

ANDROID ALKALMAZÁSFEJLESZTÉS

Kommunikáció

Mobilinternet

Wi-Fi

Letöltés



Sicz-Mesziár János

sicz-mesziar.janos@nik.uni-obuda.hu

Mezei József

mezei.jozsef@nik.uni-obuda.hu

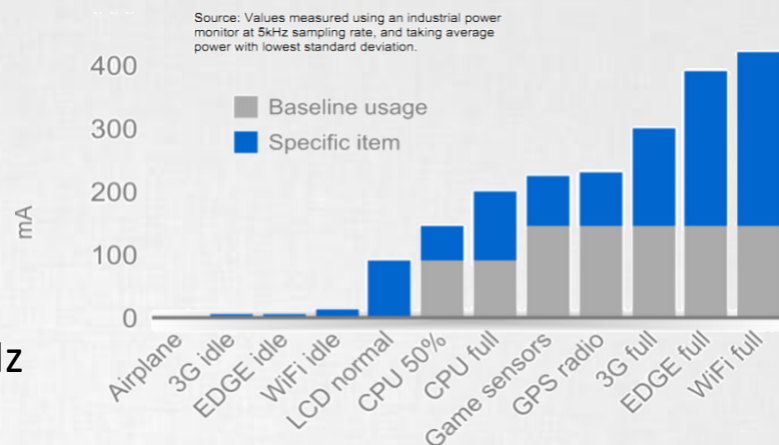
2018. november 18.

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>

URLConnection letöltés

Adott URL tartalmának letöltése egy fájlba:

```
val url = URL("http://nik.uni-obuda.hu/malk/")
val file = File(path)
val connection = (url.openConnection() as HttpURLConnection).apply {
    readTimeout = 3000
    connectTimeout = 3000
    requestMethod = "GET"
    doInput = true
    connect()
}

if (connection.responseCode == HttpURLConnection.HTTP_OK) {
    connection.inputStream.use { input ->
        file.outputStream().use { output ->
            input.copyTo(output, 1024)
        }
    }
}
```

[Bővebben >>](#)

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

Kotlin UI-safe thread pattern

```
 typealias Fail = (String) -> Unit
 typealias Success = (String, File) -> Unit
```

```
 class ImageDownloader(
     var link: String,
     var path: String
) : Thread() {
     private var callbackFail: Fail? =  null
     private var callbackSuccess: Success? =  null;
     private val handler = Handler(Looper.getMainLooper())
```

```
     override fun run() {
         try {
            // ...
             if (connection.responseCode != HttpURLConnection.HTTP_OK) {
                handler.post { callbackFail?.invoke(connection.responseMessage) }
                 return
            }
            // ...
            handler.post { callbackSuccess?.invoke(url, file) }
        }  catch (e: MalformedURLException) {
            handler.post { callbackFail?.invoke(connection.responseMessage) }
        }
    }
```

```
     fun onFail(callback: Fail) = apply { callbackFail = callback }
     fun onSuccess(callback: Success) = apply { callbackSuccess = callback }
```

```
ImageDownloader(
    "http://.../logo.png",
    "/sdcard/logo.png"
).onFail { e ->
    Toast.makeText(this, e, Toast.LENGTH_SHORT).show()
}.onSuccess { s, file ->
    image.setImageURI(Uri.fromFile(file))
}.start()
```

```
}
```


DownloadManager letöltés

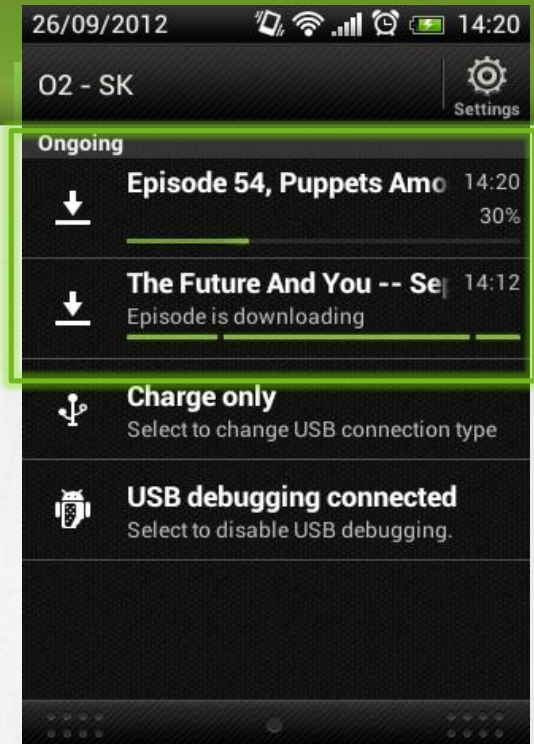
DownloadManager

- 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

```
val manager = getSystemService(DOWNLOAD_SERVICE) as DownloadManager
manager.enqueue(DownloadManager.Request(
    Uri.parse("http://.../")
).apply {
    setDescription("Leírás a letöltésről")
    setTitle("Valamilyen cím")
    setDestinationInExternalPublicDir(
        Environment.DIRECTORY_DOWNLOADS,
        "fajlnev"
    )
})
```



UDP adatátvitel

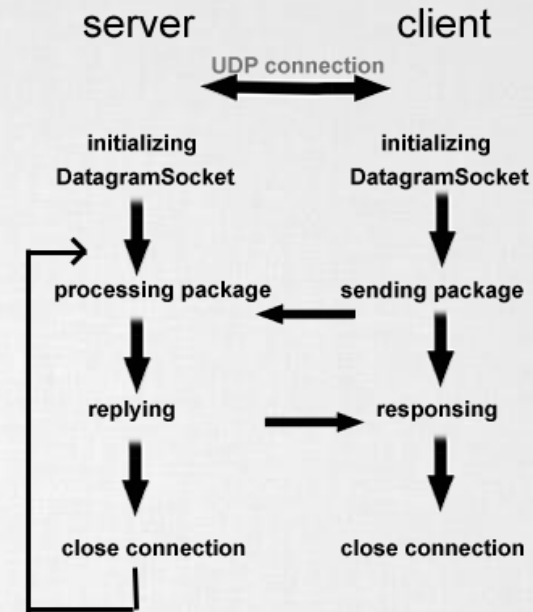
Jogosultság: android.permission.INTERNET

Szerver oldal

```
val serverPort = 50004
val ip = InetAddress.getByName("192.168.1.1")
val buf = ByteArray(1024 * 65)
val receivePacket = DatagramPacket(buf, buf.size)
val socket = DatagramSocket(serverPort)
socket.receive(receivePacket) thread
receivePacket.getData()
```

Kliens oldal

```
val msg = "Hello UDP Package"
val msgByte = msg.toByteArray()
val socket = DatagramSocket()
val serverIP = InetAddress.getByName("192.168.1.1")
socket.connect(serverIP, 50004)
socket.send(DatagramPacket(msgByte, msgByte.size))
```



[Kép forrása](#)

TCP adatátvitelt

TCP client:

```
val socket = Socket().apply {  
    connect(InetSocketAddress("valami.hu", 1000))    thread  
}  
  
val input: InputStream = socket.getInputStream()  
// Do something with input stream  
input.close();
```

TCP server:

```
val server = ServerSocket(1000);  
  
while (true) {  
    thread    val client = server.accept();    thread  
             /* Kliensenként új szál */  
}  
}
```