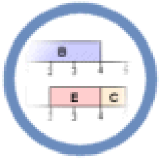


Lecture ”5”
Inter-Device Communication

<lecturer, date>

Outline

- **Android Connectivity**
  - Bluetooth
  - NFC
  - Wi-Fi P2P

Android Connectivity

- Rich APIs to let your app to connect/interact with other devices over
  - Bluetooth
  - NFC
  - Wi-Fi P2P
  - USB
  - SIP

Outline

- Android Connectivity
  - **Bluetooth**
  - NFC
  - Wi-Fi P2P

Bluetooth

- Bluetooth functionality through the Android Bluetooth APIs
- Wirelessly exchange data with other Bluetooth devices
- Point-to-point and multipoint wireless features
- Bluetooth APIs + Android app ➔
  - Scan for other Bluetooth devices
  - Query the local Bluetooth adapter for paired Bluetooth devices
  - Establish RFCOMM channles
  - Connect to other devices through service discovery
  - Transfer data to/from other devices
  - Manage multiple connectinos

Bluetooth Categories

- Classic Bluetooth
  - More battery-intensive operations e.g., streaming/communicating
- Bluetooth Low Energy (BLE)
  - Significant low power consumption
  - Devices with low power requirements e.g., proximity sensors, heart rate monitors, fitness devices
  - More about BLE: http://developer.android.com/guide/topics/connectivity/bluetooth-le.html

Communication Major Tasks

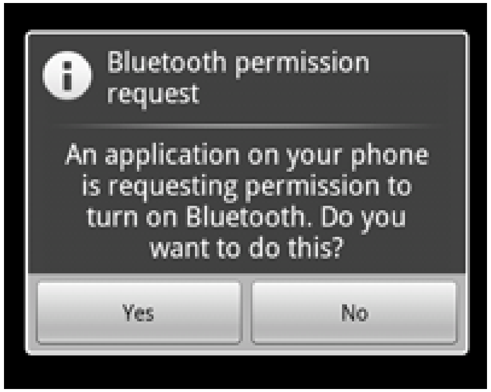
Four Major Tasks for Bluetooth Communications

1. **Setting up Bluetooth**

✓ Get the Bluetooth adapter such that your app can interact with it

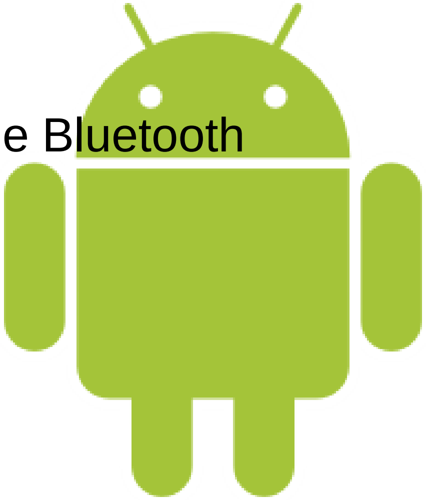
✓ Enable Blutooth

☞ Enabling *discoverability* will automatically enable Bluetooth



Bluetooth permission request

An application on your phone is requesting permission to turn on Bluetooth. Do you want to do this?



Communication Major Tasks

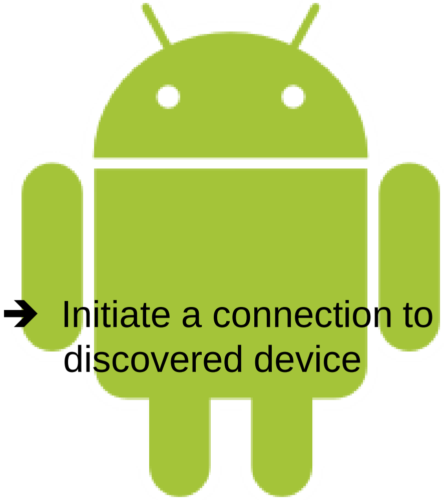
2. **Finding paired/local area devices**

✓ Quering paired devices

- Before berforming device discovery
- Is the desired device already known?

