

# Project 1

## RTP Project

PCS 2013

TA: Cheng-Ting Chang

Email: [qqting@wmnet.cs.nthu.edu.tw](mailto:qqting@wmnet.cs.nthu.edu.tw)

# Introduction

- \* RTP: A Transport Protocol for Real-Time Application
  - \* RTP : Real-time Transport Protocol
  - \* First published in 1996 as RFC 1889, superseded by RFC 3550 in 2003.
- \* RTP over UDP
  - \* A sequence number
  - \* A time stamp for synchronized play-out

# Project

- \* Goal

- \* Realize how to send voice packet over RTP.
- \* Recognize the format of RTP.
- \* Learn how to use Wireshark.

- \* Task

- \* Trace “PCS\_RTP\_Project”
- \* Modify code
- \* Use Wireshark to capture RTP/RTCP packets

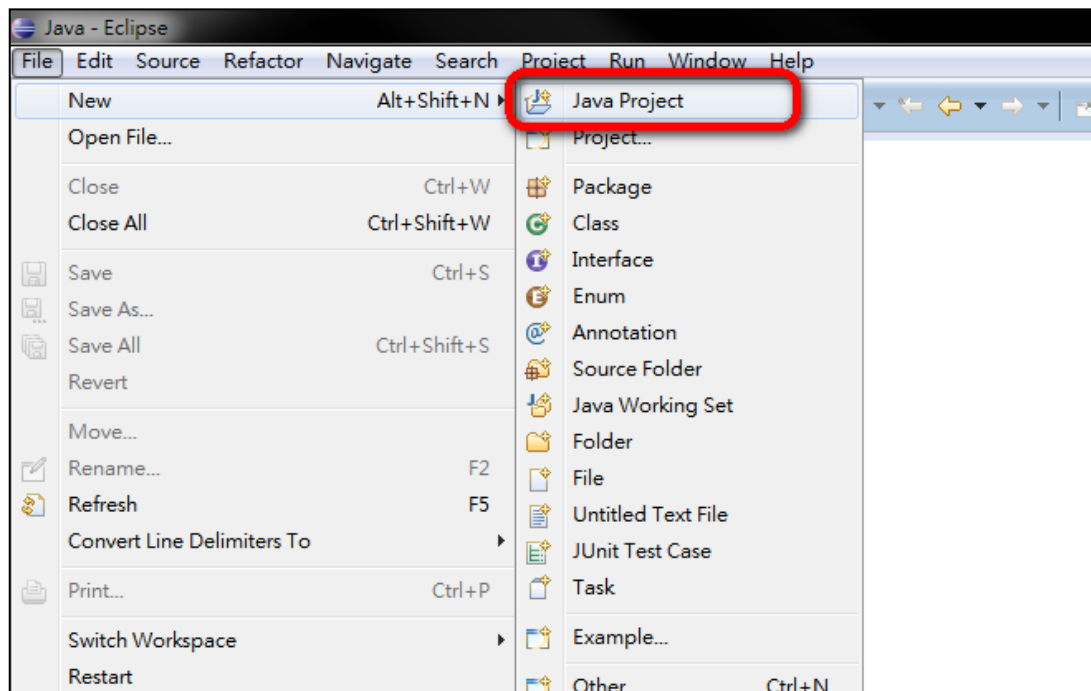
# Experiment Steps

- 1) Check your PC meets the following conditions:
  - ✓ Equipped with a speaker or headphone (喇叭或耳機)
  - ✓ Equipped with a microphone
  - ✓ Has network connection
- 2) Download PCS\_RTP\_Project.zip
  - \* If your PC has no Java development environment, you have to download and install JDK and Eclipse.

# Experiment Steps

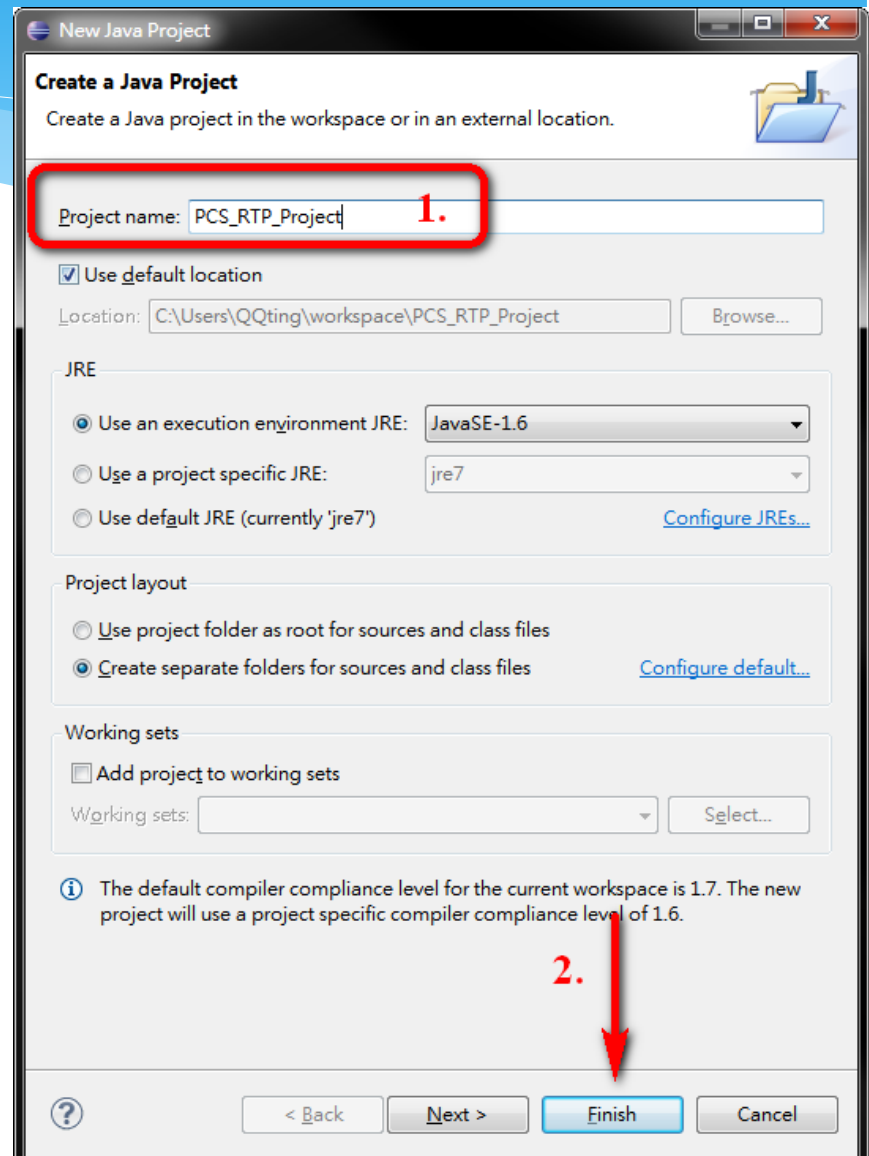
## 3) Create a new project

### a. Use Eclipse to establish a new Java project



# Experiment Steps

- 3) Create a new project
  - b. Enter the project name, and click “Finish”.



