

Energy Efficient Multi-Path Communication for Time-Critical Applications in Underwater Sensor Networks

Zhong Zhou and Jun-Hong Cui

UCONN CSE Technical Report: UbiNet-TR08-01

Last Update: March 2008

Abstract

Due to the long propagation delay and high error rate of acoustic channels, it is very challenging to provide reliable data transfer for time-critical applications in an energy-efficient way. On the one hand, traditional retransmission-upon-failure usually introduces very large end-end delay, thus is not proper for time-critical services. On the other hand, common approaches without retransmission consume lots of energy. In this paper, we propose a new multi-path power-control transmission (MPT) scheme, which can guarantee certain end-to-end packet error rate while achieving a good balance between the overall energy efficiency and the end-to-end packet delay. MPT smartly combines power control with multi-path routing and packet combining at the destination. With carefully designed power control strategies, MPT consumes much less energy than the conventional one-path transmission scheme without retransmission. Besides, since no hop-by-hop retransmission is allowed, MPT introduces much shorter delay than the traditional one-path scheme with retransmission. We conduct extensive simulations to evaluate the performance of MPT. Our results show that MPT is highly energy efficient with low end-to-end packet delays.

I. INTRODUCTION

As an emerging area, underwater sensor network has attracted rapidly growing interests in last several years [1, 7, 14, 25]. On the one hand, underwater sensor networks enable a wide range of aquatic applications, such as oceanographic data collection, pollution monitoring, offshore exploration, disaster prevention, and tactical surveillance applications. On the other hand, the adverse underwater environments pose grand challenges for efficient communication and networking.

In underwater environments, radio does not work well due to its quick absorption in water, thus acoustic channels are usually employed. The propagation speed of acoustic signals in water is about 1.5×10^3 m/s, which is five orders of magnitude lower than the radio propagation speed (3×10^8 m/s). Moreover, underwater acoustic channels are affected by many factors such as path loss, noise, multi-path fading, and Doppler spread. All these cause high error probability in acoustic channels. In short, underwater acoustic channels feature long propagation delay and high error probability. In such harsh network scenarios, it is very challenging to provide energy efficient reliable data transfer for time-critical applications (such as pollution monitoring and submarine detection). First, conventional retransmission-upon-failure approaches are hard to satisfy the delay requirements. To give a simple example, if two node are separately by 500 meters, the propagation delay between them will roughly be $\frac{500}{1500} = \frac{1}{3}$ second. Even one time retransmission-upon-

failure will additionally introduce a delay of at least $\frac{1}{3} \times 2 = \frac{2}{3}$ second¹, which is quite large for some time-critical applications. Thus, to meet certain delay requirements, less or no retransmissions are preferred.

On the other hand, however, with less or no retransmission, we usually need to increase the transmitting power of every node to reduce end-to-end packet error rate in order to meet a certain communication reliability, as often leads to more energy consumption, thus degrading the energy efficiency of the network. As its terrestrial counterpart, underwater sensor network is energy-constrained since underwater nodes are typically powered by batteries, for which replacement or recharging is very difficult if not impossible [1, 7]. Therefore, minimizing the overall energy consumption becomes one of the most important design considerations for such networks. In summary, a new transmission scheme with low delay and high energy efficiency is desired for time-critical applications in underwater sensor networks.

In this paper, we propose a new scheme, called **Multi-path Power-control Transmission (MPT)**, for time-critical applications in underwater sensor networks. MPT is a cross-layer approach. It combines power control with multi-path routing and packet combining at the destination. Distributed power control strategies at the physical layer are used to improve the overall energy efficiency. In MPT, there is no hop-by-hop retransmission, as contributes to low end-to-end delays.

In an underwater sensor network, with high probability, multi-path routing protocols can find multiple paths between any two nodes because of the relatively high node density (this assumption holds even stronger in the multiple-sink underwater network architecture [7, 31], which will be further discussed in Section III). Different paths will experience independent fading if they are node-disjoint. MPT smartly utilizes this property to provide “multi-path macro-diversity”. Specifically, in MPT, the source node transmits the same packet along multiple paths to the same destination. And the transmission power at each intermediate nodes along each path is controlled by the source nodes based on the path characteristics. Multiple copies of the packet (some of these copies may be corrupted during transmission) will arrive at the destination along different paths, and the destination then recovers the original packet by combining the received copies. Packet combining techniques such as those in [33, 10, 15] can be used here.

Multi-path schemes are commonly believed to be beneficial to load balance and network robustness [36, 13, 8], but they are usually not considered energy efficient since more nodes will be involved in a multi-path scheme than in a one-path scheme. In this paper, contrary to the common intuition, we show that in underwater fading environments, for time-critical applications, if multi-path schemes are properly combined with power control at the physical layer and packet combining at the destination, significant energy savings can be achieved while with low end-to-end delays.

Our contributions in this work are three-fold. First, we propose a novel and effective transmission scheme, MPT, for time-critical applications in underwater sensor networks. It can improve the overall energy efficiency across the whole

¹Here we assume that once a node receives an incorrect packet, it will send an NAK back to the sender to ask for retransmission. Besides the retransmission of the packet, we also need to consider the propagation time for the NAK message.

network with low end-to-end delay and high reliability. Second, for MPT, we formulate an optimization problem to optimize the energy distribution across the whole network. We propose an efficient iterative approximation algorithm to solve this complex problem. Third, contrary to the common belief, our simulation results show that if properly used, multi-path can actually reduce the total energy consumption in underwater fading environments with low end-to-end packet delays.

The rest of this paper is organized as follows. In Section II, we briefly review some related work. In Section III, we describe the network model. Then in Section IV, we present MPT in detail. After that, we formulate the energy optimization problem and describe our iterative approximation algorithm in Section V. Finally, we present simulation results in Section VI followed by our conclusions in Section VII.

II. RELATED WORK

In this section, we first summarize some recent work on underwater sensor networks. Then we briefly review some typical approaches for energy efficiency in wireless sensor networks. After that, we discuss some relevant work on multi-path routing as well as packet combining.

As a new research area, underwater sensor network has received significant research interests for the last several years. Almost every layer of the protocol stack has been tackled: medium access control (MAC) ([29, 26]); multi-hop routing ([27, 39]); localization ([6, 11]), to name a few. Different from previous work, our work in this paper proposes an energy efficient cross-layer approach for time-critical applications in underwater sensor networks.

For wireless sensor networks, generally two ways are employed to improve the overall energy efficiency. One is to devise sleeping schemes for sensor nodes [40, 4, 9]. In these schemes, sensor nodes strategically change between sleeping mode and active mode. Since nodes in sleep mode consume much less energy than in active mode, these schemes can save energy by keeping nodes in sleeping mode as long as possible. The other way is to apply power control at the physical layer to reduce the overall energy consumption for one communication event [21, 17, 38, 20]. In this type of schemes, sensor nodes can dynamically adjust their power levels during the communication process according to channel status and network conditions. Essentially, our MPT scheme belongs to the second category, but it can be combined with any sleeping scheme from the first category.

Multi-path routing has drawn extensive attention in wireless sensor networks. The dense node deployment in wireless sensor networks makes multi-path routing a natural and promising technique to cope with the unreliable network environments. Research efforts have been made using multi-path routing to improve the robustness of data delivery [36, 41], to balance traffic load and power consumption among nodes [13, 37], to reduce end-to-end delays and the frequency of route discoveries [8, 24], and to improve the network security [22, 18], etc. The focus of our work in this paper is not to propose a new multi-path routing protocol. Instead, our research leverages existing multi-path routing protocols to make end-to-end transmission more energy efficient.

The basic idea of packet combining is to combine multiple corrupted copies of one packet to recover the original one. In [33], the authors propose a way to merge two or more non-coded packets to correct errors. It is shown that with packet combining at the receiver side, the original packet can be correctly recovered even if every individual received copy of this packet is corrupted. The authors of [10] extend the packet combining scheme into multi-hop scenarios and investigate its performance in wireless sensor networks. Through experiments, they show that the packet combining scheme can achieve promising results even in multi-hop wireless networks. In [15], the authors propose a multi-path packet combining scheme for wireless multi-hop networks. Based on an analysis on the delay characteristics of multi-path transmission, they show that the optimal number of paths exists that minimizes the average end-to-end packet error rate under certain delay constraints. In our work, we also propose a simple packet combining strategy for our MPT scheme. But our main contribution does not lie in packet combining and any other packet combining techniques can be easily incorporated into our scheme.

III. NETWORK MODEL

In this paper, we consider the following *multi-sink* underwater sensor network model: underwater sensor nodes with acoustic modems are densely distributed in a 3-D aqueous space, and multiple gateway nodes with both acoustic and RF modems are strategically deployed at the water surface. Each underwater sensor node can monitor and detect environmental events locally. As shown in Fig. 1, when an underwater sensor node has data to report, it first transfers the data toward one or multiple surface gateway nodes (each is also referred to as a sink) through acoustic links. Then these surface gateway nodes relay the received data to the control center through radio links. Compared with the acoustic links in water, surface radios link are much more reliable, faster and more energy efficient. Considering that radio signal propagation is orders of magnitude faster than acoustic signal propagation, it is safe to assume that surface gateways can send packets to the control center in negligible time and with relatively small energy consumption (acoustic communications consume much more energy than radio communications [7]). In this way, all the surface gateways (or sinks) form a *virtual sink*.

This multi-sink network architecture is helpful in traffic balance and multiple-path finding, as has been studied and analyzed in [31, 42, 16]. For our MPT scheme, this multi-sink architecture can effectively help to find more paths to the (virtual) sink (since any surface gateway is counted as a sink) and can greatly reduce the packet collision probability in the MAC layer. Later, in our simulation part, we will show the impact of the surface gateways on the network performance.

IV. MPT DESCRIPTION

MPT can be divided into the following three parts: *multi-path routing*, *source initiated power-control transmission*, and *destination packet combining*. First, the source node (any underwater sensor node in our network model can be

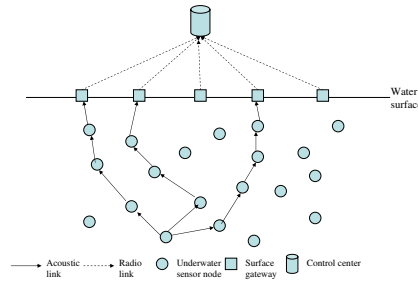


Fig. 1. Network model

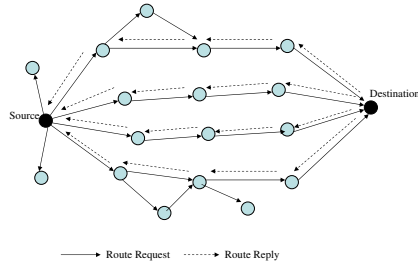


Fig. 2. Basic procedure of multi-path routing

a source node) initiates a multi-path routing process to find paths from the source to the destination (in our network model, the control center can be the destination). Through this route-finding process, the source will get to know some network parameters such as path length and the number of available paths. Based on this knowledge, the source node selects some paths and calculates the optimal transmitting power for each node along the selected paths. Then, it sends the same packet along the selected paths. Intermediate nodes on these selected paths will relay the packet with specified transmitting power parameters (carried in the packet header). When the destination receives all copies of the packet (some copies may get corrupted), it performs packet combining to recover the original packet.