✓ Device discovery ➔

- Search for Bluetooth enabled devices
- Request information
  - ➢ Device name/class
  - ➢ Unique MAC address



➔ Initiate a connection to the discovered device

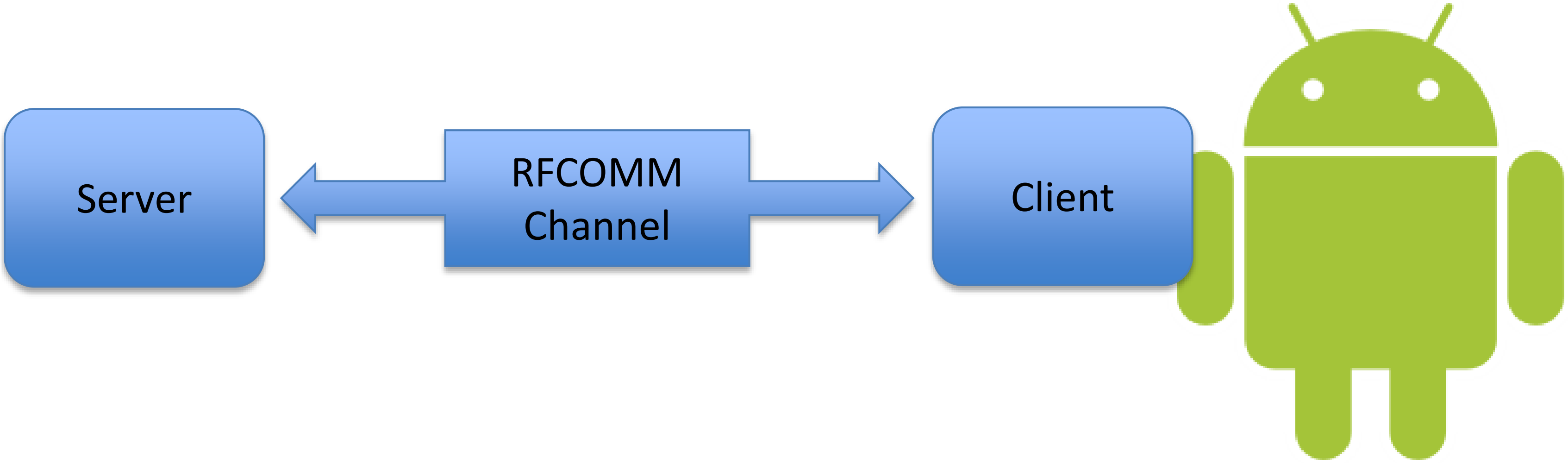
☞ Being paired vs. Being Connected

Communication Major Tasks

3. **Connecting devices**

   - Server-side : Open the server socket

   - Client-side : Initiate the connection using server MAC address



Server

RFCOMM Channel

Client

Communication Major Tasks

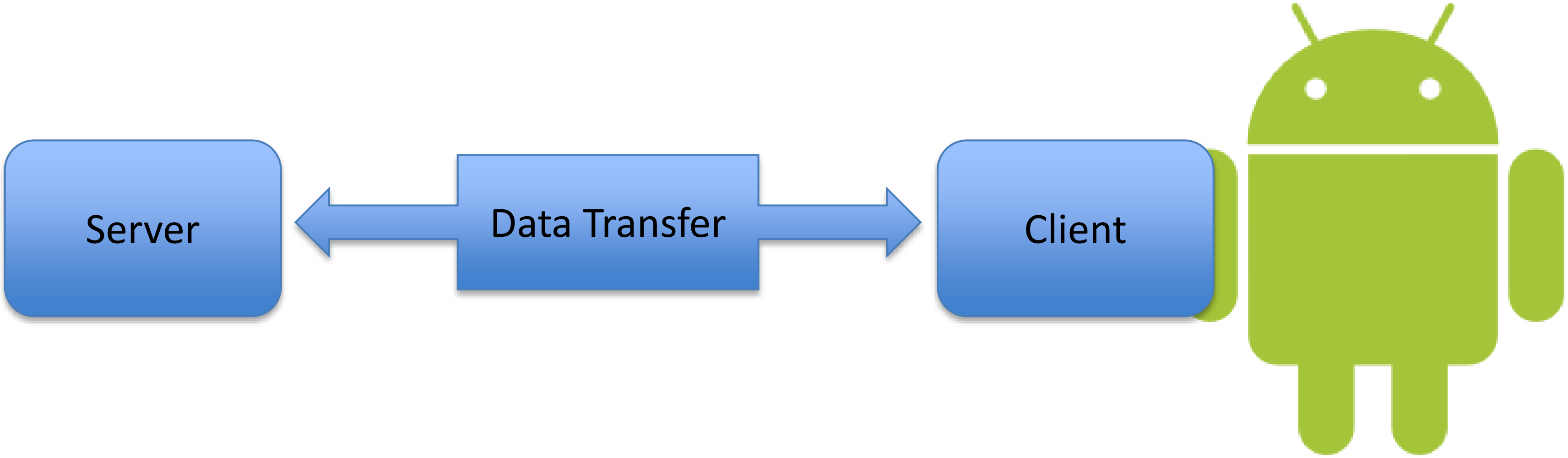
- ☞ *BluetoothSocket*
  - Server: When an incoming connection is accpeted
  - Client: When it opens an RFCOMM channel to the server
- ☞ Implementation technique: Prepare each device as a server

Communication Major Tasks

4. **Managing a connection**

   - To handle transmissions through the socket (InputStream/OutputStream)

   - Read/Write data to the streams



Server

Data Transfer

Client

Permissions

- *BLUETOOTH* : To use Bluetooth features in your app e.g., requesting/accepting a connection, transferring data

- *BLUETOOTH_ADMIN* : To initiate device discovery/manipulate Bluetotth settings

  ✓ Sloely to discover local Bluetooth devices

  ✗ Other capabilities granted should not be used except the ”*power manager*” app

☼ Bluetooth APIs available at android.bluetooth package

Outline

- Android Connectivity
  - Bluetooth
  - **NFC**
  - Wi-Fi P2P

Near Field Communication (NFC)

- Short-range wireless technologies
- Share small payloads of data an NFC tag between
  - NFC tag and Android device
  - Two Android devices
- Tags
  - Simple tag : Read/Write semantics
