# マルチビュービデオのクロスレイヤ型伝送方式に関する検討

藤橋 卓也[1] 小寺 志保[2] 猿渡 俊介[2] 渡辺 尚[3]

**概要：**
マルチビュービデオの研究は，従来，有線網などのネットワークを前提として，トラフィックの削減，応答遅延の削減，映像品質の維持に焦点が当てられてきた．しかしながら，マルチビュービデオをより多くの場面に応用するため，劣悪な環境を通したマルチビュービデオ伝送が必要となると考えられる．本稿では，劣悪な環境の例として水中音響通信を想定し，これら 3 つの要件を満たすために，MAC 層とアプリケーション層のクロスレイヤ伝送方式である Slipped-TDMA および予測伝送方式である Zaoral Streaming を提案する．Slipped-TDMA では，水中音響通信の帯域を最大限活用するため，水中音響通信における伝搬遅延およびビデオエンコーダとユーザ間で発生するトラフィックの非対称性を考慮して，タイムスロットの割り当てを行う．Zaoral Streaming では，応答遅延を削減するため，割り当てられたタイムスロットを利用し，ユーザが次に試聴する可能性が高いカメラを予測することで，ユーザがカメラを切り替える前に映像を伝送する．予測が失敗した場合は，トラフィックと応答遅延の増加及び映像品質の劣化を抑制するため，送信済みの映像を用いてユーザが必要とするカメラ映像の符号化を行う．MERL が提供するテストビデオシーケンスを利用した計算機シミュレーションにより，提案方式は単純な伝送方式と比較して応答遅延が大幅に減少することを示す．また Zaoral Streaming によって，予測失敗時でも，トラフィックの増加と映像品質の劣化を抑制することを示す．

## A Fundamental Discussion on Cross-Layer Approach for Multi-view Video

Fujihashi Takuya[1] Kodera Shiho[2] Saruwatari Shunsuke[2] Watanabe Takashi[3]

## 1. Introduction

Many applications for single-view video streaming over underwater acoustic networks are available by exploiting perceivable information of underwater: undersea explorations [1], disaster prevention [2], mine reconnaissance [2] and environmental monitoring [1, 3]. To realize the single-view video streaming over underwater, earlier studies mainly handle a low data rate of underwater acoustic networks. The underwater acoustic networks have a much lower data rate than radio networks (i.e. 20 kbps in acoustic networks while 11 Mbps in radio networks of IEEE 802.11b). The low data rate spoils video quality.

Earlier studies of underwater video streaming classify two types: tethered transmission [4–7], and improvement of acoustic wave communication [8–10]. For example, [4] uses optical fiber between an encoder node and a user node in order to realize high quality underwater video streaming. The encoder node transmits the video with the resolution of 1280×1080 pixels to the user node. [8] achieves 90 kbps in a 115 kHz acoustic band over a 200 meter vertical link under a variety of channel conditions using OFDM modulation in acoustic networks.

This paper extends single-view video to multi-view video streaming. The multi-view video streaming over underwater acoustic networks has three requirements: the reduction of traffic, the suppression of the response delay, and maintaining video quality. These requirements affect user's satisfaction and application quality.

[1] 静岡大学大学院 情報学研究科
Graduate School of Informatics, Shizuoka University, Japan
[2] 静岡大学 情報学部
Faculty of Informatics, Shizuoka University, Japan
[3] 大阪大学 情報科学研究科
Graduate School of Information Science and Technology, Osaka University, Japan

To achieve the three requirements, we propose Slipped-TDMA and Zaoral Streaming. To improve the band-utilization of multi-view video streaming over underwater acoustic networks, Slipped-TDMA, which is one-to-one MAC protocol, assigns time-slipped slots for an encoder node and a user node by exploiting asymmetric traffic, and a long propagation delay between nodes. Zaoral Streaming achieves the three requirements by two features: prediction, and Zaoral Encoding/Decoding. First, the encoder node predicts the next camera position for the user in order to reduce the response delay. Second, even when the prediction failed, the encoder node encodes correct video frames with mis-predicted frames in order to prevent the increase of the response delay, and traffic. The user node then decodes the correct video frames from the past mis-predicted frames. Evaluations using the standard MERL's benchmark test sequences show that our proposed approach significantly reduces the response delay, as compared to naive methods.

The remainder of the paper is organized as follows. Section 2 contains a summary of related work. We describe the overview of our proposed approach in Section 3. Section 4 explains the proposed Slipped-TDMA. Section 5 describes proposed Zaoral Streaming: the details of its prediction, encoding, and decoding. Evaluations appear in order to reveal the reduction of the traffic, the suppression of response delay, and maintaining video quality for each scheme in Section 6. Finally, conclusions are summarized in Section 7.

## 2. Related Work

Multi-view video streaming enables us to watch underwater objects from every angle, and freely switchable viewpoints [11–13]. Users will be able to create 3D video of underwater objects using multi-view video sequences [14]. One of the applications is measurement of the size of coral fishes because the multiple angles contribute to mitigating complexity of the coral reef's shape.

Figure 1 shows the system model of multi-view video streaming over underwater acoustic networks. We assume that several cameras are connected to a video encoder node by wire, and the encoder node is connected to a user node by acoustic waves. The encoder node periodically sends multi-view video to the user node. The user node sends camera switch requests to the encoder node. The acoustic connection has a low data rate and a long propagation delay.

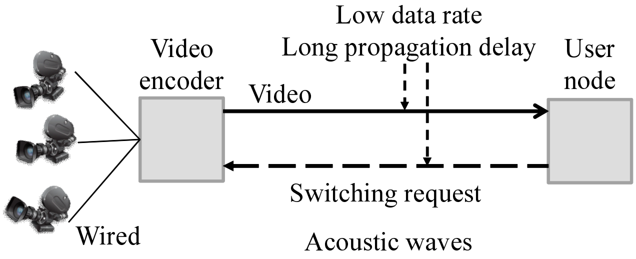


図 1 System model of multi-view video streaming over underwater acoustic networks

The multi-view video streaming over underwater acoustic networks has three requirements.

