

# ANDROID ALKALMAZÁSFEJLESZTÉS

Kommunikáció

Mobilinternet

Wi-Fi

Letöltés



[sicz-mesziar.janos@nik.uni-obuda.hu](mailto:sicz-mesziar.janos@nik.uni-obuda.hu)

Sicz-Mesziár János

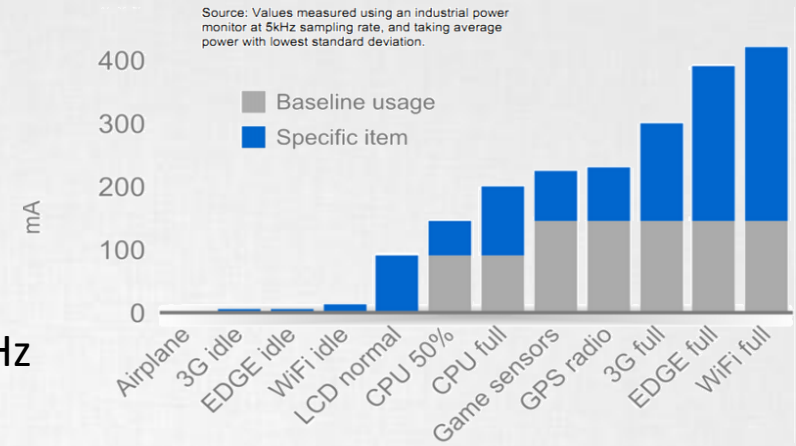
2017. április 6.

# Mobilinternet vs. Wi-Fi

## Fogyasztás →

### Frekvencia

- Wi-Fi: 2.4 / 5 Ghz
- Mobilinternet példák:  
UMTS/HSDPA/HSUPA ↔ 850/900/1900/2100 MHz  
UMTS ↔ 2100/1900/850 combo



### Sebesség tekintetében

Wi-Fi	2.4 GHz	5GHz
802.11a	54 Mbit/s	
802.11b	11 Mbit/s	
802.11g	54 Mbit/s	
802.11n	150 Mbit/s	300 Mbit/s
802.11ac		1-2 Gbit/s

Mobilinternet (átvitel technológia szerint)		2G
HSCSD	9,6 – 57,6 Kbit/s	
GPRS	< 171,2 Kbit/s	2.5G
EDGE	< 473 Kbit/s	
UMTS	1,8 – 7,2 Mbit/s	3G
HSDPA	1,8 - ... - 14,4 - 28,8 Mbit/s	3.5G
LTE	< 326 Mbit/s	4G
?	0.2 – 1 – 10 Gbit/s	5G

\* csak a fontosabbak lettek felsorolva, ennél jóval több létezik, lásd [Network Type konstansok](#) itt!

# Android támogatás

## OSI modell miatt nem szükséges különbséget tenni

- Adatátvitel módja detektálható, lásd: [Google I/O 2009](#)

## Internet használathoz jogot kell kérnünk:

- android.permission.INTERNET

## Wi-Fi

- Kezelése a [Wi-Fi API-kon](#) keresztül
- Hivatalosan ad-hoc kapcsolódás nem engedélyezett
- Wi-Fi Tethering Android 2.2 óta
- [Wi-Fi Direct](#) támogatás [Android 4.0](#) óta

## IP szabvány szállítási rétegében:

- **TCP** : kapcsolat orientált, csomag megérkezést megerősíti, és a csomagok sorrendjéről is gondoskodik
- **UDP** : gyors, apró üzenetváltások, de nem sorrendtartó, valamint a csomagok megérkezése nem garantált

# Android networking lehetőségek

## Android hivatalos API

- ~~Apache HttpClient, DefaultHttpClient~~  
Android 6.0+ deprecated  
useLibrary 'org.apache.http.legacy'
- **Http[S]URLConnection**  
<https://developer.android.com/training/basics/network-ops/connecting.html>
- **Volley**  
HTTP library that makes networking for Android apps easier  
<https://developer.android.com/training/volley/>
- **Download Manager:**  
<https://developer.android.com/reference/android/app/DownloadManager.html>

## Third-party megoldások

- **Retrofit**  
type-safe HTTP client for Android  
<http://square.github.io/retrofit/>
- **OkHttp**  
HTTP & HTTP/2 client for Android  
<http://square.github.io/okhttp/>
- **Picasso**  
powerful image downloading and caching  
<http://square.github.io/picasso/>
- **Glide**  
image loading and caching library for Android  
<https://github.com/bumptech/glide>

# HttpsURLConnection letöltés

## Adott URL tartalmának letöltése egy fájlba:

```
URL url = new URL("http://nik.uni-obuda.hu/malk/");
File malkFile = new File("/sdcard/malk.html");
HttpsURLConnection ucon = (HttpsURLConnection)url.openConnection();
ucon.setReadTimeout(3000);
ucon.setConnectTimeout(3000);
ucon.setRequestMethod("GET");
ucon.setDoInput(true);
ucon.connect();
int responseCode = connection.getResponseCode();
if (responseCode != HttpsURLConnection.HTTP_OK) {
    InputStream is = ucon.getInputStream();
    FileOutputStream fos = new FileOutputStream(malkFile);
    byte[] buffer = new byte[1024];
    int len = 0;
    while((len = is.read(buffer)) != -1)
        fos.write(buffer, 0, len);
    fos.close();
    is.close();
}
```