# Experiment Steps

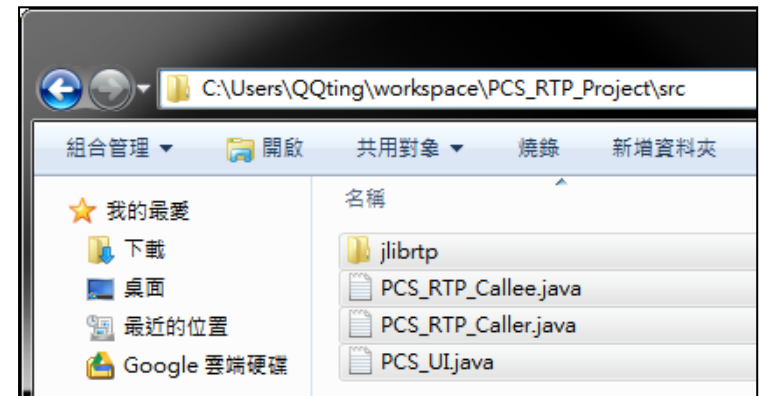
## 3) Create a new project

c. Find the project location, put the following source codes into “src” directory:

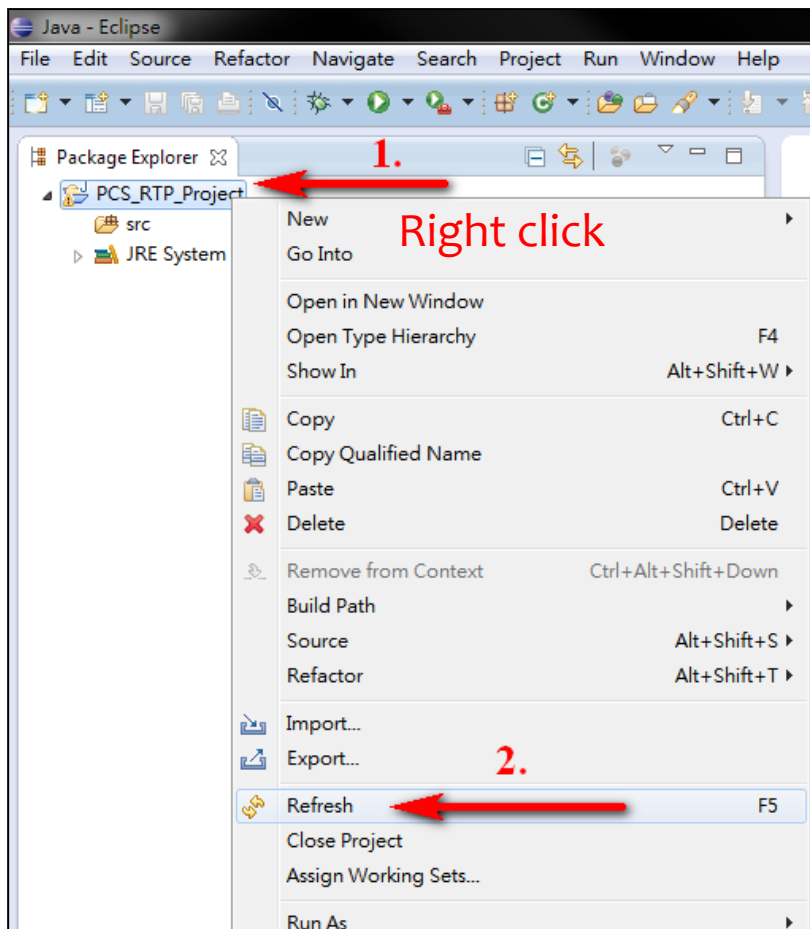
- ✓ The whole jlibrtp directory
- ✓ PCS\_RTP\_Callee.java
- ✓ PCS\_RTP\_Caller.java
- ✓ PCS\_UI.java

default location is:

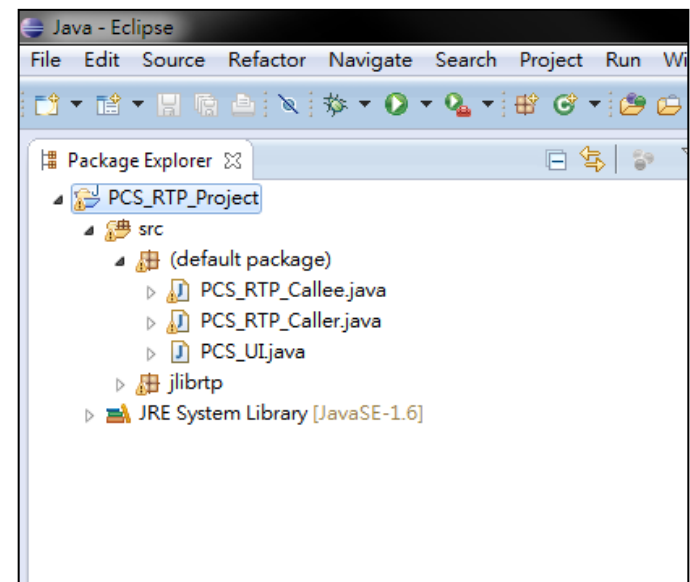
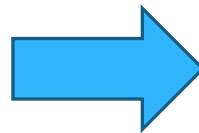
C:\Users\NAME\workspace\PCS\_RTP\_Project\src



# Experiment Steps



- 3) Create a new project
- d. Refresh your project.





# Experiment Steps

- 4) Execute this project
  - a. Before you modify the codes, this project can be built and executed successfully.

# Run Java Application (1/2)

The image shows the Eclipse IDE interface with the Package Explorer on the left. A right-click context menu is open over the 'PCS RTP Project' package. A red arrow labeled '1.' points to the right-click action, and the text 'Right click' is written in red. A red arrow labeled '2.' points to the 'Run As' option at the bottom of the menu. A blue arrow points from the 'Run As' menu item to a 'Select Java Application' dialog box. In this dialog, two items are listed under 'Matching items': 'PCS RTP\_Callee - (default package)' and 'PCS RTP\_Caller - (default package)'. Red arrows labeled '4.' point to these two items, with the word 'or' written in red between them. At the bottom of the dialog, the 'OK' button is circled in red, with a red arrow labeled '5.' pointing to it. In the background, a submenu is visible with '2 Java Application' highlighted and circled in red, with a red arrow labeled '3.' pointing to it.

