\bar{V}(\bar{\mathbf{Q}}) &= \max_{\bar{\mathbf{x}}\in\bar{\mathcal{X}}(\bar{\mathbf{Q}})} \left(C(\bar{\mathbf{x}}) + \bar{V}\left(\Gamma(\bar{\mathbf{Q}},\bar{\mathbf{x}})\right)\right) - \bar{V}_0 \\ &= \max_{\bar{\mathbf{x}}\in\bar{\mathcal{X}}(\bar{\mathbf{Q}})} \bar{V}\left(\Gamma(\bar{\mathbf{Q}},\bar{\mathbf{x}})\right) - \bar{V}_0' \end{aligned} \tag{58}$$

where $\bar{V}_0' = \bar{V}_0 - 1$ since the reward $C(\bar{\mathbf{x}})$ for a single transmission decision is 1. The optimal decision $\bar{\mathbf{x}}^*$ is given by

$$\bar{\mathbf{x}}^* = \arg\max_{\bar{\mathbf{x}}\in\bar{\mathcal{X}}(\bar{\mathbf{Q}})} \bar{V}(\Gamma(\bar{\mathbf{Q}},\bar{\mathbf{x}})). \tag{59}$$

The optimal decisions using the factorized decision space can be made if we have the knowledge of the value function $\bar{V}$ on state space $\bar{\mathcal{Q}}$. The computational complexity of determining the exact value function through dynamic programming is prohibitive for large $N$ and $G$, and therefore, we next develop an approximate value function guided by our intuition.

Since we wish to maximize the throughput, we approximate the value of a state by its potential to accommodate future transmissions within a specified time horizon given interference and half-duplex constraints. Let $Z_{jk\tau}(\bar{\mathbf{Q}}^{\{t\}})$ be a *transmission indicator function* with value 1 if a transmission from node $j$ to node $k$ is permitted in time slot $t+\tau$ given the partial schedule $\bar{\mathbf{Q}}^{\{t\}}$, and 0 otherwise. Taking constraints (8) and (10) into account and disallowing self-transmissions

$$Z_{jk\tau}(\bar{\mathbf{Q}}^{\{t\}}) = \begin{cases} 0, & \text{if } j = k \\ 0, & \text{if } \bar{Q}^{\{t\}}_{j,t+\tau} \neq 0 \\ 0, & \text{if } \bar{Q}^{\{t\}}_{k,t+\tau+D_{jk}} \neq 0 \\ 0, & \text{if } \exists i \text{ s.t. } \bar{Q}^{\{t\}}_{i,t+\tau+D_{jk}-D_{ik}} > 0 \\ 0, & \text{if } \exists i,l \text{ s.t. } \bar{Q}^{\{t\}}_{l,t+\tau+D_{ji}-D_{il}} = i \\ 1, & \text{otherwise.} \end{cases} \tag{60}$$

The transmission indicator function $Z_{jk\tau}$ captures the potential to transmit given a partial schedule and therefore may be used to approximate the value function $\bar{V}$.

Since a transmission leaves the network within $G$ time slots, it is sufficient to consider a time horizon $G$

$$\bar{V}\left(\bar{\mathbf{Q}}^{\{t\}}\right) = \sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{\tau=0}^{G} Z_{jk\tau}(\bar{\mathbf{Q}}^{\{t\}}). \tag{61}$$

The resulting algorithm incorporating the factorization of the decision space and the approximate value function is summarized as Algorithm 1.