The first requirement is the reduction of video traffic. The traffic of multi-view video is larger than that of single-view video: the traffic of $N$ views video is $N$ times larger than that of single-view video as the simple estimation. However, the underwater acoustic networks are low-bandwidth: the latest study achieves a few hundred kbps.

The second requirement is the suppression of the response delay. The response delay denotes the time from switching to a camera to displaying the video at the user node. To switch cameras freely based on the user's request is one of the advantages of multi-view video. If the response delay is high, the user might be frustrated. Especially, the long propagation delay of acoustic waves increases the response delay.

The third requirement is maintaining video quality. The video quality represents the degree of degradation of decoded video from raw video. Maintaining video quality and the reduction of video traffic, which is the first requirement, are in a trade-off relationship. If the degradation is small, and the resolution and frame rate of video are high, the video is applied to many applications because the user detects a minute change. However, high video quality induces high video traffic.

To reduce the traffic with maintaining video quality, a naive method is the use of a simple request-reply model: a user node sends a request, which includes a camera position, and an encoder node sends back only corresponding video frames. The whole network bandwidth is used for the single-view video streaming. However, the use of simple request-reply model induces long response delay because of long propagation delay in acoustic networks.

If a encoder node sends all video frames to a user node with existing multi-view video codec [15–17], video quality is high, and response delay is low because the user always has all video frames, which he/she needs. However, the

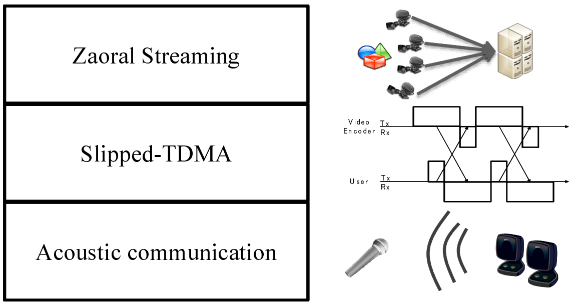


図 **2** Overview of proposed approach

transmission of all video frames induces very high traffic. For example, the traffic of H.264/AVC MVC [16] is very high: about 5 Mbps for 704 × 480, 30 fps, and 8 camera sequences [18].

To reduce the video traffic, one of the simplest methods is an encoder node degrades frame rate and quantization parameter of multi-view video, and transmits all video frames to a user node. However, not surprisingly, the degradation induces low video quality.

## 3. Overview

As mentioned in Section 2, there are three requirements for multi-view video streaming over underwater acoustic networks: the reduction of video traffic, the suppression of response delay, and maintaining video quality. To satisfy the above all requirements, we propose Slipped-TDMA and Zaoral Streaming.

Figure 2 shows the overview of our proposal. Slipped-TDMA improves the band-utilization of multi-view video streaming over underwater acoustic networks by exploiting asymmetric traffic between a user node and an encoder node. The details of the Slipped-TDMA are described in Section 4. Zaoral Streaming reduces the response delay by predicting user's behavior. Zaoral Streaming also prevents the increase of the response delay, and traffic. When the previous prediction is a failure: the encoder node encodes correct video frames with mis-predicted frames, which are already sent, and the user node decodes the correct video frames using Zaoral packets. The Zaoral packet includes the mis-predicted frames. The details of Zaoral Streaming are described in Section 5.

## 4. Slipped-TDMA

In underwater acoustic networks, conventional MAC protocols [19–27] decrease in band-utilization because of its long propagation delay. For example, Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) [19] induces a long response delay in underwater acoustic networks because nodes reserve a channel for several propagation delays, which results in large overheads. Time Division Multiple Access (TDMA) [20,21] also induces a long response delay because the large time slots required to prevent collisions. Each time slot requires large guard times that set the longest propagation delay between nodes.

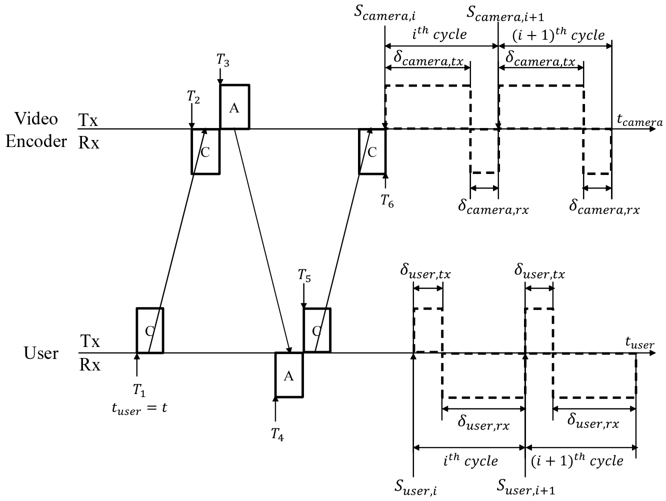


図 **3** Timing diagram of initiation phase

In view of this, we propose a new one-to-one MAC protocol called Slipped-TDMA. Slipped-TDMA consists of a initiation phase and a normal phase. The initiation phase consists of time synchronization and slot assignment. After the initiation phase, Slipped-TDMA transits to the normal phase.

### 4.1 Initiation Phase

Figure 3 depicts the timing diagram of initiation phase in Slipped-TDMA. We assume that a user node is located on the sea, and a video encoder node is located under the sea. The encoder node has uncorrected clock $t_{\mathrm{camera}}$ [s]. The user node has a perfect clock $t_{\mathrm{user}}$ [s]:

$$t_{\mathrm{camera}} = t + \theta \quad (1)$$

$$t_{\mathrm{user}} = t \quad (2)$$

where $t$ [s] is the global reference time, and $\theta$ [s] is offset of time. The encoder node and the user node exchange control packets in the initiation phase.