In the following, we describe each part of MPT in detail.

A. Multi-path Routing

We assume that some multi-path routing protocols such as those in [23, 19] are available. The basic procedure of multi-path routing is illustrated in Fig. 2. When the source node has some packets to send, it will flood a “Route Request” message to the destination. Any intermediate nodes who receive this “Route Request” for the first time will forward it. When the destination receives “Route Request” messages, it will reply with “Route Reply” messages reversely along the paths of the corresponding “Route Request” messages. The destination can also make path selection. For example, it can select node-disjoint paths and send “Route Reply” back on them. After the source node receives the “Route Reply” messages, the routes between the source and the destination are established.

From the received “Route Reply” messages, the source node gets to know some path characteristics, such as the

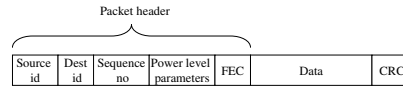


Fig. 3. Packet format

number of available paths, m , and the hop lengths of the paths. Based on this information, the source node will determine the optimal number of paths, m^* , and select m^* paths from m available paths. It also needs to calculate the optimal power level that every intermediate node on these paths should use for packet transmission. In Section V, we will show how the source node makes these decisions.

B. Source Initiated Power-control Transmission

In this phase of MPT, the same packets sent from the source node are transmitted by the intermediate nodes along all the selected paths using the specified transmission power. The packet format is shown in Fig. 3. Every packet should include the source identification (Source ID), the destination identification (Destination ID), as well as the packet sequence number (Sequence No.) in the packet header. The source node should also include power parameters in the packet header. This parameter is to specify the required power level that every intermediate node should use to relay the packet. In addition, we assume some coding schemes with strong error correction capability, such as forward error coding (FEC), are used in the header of every packet. In this way, the header part of every packet can be decoded correctly with high probability. Since the packet header is usually much smaller than the data part, the overhead incurred in the header error correction process is almost negligible. For the large data part, we do not use any error correction coding schemes because of their inefficiency in fading environments [10]. But some error detection coding schemes, such as cyclical redundant checking (CRC), are still used in the data part to check data errors.

When an intermediate node receives a packet, it will decode and check the header part. If the header is correct or can be recovered by the adopted FEC scheme, this node will relay the whole packet to its next hop with the specified power level without further checking the data part. Otherwise, it will simply drop the packet.

C. Destination Packet Combining

At the destination, after it receives one copy of the original packet from one path (the data part of this copy may be corrupted during the transmission process), it will first check whether this copy is correct or not. If there is no error with this copy, it means that the destination successfully receives the original packet. Otherwise, the destination will keep this corrupted copy in its buffer. After receiving multiple corrupted copies of the original packet, the destination will combine these copies to recover the original one.

In MPT, we use a simple packet combining technique, which is illustrated in Fig. 4. Assuming the destination

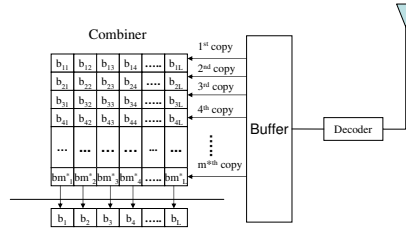


Fig. 4. Packet combining at the destination

receives m^* copies of the same packet from m^* paths, for the i -th bit in this packet, it determines its output b_i as

$$b_i = \begin{cases} 1 & \sum_{k=1}^{m^*} b_{ik} > \frac{m^*}{2}, \\ 0 & \sum_{k=1}^{m^*} b_{ik} < \frac{m^*}{2}, \end{cases} \quad (1)$$

where b_{ik} denotes the k th copy of the i th bit. To simply put, when the bits in the majority of the copies are “1”, then the corresponding bit of the original packet is decoded as “1”; otherwise, it is decoded as “0”. We choose this “majority voting” method mainly because of its simplicity. It should be noted that [33, 10] propose more complicated packet combining schemes, and these schemes can further reduce packet error rate. However, these schemes need to search through all detectable error patterns in order to recover the original packet, as will introduce significant processing delay and increase the complexity of the destination node, especially when the data packet is large. On the other hand, our simple “majority voting” scheme has a constant processing delay and is simple to implement. Our simulation results in Section VI show that even this simple packet combining technology can still achieve significant performance improvement.

V. OPTIMIZING ENERGY DISTRIBUTION

In this section, we will first describe the channel model and formulate the energy distribution optimization problem. We will then show how the source node optimally distributes energy along one path and multiple paths. Finally, we summarize the overall energy distribution process.

A. Channel Model

We adopt the acoustic channel model proposed in [35]. For an underwater acoustic channel, its average path loss is given by

$$A(d, f) = d^a \beta(f)^d, \quad (2)$$

where d is the distance of the acoustic channel; a is the spreading factor ($1 \leq a \leq 2$); and $\beta(f)$ is the absorption coefficient which is determined by the frequency of the acoustic channel f . In this paper, we assume that all nodes work in the same frequency band. Thus, $\beta(f)$ will be the same for all nodes.

As in [34, 5], we assume Rayleigh fading for underwater acoustic channels in our analysis. Further more, we use BPSK (Binary Phase Shift Keying) modulation, which is widely used in the state-of-the-art acoustic modems [12]. Thus in our system, one symbol carries information of one bit. The energy consumption for one symbol actually equals to the energy consumption for one bit. Our analysis can be easily extended to other modulations.

B. Problem Formulation

In MPT, given m available paths, n_i hops on path i ($1 \leq i \leq m$), and the required end-to-end packet error rate (PER), P_{req} , the source node needs to distribute transmission energy for each hop along the m path in order to minimize the overall energy consumption for one packet transmission. This problem can be formulated as follows:

$$\begin{aligned}
 \min \quad & \sum_{i \in (1, 2, \dots, m)} \sum_{j \in (1, 2, \dots, n_i)} E_{ij} L \\
 \text{s.t.} \quad & P_e \leq P_{req}, \\
 & 0 \leq E_{ij} \leq E_{\max}, \tag{3}
 \end{aligned}$$

where L is the packet length in bits; E_{ij} is the average transmitting energy per bit of the j th hop on the i th path; P_e is the average end-to-end packet error rate (PER); and E_{\max} is the maximal transmitting energy per bit of every node, which is stipulated by the system hardware constraints.

In Eq. (3), the first constraint specifies that the average end-to-end PER, P_e , should be smaller than the system requirement P_{req} . The second constraint is to guarantee the transmitting energy per bit of every node will not exceed its maximal allowable transmitting energy per bit E_{\max} . In this paper, we ignore the energy consumption for data receiving and processing. This is because in underwater acoustic communication, data receiving and processing consumed much less energy than data transmitting [12].

The above optimization problem is hard to solve because of the following reasons: 1) with the import of the packet combining technique, the expression for the end-to-end PER, P_e , is quite complicated and not convex; 2) In (3), there are $\sum_{i \in (1 \dots m)} n_i$ variables involved, as will increase the computation complexity of the source node.

In order to solve this complicated optimization problem, we divide the solving process into two steps. In the first step, we do optimal energy distribution among all available paths. This optimal distribution needs to guarantee that the average end-to-end PER requirement can be satisfied. In the second step, we do optimal energy distribution among all nodes along each selected path.

Next, we will first derive the optimal energy distribution for one path and get the relationship between the average energy consumption per bit for one path and its average end-to-end bit error rate (BER). Then, based on the results from one path, we will come to the energy optimization among multiple paths with certain end-to-end PER requirement.

C. Optimal Energy Distribution along One Path

Considering path i , its j th hop acoustic link can be treated a Rayleigh fading channel with additive white Gaussian noise (AWGN) on top of an average path loss which is proportional to $d_{ij}^a \beta(f)^{d_{ij}}$, where d_{ij} is the distance of the j th hop on path i . Then the average received signal noise ratio (SNR), $\bar{\gamma}_{ij}$, can be written as

$$\bar{\gamma}_{ij} = G \frac{E_{ij}}{N_0 d_{ij}^a \beta(f)^{d_{ij}}}, \quad (4)$$

where N_0 is the one-sided AWGN spectral density at the receiver, and G is a constant that is defined by the signal frequency, antenna gains, and other parameters. As such, the instantaneous received SNR, γ_{ij} , has the following distribution [32]:

$$f(\gamma_{ij}) = \frac{1}{\bar{\gamma}_{ij}} e^{-\frac{\gamma_{ij}}{\bar{\gamma}_{ij}}}. \quad (5)$$

Conditional on each instantaneous value of γ_{ij} , we have an AWGN channel. The approximate closed-form bit error rate (BER), $p_{ij}^b(\gamma_{ij})$, for uncoded BPSK signal in such an AWGN channel is provided as follows [28]:

$$p_{ij}^b(\gamma_{ij}) = \frac{1}{\sqrt{2\pi}} \int_{\sqrt{2\gamma_{ij}}}^{\infty} e^{-t^2/2} dt. \quad (6)$$

By averaging (6) over the Rayleigh fading channel, the average BER, P_{ij}^b , for one-hop transmission is obtained as [32]

$$P_{ij}^b = \int_0^{\infty} p_{ij}^b(\gamma_{ij}) f(\gamma_{ij}) d\gamma_{ij} \approx \frac{d_{ij}^a \beta(f)^{d_{ij}} N_0}{4GE_{ij}}. \quad (7)$$

Eq. (7) specifies the average BER P_{ij}^b for the j th hop on path i . Since error propagation will occur in a multi-hop network scenario, we use a Markov chain to analyze the BER for multi-hop networks. We assume each single hop transmission is communication over a binary symmetric channel. Then, the one-step transition matrix of the Markov chain \mathbf{K} is given by

$$\mathbf{K} = \begin{bmatrix} 1 - P_{ij}^b & P_{ij}^b \\ P_{ij}^b & 1 - P_{ij}^b \end{bmatrix}. \quad (8)$$

Due to the assumption of equally likely bits, the average end-to-end BER of path i from the source to the destination is given by

$$\begin{aligned} P_i^b &= \Pr(\hat{b} = 0 | b = 1) \\ &= (0, 1) \prod_{j \in (1, \dots, n_i)} \begin{pmatrix} 1 - P_{ij}^b & P_{ij}^b \\ P_{ij}^b & 1 - P_{ij}^b \end{pmatrix} (1, 0)^T \\ &= \frac{1}{2} [1 - \prod_{i \in (1, \dots, n_i)} (1 - 2P_{ij}^b)]. \end{aligned} \quad (9)$$