**Algorithm 1: Algorithm to Determine Transmissions in Time Slot $t$ and Update Partial Schedule**

**Require:** $N, G, \mathbf{D}$
**Require:** Partial schedule $\mathbf{Q}^{\{t\}}$
$\bar{\mathbf{Q}} \leftarrow \mathbf{Q}^{\{t\}}$
$u \leftarrow 0$
**while** true **do**
  Compute $\mathbf{Z}$ from $\bar{\mathbf{Q}}$ according to (60)
  $\bar{\mathcal{X}} \leftarrow \{(j,k),\ \forall\, j,k \text{ s.t. } Z_{jk0} = 1\}$
  **if** $\bar{\mathcal{X}}$ is empty **then**
    **return** $\mathbf{Q}^{\{t+1\}} \leftarrow \bar{\mathbf{Q}}$, $C_t \leftarrow u$
  **end if**
  $u \leftarrow u + 1$
  Compute $\bar{V}(\mathbf{\Gamma}(\bar{\mathbf{Q}},\bar{\mathbf{x}}))$, $\forall\ \ \bar{\mathbf{x}} \in \bar{\mathcal{X}}$ according to (61)
  $\bar{\mathbf{x}}^* \leftarrow \arg\max \bar{V}(\mathbf{\Gamma}(\bar{\mathbf{Q}},\bar{\mathbf{x}}))$
  $\bar{\mathbf{Q}} \leftarrow \mathbf{\Gamma}(\bar{\mathbf{Q}},\bar{\mathbf{x}}^*)$
**end while**

For every time slot, the algorithm sequentially selects transmission decisions from all allowable transmissions during that time slot. Every transmission reduces the potential for future transmissions; this is captured by the value function approximation. At each step, a transmission that has minimum impact on the potential future transmissions is chosen. Low-impact transmissions are those that ensure that their interference at unintended nodes largely overlaps with interference from previous transmissions. When no further transmission is possible during a time slot, the algorithm moves on to determining the transmissions in the next time slot.

The computational complexity of the algorithm is $\mathcal{O}(N^3)$, a significant improvement over the dynamic programming solution. Although the resulting algorithm uses an approximate

value function and therefore is no longer optimal, it performs very well in practice. When applied to the various illustrative network geometries presented in this paper (including the six-node network geometry presented in Example 3), the algorithm finds solutions with the optimal throughput known for those geometries.

## VII. Existence of Geometries With Good Schedules

In Section VI, we saw how to find good schedules given a network geometry. We are also interested in the converse—given a good schedule, what network geometries admit that schedule?

### A. Aliased Network Geometries

In this section, we explore this question and show that a given periodic schedule is associated with a family of delay matrices. Each delay matrix represents a potential network geometry where the schedule can be used.

Let $\mathbf{D}$ be the delay matrix and $\mathbf{Q}^{(T)}$ be a periodic schedule associated with an $N$-node network. Let $\mathbf{\Lambda}$ be an $N \times N$ symmetric matrix with all $\Lambda_{ij} \in \mathbb{Z}$ and $\Lambda_{jj} = 0$. We define a new delay matrix

$$\bar{\mathbf{D}} = \mathbf{D} + T\mathbf{\Lambda}. \tag{62}$$

From (6), we have

$$\begin{aligned} Q_{j,t-\bar{D}_{ij}} &= Q_{j,t-D_{ij}-T\Lambda_{ij}} \\ &= Q^{(T)}_{j,(t-D_{ij}-T\Lambda_{ij})} \pmod T \\ &= Q^{(T)}_{j,(t-D_{ij})} \pmod T \\ &= Q_{j,t-D_{ij}} \qquad \forall\, i,j,t. \end{aligned} \tag{63}$$

Constraints (8) and (10) are satisfied for $\mathbf{D}$ and therefore also satisfied for $\bar{\mathbf{D}}$. The schedule $\mathbf{Q}^{(T)}$ can thus also be used in a network corresponding to the delay matrix $\bar{\mathbf{D}}$. Under a periodic schedule with period $T$, the delay matrix $\bar{\mathbf{D}}$ is considered to be an *alias* of delay matrix $\mathbf{D}$.