Each control packet consists of six fields as shown in Table 1. The type field represents four kinds of the control packet: SYNC, START, NORMAL, and ERROR. The sequence number field is increased automatically when a new packet is generated. The start time field recodes the beginning of transmission time. The offset field is used for the time synchronization between the encoder node and the user node. The playback frame field recodes the

**Algorithm 1** Initiation phase at the user node

```
Transmit the control packet whose type field is SYNC
Wait for an ACK packet from the encoder node
if Receive an ACK then
    Calculate propagation delay and offset of the time
    Transmit the control packet whose type field is START
    Assign time slots
end if
```

**Algorithm 2** Initiation phase at the encoder node

```
Wait for control packets from the user node
if Receive the control packet whose type field is SYNC then
    Transmit the ACK packet
else if Receive the control packet whose type field is START then
    Synchronize the time with the user node using offset of time
    Assign time slots
end if
```

| Field | Size [bits] |
|---|---|
| Type | 8 |
| Sequence number | 8 |
| Start time | 64 |
| Offset | 16 |
| Playback frame | 16 |
| Camera position | 8 |
| Propagation delay | 32 |

**表 1** Format for the control packet

number of playback frame. The camera position field represents a camera number, which is watched by the user. The propagation delay field is used for slot assignment in Slipped-TDMA.

Algorithm 1 describes the detail of the procedures at the user node. Algorithm 2 describes the detail of the procedures at the encoder node.

First, Slipped-TDMA synchronizes the user node and the encoder node. A user node first sends a control packet whose type field is SYNC to an encoder node when $t_{\text{user}}$ is $T_1$. The control packet sets the camera position, playback frame, and propagation delay fields to zero because these fields do not use in this packet. When the encoder node receives the first bit of the control packet whose type field is SYNC, the encoder node records the received time $T_2$. The encoder node calculates temporal offset of time $T'_{\text{delay}}$ [s] between the encoder node and user node as follows:

$$T'_{\text{delay}} = T_2 - T_1. \tag{3}$$

$T'_{\text{delay}}$ includes the propagation delay and offset of time. $T'_{\text{delay}}$ substitutes the offset field in an ACK packet. The encoder node immediately returns the ACK packet whose format is the same as that of control packet to the user node. When the user node receives the ACK packet, the user node transmits a control packet whose type field is START to the encoder node for slot assignment. The propagation delay field of the packet includes a calculated propagation delay $T_{\text{delay}}$ [s]. $T_{\text{delay}}$ is calculated as follows:

$$T_{\text{delay}} = \frac{T'_{\text{delay}} + (T_4 - T_3)}{2} \tag{4}$$

where $T_3$ is the start time field in the ACK packet, and $T_4$ is the reception time of the first bit of the ACK packet. The offset field of the packet includes a calculated offset of time $\theta$. $\theta$ is calculated as follows:

$$\theta = \frac{T'_{\text{delay}} - (T_4 - T_3)}{2} \tag{5}$$

When the encoder node receives the control packet, the encoder node synchronizes time with the user node. The synchronization is done as follows:

$$t_{\text{camera}} = t_{\text{camera}} - \theta = t_{\text{user}}. \tag{6}$$

Next, Slipped-TDMA assigns time slots to the user node and the encoder node depending on the propagation delay. The dotted line in Figure 3 shows assigned time slots for transmission and reception of packets. $S_{\text{user,i}}$ [s] is the beginning of $i$th time slot for the user node. $S_{\text{camera,i}}$ [s] is the beginning of $i$th time slot for the encoder node. $i$ is more than zero. Each node calculates $S_{\text{user,i}}$, and $S_{\text{camera,i}}$ as follows:

$$S_{\text{user,i}} = T_5 + 2T_{\text{delay}} + i(\delta_{\text{user,tx}} + \delta_{\text{user,rx}}) \tag{7}$$

$$S_{\text{camera,i}} = T_6 + i(\delta_{\text{camera,tx}} + \delta_{\text{camera,rx}}) \tag{8}$$

where $T_5$ is the beginning of the transmission time of the control packet whose type field is START, $T_6$ is the reception time of the control packet whose type field is START. $\delta_{\text{user,tx}}$ [s] and $\delta_{\text{user,rx}}$ [s] are the length of each transmission slot, and reception slot for the user node. $\delta_{\text{camera,tx}}$ [s] and $\delta_{\text{camera,rx}}$ [s] are the length of each transmission slot, and reception slot for the encoder node. The user node derives $\delta_{\text{user,tx}}$, and $\delta_{\text{user,rx}}$ as follows:

$$\delta_{\text{user,tx}} = \frac{B_{\text{control}}}{R} \tag{9}$$

$$\delta_{\text{user,rx}} = 2T_{\text{delay}} - \delta_{\text{user,tx}} \tag{10}$$

where $R$ [bps] is the data rate of underwater acoustic networks, and $B_{\text{control}}$ [bits] is the control packet size. $\delta_{\text{camera,tx}}$ and $\delta_{\text{camera,rx}}$ of the encoder node correspond as $\delta_{\text{user,rx}}$ and $\delta_{\text{user,tx}}$ as follows:

$$\delta_{\text{camera,tx}} = \delta_{\text{user,rx}} \tag{11}$$

$$\delta_{\text{camera,rx}} = \delta_{\text{user,tx}} \tag{12}$$

| Field | Size [bits] |
|---|---|
| Camera position | 8 |
| Frame index | 16 |
| Readjustment flag | 1 |
| Video data | Variable |

表 **2**　Format for the video packet

**Algorithm 3** Normal phase at the encoder node

```
if $t_{camera}$ is in $\delta_{\text{camera,rx}}$ then
    Wait for a control packet whose type field is NORMAL
    if Does not receive control packet from the user node then
        Decide to set the readjustment flag field to 1 at the next transmission slot
    end if
else if $t_{camera}$ is in $\delta_{\text{camera,tx}}$ then
    if Decided to set the readjustment flag field to 1 then
        Set the readjustment flag field to 1
    end if
    Predict the user's next camera position (See Sec. 5)
    Encode the predicted camera
    Transmit video packets
    if The readjustment flag field of the video packets is 1 then
        Transit to the initiation phase
    end if
end if
```

**Algorithm 4** Normal phase at the user node

```
if $t_{user}$ is in $\delta_{\text{user,rx}}$ then
    Receive video packets
    Decode the video packets
    if Readjustment flag is 1 then
        Transit to the initiation phase
    end if
else if $t_{user}$ is in $\delta_{\text{user,tx}}$ then
    Transmit the control packet whose type field is NORMAL
end if
```

### 4.2 Normal Phase

A normal phase exchanges packets using assigned time slots between an encoder node and a user node. The user node transmits a control packet whose type field is NORMAL. The encoder node transmits video packets. Each video packet consists of four fields as shown in Table 2. The camera position field represents the camera number of encoded video. The frame index field substitutes the frame number of encoded video. The readjustment flag field is used to transit to initiation phase. The rest of video packet is actual encoded video.