Thus, to minimize the average energy consumption for one packet transmission with average end-to-end BER, P_i^b , for path i , we need to solve the following optimization problem

$$\begin{aligned} \min \quad & \sum_j E_{ij}L \\ \text{s.t.} \quad & \frac{1}{2}[1 - \prod_{j \in (1..n_i)} (1 - 2P_{ij}^b)] \leq P_i^b \\ & P_{ij}^b = \frac{N_0 d_{ij}^a \beta(f)^{d_{ij}}}{4GE_{ij}}. \end{aligned} \quad (10)$$

We use Lagrange method to solve Eq. (10) and get the following theorem

Theorem 1: For path i , with average end-to-end BER P_i^b , the optimal energy distribution along this path should satisfy

$$\frac{E_{ij}}{E_i} = \frac{G_{ij}}{G_i}, \quad (11)$$

where E_{ij} is the average energy consumption per bit of the j th hop in path i ; E_i is the average energy consumption per bit along path i ; $G_{ij} = d_{ij}^{\frac{a}{2}} \beta(f)^{\frac{d_{ij}}{2}}$, and $G_i = \sum_{j \in (1..n_i)} G_{ij}$. In addition, E_i and P_i^b have the following relationship

$$P_i^b = \frac{1}{2}(1 - e^{-\frac{A_i}{E_i}}), \quad (12)$$

where

$$A_i = \frac{N_0}{2G} \sum_{j \in (1..n_i)} d_{ij}^{\frac{a}{2}} \beta(f)^{\frac{d_{ij}}{2}} \sum_{j \in (1..n_i)} d_{ij}^{\frac{a}{2}} \beta(f)^{\frac{d_{ij}}{2}}. \quad (13)$$

With Theorem 1, given P_i^b , the source node can calculate the optimal transmission energy for each hop along path i .

Proof: see Appendix VIII-A.

D. Optimal Energy Distribution over Multiple Paths

Given m available paths, the overall energy consumption per packet, E , can be written as

$$E = \sum_{i=(1..m)} E_i L, \quad (14)$$

where E_i is the energy consumption for one bit along path i and L is the packet length in bits. Before presenting the energy distribution optimization problem for multiple paths, we first derive the average end-to-end packet error rate (PER), P_e , as follows.

Recall that we employ packet combining at the destination node. We use U to denote the event that at least one copy of a packet from m paths arrives correctly, \bar{U} to denote that all individual copies are corrupted, and V to denote the event that our packet combining technique can successfully recover the original packet. Then the average end-to-end PER, P_e , can be written as

$$P_e = (1 - \Pr(V|\bar{U})) \times \Pr(\bar{U}). \quad (15)$$

We also know that

$$\Pr(\bar{U}) = \prod_{i=(1..m)} (1 - (1 - P_i^b)^L), \quad (16)$$

where P_i^b is the average end-to-end bit error rate (BER) of path i . Then we can get

$$\Pr(V|\bar{U}) = \left\{ \sum_{k=\lfloor \frac{m}{2} + 1 \rfloor}^m \underbrace{[(1 - P_1^b) \dots (1 - P_k^b) P_{(k+1)}^b \dots P_m^b + \dots]}_{\binom{m}{k}} \right\}^L. \quad (17)$$

Now we can formulate the optimization problem of minimizing the overall energy consumption per packet as follows:

$$\begin{aligned} \min \quad & E = \sum_{i=(1,2,\dots,m)} E_i L \\ \text{s.t.} \quad & \prod_{i=(1..m)} (1 - (1 - P_i^b)^L) \times (1 - \left\{ \sum_{k=\lfloor \frac{m}{2} + 1 \rfloor}^m \underbrace{[(1 - P_1^b) \dots (1 - P_k^b) P_{(k+1)}^b \dots P_m^b + \dots]}_{\binom{m}{k}} \right\}^L) \leq P_{req}, \\ & P_i^b = \frac{1}{2} (1 - e^{-\frac{A_i}{E_i}}) \quad i \in (1, 2 \dots m), \\ & 0 \leq E_i \leq \eta_i E_{\max} \quad i \in (1, 2 \dots m). \end{aligned} \quad (18)$$

In Eq. (18), the first constraint is to specify the system requirement on end-to-end PER. The second constraint describes the relationship between the average energy consumption per bit E_i and the average end-to-end BER P_i^b for each path. The third constraint is to guarantee the transmitting energy per bit of every node will not exceed its maximal allowable transmitting energy per bit. η_i is a constant. Based on Eq. (11), it is calculated as

$$\eta_i = \min_j \frac{G_i}{G_{ij}}. \quad (19)$$

Eq. (18) is hard to solve directly because its first constraint is too complicated and not convex. Next, we propose an iterative approximation algorithm to solve it.

D.1 Iterative Algorithm for Multi-path Energy Distribution

We have noted that the complexity of Eq. (18) mainly lies in its first constraint. Thus, we focus on simplifying it in order to solve the optimization problem efficiently.

First, we set

$$C = \left\{ \sum_{k=\lfloor \frac{m}{2} + 1 \rfloor}^m \underbrace{[(1 - P_1^b) \dots (1 - P_k^b) P_{(k+1)}^b \dots P_m^b + \dots]}_{\binom{m}{k}} \right\}^L. \quad (20)$$

Then, Eq. (18) can be written as

$$\begin{aligned} \min \quad E &= \sum_{i=(1,2,\dots,m)} E_i L \\ \text{s.t.} \quad \prod_{i=(1,2,\dots,m)} (1 - [1 - \frac{1}{2}(1 - e^{-\frac{A_i}{E_i}})]^L) &\leq \frac{P_{req}}{1 - C}, \\ 0 \leq E_i &\leq \eta_i E_{\max} \quad i \in (1, 2 \dots m), \end{aligned} \quad (21)$$

which can be converted into

$$\begin{aligned} \min \quad E &= \sum_{i=(1,2,\dots,m)} E_i L \\ \text{s.t.} \quad \sum_{i=(1,2,\dots,m)} \ln(1 - [\frac{1}{2}(1 + e^{-\frac{A_i}{E_i}})]^L) &\leq \ln(\frac{P_{req}}{1 - C}), \\ 0 \leq E_i &\leq \eta_i E_{\max} \quad i \in (1, 2 \dots m). \end{aligned} \quad (22)$$

For now, we assume that C is a constant. Later, in our algorithm, we will show how to update C iteratively.

We use Taylor series expansion to the left side of the first constraint in Eq. (22) and take the first two items as its approximation². Then, by using Jensen's inequality, Eq. (22) can be approximated as follows:

$$\begin{aligned} \min \quad E &= \sum_{i=(1,2,\dots,m)} E_i L \\ \text{s.t.} \quad - \left[\sum_{i=(1,2,\dots,m)} \frac{(1 + e^{-\frac{A_i}{E_i}})}{2(1 - x_i)^{\frac{1}{L}}} \right] &\leq -m^{\frac{L-1}{L}} \times \\ \left[\sum_{i=(1,2,\dots,m)} \left(\ln(1 - x_i) + \frac{x_i}{1 - x_i} \right) - \ln\left(\frac{P_{req}}{1 - C}\right) \right]^{\frac{1}{L}}, & \\ 0 \leq E_i &\leq \eta_i E_{\max} \quad i \in (1, 2 \dots m), \end{aligned} \quad (23)$$

where x_i ($i \in (1, 2, \dots, m)$) is the Taylor expansion point for every additive item $\ln(1 - [\frac{1}{2}(1 + e^{-\frac{A_i}{E_i}})]^L)$ of the first constraint in Eq. (22). We update it iteratively in our algorithm to refine our results. The objective function of Eq. (23)

²It should be noted that the approximation ratio here is closely related to the Taylor expansion point. In our algorithm, we iteratively update this expansion point to refine the final result

is linear and its first constraint can be easily proved to be monotonically decreasing and convex when $E_i \geq \frac{A_i}{2}$ (Proof can be seen in Appendix VIII-B).

To simplify the optimization problem, we set $E_i \geq \frac{A_i}{2}$. The rationale is as follows: when $E_i \leq \frac{A_i}{2}$, from Eq. (12), we know that the corresponding end-to-end BER of path i , $P_i^b \geq \frac{1}{2}(1 - e^{-2}) = 0.4323$, which is quite high for BER and is only slightly smaller than the worst case $P_i^b = 0.5$ when $E_i = 0$. Thus the probability that the optimal solution fall in $[0, \frac{A_i}{2}]$ is quite small. Therefore, setting $\frac{A_i}{2}$ as a lower constraint for E_i will not degrade the system performance much.

Then Eq. (23) can be converted into the following convex optimization problem and can be solved efficiently by the interior point method [2].

$$\begin{aligned}
\min \quad & E = \sum_{i=(1,2,\dots,m)} E_i L \\
s.t. \quad & - \left[\sum_{i=(1,2,\dots,m)} \frac{(1 + e^{-\frac{A_i}{E_i}})}{2(1 - x_i)^{\frac{1}{L}}} \right] \leq -m^{\frac{L-1}{L}} \times \\
& \left[\sum_{i=(1,2,\dots,m)} \left(\ln(1 - x_i) + \frac{x_i}{1 - x_i} \right) - \ln\left(\frac{P_{re}}{1 - C}\right) \right]^{\frac{1}{L}}. \\
& \frac{A_i}{2} \leq E_i \leq \eta_i E_{\max} \quad i \in (1, 2, \dots, m).
\end{aligned} \tag{24}$$

Now we show our complete iterative algorithm in Algorithm 1. Clearly, our iterative algorithm includes three iterations. In the outmost iteration, we change the number of the paths and calculate the optimal energy distribution accordingly. In the middle iteration, we update x_i s in order to refine our results. In the inner iteration, we update C iteratively. We have proved that *our iterative algorithm converges in finite time*.

First, we will introduce Corollary 1.

Corollary 1: For all available m paths, if the optimal solution chooses and transmits data among j paths from them, then, these selected j paths must be the j paths with the smallest A_j s.

Proof: We use contradiction to prove this corollary. First, we sort all paths according to their A_j s and set $A_1 \leq A_2 \leq A_3 \dots \leq A_m$.

We assume that for the optimal solution, its selected j paths are $\Phi_1 = (1, 2, \dots, k, \dots, j)$, And we assume that one of the selected j paths (let's say path k) is not among the j paths with the smallest A_i s. Then, we can definitely find another path $k_1 \notin \Phi_1$ which can satisfy $A_{k_1} < A_k$. We can replace path k with path k_1 and leave others unchanged. since $A_{k_1} < A_k$, for the same end-to-end BER requirements, from Theorem 1, we know that path k_1 will consume less energy per symbol than path k . Because we let other paths unchanged, for the same end-to-end BER requirements, $\Phi_2 = (1, 2, \dots, k_1, \dots, j)$ will definitely consume less energy than $\Phi_1 = (1, 2, \dots, k, \dots, j)$. And then, the selected j paths Φ_1 are not the optimal solution, which is in contradiction with our assumption. Then, we prove corollary 1.