1. Right click

2. Run As

3. 2 Java Application

4. PCS RTP\_Callee - (default package) or PCS RTP\_Caller - (default package)

5. OK

# Run Java Application (2/2)

- \* This program is successfully built if the following window appears.

The image shows two side-by-side Java application windows. The left window is titled 'This is Caller!' and the right window is titled 'This is Callee!'. Both windows contain a table with five rows of input fields. The first row is 'Remote IP' with the value '0.0.0.0'. The next three rows are 'Remote RTP port', 'Remote RTCP port', 'Local RTP port', and 'Local RTCP port', all with the value '0'. At the bottom of each window is a button: 'Dial' for the caller and 'Answer' for the callee.

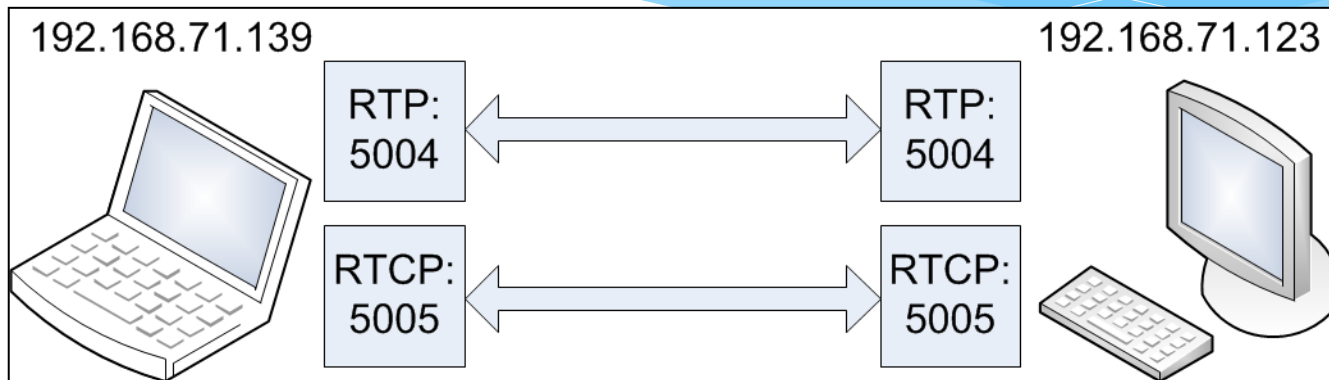
Field	Value
Remote IP	0.0.0.0
Remote RTP port	0
Remote RTCP port	0
Local RTP port	0
Local RTCP port	0
Dial	

Field	Value
Remote IP	0.0.0.0
Remote RTP port	0
Remote RTCP port	0
Local RTP port	0
Local RTCP port	0
Answer	

# Experiment Steps

- 5) Choose one of the following way to test this project
  - a. Use 2 different PCs to build and run the program, one of them runs the caller, and another runs the callee.
  - b. You can export your project into a runnable JAR file, and put this JAR file in another PC to run. (Note: The another PC must have Java Runtime Environment, JRE)
  - c. Run both caller and callee in the same PC.

# Two Different PCs to test



Two software windows are shown side-by-side. The left window is titled 'This is Caller!' and the right window is titled 'This is Callee!'. Both windows have a table-like interface for configuring ports and a button at the bottom.

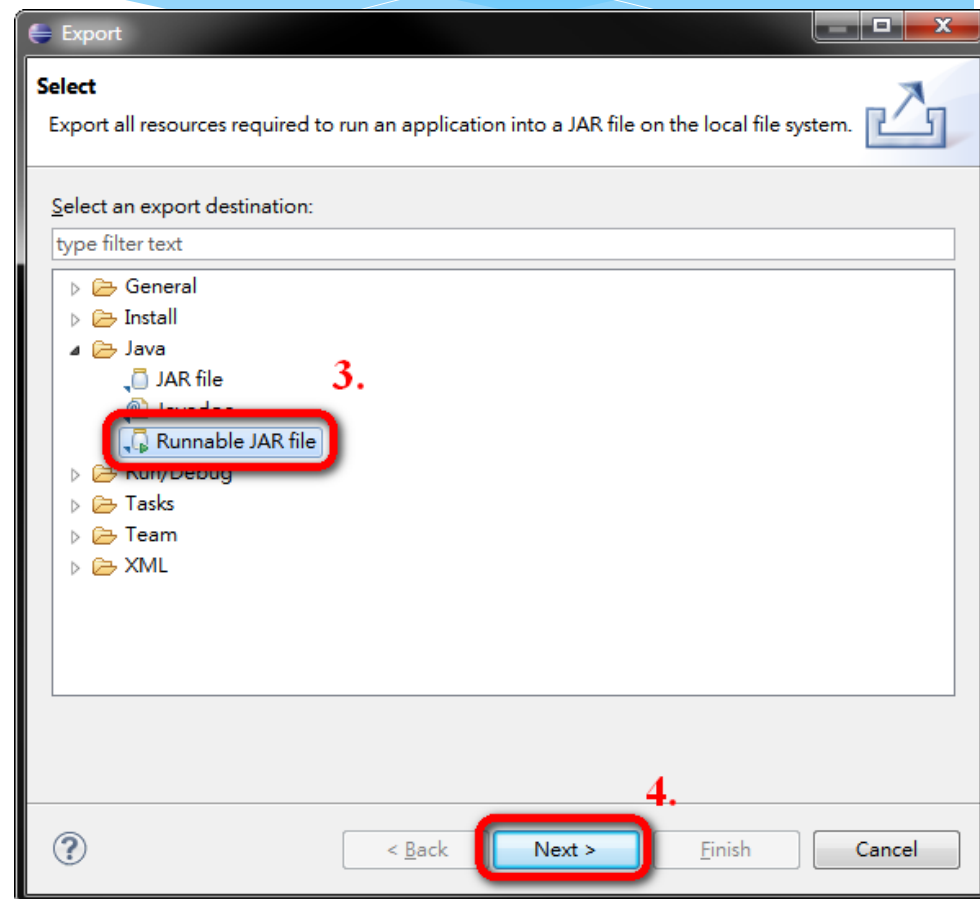
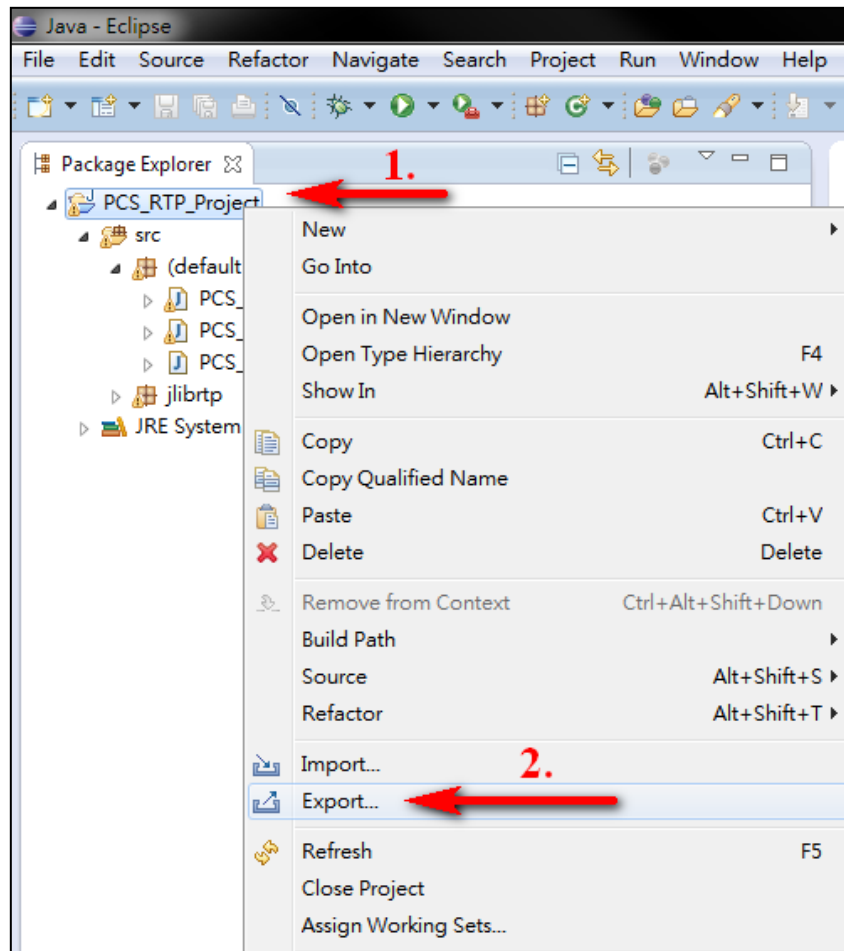
Field	Value
Remote IP	192.168.71.123
Remote RTP port	5004
Remote RTCP port	5005
Local RTP port	5004
Local RTCP port	5005