Since there are an infinite number of matrices $\mathbf{\Lambda}$ with $\Lambda_{jt} \in \mathbb{Z}$, a delay matrix $\mathbf{D}$ defines a family of delay matrices that may correspond to networks that admit the same periodic schedule $\mathbf{Q}^{(T)}$. We represent this family of delay matrices by a *fundamental delay matrix* $\mathbf{D}^{(T)}$ such that $0 \le D^{(T)}_{jt} < T,\ \forall\, j,t$. Any other delay matrix $\mathbf{D}$ in the family is related to the fundamental delay matrix $\mathbf{D}^{(T)}$ such that

$$D^{(T)}_{jt} = D_{jt} \pmod T. \tag{64}$$

Not all delay matrices in a family are EDMs. The delay matrices that are 3-D EDMs correspond to network geometries that can exist in 3-D. If a delay matrix is not a 3-D EDM, we can find a 3-D EDM closest to the given delay matrix and use an appropriate $\rho$-schedule with it. A generating list of 3-D relative node locations that most closely matches a given delay matrix can be obtained by *list reconstruction* techniques [23, pp. 295–297]. A new delay matrix can be computed from the generating list—by definition this delay matrix is a 3-D EDM. As shown in Example 3 (Section VII-C), this approach can be used to find 3-D network geometries that admit a known high-throughput schedule.

### B. Odd–Even Schedules

Since a given schedule can be used in many network geometries, we can start with a known good schedule and find a suitable geometry for a network to ensure that a high throughput can be achieved. To explore this idea, we study a special family of period-2 perfect schedules for any even $N$. We then show that there indeed exist a large number of network geometries that can achieve high throughput from the use of such schedules.

The four-node network geometry in Section V-F corresponds to a perfect per-node fair period-2 schedule where all nodes transmit in one time slot and receive during the next time slot without causing any collisions. This inspires us to develop a generalized schedule $\mathbf{Q}^{(2)}$ for an $N$-node network (for even $N$) that is able to have all nodes transmit simultaneously without causing collisions. We call this schedule the *odd–even schedule* as all nodes transmit during the even time slots and receive during the odd time slots. The schedule $\mathbf{Q}^{(2)}$ and the tridiagonal fundamental delay matrix $\mathbf{D}^{(2)}$ that defines the family of delay matrices that admit the schedule are shown as follows:

$$\mathbf{D}^{(2)} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{65}$$

$$\mathbf{Q}^{(2)} = \begin{bmatrix} 2 & -2 \\ 1 & -1 \\ 4 & -4 \\ 3 & -3 \\ 6 & -6 \\ 5 & -5 \\ \vdots & \vdots \\ N & -N \\ N-1 & -(N-1) \end{bmatrix}. \tag{66}$$

$N$-node networks corresponding to any of the delay matrices in the family defined by (65) can use the perfect schedule given by (66) to achieve a throughput of $N/2$. For $N = 2$, this gives us the two-node network in Example 1, and for $N = 4$, this yields the four-node stretched tetrahedron network in Section V-F.

### C. Odd–Even $\rho$-Schedules

For even $N \ge 6$, do any delay matrices in the family defined by (65) correspond to network geometries in 3-D? Equivalently, are any of the delay matrices in the family 3-D EDMs? In [17], Mathar and Zivkovic suggest a negative answer to this question. However, we show that there are network geometries that have delay matrices $\mathbf{D}'$ that are close to some of the delay matrices in the family defined by $\mathbf{D}^{(2)}$ in (65). These geometries can admit high-throughput schedules. The $\rho$-schedule given by $\mathbf{Q}^{(2)}$ can be used for the network corresponding to delay matrix $\mathbf{D}'$ to yield a throughput $S$ given by

$$S = (1 - \rho^- - \rho^+)N/2 \tag{67}$$

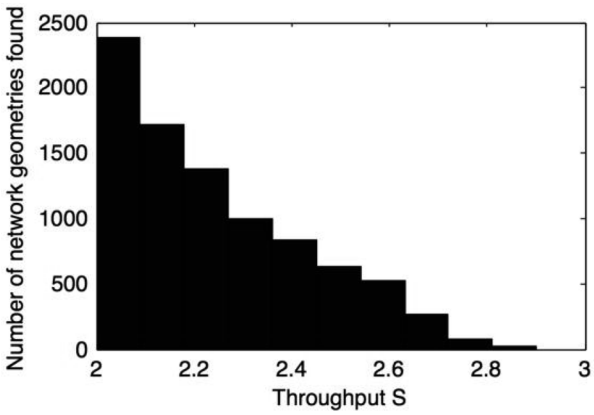


Fig. 5. Histogram of the throughput of six-node networks satisfying an odd–even $\rho$-schedule.