Algorithm 1 *Iterative Algorithm*

- 1: Sort all m paths according to their A_i and have $A_1 \leq A_2 \leq A_3 \dots \leq A_m$.
 - 2: $k = 1$ //Initialize the counter of the paths
 - 3: **repeat**
 - 4: $x_i = 1 - P_{req}, i \in (1, 2, \dots, k)$ //Initialize the Taylor expansion points
 - 5: **repeat**
 - 6: $E_{current,k} = \sum_{i \in (1, 2, \dots, k)} n_i E_{\max} L$ // $E_{current,k}$ is used to store the minimal energy consumption per bit with k paths. And it is initialized to its maximal value
 - 7: $C_{current} = 0$ // C is initialized to 0
 - 8: **repeat**
 - 9: Solve Eq. (24) and get the corresponding optimal energy distribution $(E_1, E_2, E_3, \dots, E_k)$ and BER $(P_1^b, P_2^b, P_3^b, \dots, P_k^b)$ of the selected k paths
 - 10: $C_{prev} = C_{current}$
 - 11: With $(P_1^b, P_2^b, P_3^b, \dots, P_k^b)$, calculate $C_{current}$ according to Eq. (20)
 - 12: **until** $|C_{prev} - C_{current}| < \delta_1$ // δ_1 is a predefined threshold
 - 13: $E_{prev,k} = E_{current,k}$
 - 14: $E_{current,k} = \sum_{i \in (1, 2, \dots, k)} E_i L$
 - 15: $x_i = [\frac{1}{2}(1 + e^{-\frac{A_i}{E_i}})]^L, i \in (1, 2, \dots, k)$
 - 16: **until** $|E_{current,k} - E_{prev,k}| < \delta_2$ // δ_2 is a predefined threshold
 - 17: $k = k + 1$
 - 18: **until** $k > m$ //Finish all m paths
 - 19: Compare all $E_{current,k}, k \in (1, 2, \dots, m)$ and select the smallest one, which corresponds to the final solution
-

Theorem 2: For the optimization problem (24) with the same $x_i \quad i \in (1, 2, \dots, m)$, let C take value C_1 and C_2 , corresponding to C_1 and C_2 , the optimal solutions are $(E_{1,C_1}, E_{2,C_1}, \dots, E_{m,C_1})$ and $(E_{1,C_2}, E_{2,C_2}, \dots, E_{m,C_2})$.

If $C_1 > C_2$, Then

$$(E_{1,C_1}, E_{2,C_1}, \dots, E_{m,C_1}) \succeq (E_{1,C_2}, E_{2,C_2}, \dots, E_{m,C_2}),$$

While if $C_1 < C_2$, Then

$$(E_{1,C_1}, E_{2,C_1}, \dots, E_{m,C_1}) \preceq (E_{1,C_2}, E_{2,C_2}, \dots, E_{m,C_2})$$

where \succeq means component wise bigger or equal and \preceq means component wise smaller or equal.

Proof: See Appendix VIII-C.

Corollary 2: For every x_i s update, we will get at least the same good result in the next iteration as that in the

previous one.

Proof: In our algorithm, we set $x_i = [\frac{1}{2}(1 + e^{-\frac{A_i}{E_i}})]^L$, which equals to the average probability that the packet from path i with energy E_i per bit has been successfully received by the sink node. At the $i - 1$ th iteration, the optimal solution is $(E_{(1,x_{i-1})}, E_{(2,x_{i-1})}, \dots, E_{(j,x_{i-1})})$. And we calculate $x_i = [\frac{1}{2}(1 + e^{-\frac{A_i}{E_{(i,x_{i-1})}}})]^L$ for the next iteration. Based on these new values of x_i s, we get new formations of (23). Because $\ln(1 - x)$ is a monotonically decreasing concave function, it can be easily verified that the optimal solution of the last iteration $(E_{(1,x_{i-1})}, E_{(2,x_{i-1})}, \dots, E_{(j,x_{i-1})})$ must be a feasible solution (may not be the optimal one) of the current iteration. And thus, we can get that the optimal solution of the next iteration must be at least as good as that of the previous one.

Theorem 3: Our algorithm converges in finite time.

Proof: See Appendix VIII-D.

Our algorithm includes multiple convex optimization processes. Although the computational complexity of convex optimization is hard to judge, efficient interior-point algorithm can be used here. Further, it should be noted that every intermediate solution of our algorithm can satisfy the the system end-to-end PER requirement, though they are not optimal. Thus, in practice, we can stop our algorithm whenever the required energy efficiency is reached. In Section VI, we will explore the trade-off between energy efficiency and computational complexity in our algorithm

E. Overall Energy Distribution Process

Considering the optimization algorithm discussed in this section, now we summarize the overall energy distribution process as follows:

- 1) Through the source-initiated multi-path routing process, the source node gets to know all needed information such as the number of available paths, the number of hops for each path, and the per-hop distance.
- 2) At the source node, the iterative algorithm is performed for multi-path energy distribution. Then for every selected path, the source node sends out messages with two additional fields in the packet header. One field is to specify the optimal overall energy along this path E_i and the other is to specify G_i .
- 3) For each intermediate node j along path i , it has recorded its distance to its next hop in the path during the routing process. According to Eq. (11), it calculates its transmitting energy as follows:

$$E_{ij} = \frac{G_{ij}}{G_i} \times E_i. \quad (25)$$

Then it transmits packets with energy E_{ij} per bit.

- 4) At the destination, after it receives a copy of a packet, it first judges whether this copy is correct or not. If correct, this packet will be passed to the application layer without any delay. Otherwise, it will wait for other copies and do packet combining to recover the original packet.

VI. PERFORMANCE EVALUATION

In this section, we evaluate the performance of MPT through simulations.

A. Simulation Settings

Based on NS-2, we implement a simulation package for underwater sensor networks. Following the multiple-sink underwater sensor network model, the simulated network settings are as follows: 512 underwater sensor nodes are deployed in a 3-Dimensional space of $4000m \times 4000m \times 2000m$; and 36 surface gateways are deployed in a 2-Dimensional area of $4000m \times 4000m$ at the water surface. The transmission range of underwater sensor nodes is set to $600m$ and the data rate is set to be 10Kbps. We use the same broadcast MAC protocol as in [39]. In this MAC protocol, when a node has packets to send, it first senses the channel. If the channel is free, it broadcasts the packets. Otherwise, it backs off. Packets will be dropped if the node backs off 5 times. Since there is no collision resolution in this broadcast MAC protocol, the performance of our scheme might be degraded. But for underwater sensor networks with low data generation rate, the collision probability will be very small. Thus, in our simulations, we set the data generation rate as 1 packet every 10 seconds to minimize the effect of MAC protocols. Further, the packet size is set to be 200 bytes. As for the routing protocol, we modify AODV (Ad hoc On-Demand Distance Vector) [30] and make it support multiple path routing. Each simulation lasts for 10000s. Thus, each node generate about 1000 packets in each simulation. We run simulations for 100 times, and take the average as our final results.

For comparison, we implement two other schemes in the same underwater network settings. One scheme is one-path transmission with power control but without retransmission (referred to as *one-path without retransmission* for short), and the other scheme is one-path transmission with retransmission and power control (referred to as *one-path with retransmission* for short). In the one-path without retransmission scheme, first, through a routing process, the source node finds the most energy-efficient path and transmits its packets with power control to guarantee the end-to-end packet error rate. No retransmission is performed upon transmission failure. For the one-path with retransmission scheme, it works as follows: first, the source node finds the most energy-efficient path by its routing process and then transmits packets with power control along this path. Retransmission is allowed upon failure (i.e., if the sender does not receive an ACK for a packet from the receiver after time t_r (in our simulations, we set $t_r = 1s$), it will retransmit the packet). We set the maximal times of retransmission, n_r . After retransmitting a packet for n_r times, a node will simply drop this packet.

For the one-path without retransmission scheme, the optimal energy distribution problem has been actually solved in Section V-C. For the one-path with retransmission scheme, we have also formulated and solved the optimal transmitting energy distribution problem (See Appendix VIII-E for details).

For all schemes, we measure two metrics: *average energy consumption per packet* and *average end-to-end packet*

delay. For the first metric, due the huge energy consumption of the one-path without retransmission scheme, to better present the comparison results, we covert the original energy consumption per packet E , measured in micro joules (mj), into log scale: $10 * \log(E/1mj)$. The second metric is measured in seconds.

B. Results and Analysis

B.1 Impact of End-to-End PER

Fig. 5 shows the impact of end-to-end PER on various schemes. From this figure, we can observe that with the increase of end-to-end PER, the average energy consumption per packet will decrease sharply. Compared with one path without retransmission, MPT always consumes much less energy. While compared with one path with retransmissions, MPT has comparable energy efficiency when the end-to-end PER requirement is high. When the end-to-end PER requirement is relatively low (In our simulation, when the end-to-end PER is larger than 0.1), MPT has better energy efficiency over one path with retransmissions. The energy benefits of “MPT” comes from the following two aspects. First, it takes advantage of multiple paths to transmit the same packet simultaneously. So, when one path is in deep fading and fails its transmission, other paths perhaps can transmit the packet correctly. This actually can provide some “macro-diversity” benefits in the underwater fading environment. Second, with packet combining at the destination, even when all paths are in relatively bad conditions and all received copies of the packet are wrong, the destination still has some probability to recover the original packets. Thus, MPT is highly energy efficient.

As for the average delay per packet, our MPT scheme is a little higher(Here, about 2 seconds) than one path without retransmission. This is in most part because of the packet combining delay at the destination. While compared with one path with retransmissions, MPT has much lower end-to-end delay. For example, compared with one path with retransmission when $n_r = 3$, the average end-to-end delay of MPT is about 5 seconds smaller. This is because of the long propagation delay and the low bandwidth of UWSN, which makes the retransmission quite time-consuming. Fig.5 clearly shows us that our scheme can achieve high energy efficiency with small end-to-end delay under certain end-to-end PER requirements.

B.2 Impact of Node Density

Here, we gradually change the average node density of the network ³by changing the maximal transmitting range of every node from 450 to 750. And thus, the average node density change from 4 to 18. And the average end-to-end PER is set to be 0.05.

Fig.6(a) clearly shows us with the increase of node density, the overall energy consumption of MPT will decrease. This is because more paths to the sink node will be found with the increase of node density, which contributes to the

³the average node density is defined as the average number of one-hop neighbors of a node. And two nodes are neighbors if they can reach each other with their maximal transmitting power.

