Networking in Android

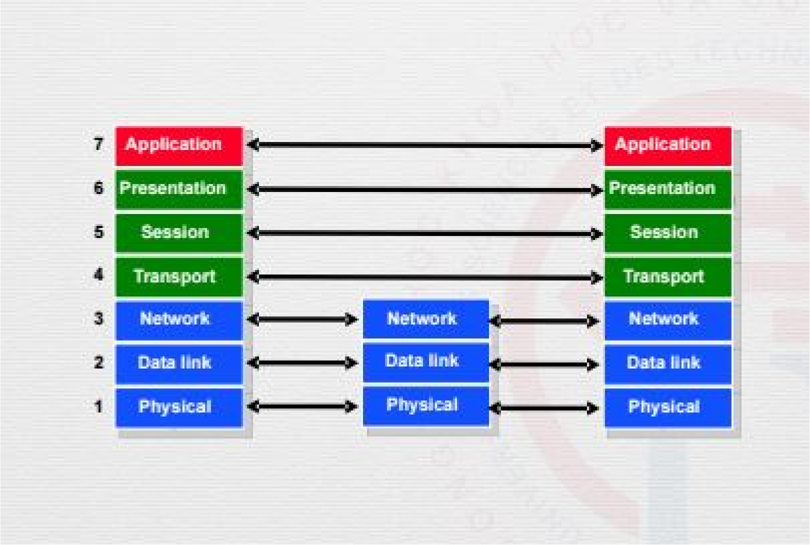

Figure 1: Network Layers

Contents

- Permissions
- Embedded package/class
- External library
- Data representation

Permissions

1. What?
   - Android has privilege-separation
   - Sandboxing
     – System user ID
     – System group ID
   - Specific actions require permissions
2. Why ?

- Privacy is an important aspect

- Permission is a way to implement/improve security and privacy

- Each "sensitive" action requires a separated permission – Read external storage – Write external storage – Read contact list

3. How ?

- Marshmallow+ has two main levels of permissions:

- Normal: no effect on user privacy, requires user confirmation

- "Dangerous": affect user privacy or device operations, requires confirmation

- Normal level
  – Internet access
  – Read network state
  – Set timezone, set wallpaper…

- Dangerous level
  – Read/write external storage
  – Access contact list
  – Access phone (make phone calls, receive calls, call log)
  – Send / receive SMS
  – Calendar, events
  – Microphone
  – Camera
  – Location

- Define what permissions are needed in the manifest
- For internet access

`xml <uses-permission android:name="android.permission.INTERNET" / }`

- Normal permission, so no need special treatment – Request permissions at runtime or reducing targetSdkVersion

Embedded package

1.Create URL from string

```
    URL url = new URL("http://ict.usth.edu.vn/wp-content/uploads/usth/usthlogo.png");
```

2. Make a request to server

```
    HttpURLConnection connection = (HttpURLConnection) url.openConnection();
    connection.setRequestMethod("GET");
    connection.setDoInput(true);
    // allow reading response code and response dataconnection.
    connection.connect();
```

3. Receive response

```
int response = connection.getResponseCode();
 Log.i("USTHWeather", "The response is: " + response);
 InputStream is = connection.getInputStream();
```

4. Process response

- Different response type requires different data treatment
- Image: transform to bitmap
- JSON/XML : parsing (later. . .)
- Decode data to bitmap

```
Bitmap bitmap = BitmapFactory.decodeStream(is);
```

- Show it

```
ImageView logo = (ImageView) findViewById(R.id.logo);
logo.setImageBitmap(bitmap);
```

- Don’t forget to disconnect

```
connection.disconnect();
```

External library

1. Volley

- An Android HTTP Client library
- Made within AOSP [Android OpenSource Project]
- Simple to use
- Powerful
- Extendable
- Cache
- Maintained by Google

2. Usage

- Add INTERNET permission, if you haven’t done so

- Clone volley repository

```
git clone https://android.googlesource.com/platform/frameworks/volley
```

- Add volley as module
- Right click project, open module settings, “+”
- Import Gradle Project

1. Create request queue (one per app)

2. Create request with listeners

3. Add request to queue

Data Representation

1.JSON * Can represent structured data
* Simple to use
* Less verbose
* Getting more attraction Example

```
{
"title": "Yahoo! Weather",
"width": "142",
"height": "18",
"link": "http://weather.yahoo.com",
"url": "http://l.yimg.com/a/i/brand/purplelogo//uh/us/news-wea.gif"
}
```