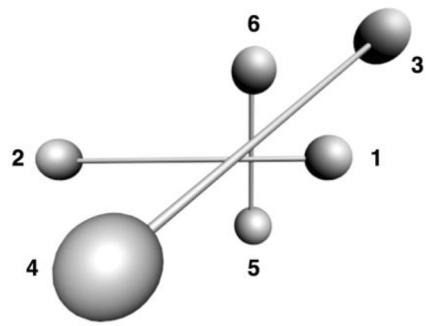


Fig. 6. A six-node network satisfying an odd–even $\rho$-schedule.

$$\rho^- = -\min_{i,j}(D'_{ij} - D_{ij}) \tag{68}$$

$$\rho^+ = \max_{i,j}(D'_{ij} - D_{ij}) \tag{69}$$

provided $\rho^- + \rho^+ \leq 1$.

The general form of the delay matrices $\bar{\mathbf{D}}$ in the family defined by $\mathbf{D}^{(2)}$ is obtained by setting $T = 2$ in (62)

$$\bar{\mathbf{D}} = \mathbf{D}^{(2)} + 2\mathbf{\Lambda}. \tag{70}$$

There exists a 3-D EDM $\mathbf{D}'$ close to each delay matrix $\bar{\mathbf{D}}$ obtained for different $\mathbf{\Lambda}$. Given $\bar{\mathbf{D}}$, $\mathbf{D}'$ and the corresponding 3-D network geometry can be found using the technique outlined in Section VII-A. We can thus search for $N$-node network geometries with high throughput by varying $\mathbf{\Lambda}$. For small $N$ and a small maximum network size, a brute force search over all possible $\mathbf{\Lambda}$ is computationally feasible. The search for a six-node network with maximum size of approximately 6 resulted in 8842 network geometries with throughput $S > 2$. A histogram of the throughput of these networks is shown in Fig. 5 and the network with the highest throughput is described in detail in the example below.

*Example 3:* Consider a six-node network with nodes located at the coordinates given by the columns of matrix $\bar{\mathbf{X}}$, as shown in Fig. 6, in an environment with a propagation speed $c$

$$\bar{\mathbf{X}} = c\begin{bmatrix} 0 & -4.83 & 0.84 & -3.20 & -1.23 & -1.08 \\ 0 & 1.28 & 0.61 & -2.23 & 0.19 & 1.15 \\ 0 & -0.15 & 1.76 & 0.90 & -1.64 & 1.24 \end{bmatrix}. \tag{71}$$

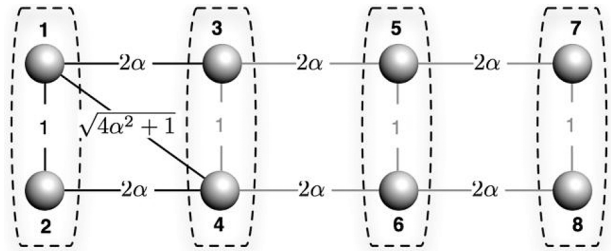


Fig. 7. A 2-D $N$-node network for even $N$.

The delay matrix $\bar{\mathbf{D}}$ for this network is

$$\bar{\mathbf{D}} = \begin{bmatrix} 0 & 5.00 & 2.04 & 4.00 & 2.06 & 2.01 \\ 5.00 & 0 & 6.02 & 4.01 & 4.05 & 4.01 \\ 2.04 & 6.02 & 0 & 5.01 & 4.00 & 2.06 \\ 4.00 & 4.01 & 5.01 & 0 & 4.02 & 4.00 \\ 2.06 & 4.05 & 4.00 & 4.02 & 0 & 3.04 \\ 2.01 & 4.01 & 2.06 & 4.00 & 3.04 & 0 \end{bmatrix}. \tag{72}$$