  - More complex : Math operation, Cryptographic hardware
  - Most sophisticated : Operating environemnt

  ☞ Data format : NDEF (NFC Data Enchange Format)

Modes of Operation

Three Modes of Operation

1. Reader/Writer mode : To read and/or write passive NFC tags

2. P2P : To exchange data with other NFC peers (Andriod Beam)

3. Card emulation mode : To allow NFC device act as an NFC card

Outline

- Android Connectivity
  - Bluetooth
  - NFC
  - **Wi-Fi P2P**

Wi-Fi P2P

Wi-Fi Peer-to-Peer (Wi-Fi P2P)

- Android 4.0 (API level 14) or later
- To connect directly to each other via Wi-Fi
- Without intermediate access point
- Distance much longer than a Bluetooth connection
- Applications for sharing data e.g., multiplayer game, photo sharing

APIs Components

APIs Three Main Components

1. Methods to discover, request and connect peers (*WiFiP2pManager* class)

2. Listeners to notify success/failure of *WiFiP2pManager* calls (Passed in as method’s parameter)

3. Broadcast intents to notify events detected by Wi-Fi P2P framework e.g., dropped connection, newly discovered peer

☼ You often use these three main components of APIs together

Creating a Broadcast Receiver

Creating a Broadcast Receiver for Wi-Fi P2P Intents

- Broadcast receiver
  - To receive intents broadcast by the Android system
  - Your application can respond to events you are interested in

☞ Basic steps

✓ Create a class to have parameters for *WiFiP2pManager, WiFiP2pManager.Channel*

✓ Check the intents you are interested in *OnReceive()*, carry on any necessary actions depending on the received intent

