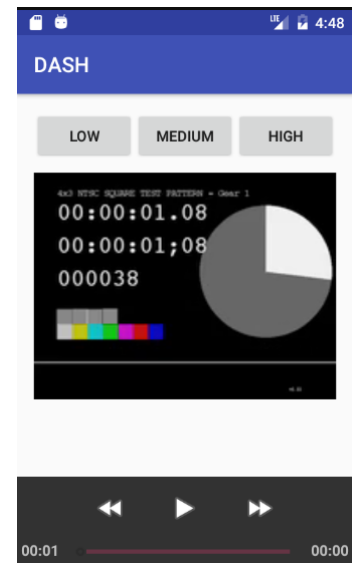


DASH (Dynamic Adaptive Streaming over HTTP) streaming system

Goals:

1. Upload a MP4 video at least 3 mins to a web server.
2. Split the MP4 file into streamlets, i.e., 10 second long video files.
3. Transcode the streamlets into 3 different streamlets (e.g., low, medium, high quality).
4. Create a playlist (.m3u8) on the web server.
5. Implement a simple Android DASH media player to play your video.
6. Users can switch among quality levels using your



A simple DASH player & web interface.

GUI.

Notes:

- On the web server, write scripts in the **Python** or **PHP** language to
 - 1) Allow users to upload the video.
 - 2) Split the video into streamlets.
 - 3) Transcode the streamlets to 3 different qualities.
 - 4) Generate the playlists (.m3u8) on the web server.
- Use **FFmpeg** to transcoder.
- Use **Java** and Android Studio IDE to write your Android App.
- You need to know some basic Linux commands: ls, mv, cd, chmod, ...

You need to submit a .zip file that contains the following files to the FTP

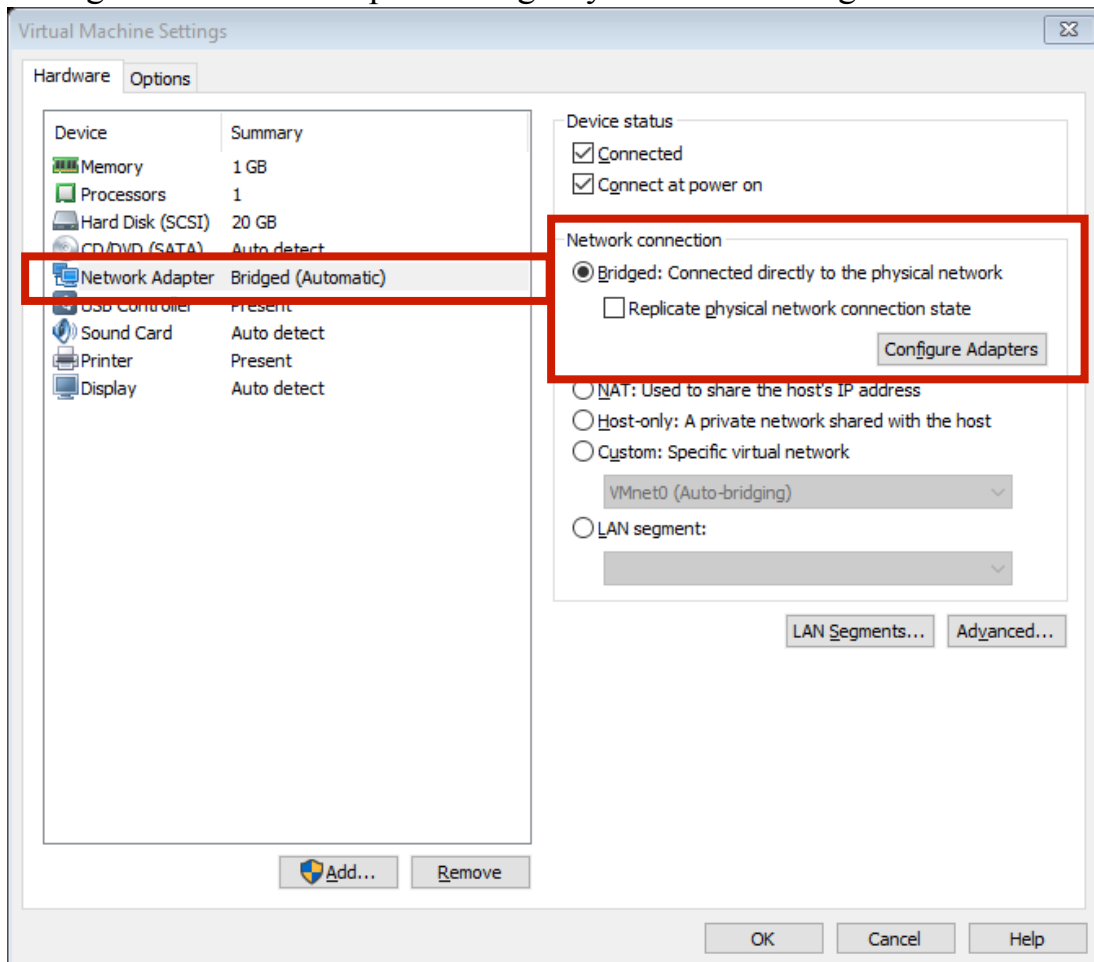
(The information of FTP will be posted on Facebook.)