For a periodic schedule with period $T = 2$, the fundamental delay matrix $\bar{\mathbf{D}}^{(2)}$ is obtained by applying the modulo 2 operation on all entries of $\bar{\mathbf{D}}$. Comparing the entries in matrix $\bar{\mathbf{D}}^{(2)}$ with those from the prototype odd–even distance matrix $\mathbf{D}^{(2)}$ of the form (65), we have $\rho^- = 0$ and $\rho^+ = 0.06$. Therefore, we can apply a $\rho$-schedule to get a throughput $S = 2.82$.

### *D. Large Networks With Good Schedules*

For any even number of nodes $N \geq 4$, it is possible to construct a 2-D network geometry that achieves the $N/2$ upper bound as closely as desired, provided we allow the size of the network to grow arbitrarily. To construct such a network, we place the $N$ nodes pairwise, as shown in Fig. 7. The delay matrix of this network is denoted by $\bar{\mathbf{D}}$. We place each pair such that the delay between the nodes of the pair is 1, i.e., $\bar{D}_{j,j+1} = 1$ for odd $j$, and the delay between adjacent pairs $\bar{D}_{j,j+2} = 2\alpha,\ \forall\ j,\ \alpha \in \mathbb{Z}^+$. Thus, the delay $\bar{D}_{ij} = \alpha|i-j|$ is even for odd $i$ and $j$, or even $i$ and $j$. If $i$ is odd and $j$ is even

$$\bar{D}_{ij} = \sqrt{\bar{D}^2_{i+1,j} + 1}. \tag{73}$$

When the difference between $i$ and $j$ is large, $\bar{D}_{i+1,j}$ is large and $\bar{D}_{ij} \approx \bar{D}_{i+1,j} = \alpha|i-j+1|$, which is even. The largest deviation $\delta$ from an even delay occurs when $i - j + 1 = 2$

$$\delta = \sqrt{4\alpha^2+1} - 2\alpha. \tag{74}$$

By applying the modulo 2 operation on all entries of $\bar{\mathbf{D}}$, we get $\bar{\mathbf{D}}^{(2)}$. Comparing the entries in $\bar{\mathbf{D}}^{(2)}$ with those from the prototype odd–even distance matrix $\mathbf{D}^{(2)}$ from (65), we have $\rho^- = 0$ and $\rho^+ = \delta$. We can thus apply a $\rho$-schedule to get throughput

$$S = (1-\delta)\frac{N}{2} = \frac{N}{2}(1 - \sqrt{4\alpha^2+1} + 2\alpha). \tag{75}$$

Even for $\alpha = 1$, the throughput $S = N(3-\sqrt{5})/2$, which is 76% of the $N/2$ upper bound.

Since this 2-D $N$-node network consists of $N/2$ pairs of nodes with distance $2\alpha$ between each pair, the size $G$ of the network is given by

$$G = \sqrt{1+\gamma^2} \tag{76}$$

where $\gamma = 2\alpha((N/2)-1) = \alpha(N-2)$.

As $\alpha$ becomes large, this network can achieve a throughput arbitrarily close to the $N/2$ upper bound. However, the size $G$ also increases without bound with $\alpha$. It is then natural to ask what throughput can be achieved in networks of bounded size. Recall that the size $G$ of a network is defined in terms of the slot length $\tau$. For any finite physical space, an arbitrarily high throughput can be achieved by letting the slot length $\tau \to 0$. Typically, the minimum message duration that can be supported by a real system sets a lower limit on slot length and an upper limit on network size.