Algorithm 3 describes the details of the procedures at the encoder node. Algorithm 4 describes the details of the procedures at the user node.

When $t_{camera}$ is in $\delta_{\text{camera,rx}}$, an encoder node waits for a control packet whose type field is NORMAL from a user node. When the encoder node receives the control packet in the reception slot, each field of the control packet is exploited for the next transmission slot. If the encoder node does not receive the control packet, the encoder node decides to set the readjustment flag field of video packets to 1 at the next transmission slot. When $t_{camera}$ is in $\delta_{\text{camera,tx}}$, the encoder node transmits video packets to the user node. To transmit the video packets, the encoder node predicts a user's next camera position. The details of the prediction are described in Section 5. After the prediction, the encoder node encodes the predicted camera. If the encoder node decided to set the readjustment flag field to 1 in the previous reception slot, the encoder node first sets the readjustment flag field to 1, and the encoder node transits to the initiation phase.

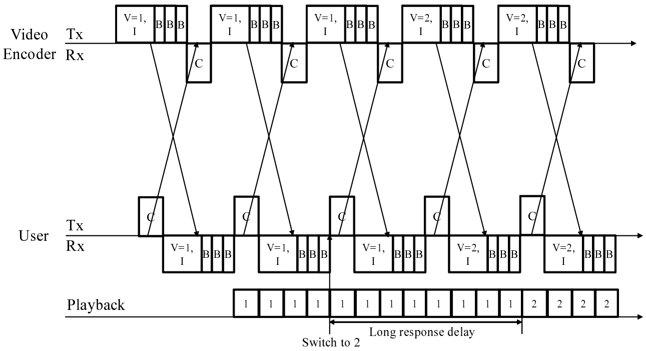


図 **4**　Naive method

When $t_{user}$ is in $\delta_{\text{user,rx}}$, the user node waits for video packets from the encoder node. When the user node receives video packets, the user node decodes the video. After the video decoding, the user node checks the readjustment flag field. If the readjustment flag field is 1, the user node transits to the initiation phase. When $t_{user}$ is in $\delta_{\text{user,tx}}$, the user node transmits the control packet whose type field is NORMAL to the encoder node.

## 5. Zaoral Streaming

Slipped-TDMA improves band-utilization of multi-view video streaming over underwater acoustic networks. However, the response delay is still long. Figure 4 shows a timing diagram of a naive method. $V$ is a camera position, and $I$ and $B$ represent an encoded video frame that is standardized in H.264/AVC. The values in the underneath boxes are the camera position of the playback frame. The naive method exchanges control packets and video packets using assigned time slots between an encoder node and a user node. When the user switches to a camera, the user node has to spend much time for waiting to playback the camera.

To overcome the problem, we propose Zaoral Streaming. Zaoral Streaming consists of a prediction, and Zaoral

Encoding/Decoding. The prediction reduces the response delay by predicting the next camera position for the user. Zaoral Encoding/Decoding prevents the increase of the response delay, and traffic.

### 5.1 Prediction

Figure 5 shows a prediction example in the normal phase. An encoder node predicts the next camera position for a user at the beginning of each transmission slot. The prediction is done every one Group of Picture (GOP). Figure 5 assumes that the encoder node predicts that the user will gaze camera 1 at the first GOP, and switch to camera 2 at the second GOP. To predict the camera position from the previous control packets, the encoder node uses two types of solution depending on view-switching models of the user.

**1)** Kalman Filter:
Kalman Filter supports that estimations of future state even when the precise nature of the modeled system is unknown [28,29]. When the user switches cameras linearly as shown in Figure 6 (a), Kalman Filter predicts the next camera position with a high probability. Figure 6 (a) assumes that the user switches views with the same camera-switching speed until the end of the video.

**2)** Bayesian Estimation:
Bayesian Estimation is suitable for estimations of future states in nonlinear situation such as a high bias model [30, 31]. Bayesian Estimation stores all of previous camera positions at the encoder node in order to enhance the prediction accuracy. Figure 6 (b) shows one of the high bias models. The user switches cameras to find a favorite camera, and gazes the camera for a while.

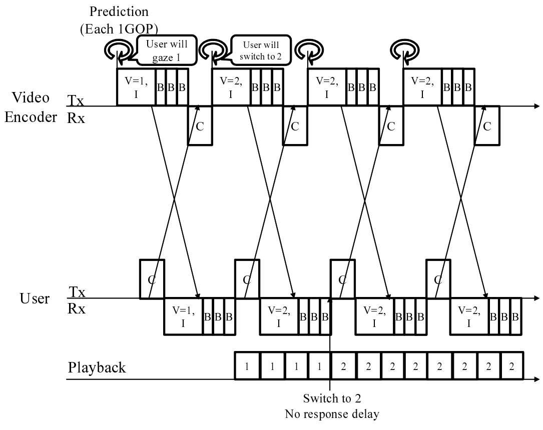
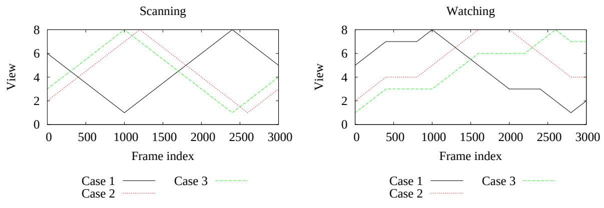


**図 5** Zaoral Streaming: Prediction



(a) (b)

**図 6** Samples of view-switching model of the user. (a) Linear model. (b) High bias model.

After the prediction, the encoder node transmits the predicted camera's video packets to the user node in the transmission slot. After receiving the video packets in a reception slot, the user node checks if the prediction was correct by reference to buffered video packets at the beginning of the next transmission slot. When the user switches to camera $V$ at the beginning of the next transmission slot, the user node searches for each buffered video packet whose camera position field corresponds as $V$. If a buffered video packet satisfies the condition, the user node detects that the prediction was correct, and the user node playbacks the camera's video soon.