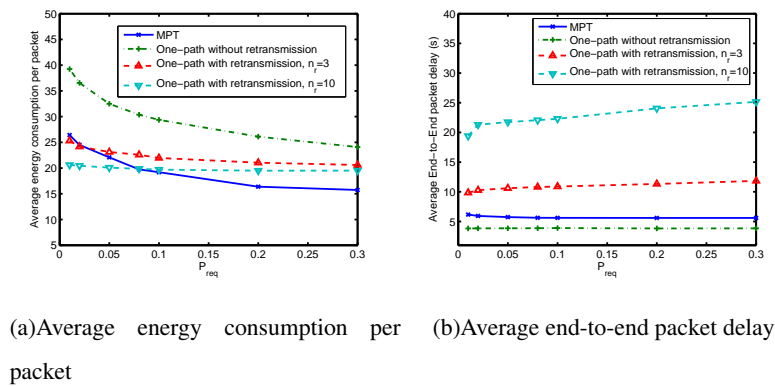


Fig. 5. Performance with varying end-to-end PER

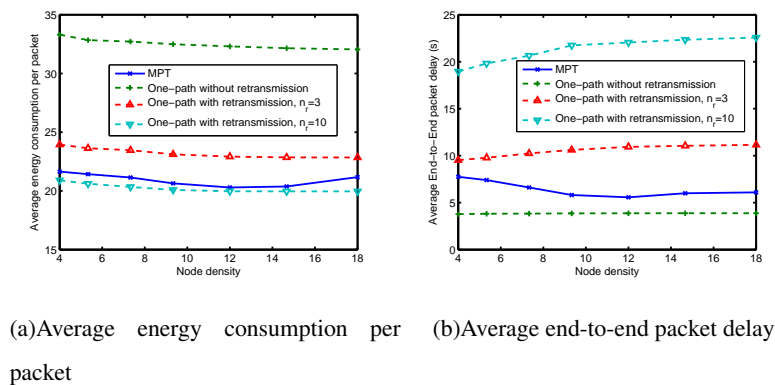


Fig. 6. Performance with varying node density

decrease of energy consumption. But the decrease will slow down in the high node density region. This is because in the high node density region, most usable energy efficient paths have already been found, continuing increase the maximal transmission range will not help much in energy consumption reduction. Besides, higher node density will introduce more collision in the network, which will degrade the network's energy efficiency in practice.

As for the average end-to-end delay, From Fig.6(b), we can see that in the low node density region(here, from 4 to 12), the average end-to-end delay of our MPT scheme will decrease with the increase of node density. This is because more path with shorter end-to-end delay will be found with the increase of node density. Correspondingly, the overall delay will decrease. But, in the high node density region, the average delay of our scheme will increase slightly, we believe this increase is brought about by the backoff in the MAC layer. But since our MPT is aimed for underwater sensor network with low traffic (Here, 1 packet 10 second), the contention in the MAC layer is not significant and thus, the increase in delay here is just a minor.

B.3 Impact of Surface Gateways

As discussed in Section III, one important network parameter of our MPT scheme is the number of surface buoys in the network. With more surface buoys, every sensor node has better chance to find more paths to the data sink, Thus,

better energy efficiency seems can be achieved. While on the other hand, with more surface buoys and more available paths, higher collision probability will be introduced in the networks and thus more energy and time will be wasted in the collision resolution process.

Here, we gradually change the number of surface buoys from 4 to 64. Fig. 7(a) clearly shows us that with the increase of surface buoys, the average overall energy consumption per packet will decrease. And the decreasing rate will slow down with the increase of surface buoy. This is because of the higher collision probability introduced by more surface buoys and more paths.

From Fig. 7(b), we can see that the end-to-end packet delay changes drastically with the number of surface buoys. This is because on one hand, with more surface buoys, every node has higher probability to find shorter path to the surface buoy, which will contributes to a shorter end-to-end packet delay. While on the other side, with more surface buoys and more available paths, higher collision probability will be introduced in the networks and thus more time will be wasted in the collision resolution. Fig. 7(b) shows us that in our simulation setting, when the surface buoys reach 36, the end-to-end delay reaches its minimum.

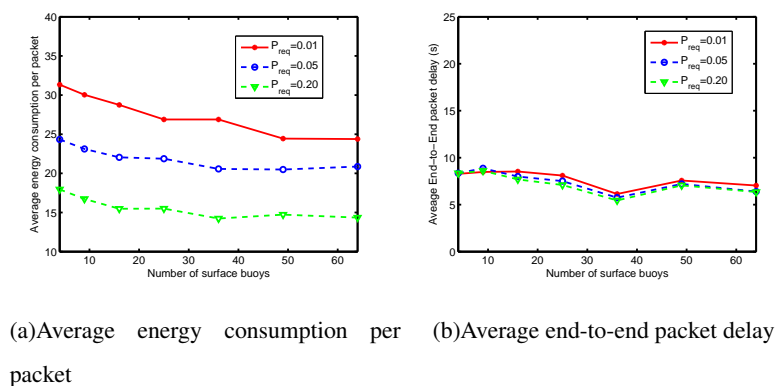


Fig. 7. Performance with varying number of surface gateways

B.4 Convergence Property

In this set of simulation, we investigate the convergence property of our proposed iterative algorithm. we gradually change δ_1 from 0.0005 to 0.02 to see its impacts on the system's performance.

Fig. 8 clearly show us that with the increase of δ_1 , the average energy consumption will increase slowly with δ_1 . This is because with the increase of δ_1 , our iterative algorithm will deviate away slowly from the optimal point and thus the average energy consumption will increase. But on the other hand, with the increase of δ_1 , we can clearly see that the number of iterations in our iterative algorithm will decrease, sharply at the start and then slow down. In practice, we can control the tradeoff between the computation complexity and the energy consumption by changing this threshold δ_1 . Fig. 8(b) also shows us that our algorithm will converge well fast. Averagely, no more than 45

times of iterations(in one iteration, we solve one convex optimization problem) are needed when $PER = 0.01$ and $\delta_1 = 0.0001$.

We also investigate the impact of δ_2 on our system's performance. we ignore it here for space saving.

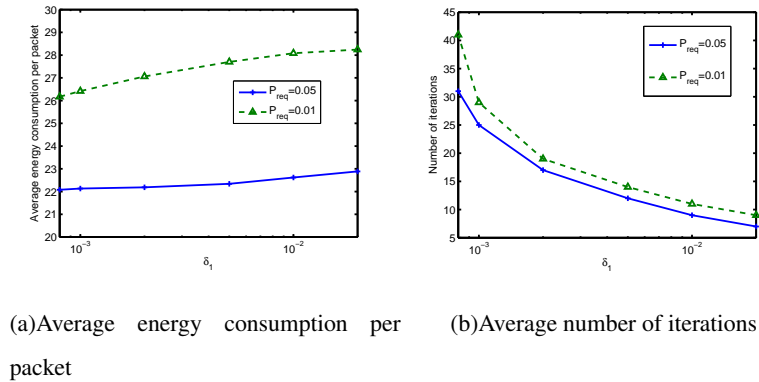


Fig. 8. Performance with varying δ_1

VII. CONCLUSIONS

In this paper, we propose a novel Multi-path Power-control Transmission scheme, MPT, for time-critical applications in underwater sensor networks. MPT combines the power control strategies with multi-path routing protocols and packet combining at the destination. Without retransmission at the intermediate nodes, MPT can achieve low end-to-end packet delay. With power control at the physical layer, MPT can achieve relatively high energy efficiency. For time-critical applications in energy constrained underwater sensor networks, MPT is a promising transmission scheme for a good balance between packet delay and energy efficiency.

Future Work: We would like to pursue our future work in two directions: 1) Explore MPT in other long-delay and error-prone networks; 2) Implement MPT in real underwater sensor network testbeds and investigate the practical design parameters.

REFERENCES

- [1] I. F. Akyildiz, D. Pompili, and T. Melodia. Underwater acoustic sensor networks: Research challenges. *Ad Hoc Networks (Elsevier)*, 3(3):257–279, March 2005.
- [2] M. A. Bhatti. *Practical optimization methods with mathematical applications*. New York springer, 2000.
- [3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2000.
- [4] Q. Cao, T. Abdelzاهر, T. He, and J. Stankovic. Towards optimal sleep scheduling in sensor networks for rare-event detection. In *Proceedings of ACM/IEEE IPSN*, volume 1, pages 20–27, April 2005.
- [5] C. Carbonelli and U. Mitra. Cooperative multihop communication for underwater acoustic networks. In *Proceedings of the 1st ACM international workshop on Underwater networks*, volume 1, pages 97–100, Sept. 2006.
- [6] X. Cheng, H. Shu, Q. Liang, and D. H.-C. Du. Slient positioning in underwater acoustic sensor networks. *To appear: IEEE Transactions on Vehicular Technology*, May 2008.

- [7] J.-H. Cui, J. Kong, M. Gerla, and S. Zhou. Challenges: building scalable mobile underwater wireless sensor networks for aquatic applications. *IEEE Network, Special Issue on Wireless Sensor Networking*, 20(3):12–18, May 2006.
- [8] S. K. Das, A. Mukherjee, S. Bandyopadhyay, D. Saha, and K. Paul. An adaptive framework for QoS routing through multiple paths in ad hoc wireless networks. *Journal of Parallel and Distributed Computing*, 63(2):141–153, Feb. 2003.
- [9] Y. Dong and D. K. Y. Yau. Adaptive sleep scheduling for energy-efficient movement-predicted wireless communication. In *Proceedings of 13th IEEE International Conference on Network Protocols (ICNP)*, pages 391–400, Nov 2005.
- [10] H. Dubois-Ferriere, D. Estrin, and M. Vetterli. Packet combining in sensor networks. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 102–115, Nov. 2005.
- [11] M. Erol, L. F. M. Vieira, and M. Gerla. AUV-aided localization for underwater sensor networks. In *Proceedings of International Conference on Wireless Algorithms, Systems and Applications (WASA)*, volume 1, pages 44–51, Aug 2007.
- [12] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball. The WHIO micro-modem: An acoustic communications and navigation system for multiple platforms. In *Proceedings of MTS/IEEE OCEANS*, volume 2, pages 1086–1092, June 2005.
- [13] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-resilient, energy efficient multipath routing in wireless sensor networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(4):11–25, 2001.
- [14] J. Heidemann, Y. Li, A. Syed, J. Wills, and W. Ye. Research challenges and applications for underwater sensor networking. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, pages 228–235, 2006.
- [15] Y. Hirayama, H. Okada, T. Yamazato, and M. Katayama. Time-dependent analysis of the multiple-route packet combining scheme in wireless multihop networks. *International Journal of Wireless Information Networks*, 12(1):35–45, Jan 2005.
- [16] S. Ibrahim, J.-H. Cui, and R. Ammar. Surface Gateway Deployment for Underwater Sensor Networks. In *Proceedings of IEEE Military Communications Conference*, Orlando, Atlanta, USA, October 2007.
- [17] M. Kubisch, H. Karl, A. Wolisz, L. C. Zhong, and J. Rabaey. Distributed algorithms for transmission power control in wireless sensor networks. In *Proceedings of IEEE Wireless Communications and Networking Conference(WCNC)*, volume 1, pages 558–563, March 2003.
- [18] S.-J. Lee and M. Gerla. Split multipath routing with maximally disjoint paths in ad hoc networks. In *Proceedings of IEEE International Conference on Communications*, volume 10, pages 3201–3205, 2001.
- [19] R. Leung, J. Liu, E. Poon, A.-L. C. Chan, and B. Li. MP-DSR: A QoS-Aware Multi-Path Dynamic Source Routing Protocol for Wireless Ad-Hoc Networks. In *proceedings of IEEE International Conference on Local Computer Networks*, Los Alamitos, CA, USA, 2001. IEEE Computer Society.
- [20] L. Lin, X. Lin, and N. B. Shroff. Low-complexity and distributed energy minimization in multi-hop wireless networks. In *Proceedings of IEEE INFOCOM*, pages 1685–1693, May 2007.
- [21] S. Lin, J. Zhang, G. Zhou, L. Gu, T. He, and J. A. Stankovic. ATPC: Adaptive transmission power controls for wireless sensor networks. In *Proceedings of the 4th ACM International Conference on Embedded Networked Sensor Systems (SenSys 2006)*, pages 223–236, Nov 2006.
- [22] W. Lou, W. Liu, and Y. Fang. Spread: Enhancing data confidentiality in mobile ad hoc networks. In *Proceedings of IEEE INFOCOM*, volume 4, pages 2404–2413, 2004.
- [23] M. Marina and S. Das. On-demand multi path distance vector routing in ad hoc networks. In *Proceedings of International Conference on Network Protocols(ICNP)*, pages 0–14, Los Alamitos, CA, USA, 2001. IEEE Computer Society.
- [24] A. Nasipuri and R. Castaneda. Performance of multipath routing for on-demand protocols in mobile ad hoc networks. *Journal of Parallel and Distributed Computing*, 6(2):339–349, Feb. 2001.