We have already seen bounded network geometries with better than unity throughputs. Consider the four-node regular tetrahedron network from Section V-E. The network has a size $G = 1$ and a perfect per-node fair schedule shown in (42) with throughput $S = 2$. The three-node equilateral triangle network from Section V-B has a size $G = 1$ and a perfect per-link fair schedule shown in (45) with throughput $S = 3/2$. The 2-D $N$-node network described above has a size $G$ given by (76). Using the odd–even per-node fair $\rho$-schedule with this network, we get a throughput $S$ given by (75). Writing $S$ in terms of $G$, we get

$$S = \frac{N}{2}\left(1 - \sqrt{4\frac{G^2-1}{(N-2)^2}+1} + 2\frac{\sqrt{G^2-1}}{N-2}\right). \tag{77}$$

Equation (77) directly relates the throughput $S$ to the number of nodes $N$ and the size of the network $G$. For a given number of nodes, increasing size will allow throughput to be increased toward the $N/2$ bound.

## VIII. Conclusion and Future Work

In this paper, we find, rather surprisingly, that large propagation delays in underwater networks, rather than being harmful, lead to significant performance gains as compared to wireless networks with negligible propagation delays. This result relies on designing transmission schedules with two properties. The first is that interfering packets overlap in time at unintended nodes and desired packets are interference free at the intended node. The intuition behind *interference alignment by delay* is that steering the interference to overlap at unintended nodes leaves other times slots interference free and suitable for successful receptions. The second is to utilize the interference laden time slots for transmission. We showed that the application of these ideas results in schedules with throughputs better than that of schedules for a zero-delay network. Specifically, we showed that the upper bound on the throughput of a large propagation delay network of $N$ nodes is $N/2$, meaning that the potential gains grow without bound. We systematically explored the value of propagation delay by constructing network topologies which admit schedules that achieve the $N/2$ upper bound exactly, asymptotically, or approximately. By utilizing the intuition above, we presented a computationally efficient algorithm that generates high-throughput schedules given any arbitrary network geometry.

Our research is but a step in the direction of understanding and exploiting large propagation delays in underwater communication networks. To isolate and understand the impact of large propagation delays, we have made several assumptions. The assumption of a single collision domain network (i.e., fully connected network) is valid in small underwater acoustic networks, but it may not hold in all scenarios, such as partially connected networks. It is worth noting that the single collision domain results are a lower bound to the more general arbitrarily connected network scenario, whose understanding would help in extending the findings to multihop networks. For example, our scheduling algorithm can easily be adapted for use in multihop networks by limiting the interference constraint to a small number of slots. Furthermore, power control can easily be incorporated into underwater protocols to limit interference range and minimize energy consumption. Other assumptions include a time-division approach to scheduling, specifically constructed network geometries and constrained traffic patterns in which only certain source–destination pairs (i.e., the throughput maximizing ones) are allowed. Relaxing these assumptions leads to exciting new problems. How can the intuition gained from the current paper be translated into a random access scenario, as compared to time-division scheduling? Finally, what are the gains of large propagation delays in networks with given geometries, predefined traffic patterns, and stochastic packet arrival processes?

## References


[1] T. Rappaport, *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ: Prentice-Hall, 2001, pp. 400–401.

[2] A. Leon-Garcia and I. Widjaja, *Communication Networks: Fundamental Concepts and Key Architectures*. New York: McGraw-Hill, 2004, pp. 297–299.

[3] C. Barakat, E. Altman, W. Dabbous, and S. Inria, "On TCP performance in a heterogeneous network: A survey," *IEEE Commun. Mag.*, vol. 38, no. 1, pp. 40–46, Jan. 2000.

[4] C. Fullmer and J. Garcia-Luna-Aceves, "Floor acquisition multiple access (FAMA) for packet-radio networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 25, no. 4, pp. 262–273, 1995.

[5] P. Viswanath, D. Tse, and R. Laroia, "Opportunistic beamforming using dumb antennas," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1277–1294, Jun. 2002.

[6] B. Peleato and M. Stojanovic, "Distance aware collision avoidance protocol for ad-hoc underwater acoustic sensor networks," *IEEE Commun. Lett.*, vol. 11, no. 12, pp. 1025–1027, Dec. 2007.

[7] K. Kredo, P. Djukic, and P. Mohapatra, "STUMP: Exploiting position diversity in the staggered TDMA underwater MAC protocol," in *Proc. IEEE INFOCOM* , Apr. 2009, pp. 2961–2965.