### 5.2 Zaoral Encoding

When a prediction is a failure, Zaoral Streaming uses Zaoral Encoding and Decoding. Figure 7 shows a case that an encoder node predicts that a user will continue to gaze camera 1 at second GOP, and the user actually switches to camera 2 at the same GOP.

After a user node detects a failure prediction at the beginning of a transmission slot, the user node transmits a control packet whose type field is ERROR to an encoder node in the transmission slot. When the encoder node receives the control packet in a reception slot, the encoder node detects the failure prediction by reference to the type field in the control packet, and the encoder node

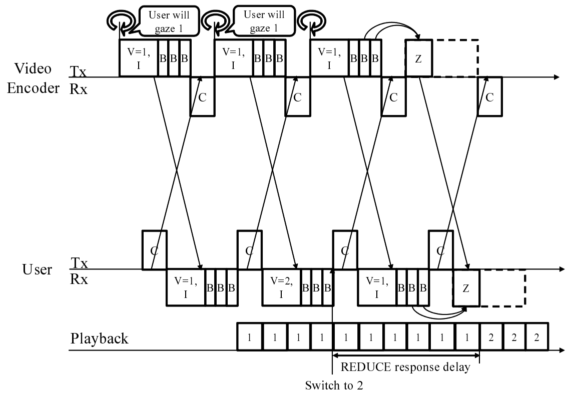


**図 7** Zaoral Streaming: Zaoral Encoding and Decoding

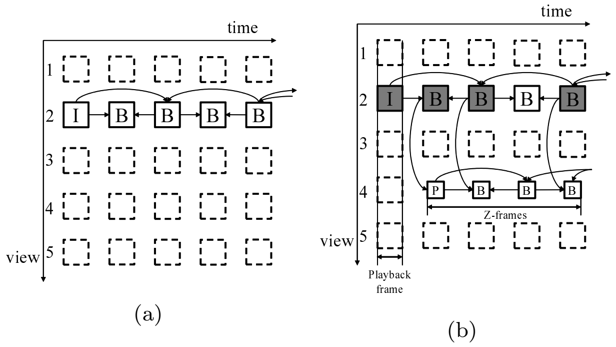


図 8 Prediction structure in Zaoral Streaming. (a) Prediction (b) Zaoral Encoding

begins Zaoral Encoding at the next transmission slot. Zaoral Encoding first detects correct video frames for the user node according to the camera position, and playback frame fields in the control packet. The correct frames are video frames whose camera position corresponds as the camera position field, and GOP number corresponds as that of playback frame field. The correct video frames are called Z-frames.

After the detection, Zaoral Encoding encodes Z-frames with mis-predicted frames, which are already sent. Each Z-frame is predicted from a mis-predicted frame in the same instant. Figure 8 shows prediction structure of Zaoral Streaming. Figure 8 (a) shows prediction structure of a prediction when an encoder node predicts that the next camera position for a user is 2. The prediction structure is based on H.264/AVC MVC [16]. Figure 8 (b) shows prediction structure of Zaoral Encoding when a correct camera position is 4, and mis-predicted frames are the gray frames.

After Zaoral Encoding, the encoder node transmits Zaoral packets to the user node in the transmission slot. Each Zaoral packet includes the encoded Z-frames. Transmission order of the Z-frames depends on encoding dependency in order to decode the Z-frames smoothly at the user node.

### 5.3 Zaoral Decoding

A user node decodes all received video frames even if the video frames are not need for the user node. The reason is to decode Z-frames immediately when the user node receives Zaoral packets from an encoder node. When the user node receives Zaoral packets from the encoder node, the user node begins Zaoral Decoding in order to decode Z-frames. Zaoral Decoding decodes Z-frames by a standard H.264/AVC decoder because order of Z-frames is based on encoding dependency of Z-frames. If the user node receives the Z-frames while the user node decodes received mis-predicted frames, Zaoral Decoding first decodes the mis-predicted frames, which are predictors of an anchor frame of Z-frames, and Zaoral Decoding decodes the mis-predicted frames and the Z-frames in parallel. After the decoding, the user node playbacks required camera's video.

## 6. Evaluation

### 6.1 Evaluation Setting

To evaluate the response delay of our proposal, we define video and network parameters in OMNeT++ [32]. Evaluation results were obtained using multi-view video test sequence "Exit" with resolution of 144×176. The test sequence is provided by MERL [33]. Encoder implemented by the modified open source project JMVC [34] is used to encode the multi-view video sequence. The number of cameras is 8, and the quantization parameter is 32. Each camera is encoded at a frame rate of 15 fps. The GOP is set to 8 frames. We assume a user switches cameras during 250 frames.

The evaluation assumes that one video encoder node is located under the sea, and one user node is located on the sea. We define that the speed of sound is 1,500 m/s. A data rate of acoustic networks is 90 kbps, and a frequency is 115 kHz [8].

We evaluate a response delay, traffic, and video quality of four schemes: naive-I, naive, proposal w/o ZE, and proposal.

**1)** Naive-I
Naive-I is the simplest method for multi-view video streaming over underwater acoustic networks. All video frames are encoded by H.264/AVC I-frames in order to maintain video quality. Naive-I is a baseline for performance without our proposal.