- [25] J. Partan, J. Kuose, and B. N. Levine. A survey of practical issues in underwater networks. In *Proceedings of the 1st ACM international workshop on Underwater networks*, pages 17–24, Sept. 2006.
- [26] B. Peleato and M. Stojanovic. Mac protocol for ad-hoc underwater acoustic sensor networks. In *Proceedings of the 1st ACM international workshop on Underwater networks*, pages 113–115, Sept. 2006.
- [27] D. Pompili, T. Melodia, and I. F. Akyildiz. Routing algorithms for delay-insensitive and delay-sensitive applications in underwater sensor networks. In *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (Mobicom)*, pages 298–309, Sep 2006.
- [28] J. G. Proakis. *Digital Communications*. McGraw-Hill, 2000.
- [29] V. Rodoplu and M. K. Park. An energy-efficient MAC protocol for underwater wireless acoustic networks. In *Proceedings of MTS/IEEE OCEANS*, volume 2, pages 1198–1203, Sept. 2005.
- [30] E. M. Royer and C. E. Perkins. Ad-hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, volume 2, pages 90–100, Feb 1999.
- [31] W. K. Seah and H.-X. Tan. Multipath Virtual Sink Architecture for Underwater Sensor Networks. In *Proceedings of OCEANS 2006 Asia Pacific Conference*, Singapore, May 2006.
- [32] M. K. Simon and M. S. Alouini. *Digital Communication over Fading Channels: A unified approach to performance analysis*. John Wiley & Sons, 2000.
- [33] P. S. Sindhu. Retransmission error control with memory. *IEEE Transactions on Communications*, 25(5):473–479, May 1977.
- [34] M. Stojanovic. Recent advances in high-speed underwater acoustic communications. *IEEE Journal of Oceanic Engineering*, 21(2):125–136, Apr. 1996.
- [35] M. Stojanovic. On the relationship between capacity and distance in an underwater acoustic communication channel. In *Proceedings of the 1st ACM international workshop on Underwater networks*, volume 1, pages 41–47, Sept. 2006.
- [36] A. Tsirigos and Z. J. Haas. Multipath routing in the presence of frequent topological changes. *IEEE Communication Magazine*, 39(11):132–138, Nov. 2001.
- [37] K. Wu and J. Harms. Performance study of a multipath routing method for wireless mobile ad hoc networks. In *Proceedings of the 9th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pages 99–107, 2001.
- [38] Y. Xi and E. M. Yeh. Distributed algorithms for spectrum allocation, power control, routing and congestion control in wireless networks. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc)*, pages 180–189, Sep 2007.
- [39] P. Xie, L. Lao, and J.-H. Cui. VBF: vector-based forwarding protocol for underwater sensor networks. In *Proceedings of IFIP Networking*, May 2006.
- [40] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocols for wireless sensor networks. In *Proceedings of IEEE INFOCOM*, volume 1, pages 1567–1576, New York, NY, USA, June 2002.
- [41] Z. Ye, S. V. Krishnamurthy, and S. K. Tripathi. A framework for reliable routing in mobile ad hoc networks. In *Proceedings of IEEE INFOCOM*, volume 1, pages 270–280, 2003.
- [42] Z. Zhou, J.-H. Cui, and S. Zhou. Localizaiton for large scale underwater sensor networks. In *Proceedings of IFIP Networking*, May 2007.

VIII. APPENDIX

A. Proof for Theorem 1

Under certain end-to-end BER requirement P_i^b , the optimal energy distribution along one path can be formulated as follows:

$$\begin{aligned} \min \quad & \sum_j E_{ij} \\ \text{s.t.} \quad & \frac{1}{2} [1 - \prod_{j \in (1 \dots n_i)} (1 - 2P_{ij}^b)] \leq P_i^b, \\ & P_{ij}^b = \frac{N_0 d_{ij}^a \beta(f)^{d_{ij}}}{4GE_{ij}}, \end{aligned}$$

which is equivalent to

$$\begin{aligned} \min \quad & \sum_j E_{ij} \\ \text{s.t.} \quad & \frac{1}{2} [1 - \prod_{j \in (1 \dots n_i)} (1 - 2 \frac{N_0 d_{ij}^a \beta(f)^{d_{ij}}}{4GE_{ij}})] = P_i^b. \end{aligned} \quad (26)$$

For this optimization problem, its lagrange function is

$$\begin{aligned} f(E_{ij}) = & \sum_j E_{ij} \\ & + \lambda \left(\frac{1}{2} [1 - \prod_{j \in (1 \dots n_i)} (1 - 2 \frac{N_0 d_{ij}^a \beta(f)^{d_{ij}}}{4GE_{ij}})] - P_i^b \right). \end{aligned} \quad (27)$$

At the optimal point, for the m_1 th hop, the transmitting energy per symbol E_{im_1} should satisfy

$$\frac{\partial f(E_{ij})}{\partial E_{im_1}} = 0. \quad (28)$$

We can then get that

$$\begin{aligned} \frac{4GE_{im_1}^2}{N_0 d_{im_1}^a \beta(f)^{d_{im_1}}} = \\ \lambda \prod_{j=(1, \dots, m_1-1, m_1+1, \dots, n_i)} (1 - 2 \frac{N_0 d_{ij}^a \beta(f)^{d_{ij}}}{4GE_{ij}}). \end{aligned} \quad (29)$$

In the same way, for the m_2 th hop, we can get that

$$\begin{aligned} \frac{4GE_{im_2}^2}{N_0 d_{im_2}^a \beta(f)^{d_{im_2}}} = \\ \lambda \prod_{j=(1, \dots, m_2-1, m_2+1, \dots, n_i)} (1 - 2 \frac{N_0 d_{ij}^a \beta(f)^{d_{ij}}}{4GE_{ij}}). \end{aligned} \quad (30)$$

From the above, we can get that

$$\frac{E_{im_1}^2}{E_{im_2}^2} = \frac{d_{im_1}^a \beta(f)^{d_{im_1}} (1 - 2 \frac{N_0 d_{im_1}^a \beta(f)^{d_{im_1}}}{4GE_{im_1}})}{d_{im_2}^a \beta(f)^{d_{im_2}} (1 - 2 \frac{N_0 d_{im_2}^a \beta(f)^{d_{im_2}}}{4GE_{im_2}})}. \quad (31)$$

Since $P_{im}^b = \frac{k_1 d_{im}^a \beta(f)^{d_{im}}}{E_{im}}$, Eq. (31) can be converted to

$$\frac{E_{im_1}^2}{E_{im_2}^2} = \frac{d_{im_1}^a \beta(f)^{d_{im_1}} (1 - 2P_{im_1}^b)}{d_{im_2}^a \beta(f)^{d_{im_2}} (1 - 2P_{im_2}^b)}. \quad (32)$$

Because P_{im}^b indicates the average BER of the m th link of the i th path. It should be much smaller than 1 for all qualified paths. We can approximate the above equation as,

$$\frac{E_{im_1}}{E_{im_2}} = \left(\frac{d_{im_1}^{\frac{a}{2}} \beta(f)^{\frac{d_{im_1}}{2}}}{d_{im_2}^{\frac{a}{2}} \beta(f)^{\frac{d_{im_2}}{2}}} \right). \quad (33)$$

And then, we can get that

$$\frac{E_{im_1}}{E_i} = \frac{d_{im_1}^{\frac{a}{2}} \beta(f)^{\frac{d_{im_1}}{2}}}{\sum_{j \in (1 \dots n_i)} d_{ij}^{\frac{a}{2}} \beta(f)^{\frac{d_{ij}}{2}}}. \quad (34)$$

The constraint of Eq. (26) can be rewritten as

$$\prod_{j \in (1 \dots n_i)} \left(1 - 2 \frac{N_0 d_{ij}^a \beta(f)^{d_{ij}}}{4GE_{ij}} \right) = 1 - 2P_i^b. \quad (35)$$

If $P_i^b < \frac{1}{2}$, we can take log on both side of (35) and get that

$$\sum_{j \in (1 \dots n_i)} \ln \left(1 - 2 \frac{N_0 d_{ij}^a \beta(f)^{d_{ij}}}{4GE_{ij}} \right) = \ln(1 - 2P_i^b). \quad (36)$$