[8] X. Guo, M. Frater, and M. Ryan, "Design of a propagation-delay-tolerant MAC protocol for underwater acoustic sensor networks," *IEEE J. Ocean. Eng.*, vol. 34, no. 2, pp. 170–180, Apr. 2009.

[9] H.-H. Ng, W.-S. Soh, and M. Motani, "BiC-MAC: Bidirectional-concurrent MAC protocol with packet bursting for underwater acoustic networks," in *Proc. OCEANS Conf.*, Sep. 2010, DOI:10.1109/OCEANS.2010.5664567.

[10] B. Hou, O. Hinton, A. Adams, and B. Sharif, "An time-domain-oriented multiple access protocol for underwater acoustic network communications," in *Proc. MTS/IEEE OCEANS Conf., Riding the Crest Into the 21st Century*, 1999, vol. 2, pp. 585–589.

[11] V. Cadambe and S. Jafar, "Interference alignment and degrees of freedom of the $k$-user interference channel," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3425–3441, Aug. 2008.

[12] V. Cadambe and S. Jafar, "Degrees of freedom of wireless networks-what a difference delay makes," in *Conf. Record 41st IEEE Asilomar Conf. Signals Syst. Comput.*, 2007, pp. 133–137.

[13] R. Mathar and G. Bocherer, "On spatial patterns of transmitter-receiver pairs that allow for interference alignment by delay," in *Proc. IEEE 3rd Int. Conf. Signal Process. Commun. Syst.*, 2009, DOI: 10.1109/ICSPCS.2009.5306396.
[14] F. Blasco, F. Rossetto, and G. Bauch, "Time interference alignment via delay offset for long delay networks," 2011 [Online]. Available: http://arxiv.org/abs/1111.3204
[15] L. Badia, M. Mastrogiovanni, C. Petrioli, S. Stefanakos, and M. Zorzi, "An optimization framework for joint sensor deployment, link scheduling and routing in underwater sensor networks," in *Proc. 1st ACM Int. Workshop Underwater Netw.*, 2006, pp. 56–63.
[16] A. A. Syed, W. Ye, J. Heidemann, and B. Krishnamachari, "Understanding spatio-temporal uncertainty in medium access with ALOHA protocols," in *Proc. 2nd Workshop Underwater Netw.*, 2007, pp. 41–48.
[17] R. Mathar and M. Zivkovic, "How to position $n$ transmitter-receiver pairs in $n-1$ dimensions such that each can use half of the channel with zero interference from the others," in *Proc. IEEE Global Telecommun. Conf.*, 2009, DOI: 10.1109/GLOCOM.2009.5425222.
[18] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.
[19] J. Preisig, "Acoustic propagation considerations for underwater acoustic communications network development," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 11, no. 4, pp. 2–10, 2007.
[20] T. B. Koay, P. J. Seeking, M. Chitre, S. P. Tan, and M. Hoffmann-Kuhnt, "Advanced PANDA for high speed autonomous ambient noise data collection and boat tracking—System and results," in *Proc. Asia Pacific OCEANS Conf.*, May 2006, DOI: 10.1109/OCEANSAP.2006.4393919.
[21] A. Syed, W. Ye, and J. Heidemann, "Comparison and evaluation of the T-Lohi MAC for underwater acoustic sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 9, pp. 1731–1743, Dec. 2008.
[22] Y. Han, J. Deng, and Z. Haas, "Analyzing multi-channel medium access control schemes with aloha reservation," *IEEE Trans. Wireless Commun.*, vol. 5, no. 8, pp. 2143–2152, Aug. 2006.
[23] J. Dattorro, "Euclidean distance matrix," in *Convex Optimization & Euclidean Distance Geometry*. Palo Alto, CA: Meboo, 2005, ch. 4, pp. 219–314.
[24] W. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. New York: Wiley-Interscience, 2007.
[25] M. Chitre, M. Motani, and S. Shahabudeen, "A scheduling algorithm for wireless networks with large propagation delays," in *Proc. IEEE OCEANS Conf.*, Sydney, Australia, May 2010, DOI:10.1109/OCEANSSYD.2010.5603623.