**2)** Naive
Naive is a simple method for multi-view video streaming over underwater acoustic networks. Video frames are encoded by one I-frame and seven B-frames in one GOP in order to reduce the traffic, and response delay. Naive is a baseline for performance without Zaoral Streaming.

**3)** Proposal w/o ZE
Proposal w/o ZE is based on our proposal as shown in Section 4. The proposal w/o ZE supports prediction in Zaoral Streaming. If a prediction at an encoder node is a failure, the encoder node encodes Z-frames without mis-predicted frames. We used Za-

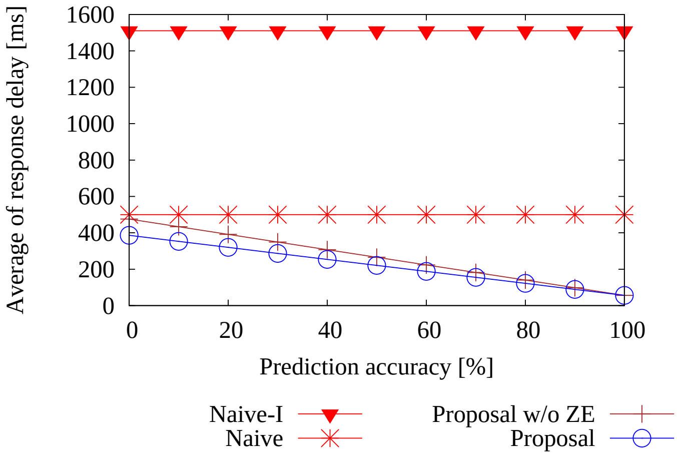


図 **9** Prediction accuracy v.s. Response delay.

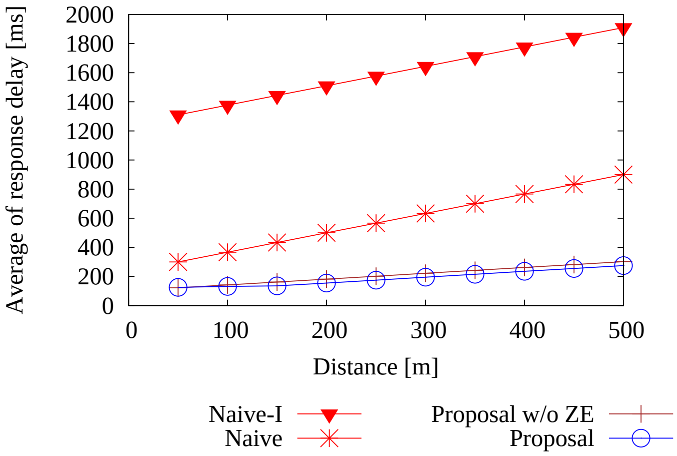


図 **10** Distance v.s. Response delay.

oral Streaming w/o ZE as a baseline for performance evaluation without Zaoral Encoding/Decoding.

4) Proposal
The proposal is our proposal as shown in Section 4 and Section 5. The proposal supports Slipped-TDMA and Zaoral Streaming in multi-view video streaming over underwater acoustic networks.

### 6.2 Response Delay

Figure 9 shows results of a response delay for different prediction accuracy when the communication distance between an encoder node and a user node is 200m. When the prediction accuracy is 100%, the response delay of our proposal is approximately 96.2% shorter than that of naive-I, and 88.8% lower than that of naive. The proposal greatly achieves a lower response delay than the naive methods because the encoder node predicts the next camera position for the user. When the prediction accuracy is 10%, the response delay of the proposal is approximately 76.6% shorter than that of naive-I, 29.4% shorter than that of naive, and 18.6% shorter than that of proposal w/o ZE. The proposal has a lower response delay than the naive methods and the proposal w/o ZE even if prediction accuracy becomes lower. The proposal encodes Z-frames with mis-predicted frames in order to prevent the increase of the response delay.

Figure 10 shows results of a response delay for different communication distances between an encoder node and a user node when prediction accuracy is 70%. Our proposal decreases the response delay by 90.4% compared to naive-I, and 64.0% compared to naive when the communication distance is 100m. The response delay of naive methods increases linearly as the communication distance increases. The encoder node transmits video packets after receiving a control packet from the user node. The response delay of the proposal is approximately 85.6% shorter than that of naive-I, 69.4% shorter than that of naive, and 8.9% shorter than that of proposal w/o ZE when the distance is 500m. With the increase of the communication distance, the encoder node transmits many predicted video packets before receiving the control packet. The proposal reduces video packet size by exploiting the predicted frames.

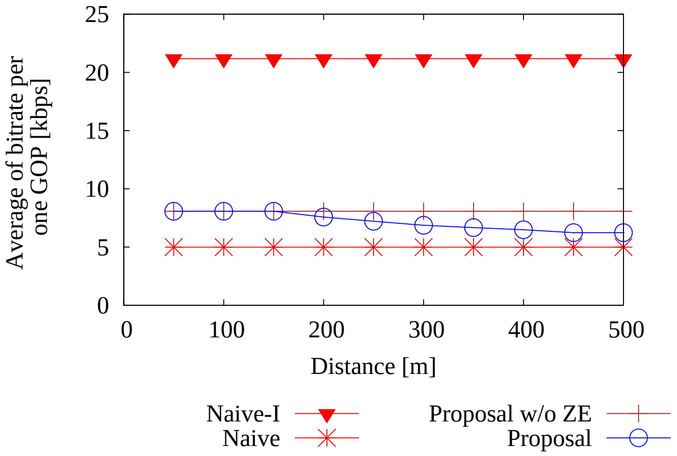


図 **11** Distance v.s. Traffic.