For $\ln(1 - x)$, when x is small, it can be approximated as $\ln(1 - x) = -x$. Since $P_{ij}^b = \frac{N_0 d_{ij}^a \beta(f)^{d_{ij}}}{4GE_{ij}}$, which indicates the average bit error rate of the j th hop in the i th path, it should be much small than 1 for all qualified links. Then, we can get that

$$\begin{aligned} -2 \frac{N_0}{4G} \sum_{j \in (1 \dots n_i)} \frac{d_{ij}^a \beta(f)^{d_{ij}}}{E_{ij}} &= \ln(1 - 2P_i^b) \\ \sum_{j \in (1 \dots n_i)} \frac{d_{ij}^{\frac{a}{2}} \beta(f)^{\frac{d_{ij}}{2}} \sum_{j \in (1 \dots n_i)} d_{ij}^{\frac{a}{2}} \beta(f)^{\frac{d_{ij}}{2}}}{E_i} &= \frac{\ln(1 - 2P_i^b)}{-2 \frac{N_0}{4G}} \\ \frac{1}{E_i} \sum_{j \in (1 \dots n_i)} d_{ij}^{\frac{a}{2}} \beta(f)^{\frac{d_{ij}}{2}} \sum_{j \in (1 \dots n_i)} d_{ij}^{\frac{a}{2}} \beta(f)^{\frac{d_{ij}}{2}} &= \frac{\ln(1 - 2P_i^b)}{-2 \frac{N_0}{4G}} \\ E_i &= \sum_{j \in (1 \dots n_i)} d_{ij}^{\frac{a}{2}} \beta(f)^{\frac{d_{ij}}{2}} \sum_{j \in (1 \dots n_i)} d_{ij}^{\frac{a}{2}} \beta(f)^{\frac{d_{ij}}{2}} \frac{-2 \frac{N_0}{4G}}{\ln[(1 - 2P_i^b)]}. \end{aligned} \quad (37)$$

We set

$$A_i = 2 \frac{N_0}{4G} \sum_{j \in (1 \dots n_i)} d_{ij}^{\frac{\alpha}{2}} \beta(f)^{\frac{d_{ij}}{2}} \sum_{j \in (1 \dots n_i)} d_{ij}^{\frac{\alpha}{2}} \beta(f)^{\frac{d_{ij}}{2}}. \quad (38)$$

And then, we can get that

$$E_i = \frac{-A_i}{\ln[(1 - 2P_i^b)]} \quad (39)$$

and

$$P_i^b = \frac{1}{2} (1 - e^{-\frac{A_i}{E_i}}). \quad (40)$$

B. Proof for the convexity of the first constraint in Eq. (23)

Set the left side of the first constraint in Eq. (23) as $g(E_1, \dots, E_i, \dots, E_m)$,

$$g(E_1, \dots, E_i, \dots, E_m) = - \left[\sum_{i=(1,2,\dots,m)} \frac{(1 + e^{-\frac{A_i}{E_i}})}{2(1 - x_i)^{\frac{1}{L}}} \right] \quad (41)$$

take derivative to E_i and we can get that

$$\frac{\partial g(E_1, \dots, E_i, \dots, E_m)}{\partial E_i} = - \frac{\frac{A_i}{E_i^2} e^{-\frac{A_i}{E_i}}}{2(1 - x_i)^{\frac{1}{L}}}. \quad (42)$$

Since its first derivative are always negative when $E_i > 0$ and $A_i > 0$, $g(E_1, \dots, E_i, \dots, E_m)$ is a monotonically decreasing function of E_i . Its second order derivative is.

$$\frac{\partial^2 g(E_1, \dots, E_i, \dots, E_m)}{\partial^2 E_i} = \frac{\frac{A_i e^{-\frac{A_i}{E_i}}}{E_i^4} (2E_i - A_i)}{2(1 - x_i)^{\frac{1}{L}}}. \quad (43)$$

If $E_i > \frac{A_i}{2}$, $\frac{\partial^2 g(E_1, \dots, E_i, \dots, E_m)}{\partial^2 E_i} > 0$. And it can be easily verified that $\frac{\partial^2 g(E_1, \dots, E_i, \dots, E_m)}{\partial E_i \partial E_j} = 0$. The Hessian matrix of $g(E_1, \dots, E_i, \dots, E_m)$ is positive semidefinite. And then, we can conclude that that $g(E_1, \dots, E_i, \dots, E_m)$ is a convex function of E_i when $E_i > \frac{A_i}{2} \forall i \in (1, 2 \dots m)$ [3].

C. Proof for Theorem 2

First, let's set $h_i = \frac{1}{2(1-x_i)^{\frac{1}{L}}}$ and

$$H = -m^{\frac{L-1}{L}} \times \left[\sum_{i=(1,2,\dots,m)} \left(\ln(1 - x_i) + \frac{x_i}{1 - x_i} \right) - \ln\left(\frac{P_{req}}{1 - C}\right) \right]^{\frac{1}{L}}$$

Eq. (24) can be written as

$$\begin{aligned} \min \quad & E_M = \sum_{i=(1,2,\dots,m)} E_i L \\ \text{s.t.} \quad & - \left[\sum_{i=(1,2,\dots,m)} h_i (1 + e^{-\frac{A_i}{E_i}}) \right] = H \\ & \frac{A_i}{2} \leq E_i \leq \eta_i E_{\max} \quad i \in (1, 2, \dots, m) \end{aligned}$$

And its lagrange function is as follows,

$$\begin{aligned} f(E_i) = & \sum_i E_i L + \lambda \left[\sum_i h_i (1 + e^{-\frac{A_i}{E_i}}) + H \right] \\ & + \sum_i u_i \left(-E_i + \frac{A_i}{2} \right) + \sum_i v_i (E_i - k_i E_{\max}) \end{aligned} \quad (44)$$

where λ , u_i and v_i are lagrange multipliers. Take derivative of $f(E_i)$ to E_i , when E_i get its optimal value, the following equation must be satisfied.

$$\begin{aligned} \frac{\partial f(E_i)}{\partial E_i} &= L + \lambda \left(-h_i \frac{A_i e^{-\frac{A_i}{E_i}}}{E_i^2} \right) - u_i + v_i = 0 \\ \lambda &= \frac{L + v_i - u_i}{\frac{-h_i A_i e^{-\frac{A_i}{E_i}}}{E_i^2}} \end{aligned} \quad (45)$$

$$\text{set } g(E_i) = \frac{-h_i A_i e^{-\frac{A_i}{E_i}}}{E_i^2},$$

$$\lambda = \frac{L + v_i - u_i}{g(E_i)}. \quad (46)$$

It can be easily verified that $g(E_i) < 0$ and $g(E_i)$ is a monotonically increasing function of E_i .

Corresponding to C_1 and C_2 , we assume that their lagrange multiplier λ in (44) to be λ_{C_1} and λ_{C_2} respectively. We define E_{i,C_i} as the optimal transmitting energy per bit on path i when $C = C_i$, And define P_{i,C_i}^b as the optimal bit error rate of path i when $C = C_i$.

First, let's set $C_1 > C_2$. It should be noted that C_i actually equals to the probability that the destination can correctly receive every bit of a packet from more than one half paths. If $C_1 > C_2$, there exists at least one path i which can satisfy $P_{i,C_1}^b < P_{i,C_2}^b$. This can be verified by contradiction. If all paths satisfy $P_{i,C_1}^b > P_{i,C_2}^b \quad i \in (1, 2..j)$, every path has worse end-to-end PER performance when $C = C_1$ than $C = C_2$. From the definition of C_i , we can easily conclude that $C_1 < C_2$, which is in contradiction with the previous assumption that $C_1 > C_2$. So, there at least exists one path i with $P_{i,C_1}^b < P_{i,C_2}^b$. Correspondingly, $E_{i,C_1} > E_{i,C_2}$.

Let's assume that when $C_1 > C_2$, for the i th path, $P_{i,C_1}^b < P_{i,C_2}^b$ and $E_{i,C_1} > E_{i,C_2}$. In the following, we will prove the conclusion of Theorem 2 in different situations. Each situation corresponds to different values of E_i , which will activate or deactivate different constraints in Eq. (24) at the optimal point.

- (Case 1): When $C = C_1$, at the optimal point, constraint v_i is active ($E_{i,C_1} = \eta_i E_{\max}, v_i > 0$), constraint u_i is inactive ($E_{i,C_1} > \frac{A_i}{2}, u_i = 0$). While when $C = C_2$, at the optimal point, constraint v_i is inactive ($E_{i,C_2} < \eta_i E_{\max}, v_i = 0$), constraint u_i is inactive ($E_{i,C_2} > \frac{A_i}{2}, u_i = 0$). Because $E_{i,C_1} > E_{i,C_2}$ and $g(E_i)$ is a monotonically increasing function of E_i with $g(E_i) < 0$, according to Eq. (46), we can get that in this case $\lambda_{C_1} < \lambda_{C_2}$.
- (Case 2): When $C = C_1$, at the optimal point, constraint v_i is active ($E_{i,C_1} = \eta_i E_{\max}, v_i > 0$), constraint u_i is inactive ($E_{i,C_1} > \frac{A_i}{2}, u_i = 0$). While when $C = C_2$, at the optimal point, constraint v_i is inactive ($E_{i,C_2} < \eta_i E_{\max}, v_i = 0$), constraint u_i is active ($E_{i,C_2} = \frac{A_i}{2}, u_i > 0$). Because $E_{i,C_1} > E_{i,C_2}$ and $g(E_i)$ is a monotonically increasing function of E_i with $g(E_i) < 0$, according to eq. (46), we can get that in this case $\lambda_{C_1} < \lambda_{C_2}$.
- (Case 3): When $C = C_1$, at the optimal point, constraint v_i is inactive ($E_{i,C_1} < k_i E_{\max}, v_i = 0$), constraint u_i is inactive ($E_{i,C_1} > \frac{A_i}{2}, u_i = 0$). While when $C = C_2$, at the optimal point, constraint u_i is inactive ($E_{i,C_2} < k_i E_{\max}, v_i = 0$), constraint u_i is active ($E_{i,C_2} = \frac{A_i}{2}, u_i > 0$). Because $E_{i,C_1} > E_{i,C_2}$ and $g(E_i)$ is a monotonically increasing function of E_i with $g(E_i) < 0$, according to eq. (46), we can get that in this case $\lambda_{C_1} < \lambda_{C_2}$.
- (Case 4): under $C = C_1$, at the optimal point, constraint v_i is inactive ($E_{i,C_1} < \eta_i E_{\max}, v_i = 0$), constraint u_i is inactive ($E_{i,C_1} > \frac{A_i}{2}, u_i = 0$). While when $C = C_2$, at the optimal point, constraint v_i is inactive ($E_{i,C_2} < \eta_i E_{\max}, v_i = 0$), constraint u_i is inactive ($E_{i,C_2} = \frac{A_i}{2}, u_i = 0$). Because $E_{i,C_1} > E_{i,C_2}$ and $g(E_i)$ is a monotonically increasing function of E_i with $g(E_i) < 0$, according to Eq. (46), we can get that in this case $\lambda_{C_1} < \lambda_{C_2}$.

Since $E_{i,C_1} > E_{i,C_2}$, there will be no other case. And we can conclude that when $C_1 > C_2$, $\lambda_{C_1} < \lambda_{C_2}$.