**Mandar Chitre** (M'03–SM'11) received B.Eng. (honors) and M.Eng. degrees in electrical engineering from the National University of Singapore (NUS), Singapore, in 1997 and 2000, respectively, the M.Sc. degree in bioinformatics from the Nanyang Technological University (NTU), Singapore, in 2004, and the Ph.D. degree from NUS in 2006.

From 1997 to 1998, he worked with the Acoustic Research Laboratory (ARL), NUS, as a Research Engineer. From 1998 to 2002, he headed the technology division of a regional telecommunications solutions company. In 2003, he rejoined ARL, initially as the Deputy Head (Research) and is now the Head of the laboratory. He also holds a joint appointment with the Department of Electrical and Computer Engineering at NUS as an Assistant Professor. His current research interests are underwater communications, autonomous underwater vehicles, and underwater signal processing.

Dr. Chitre has served on the technical program committees of the following conferences: IEEE OCEANS Conference, International Conference on Underwater Networks and Systems (WUWNet), Defence Technology Asia International Conference & Exhibition (DTA), and Waterside Security International Conference (WSS), and has served as reviewer for many international journals. He was the Chairman of the Student Poster Committee for the 2006 IEEE OCEANS Conference in Singapore. In the past years, he has served as the Vice Chairman, Secretary, and Treasurer for the IEEE Oceanic Engineering Society (Singapore chapter) and is currently the IEEE Technology Committee Co-Chair of Underwater Communication, Navigation, and Positioning. He also serves as a Technical Co-Chair for the 2012 IEEE International Conference on Communication Systems.

**Mehul Motani** (S'92–M'00) received the B.E. degree from Cooper Union, New York, NY, in 1992, the M.S. degree from Syracuse University, Syracuse, NY, in 1995, and the Ph.D. degree from Cornell University, Ithaca, NY, in 2000, all in electrical and computer engineering.

He is currently an Associate Professor in the Electrical and Computer Engineering Department, National University of Singapore (NUS), Singapore. He has held a Visiting Fellow appointment at Princeton University, Princeton, NJ. Previously, he was a Research Scientist at the Institute for Infocomm Research in Singapore for three years and a Systems Engineer at Lockheed Martin, Syracuse, NY, for over four years. His research interests are in the area of wireless networks. Recently, he has been working on research problems which sit at the boundary of information theory, networking, and communications, with applications to mobile computing, underwater communications, sustainable development, and societal networks.

Dr. Motani has received the Intel Foundation Fellowship for his Ph.D. research, the NUS Faculty of Engineering Innovative Teaching Award, and placement on the NUS Faculty of Engineering Teaching Honours List. He has served on the organizing committees of IEEE International Symposium on Information Theory (ISIT), IEEE Wireless Network Coding Conference (WiNC), and IEEE International Conference on Communication Systems (ICCS), and the technical program committees of ACM International Conference on Mobile Computing and Networking (MobiCom), IEEE International Conference on Computer Communications (Infocom), IEEE International Conference on Network Protocols (ICNP), IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), and several other conferences. He participates actively in the IEEE and the Association for Computing Machinery (ACM) and has served as the Secretary of the IEEE Information Theory Society Board of Governors. He is currently an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION THEORY and an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS.

**Shiraz Shahabudeen** (S'05–M'06) received the B.Eng. degree in electrical engineering from the National University of Singapore (NUS), Singapore, in 1998 and the M.S. degree in telecommunication engineering from Melbourne University, Melbourne, Vic., Australia, in 2003.

He worked in the telecommunication software industry from 1998 to 2002 and at the Infocomm Development Authority of Singapore (IDA) from 2003 to 2004 as a Wireless Technology Specialist. He worked with the Acoustic Research Laboratory (ARL), NUS, from 2004 until 2010. His current research interests are underwater acoustic communications, networking, and autonomous underwater vehicles.