1. A report.
2. Android source code.
3. Web site source code

Deadline: Mar.30(四) 23:59:59

Guides:

1. If you don't have your own server, you can use **VMware Workstation player** to host your server. And you have to run the Android emulator on the same PC to connect the VM.
2. Change the network adapter setting of your VM to Bridge.



3. You can install **Ubuntu 16.04** (or other OS you are familiar with) on your server.
4. Install Web server (**Apache2**, **Nginx** or others)
5. Install **PHP** or **Python**

```
sudo apt-get install apache2 php libapache2-mod-php
```

The example commands use on Ubuntu 16.04. (Apache2 and PHP 7.0.15)

6. Install FFmpeg

```
sudo apt-get install ffmpeg
```

The example commands use on Ubuntu 16.04. (FFmpeg version: 2.8.11)

7. You should be able to see the default web page on your PC.

Use ifconfig command to check the IP of your VM.

The screenshot shows a terminal window with the following output for the 'ifconfig' command:

```
nmsl@ubuntu:~$ ifconfig
ens33
Link encap:Ethernet HWaddr 00:0c:29:78:a1:da
inet addr:192.168.1.99 Bcast:192.168.1.255 Mask:255.255.255.0
inet6 addr::::69a4:a726:bc9c/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:51 errors:0 dropped:0 overruns:0 carrier:0
TX packets:81 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:10274 (10.2 KB) TX bytes:22156 (22.1 KB)

lo
Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr:::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:65536 Metric:1
RX packets:201 errors:0 dropped:0 overruns:0 frame:0
TX packets:201 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1
RX bytes:14717 (14.7 KB) TX bytes:14717 (14.7 KB)

nmsl@ubuntu:~$
```

The browser window displays the Apache2 Ubuntu Default Page, which includes a red banner that says "It works!" and a "Configuration Overview" section. The "Configuration Overview" section contains the following text:

Configuration is different from the upstream default configuration, and split interaction with Ubuntu tools. The configuration system is **fully** **bc/apache2/README.Debian.gz**. Refer to this for the full or the web server itself can be found by accessing the **manual** if the ed on this server.

apache2 web server installation on Ubuntu systems is as follows:

configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.

- ports.conf is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.

8. For php to execute FFmpeg commands, you can see here:

<https://trac.ffmpeg.org/wiki/PHP>

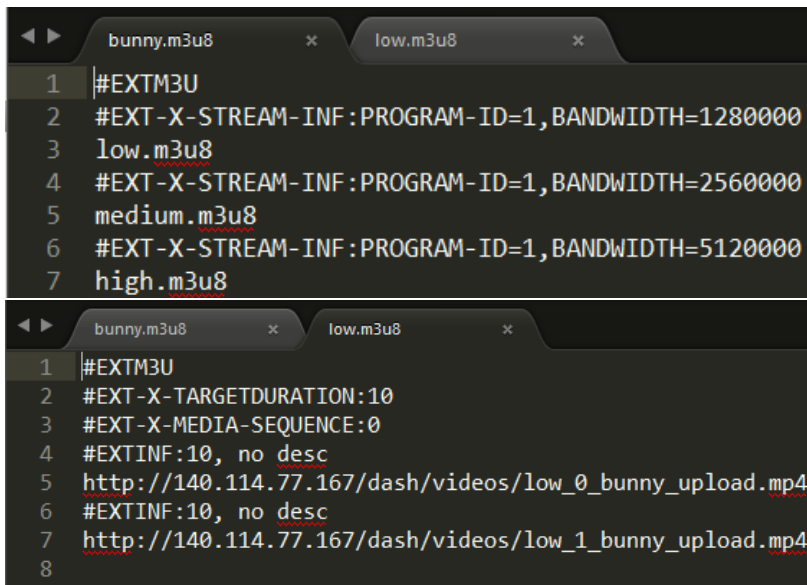
9. Your DASH media player need to read the .m3u8 playlist.

For .m3u8 format, you can see here:

https://developer.apple.com/library/content/technotes/tn2288/_index.html

Hint:

- There are some samples of playlists on our server. You can use it to test your program.
<http://140.114.77.167/dash/playlists/bunny.m3u8>
<http://140.114.77.167/dash/playlists/low.m3u8>
- The playlists may not work on some players because the HLS (Http Live Streaming) is not support mp4 files. So you are going to build your own player.



```
1 #EXTM3U
2 #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=1280000
3 low.m3u8
4 #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=2560000
5 medium.m3u8
6 #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=5120000
7 high.m3u8

1 #EXTM3U
2 #EXT-X-TARGETDURATION:10
3 #EXT-X-MEDIA-SEQUENCE:0
4 #EXTINF:10, no desc
5 http://140.114.77.167/dash/videos/low_0_bunny_upload.mp4
6 #EXTINF:10, no desc
7 http://140.114.77.167/dash/videos/low_1_bunny_upload.mp4
8
```

Grading policy:

- (1 point) User can upload the video via your web site.
- (1 point) Your web scripts can split a video into streamlets.
- (1 point) Your web scripts can transcode the video into three qualities.
- (1 point) Your web scripts can generate playlists file in .m3u8 format.
- (2 point) Your Android app can read the playlists via internet.
- (2 point) Your Android app can play streamlets automatically.
- (2 point) Your Android app allows users to switch the quality.