And next, we will prove that if $C_1 > C_2$, there exists no path j with $E_{j,C_1} < E_{j,C_2}$. We use contradiction to prove this conclusion.

If there exists path j which satisfy $E_{j,C_1} < E_{j,C_2}$, For this path j , In the same way, we can get the following cases .

- (Case 1): When $C = C_1$, at the optimal point, constraint u_j is active ($E_{j,C_1} = \frac{A_j}{2}, u_j > 0$), constraint v_j is inactive ($E_{j,C_1} < \eta_j E_{\max}, v_j = 0$). While when $C = C_2$, at the optimal point, constraint u_j is inactive ($E_{j,C_2} > \frac{A_j}{2}, u_j = 0$), constraint v_j is inactive ($E_{j,C_2} < \eta_j E_{\max}, v_j = 0$). Since $g(E_j)$ is a monotonically increasing function and $g(E_j) < 0$, according to Eq. (46), we can get that in this case $\lambda_{C_1} > \lambda_{C_2}$.
- (Case 2): When $C = C_1$, at the optimal point, constraint u_j is inactive ($E_{j,C_1} > \frac{A_j}{2}, u_j = 0$), constraint v_j is inactive ($E_{j,C_1} < \eta_j E_{\max}, v_j = 0$). While when $C = C_2$, at the optimal point, constraint u_j is inactive ($E_{j,C_2} > \frac{A_j}{2}, u_j = 0$), constraint v_j is inactive ($E_{j,C_2} < \eta_j E_{\max}, v_j = 0$). Since $g(E_j)$ is a monotonically increasing function and $g(E_j) < 0$, according to Eq. (46), we can get that in this case $\lambda_{C_1} > \lambda_{C_2}$.
- (Case 3): When $C = C_1$, at the optimal point, constraint u_j is active ($E_{j,C_1} = \frac{A_j}{2}, u_j \geq 0$), constraint v_j is inactive ($E_{j,C_1} < \eta_j E_{\max}, v_j = 0$). While when $C = C_2$, at the optimal point, constraint u_j is inactive ($u_j = 0$), constraint v_j is active ($E_{j,C_2} = \eta_j E_{\max}, v_j = 0$). Since $g(E_j)$ is a monotonically increasing function and $g(E_j) < 0$, according to Eq. (46), we can get that in this case $\lambda_{C_1} > \lambda_{C_2}$.

• (Case 4): When $C = C_1$, at the optimal point, constraint u_j is inactive ($E_{j,C_1} > \frac{A_i}{2}, u_j = 0$), constraint v_j is inactive ($E_{j,C_1} < \eta_j E_{\max}, v_j = 0$). While when $C = C_2$, at the optimal point, constraint u_j is inactive ($E_{j,C_2} > \frac{A_i}{2}, u_j = 0$), constraints v_j is active ($E_{j,C_2} = \eta_j E_{\max}, v_j = 0$). Since $g(E_j)$ is a monotonically increasing function and $g(E_j) < 0$, according to Eq. (46), we can get that in this case $\lambda_{C_1} > \lambda_{C_2}$.

Thus, under all situations, if there exist one path j with $E_{j,C_1} < E_{j,C_2}$, we conclude that $\lambda_{C_1} > \lambda_{C_2}$. This is in contradiction with our previous conclusion $\lambda_{C_1} < \lambda_{C_2}$. So, there exist no path j with $E_{j,C_1} < E_{j,C_2}$ when $C_1 > C_2$.

And then, we get that if $C_1 > C_2, (E_{1,C_1}, E_{2,C_1}, \dots, E_{j,C_1}) \succeq (E_{1,C_2}, E_{2,C_2}, \dots, E_{j,C_2})$. Correspondingly, we can get that $(P_{b1,C_1}, P_{b2,C_1}, \dots, P_{bj,C_1}) \preceq (P_{b1,C_2}, P_{b2,C_2}, \dots, P_{bj,C_2})$.

In the same way, we can prove that if $C_1 < C_2, (E_{1,C_1}, E_{2,C_1}, \dots, E_{j,C_1}) \preceq (E_{1,C_2}, E_{2,C_2}, \dots, E_{j,C_2})$ and $(p_{b1,C_1}, p_{b2,C_1}, \dots, p_{bj,C_1}) \succeq (p_{b1,C_2}, p_{b2,C_2}, \dots, p_{bj,C_2})$.

And then, we get Theorem 2.

D. Proof for Theorem 3

In our algorithm, we first sort all links according to their “ A_i ”s. And then, we calculate the optimal energy distribution for all first “ k ” paths and select the best solution. If we can prove that for any “ k ” paths, the proposed algorithm will converge in finite time. Then, our algorithm will converge.

For the calculation of k paths, it includes two iterations. The outer iteration is to update the x_i , and the inner iteration is to update the C . For the outer iteration, from corollary 2, we know that every iteration will get results at least as good as its previous one. So, it will converge in finite time.

Thus, if we can prove that the inner iteration which is used to update C also converge in finite time, we can prove that our algorithm will converge.

Let’s set at “ $j-1$ ” iteration $C = C_{j-1}$, the optimization problem is as follows,

$$\begin{aligned}
\min \quad & E_M = \sum_{i=(1,2,\dots,k)} E_i L \\
s.t. \quad & - \left[\sum_{i=(1,2,\dots,k)} \frac{(1 + e^{-\frac{A_i}{E_i}})}{2(1 - x_i)^{\frac{1}{L}}} \right] \leq -j^{\frac{L-1}{L}} \times \\
& \left[\sum_{i=(1,2,\dots,k)} \left(\ln(1 - x_i) + \frac{x_i}{1 - x_i} \right) - \ln\left(\frac{Pre}{1 - C_{j-1}}\right) \right]^{\frac{1}{L}} \\
& \frac{A_i}{2} \leq E_i \leq \eta_i E_{\max} \quad i \in (1, 2, \dots, k)
\end{aligned} \tag{47}$$

And we can get the corresponding optimal solution $P_{1,C_{j-1}}^b, P_{2,C_{j-1}}^b, \dots, P_{k,C_{j-1}}^b$. Then, we can calculate C_j for the

next iteration.

$$C_j = \left[\sum_{n=\lfloor \frac{k}{2} + 1 \rfloor}^k (1 - P_{1,C_{j-1}}^b) \right. \\ \left. \times \underbrace{\dots (1 - P_{n,C_{j-1}}^b) P_{n+1,C_{j-1}}^b \dots P_{k,C_{j-1}}^b + \dots}_{\binom{k}{n}} \right]^L \quad (48)$$

If $C_j = C_{j-1}$, then, we got the optimal solution with k paths.

If $C_j > C_{j-1}$, then, at the j th iteration, we solve

$$\begin{aligned} \min \quad E_M &= \sum_{i=(1,2,\dots,k)} E_i L \\ \text{s.t.} \quad & - \left[\sum_{i=(1,2,\dots,k)} \frac{(1 + e^{-\frac{A_i}{E_i}})}{2(1 - x_i)^{\frac{1}{L}}} \right] \leq -j^{\frac{L-1}{L}} \times \\ & \left\{ \sum_{i=(1,2,\dots,k)} \left(\ln(1 - x_i) + \frac{x_i}{1 - x_i} \right) - \ln\left(\frac{P_{re}}{1 - C_j}\right) \right\}^{\frac{1}{L}} \\ & \frac{A_i}{2} \leq E_i \leq \eta_i E_{\max} \quad i \in (1, 2, \dots, k) \end{aligned} \quad (49)$$

And we can get the solution $P_{1,C_j}^b, P_{2,C_j}^b, \dots, P_{k,C_j}^b$. Then, we can calculate C_{j+1} for the next iteration.

$$C_{j+1} = \left[\sum_{n=\lfloor \frac{k}{2} + 1 \rfloor}^k (1 - P_{1,C_j}) \right. \\ \left. \times \underbrace{\dots (1 - P_{n,C_j}^b) P_{n+1,C_j}^b \dots P_{k,C_j}^b + \dots}_{\binom{k}{n}} \right]^L \quad (50)$$

Since C equals to the probability that more than one half paths can correctly transmit all bits in one packet. If all P_i^b increase, which means that for all path, its bit error rate will increase, C will decrease. While if all P_i^b decrease, C_i will increase. Since $C_k > C_{k-1}$, from Theorem 2, we know that for all path, $P_{i,C_k}^b > P_{i,C_{k-1}}^b$, which leads to $C_{k+1} < C_k$. And in the same way, we can prove that if $C_k < C_{k-1}$, then $C_{k+1} > C_k$.

The convergence process is shown in Fig. 9.

When the selected number of paths is k .

At the first iteration, we set $C = C_0 = 0$. And we get the corresponding solution $P_{1,C_0}^b, P_{2,C_0}^b, \dots, P_{k,C_0}^b$, and calculate C_1 . it is apparent that $C_1 \geq C_0$,

At the second iteration, with C_1 , we get the corresponding solution $P_{1,C_1}^b, P_{2,C_1}^b, \dots, P_{k,C_1}^b$, and calculate C_2 . From above, we know that $C_2 \leq C_1$. Since $C_0 = 0$ and $C_2 \leq C_1$, the gap between C_2 and C_1 is smaller than the gap between C_1 and C_0 , as shown in Fig. 9.

$(1 - P_i) + 2P_i(1 - P_i) + 3P_i^2(1 - P_i) + \dots + MP_i^{(M-1)}(1 - P_i) = \frac{1+P_i^M(M(P_i-1)-1)}{1-P_i}$ since P_i will be much smaller than 1, $n_i \approx \frac{1}{1-P_i}$. we use E_i to demote the transmission (retransmission) energy per symbol for the i th hop. And the average overall transmitting energy (including the retransmission) per packet for the i th hop will be $E_i \times n_i \times L$. Then, we can formulate the following optimization problem to minimize the overall energy consumption per packet(Here, we ignore the energy for ACK transmission since ACK packet is usually short.)

$$\begin{aligned}
& \min \sum_i LE_i \left(\frac{1}{1-P_i} \right) \\
s.t \quad & 1 - \prod_i (1 - P_i^M) \leq P_e \\
& 0 \leq E_i \leq E_{\max} \\
& P_i = 1 - \left(1 - \frac{N_0 d_i^a \beta(f)^{d_i}}{4GE_i} \right)^L \approx L \frac{N_0 d_i^a \beta(f)^{d_i}}{4GE_i}
\end{aligned} \tag{51}$$

This problem can be converted to

$$\begin{aligned}
& \min \sum_i \frac{L^2 N_0 d_i^a \beta(f)^{d_i}}{4GP_i(1-P_i)} \\
s.t \quad & \sum_i \ln(1 - P_i^M) \geq \ln(1 - P_e) \\
& \frac{LN_0 d_i^a \beta(f)^{d_i}}{4GE_{\max}} \leq P_i.
\end{aligned} \tag{52}$$

It can be easily verified that this problem is a convex problem and can be solved efficiently.