Dial

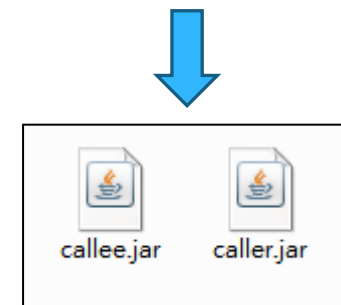
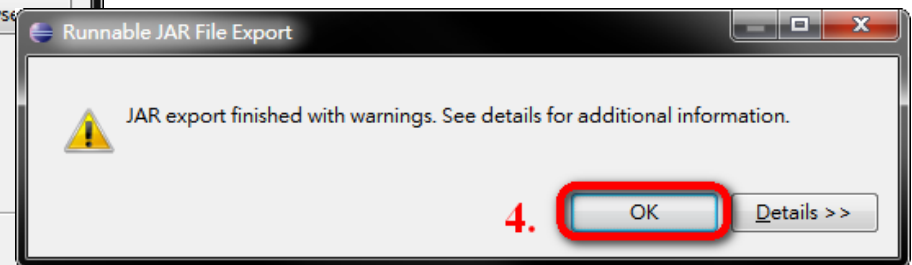
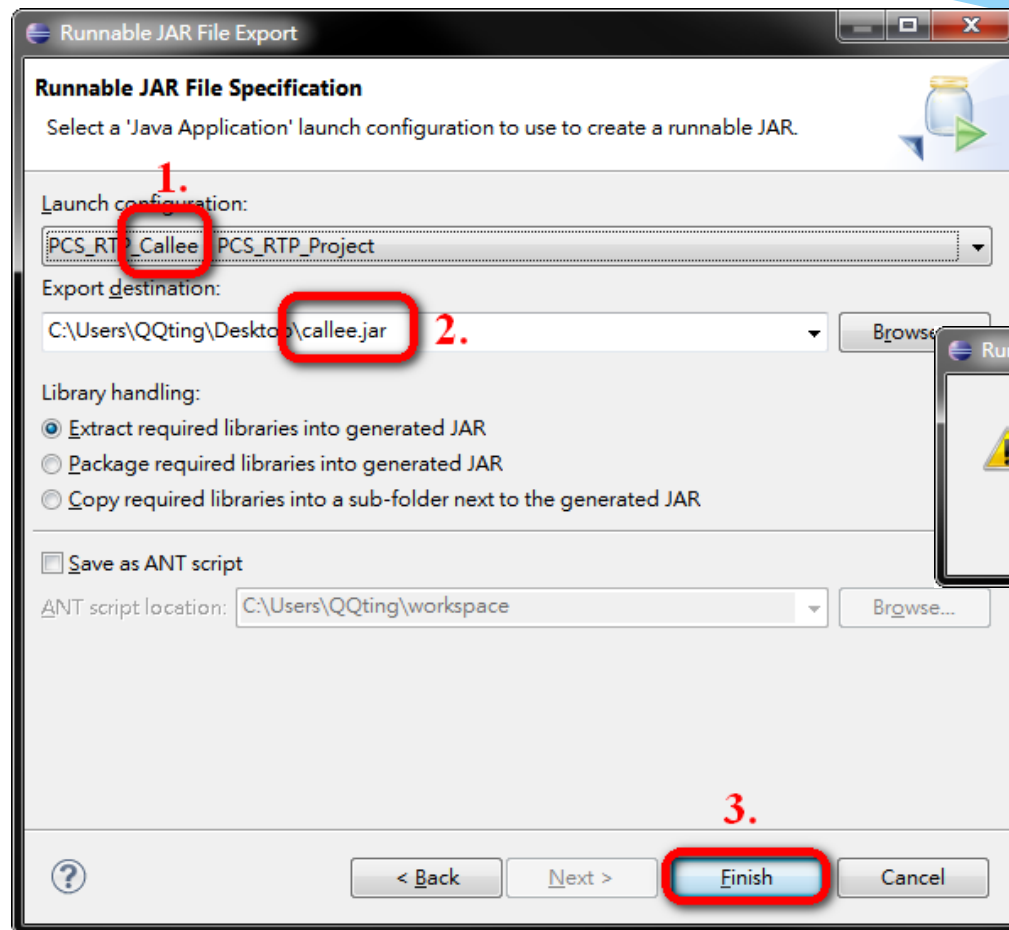
Field	Value
Remote IP	192.168.71.139
Remote RTP port	5004
Remote RTCP port	5005
Local RTP port	5004
Local RTCP port	5005