### 6.3 Traffic

Figure 11 shows traffic of each scheme for different communication distances between an encoder node and a user node. Even if the communication distance increases, the traffic of naive methods and proposal w/o ZE is constant. Naive-I encodes all video frames into I-frames in order to maintain the video quality. Proposal w/o ZE encodes Z-frames independently of mis-predicted frames. Naive has lower traffic than the proposal because the encoder node

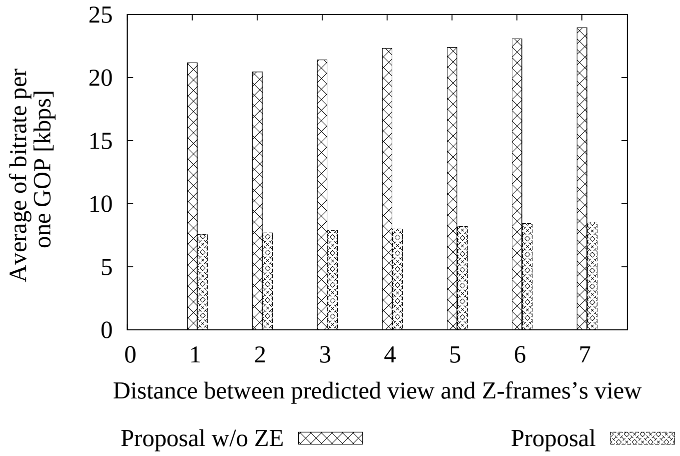


図 **12** Camera distance between mis-predicted frames and Z-frames v.s. Traffic.

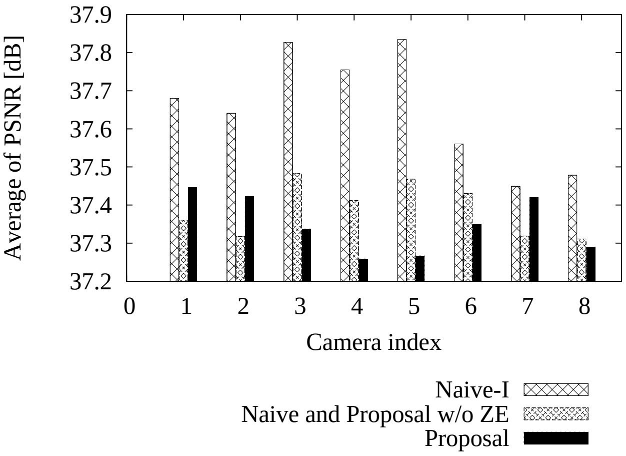


図 **13** Camera number v.s. Video quality.

transmits video frames once according to a received control packet.

When the distance is short until 150 m, the traffic of our proposal is the same as that of proposal w/o ZE. The encoder node does not have enough time to transmit predicted video frames to the user node before receiving a control packet. When the distance is 300m, the traffic of the proposal is approximately 68.5% lower than that of naive-I, and 16.1% lower than that of proposal w/o ZE. The proposal transmits many predicted video frames to the user node as the distance increases, and the proposal encodes Z-frames with mis-predicted frames.

Figure 12 shows traffic of each scheme for different camera distances between mis-predicted frames and Z-frames when the communication distance between an encoder node and a user node is 400m. We assume that the encoder node predicts the next camera position for the user is 1 before receiving a control packet. Our proposal reduces the traffic by 26.2% compared to proposal w/o ZE when the camera distance is 1, and the proposal reduces the traffic by 20.5% compared to proposal w/o ZE when the camera distance is 7. Even when the camera number of Z-frames is an opposite camera (i.e. camera 8), the proposal reduces the traffic. However, the traffic reduction between the proposal and the proposal w/o ZE decreases with the increase of the camera distance. The encoder node does not reduce the traffic much because camera 1 and 8 do not have similar information.

### 6.4 Video Quality

Figure 13 shows video quality of each scheme for different camera numbers. Video quality of our proposal is mean video quality of Z-frames, which are encoded with different mis-predicted video frames. The proposal degrade the video quality by 0.30 dB compared to naive-I, and 0.03 dB compared to naive and proposal w/o ZE on average. Naive-I has the highest video quality every camera number because Naive-I encodes all video frames into I-frames. Naive and proposal w/o ZE has the same video quality because the naive and the proposal w/o ZE encode video frames without mis-predicted frames. The proposal has lower video quality than naive-I. However, the proposal does not degrade the video quality very much.

## 7. Conclusion

This paper proposes Slipped-TDMA and Zaoral Streaming in order to achieve the reduction of video traffic, the suppression of the response delay, and maintaining video quality for multi-view video streaming over underwater acoustic networks. To exploit asymmetric traffic and a long propagation delay between nodes, Slipped-TDMA assigns time-slipped slots for an encoder node and a user node in order to improve band-utilization. Zaoral Streaming's design consists of two key components: prediction, and Zaoral Encoding/Decoding. The prediction reduces the response delay by predicting the next camera position for the user. Zaoral Encoding prevents the increase of the response delay and traffic by encoding Z-frames with mis-predicted video frames when the previous prediction is a failure. The evaluation shows that our proposal achieves low traffic, a short response delay, and small degradation of video quality, as compared to naive methods and proposal w/o Zaoral Encoding. When the distance between the encoder node and the user node is 500m, the response delay of the proposal is approximately 85.6% shorter than that of naive-I, 69.4% shorter than that of naive, and 8.9%

shorter than that of proposal w/o Zaoral Encoding.

## Acknowledgment

This study was partially supported by research grants from Suzuki Foundation and The Telecommunications Advancement Foundation.

### 参考文献

[1] Michiya, S., Takashi, S. and Toshio, T.: Digital Acoustic Image Transmission System For Deep-sea Research Submersible, *IEEE OCEANS*, pp. 567–570 (1992).

[2] Akyildiz, I. F., Pompili, D. and Melodia, T.: State-of-the-art In Protocol Research For Underwater Acoustic Sensor Networks, *ACM WUWNet*, pp. 7–16 (2006).

[3] Pompili, D. and Akyildiz, I. F.: A Cross-layer Communication Solution For Multimedia Applications In Underwater Acoustic Sensor Networks, *IEEE MASS*, pp. 275–284 (2008).

[4] Shiau, Y.-H., Lin, S.-I., Lin, F.-P. and Chen, C.-C.: Real-Time Fish Obesrvation And Fish Category Database Construction, *International Journal Of Advanced Computer Science And Applications*, Vol. 3, No. 4, pp. 45–49 (2012).