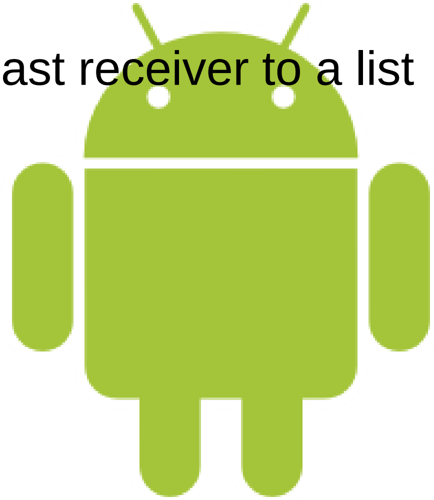
Creating a Wi-Fi P2P Application

1. **Initial set up**
   - To create and register a broadcast receiver
   - Start using Wi-Fi P2P APIs
1. Discovering peers
2. Connecting to a peer
3. Transferring data to a peer

Creating a Wi-Fi P2P Application

2. **Discovering peers**

   - To detect available peers in range
   - A success/failure is communicated to your application
   - If the dicovery proceeds, you can listen in a broadcast receiver to a list of peers



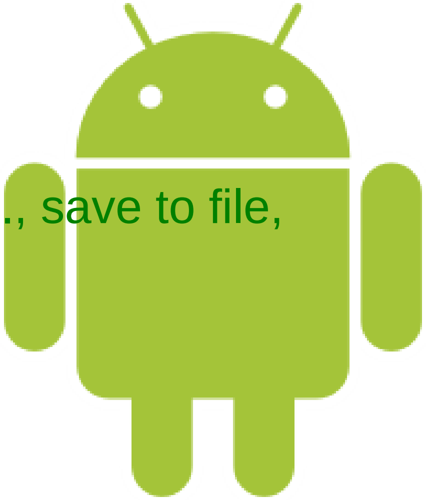
Creating a Wi-Fi P2P Application

3. **Connecting to a peer**
    - Can figure out the device you want to connect to after obtaining a list of peers
    - Notifies of a connection success/failure

4. **Transferring data to a peer**
    - Once connection is established
    - Transfer data between devices with sockets

Creating a Wi-Fi P2P Application

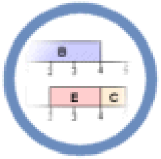
- Basic steps
  - ✓ Create a *ServerSocket* that waits for a connection from client
  - ✓ Create a client *Socket* (IP address and port of the server)
  - ✓ Send data from the client to the server
  - ✓ Server carries out actions on received data e.g., save to file, present to user



References

- Android: http://developer.android.com/
- Android Connectivity:
  - http://developer.android.com/guide/topics/connectivity/index.html
  - https://www.eecs.berkeley.edu/~daw/papers/intents-mobisys11.pdf
- Android Connectivity Training:

http://developer.android.com/training/building-connectivity.html



Lab ”5”
Inter-Device Communication

<lecturer, date>

Lab ”5”

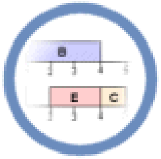
- Your task in this lab is as follows:
  - Get familiar with Wi-Fi P2P APIs, *WiFiP2pManager* class (methods, listeners, intents) in Android
  - Create a broadcast receiver for Wi-Fi P2P intents in Android

Seminar ”5”
Inter-Device Communication

<lecturer, date>

Seminar ”5”

- Discuss the steps of creating P2P connections with Wi-Fi, e.g., setting up permissions/broadcast receiver, peer discovery, …



Mini-Project ”5” Inter-Device Communication

<lecturer, date>

Mini-Project ”5”

- Make a short report of Session Initiation Protocol (SIP) connectivity in Android and discuss how it works, the procedure to make and receive calls, the approach requirements and limitations.

  Resources:

  – http://developer.android.com/guide/topics/connectivity/sip.html