Answer

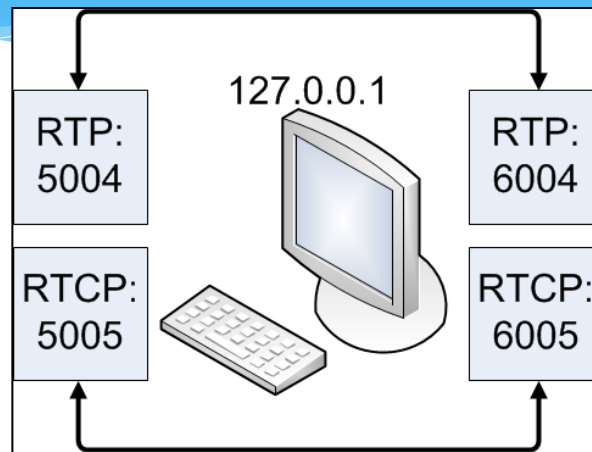
# Export Java Application (1/2)



# Export Java Application (2/2)



# Caller and Callee in one PC



This is Caller!	
Remote IP	127.0.0.1
Remote RTP port	5004
Remote RTCP port	5005
Local RTP port	6004
Local RTCP port	6005
<input type="button" value="Dial"/>	

This is Callee!	
Remote IP	127.0.0.1
Remote RTP port	6004
Remote RTCP port	6005
Local RTP port	5004
Local RTCP port	5005
<input type="button" value="Answer"/>	



# Experiment Steps

## 6) Test this project

- a. Assign appropriate IP and RTP/RTCP port to the running program.
- b. Click the “Dial” button of the caller and the “Answer” button of the callee
- c. Before you modify the codes, the callee should hear the voice from the caller. (but Caller can’t hear Callee)

# Experiment Steps

## 7) Complete this project

- a. Trace PCS\_RTP\_Caller.java and PCS\_RTP\_Callee.java, and then modify the codes to make Caller and Callee can talk to each other.
- b. Hints:
  - ✓ Trace **main function** first, and then trace the functions used in the main function.
  - ✓ Trace codes in PCS\_RTP\_Caller.java for recording voice and PCS\_RTP\_Callee.java for playing voice.

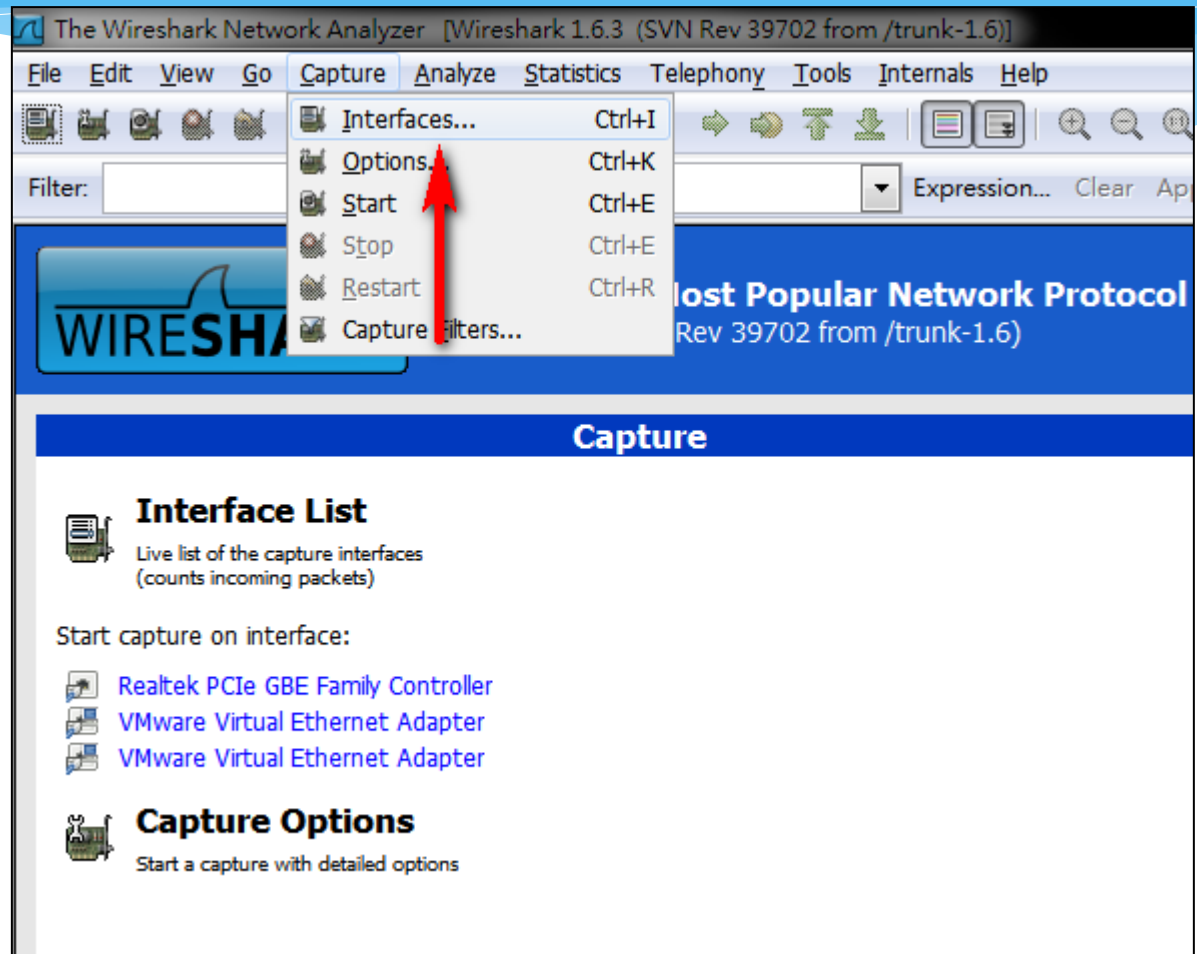
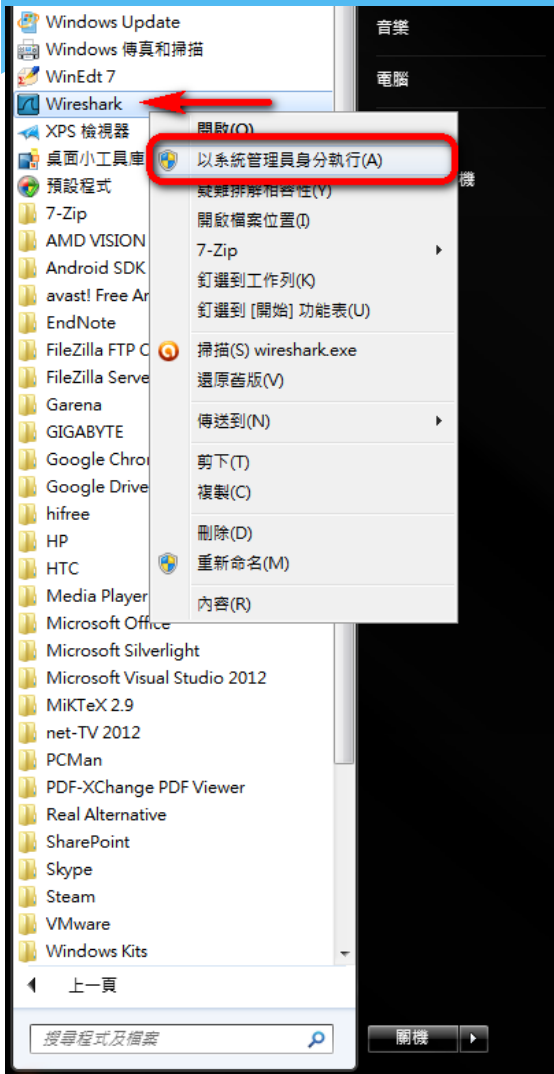
# Experiment Steps

