Carnegie Mellon University

JackTrip

A efficient solution for remote audio transmission

Dalong Cheng<dalongc@andrew.cmu.edu>

April 5, 2013
Presentation Outline

1. Keyword, Concept Review
2. Challenges for Audio Transmission
3. JackTrip’s Solution
   ▶ Motivation
   ▶ Architecture
   ▶ Buffer Design
   ▶ Redundancy Design
4. A JackTrip’s Extension
   ▶ Motivation
   ▶ Evaluation Metrics
   ▶ Design
   ▶ Implementation
UDP

- Single socket to receive messages
- No guarantee of delivery
- Not necessarily in-order delivery
- Datagram – independent packets
- Must address each packet

Postal Mail

- Single mailbox to receive letters
- Unreliable 😞
- Not necessarily in-order delivery
- Letters sent independently
- Must address each reply

Example UDP applications
Multimedia, voice over IP
**TCP**

- Reliable – guarantee delivery
- Byte stream – in-order delivery
- Connection-oriented – single socket per connection
- Setup connection followed by data transfer

**Telephone Call**

- Guaranteed delivery
- In-order delivery
- Connection-oriented
- Setup connection followed by conversation

**Example TCP applications**

Web, Email, Telnet
LAN & WAN

Figure: LAN & WAN
Process & Thread

Figure: Process Model
Process & Thread

Stack ↓ ↓ ↓ ↓ Heap
  Stack
  Stack
  Data
  Code

Registers

Figure: Process with Multi-thread
Scheduling & Block & Non-Block

Figure: Architecture of HCMP
Classic Multi-Thread Program Model

Figure: Single-Producer & Consumer Model

- More than one consumer thread (Thread Pool)
- What if more than one producer and consumer
Distributed Flavor

Figure: Architecture of HCMP

- A single node to receive all request
- Add a “dispatcher” to assign work
“Universal” Distributed System Design Principle

1. Add another layer of indirection & abstraction
2. Add cache & buffer
Problem Definition

1. WAN, Bi-directional, Audio, Transmission
2. Similar Problems (Not Same!)
   ▶ LAN Area Studio System
   ▶ Uni-directional WAN Streaming
Objective

1. Latency Critical (25ms is upper bound!)
2. Maximize Audio Quality (Tradeoff)

Possible Solution

1. Compressed Audio
2. One-way recording techniques
Audio Latency

1. Acoustic Delay
2. Analogue to Digital, Digital to Analogue Delay
3. Transmission Delay
4. Settings Chosen for Audio Quality
   (Etc. Sample Rate, Buffer, Packet Size)
   - 96kHz Sample Rate, Audio Buffer 64 samples, deliver
     \[ \frac{64}{96000} \times 1000 = 0.67\text{ms} \]
Basic Approach

- Transfer Uncompressed Audio (require large bandwidth)
- Use UDP instead of TCP
  - TCP, Retransmission, Fix-Order, Ack-Scheme
  - Have to deal with UDP lost packet :(
- Ring buffer
Architecture - Sender

- Block if empty
- Ring buffer for share

Figure: Sender
Architecture - Receiver

- Never block
- Ring buffer for share

Figure: Receiver
Figure: Architecture

Fig. 1. JackTrip architecture overview.
Ring Buffer Analysis

- Ring buffer is to Inter-thread Communication
- Initialization
- What is ideal situation, What is two possible consequence
Consequence 1 - Underflow

- $V_{\text{read}} > V_{\text{write}}$
- Reason: packet lost or network jitter
- Solution: send slicent or last received packet

**Figure:** Ring Buffer Underflow
Consequence 2 - Overflow

- $V_{write} > V_{read}$
- Reason: clock drift, peer’s clock faster than local clock
- Solution: reset read position (drop samples)

**Figure**: Ring Buffer Overflow
UDP’s Price - Redundancy

Figure: Redundancy
A Interesting Extension

Think of Such Scenario

- You have a musician friend
- You build a tool for him to use
- He needs to tune parameters every time using the tool
- He doesn't understand those parameters
Evaluation Metric

Figure: Karplus - Strong string model

- Pinging the network acoustically
- Ipsilateral be server side
- Contralateral be client side
Server Architecture 1

Figure: Server Architecture 1
Server Architecture 2

**Figure:** Server Architecture 2
Summary

- Multi-thread model in audio
- Ring buffer
- Use ear to facilitate tuning parameters