[Bővebben >>](#)

Ne felejtünk el jogosultságot kérni :  
`android.permission.INTERNET`  
`android.permission.WRITE_EXTERNAL_STORAGE`



# DownloadManager letöltés

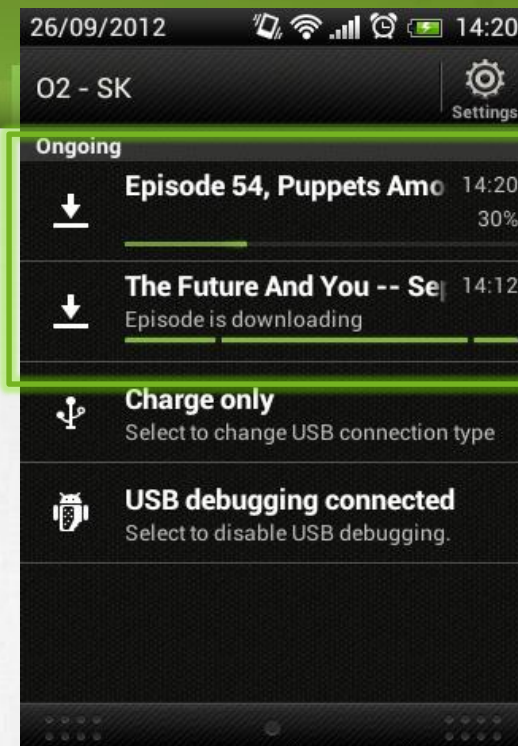
## DownloadManager

- API level 9-től, Gingerbread óta érhető el
- Letöltés jelzése a Notification bar-on
- Nem kell kézzel megírni a letöltések kezelését: szálak kezelése, streamek, letöltés folytatása, százalékos visszajelzés

Ugyanúgy kell jogosultság:

**android.permission.INTERNET**

```
DownloadManager.Request request =  
    new DownloadManager.Request(Uri.parse("http://.../"));  
request.setDescription("Leírás a letöltésről");  
request.setTitle("Valamilyen cím");  
request.setDestinationInExternalPublicDir(  
    Environment.DIRECTORY_DOWNLOADS, „fajlnev.kit");  
DownloadManager manager =  
    (DownloadManager) getSystemService(DOWNLOAD_SERVICE);  
manager.enqueue(request);
```



# UDP adatátvitel

**Jogosultság: android.permission.INTERNET**

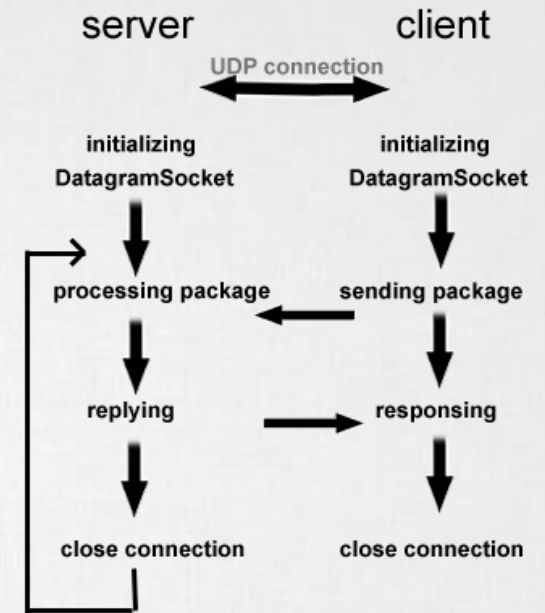
## Szerver oldal

```
int serverPort = 50004;
InetAddress ip =
    InetAddress.getByName("192.168.1.1");
byte[] buf = new byte[1024 * 65];
DatagramPacket receivePacket =
    new DatagramPacket(buf, buf.length);
DatagramSocket socket = new
    DatagramSocket(serverPort);
socket.receive(receivePacket);
receivePacket.getData();
```

thread

## Kliens oldal

```
String msg = "Hello UDP Package";
byte[] msgByte = msg.getBytes();
DatagramSocket socket = new DatagramSocket();
InetAddress serverIP = InetAddress.getByName("192.168.1.1");
socket.connect(serverIP, 50004);
socket.send(new DatagramPacket(msgByte, msgByte.length));
```



[Kép forrása](#)

Android 1.5 alatt még bug-os volt.

# TCP adatátvitelt

## TCP client:

```
Socket socket = new Socket();
socket.connect(new InetSocketAddress(/* Cím */,
                                   /* port */));

byte[] buffer = new byte[1024];
InputStream in = socket.getInputStream();
int len = 0;
while((len = in.read(buffer)) != -1) {
    /* valamit csinálunk */
}
in.close();
```

thread

## TCP server:

```
ServerSocket server = new ServerSocket(/* port */);
while(true) {
    Socket client = server.accept();
    /* Klient szokás új szálon kezelni a
       továbbiakban */
}
```

thread