- 8) Use Wireshark to analyze RTP and RTCP packets
  - a. Download and install Wireshark.
  - b. Capture the UDP packets whose address includes the remote PC. (`ip.addr == REMOTE_IP_ADDRESS`)
  - c. Decode the UDP packets on port 5004 as RTP.
  - d. Decode the UDP packets on port 5005 as RTCP.

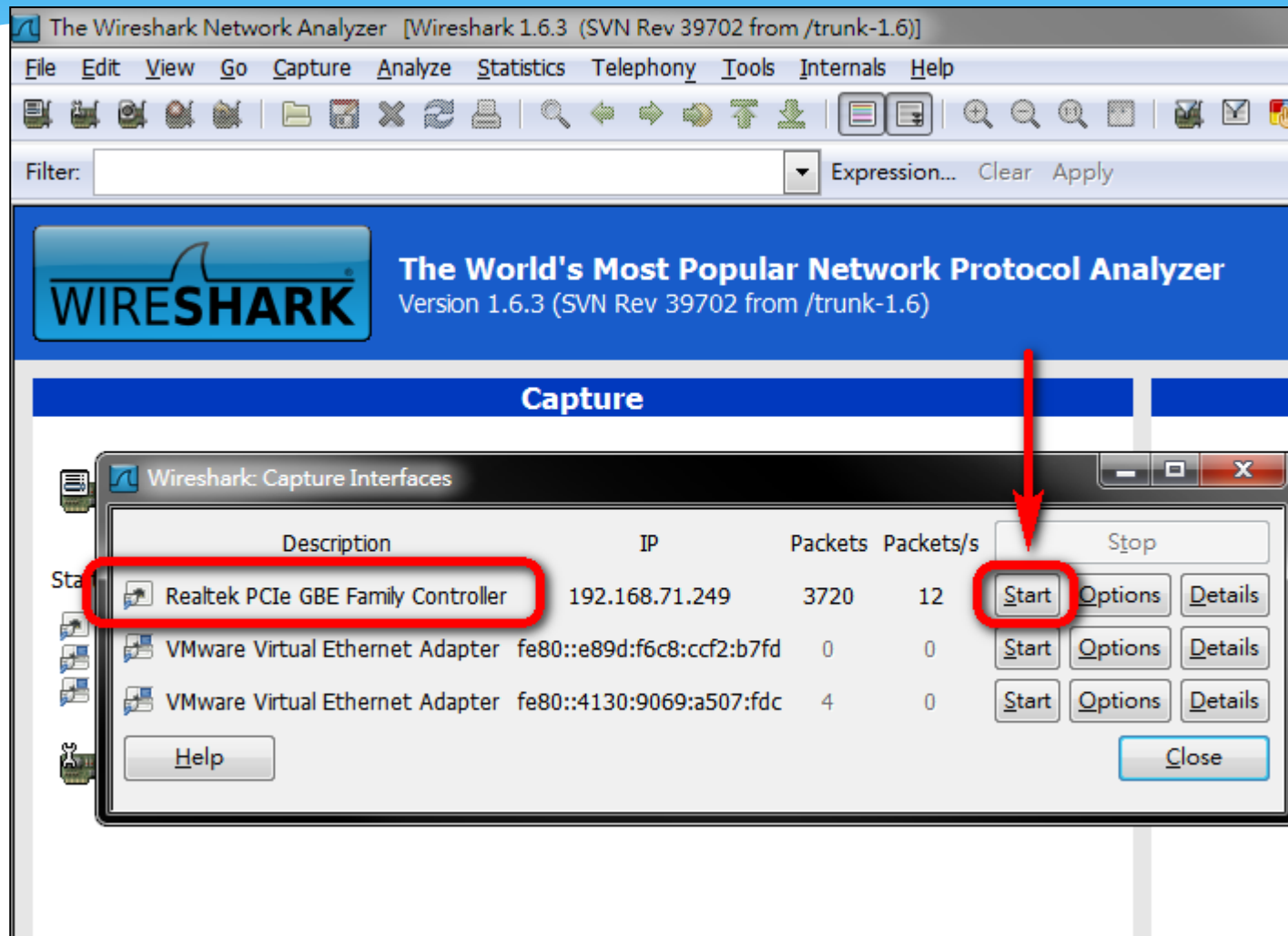
## Note:

- a. `ip.addr` can't be `127.0.0.1` because Wireshark can't capture the packets from this address.
- b. 5004 and 5005 are determined by your input of program.