[5] Mineo, O., Susumu, M. and Takao, S.: Fundamental Study To Estimate Fish Biomass Around Coral Reef Using 3-dimensional Underwater Video System, *IEEE OCEANS*, pp. 1389–1392 (2000).

[6] Yavuz, K. and Ilkcay, U.: 3D Reconstruction Of Underwater Scenes From Uncalibrated Video Sequences, *IEEE SIU*, pp. 105–108 (2009).

[7] Li, Q.-Z., Liu, J.-X., Zang, A.-Y., Wang, Z.-Q. and Wang, W.-J.: A DM642-Based Underwater Video Coding System, *INC, IMS and IDC*, pp. 1567–1572 (2009).

[8] Jordi, R., Sura, D. and Stojanovic, M.: Underwater Wireless Video Transmission For Supervisory Control And Inspection Using Acoustic OFDM, *IEEE OCEANS*, pp. 1–9 (2010).

[9] Pelekanakis, C., Stojanovic, M. and Freitag, L.: High Rate Acoustic Link For Underwater Video Transmission, *IEEE OCEANS*, pp. 1091–1097 (2003).

[10] Vall, L. D., Sura, D. and Stojanovic, M.: Towards Underwater Video Transmission, *ACM WUWNet*, pp. 1–5 (2011).

[11] Tanimoto, M.: Overview Of Free Viewpoint Television, *Signal Processing: Image Communication*, Vol. 21, No. 6, pp. 454–461 (2006).

[12] Kimata, H., Kitahara, M., Kamikura, K., Yashima, Y., Fujii, T. and Tanimoto, M.: Low-delay Multiview Video Coding For Free-viewpoint Video Communication, *Systems And Computers In Japan*, Vol. 38, No. 5, pp. 14–29 (2007).

[13] Wilburn, B., Smulski, M., Lee, K. and Horowitz, M.: Light Field Video Camera, *Media Processors*, pp. 29–36 (2002).

[14] Do, L., Zinger, S. and Peter, D. W.: Conversion Of Free-viewpoint 3D Multi-view Video For Stereoscopic Displays, *IEEE ICME*, pp. 1730–1734 (2010).

[15] Joint Video Team ISO/IEC JTC1/SC29/WG11 MPEG2005/N7567: *Updated Call For Proposals On Multi-view Video Coding* (2005).

[16] Text Of ISO/IEC 14496-10:2008/FDAM 1 ISO/IEC JTC1/SC29/WG11: *Multiview Video Coding* (2008).

[17] Muller, K., Merkle, P., Schwarz, H., Hinz, T., Smolic, A. and Wiegand, T.: Multi-view Video Coding Based On H.264/AVC Using Hierarchical B-frames, *IEEE PCS*, pp. 385–390 (2006).

[18] Vetro, A., Pandit, P., Kimata, H., Smolic, A. and Wang, Y. K.: *Joint Draft 8.0 On Multi-view Video Coding* (2008).

[19] Sun, M.-T., Huang, L., Arora, A. and Lai, T.-H.: Reliable MAC Layer Multicast In IEEE 802.11 Wireless Networks, *International Conference On Parallel Processing*, pp. 527–536 (2002).

[20] Kimura, Y. and Yamauchi, I.: A Layer 2 Multicast With Improved Reliability Over The TDMA Cellular Radio Networks, *IFIP International Conference on Personal Wireless Communications*, pp. 24–28 (1999).

[21] Partan, J., Kurose, J. and Levine, B. N.: A Survey Of Practical Issues In Underwater Networks, *ACM SIGMOBILE Mobile Computing And Communications Review*, Vol. 11, No. 4, pp. 23–33 (2007).

[22] Abramson, N.: The Aloha System - Another Alternative For Computer Communications, *AFIPS*, pp. 281–285 (1970).

[23] Roberts, L. G.: Aloha Packet System With And Without Slots And Capture, *ACM SIGCOMM Computer Communication Review*, Vol. 5, No. 2, pp. 28–42 (1975).

[24] Karn, P.: MACA-A New Channel Access Method For Packet Radio, *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pp. 134–140 (1990).

[25] Bharghavan, V., Demers, A., Schenker, S. and Zhang, L.: MACAW: A Media Access Protocol for Wireless LAN's, *ACM SIGCOMM*, pp. 212–225 (1994).

[26] Fullmer, C. L. and Garcia-Luna-Aceves, J. J.: Floor Acquisition Multiple Access (FAMA) For Packet-Radio Networks, *ACM SIGCOMM*, pp. 262–273 (1995).

[27] Syed, A. A., Ye, W., Heidemann, J. and Krishnamachari, B.: Understanding Spatio-temporal Uncertainty In Medium Access With ALOHA Protocols, *ACM WUWNet*, pp. 41–48 (2007).

[28] Welch, G. and Bishop, G.: *An Introduction To The Kalman Filter* (2006).

[29] Kalman, R. E.: A New Approach to Linear Filtering and Prediction Problems, *Transaction Of The ASME Journal Of Basic Engineering*, No. 82, pp. 35–45 (1960).

[30] N. Gordon, D. S. and Smith, A.: Novel Approach To Nonlinear And Non-gaussian Bayesian State Estimation, *Iee Proceedings F Radar and Signal Processing*, Vol. 140, No. 2, pp. 107–113 (1993).

[31] Sanjeev, A., Maskell, S., Gordon, N. and Clapp, T.: A Tutorial On Particle Filters For Online Nonlinear/Non-Gaussian Bayesian Tracking, *IEEE Transactions On Signal Processing*, Vol. 50, No. 2, pp. 174–188 (2002).

[32] Varga, A.: *The OMNET++ Discrete Event Simulation System* (2006).

[33] ISO/IEC JTC1/SC29/WG11: *Multiview Video Test Sequences from MERL* (2005).

[34] Joint Video Team Of ITU-T VCEG And ISO/IEC MPEG: *JMVC (Joint Multiview Video Coding) Software* (2008).