# Wireshark – Getting Started



# Wireshark - Capture Device



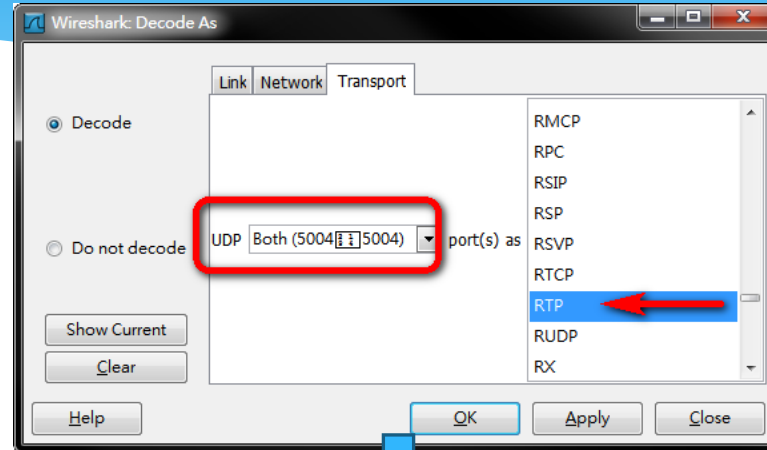
# Wireshark – Decode RTP/RTCP

The image shows the Wireshark 1.8.2 interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Tools, Internals, and Help. The filter bar shows 'ip.addr == 140.114.71.200' with a red arrow pointing to it labeled '1. remote IP'. Below the filter bar is a table of captured packets. The table has columns: No., Time, Source, Destination, Protocol, Length, and Info. The packets are all UDP from 140.114.71.200 to 140.114.71.139. A red arrow points to the '140.114.71.139' destination IP in the first packet row, labeled '3. right click'. A right-click context menu is open over the first packet. The menu items are: Mark Packet (toggle), Ignore Packet (toggle), Set Time Reference (toggle), Time Shift..., Edit or Add Packet Comment..., Manually Resolve Address, Apply as Filter, Prepare a Filter, Conversation Filter, Colorize Conversation, SCTP, Follow TCP Stream, Follow UDP Stream, Follow SSL Stream, Copy, Decode As..., Print..., and Show Packet in New Window. A red arrow points to the 'Decode As...' option, labeled '4. decode'. The packet details pane on the left shows the structure of the first packet (Frame 623): Ethernet II, Internet Protocol Version 4, User Datagram Protocol, and Data (1036 bytes). A red box highlights the 'Destination port: avt-profile-1 (5004)' in the UDP section, with a red arrow pointing to it labeled '2. check port'.

No.	Time	Source	Destination	Protocol	Length	Info
623	26.5441020	140.114.71.200	140.114.71.139	UDP		
624	26.5451070	140.114.71.200	140.114.71.139	UDP		
625	26.5461040	140.114.71.200	140.114.71.139	UDP		
626	26.5470960	140.114.71.200	140.114.71.139	UDP		
627	26.5471010	140.114.71.200	140.114.71.139	UDP		
628	26.5480950	140.114.71.200	140.114.71.139	UDP		
629	26.5491010	140.114.71.200	140.114.71.139	UDP		
635	26.6747260	140.114.71.200	140.114.71.139	UDP		
636	26.6755000	140.114.71.200	140.114.71.139	UDP		

Frame 623: 1078 bytes on wire (8624 bits), 1078 bytes captured (8624 bits) on interface 0  
Ethernet II, Src: Giga-Byt\_55:ec:10 (50:e5:49:55:ec:10), Dst: 08:00:27:00:00:00 (8:00:27:00:00:00)  
Internet Protocol Version 4, Src: 140.114.71.200 (140.114.71.200), Dst: 140.114.71.139 (140.114.71.139)  
User Datagram Protocol, Src Port: avt-profile-1 (5004), Dst Port: avt-profile-1 (5004)  
Source port: avt-profile-1 (5004)  
Destination port: avt-profile-1 (5004)  
Length: 1044  
Checksum: 0x8cbe [validation disabled]  
Data (1036 bytes)

# Wireshark – Decode RTP



No.	Time	Source	Destination	Protocol	Length	Info
623	26.5441020	140.114.71.200	140.114.71.139	RTP	1078	PT=ITU-T G.711 PCMU, SSRC=0x23068FB8, Seq=14734, Time=2171651078
624	26.5451070	140.114.71.200	140.114.71.139	RTP	1078	PT=ITU-T G.711 PCMU, SSRC=0x23068FB8, Seq=14735, Time=2171651085
625	26.5461040	140.114.71.200	140.114.71.139	RTP	1078	PT=ITU-T G.711 PCMU, SSRC=0x23068FB8, Seq=14736, Time=2171651086
626	26.5470960	140.114.71.200	140.114.71.139	RTP	1078	PT=ITU-T G.711 PCMU, SSRC=0x23068FB8, Seq=14737, Time=2171651086
627	26.5471010	140.114.71.200	140.114.71.139	RTP	1078	PT=ITU-T G.711 PCMU, SSRC=0x23068FB8, Seq=14738, Time=2171651087
628	26.5480950	140.114.71.200	140.114.71.139	RTP	1078	PT=ITU-T G.711 PCMU, SSRC=0x23068FB8, Seq=14739, Time=2171651088
629	26.5491010	140.114.71.200	140.114.71.139	RTP	1078	PT=ITU-T G.711 PCMU, SSRC=0x23068FB8, Seq=14740, Time=2171651089
635	26.6747260	140.114.71.200	140.114.71.139	RTP	1078	PT=ITU-T G.711 PCMU, SSRC=0x23068FB8, Seq=14741, Time=2171651215
636	26.6755900	140.114.71.200	140.114.71.139	RTP	1078	PT=ITU-T G.711 PCMU, SSRC=0x23068FB8, Seq=14742, Time=2171651215

Frame 623: 1078 bytes on wire (8624 bits), 1078 bytes captured (8624 bits) on interface 0

Ethernet II, Src: Giga-Byt\_55:ec:10 (50:e5:49:55:ec:10), Dst: Giga-Byt\_55:ec:22 (50:e5:49:55:ec:22)

Internet Protocol Version 4, Src: 140.114.71.200 (140.114.71.200), Dst: 140.114.71.139 (140.114.71.139)

User Datagram Protocol, Src Port: avt-profile-1 (5004), Dst Port: avt-profile-1 (5004)

Real-Time Transport Protocol

- 10... = Version: RFC 1889 Version (2)
- ..0. = Padding: False
- ...0 = Extension: False
- .... 0000 = Contributing source identifiers count: 0
- 0... = Marker: False
- Payload type: ITU-T G.711 PCMU (0)
- Sequence number: 14734
- Timestamp: 2171651078
- Synchronization Source identifier: 0x23068fb8 (587632568)
- Payload: 26ff2fff23ff33ff57ff49ff37ff54ff37ff27ff44ff26ff...

# Wireshark – Decode RTPC

Capturing from Realtek PCIe GBE Family Controller: \Device\NPF\_{569FF1D2-CCA2-4EB5-BE29-6C8D0C54C60F} [Wireshark 1.8.6 (SVN Rev 48142)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: `ip.addr == 140.114.71.200` Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
759	28.0608050	140.114.71.200	140.114.71.139	RTP	1078	PT=ITU-T G.711 PCMU,
760	28.0615670	140.114.71.200	140.114.71.139	RTP	1078	PT=ITU-T G.711 PCMU,
768	28.1194870	140.114.71.139	140.114.71.200	RTP	1078	PT=ITU-T G.711 PCMU,
769	28.1204070	140.114.71.139	140.114.71.200	RTP	1078	PT=ITU-T G.711 PCMU,
772	28.1462660	140.114.71.200	140.114.71.139	UDP	94	Sender Report Source c
776	28.1868040	140.114.71.200	140.114.71.139	RTP	1078	PT=ITU-T G.711 PCMU,
777	28.1875970	140.114.71.200	140.114.71.139	RTP	1078	PT=ITU-T G.711 PCMU,
778	28.2455600	140.114.71.139	140.114.71.200	RTP	1078	PT=ITU-T G.711 PCMU,

Frame 772: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface 0  
Ethernet II, Src: Giga-Byt\_55:ec:10 (50:e5:49:55:ec:10), Dst: Giga-Byt\_55:ec:22 (50:e5:49:55:ec:22)  
Internet Protocol Version 4, Src: 140.114.71.200 (140.114.71.200), Dst: 140.114.71.139 (140.114.71.139)  
User Datagram Protocol, Src Port: avt-profile-2 (5005), Dst Port: avt-profile-2 (5005)  
Source port: avt-profile-2 (5005)  
Destination port: avt-profile-2 (5005)  
Length: 60  
Checksum: 0xee92 [validation disabled]  
Data (52 bytes)

Capturing from Realtek PCIe GBE Family Controller: \Device\NPF\_{569FF1D2-CCA2-4EB5-BE29-6C8D0C54C60F} [Wireshark 1.8.6 (SVN Rev 48142)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: `ip.addr == 140.114.71.200` Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
759	28.0608050	140.114.71.200	140.114.71.139	RTP	1078	PT=ITU-T G.711 PCMU, SSR
760	28.0615670	140.114.71.200	140.114.71.139	RTP	1078	PT=ITU-T G.711 PCMU, SSR
768	28.1194870	140.114.71.139	140.114.71.200	RTP	1078	PT=ITU-T G.711 PCMU, SSR
769	28.1204070	140.114.71.139	140.114.71.200	RTP	1078	PT=ITU-T G.711 PCMU, SSR
772	28.1462660	140.114.71.200	140.114.71.139	RTCP	94	Sender Report Source c
776	28.1868040	140.114.71.200	140.114.71.139	RTP	1078	PT=ITU-T G.711 PCMU, SSR
777	28.1875970	140.114.71.200	140.114.71.139	RTP	1078	PT=ITU-T G.711 PCMU, SSR
778	28.2455600	140.114.71.139	140.114.71.200	RTP	1078	PT=ITU-T G.711 PCMU, SSR
779	28.2459700	140.114.71.139	140.114.71.200	RTP	1078	PT=ITU-T G.711 PCMU, SSR

Frame 772: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface 0  
Ethernet II, Src: Giga-Byt\_55:ec:10 (50:e5:49:55:ec:10), Dst: Giga-Byt\_55:ec:22 (50:e5:49:55:ec:22)  
Internet Protocol Version 4, Src: 140.114.71.200 (140.114.71.200), Dst: 140.114.71.139 (140.114.71.139)  
User Datagram Protocol, Src Port: avt-profile-2 (5005), Dst Port: avt-profile-2 (5005)  
Real-time Transport Control Protocol (Sender Report)  
10.. .... = Version: RFC 1889 Version (2)  
..0. .... = Padding: False  
...0 0000 = Reception report count: 0  
Packet type: Sender Report (200)  
Length: 6 (28 bytes)  
Sender SSRC: 0x23068fb8 (587632568)  
Timestamp, MSW: 3572665085 (0xd4f28afd)  
Timestamp, LSW: 2224793058 (0x849ba5e2)  
[MSW and LSW as NTP timestamp: Mar 19, 2013 06:58:05.517999000 (UTC)]  
RTP timestamp: 2171652686  
Sender's packet count: 31  
Sender's octet count: 31  
Real-time Transport Control Protocol (Source description)  
10.. .... = Version: RFC 1889 Version (2)  
..0. .... = Padding: False  
...0 0001 = Source count: 1  
Packet type: source description (202)  
Length: 5 (24 bytes)  
Chunk 1, SSRC/CSRC 0x23068fb8  
[RTCP frame length check: OK - 52 bytes]



# Requirement

- \* Caller and Callee can talk to each other (50%)
- \* Report requirements (3 ~ 5 pages) (50%)
  - \* Screenshots for program running windows of Caller and Callee.
  - \* Screenshots for RTP and RTCP packets format in Wireshark.
  - \* Explain the process of program execution.
  - \* Your thoughts after this project.
  - \* Possible extensions.

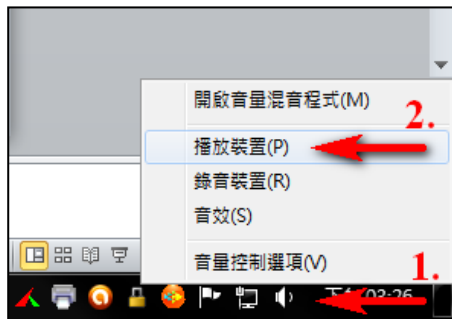
# Turn In

- \* iLMS
  - \* <http://lms.nthu.edu.tw/>
- \* Wrap the following files into 學號\_姓名.zip
  - \* Report
  - \* PCS\_RTP Caller.java and PCS\_RTP Callee.java
- \* Due Date
  - \* 2013/04/03 23:59:59

# 補充說明 (1/2)

- \* 如何檢查播放裝置與錄音裝置是否正常?(參考下頁圖)
  1. 開啟播放/錄音裝置。
  2. 放個音樂檢查正在使用的播放裝置是否有音量起伏。
  3. 確認麥克風已開啟，對麥克風說話檢查錄音裝置是否有音量起伏。
  4. 若修改程式前Callee聽不到Caller講話，可能是程式抓到的裝置並非您正在使用的裝置，請將其餘非使用中的裝置停用後，重開Caller與Callee再行測試。

# 補充說明 (2/2)



Right click





Thank you!

[qqting@wmnet.cs.nthu.edu.tw](mailto:qqting@wmnet.cs.nthu.edu.tw)