

Mobilne komunikacije

Mobilne aplikacije

Stevan Gostojić

Fakultet tehničkih nauka, Novi Sad

29. novembar 2022.

Agenda

- 1 Radio komunikacije
- 2 Celularna mreža
- 3 Telefonija
- 4 SMS

Radio talasi

- Radio talasi su oscilacije elektromagnetnog polja u vremenu i prostoru.
- Slabljenje radio talasa zavisi od njegove frekvencije i karakteristika medijuma kroz koji se prostire.

Radio spektar

- Na osnovu karakteristika koje utiču na prenos radio talasa, radio spektar može se grubo podeliti na više opsega:
 - Vrlo niske frekvencije (VNF / VLF) od 3 do 30 kHz
 - Niske frekvencije (NF / LF) od 30 do 300 kHz
 - Srednje frekvencije (SF / MF) od 300 do 3000 kHz
 - Visoke frekvencije (VF / HF) od 3 do 30 MHz
 - Vrlo visoke frekvencije (VVF / VHF) od 30 do 300 MHz
 - Ultra visoke frekvencije (UVF / UHF) od 300 do 3000 MHz
 - Supervisoke frekvencije (SVF / SHF) od 3 do 30 GHz
 - Ekstremno visoke frekvencije (EVF / EHF) od 30 do 300 GHz

Radio uređaji



Figure 1: RTU-100.



Figure 2: NEXUS 5.

Radio urađaji

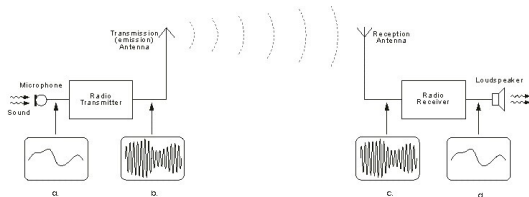


FIG. 2.1. Radio Transmission Block-Diagram

Figure 3: Prijemnik i predajnik.

- Pošiljalac
- Predajnik
- Antena
- Medijum
- Antena
- Prijemnik
- Primalac

Analogni signal



Figure 4: Analogni signal.

- Analogni signal je kontinualni signal koji informacije prenosi kao promena amplitude, frekvencije i faze.

Digitalni signal

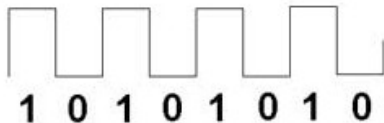


Figure 5: Digitalni signal.

- Digitalni signal je diskretan signal koji informacije prenosi kao niz znakova (logičnih nula i jedinica).

Modulacija

Modulacija je proces kojim se signal prilagođava karakteristikama prenosnog medijuma (transponuje se u područje frekvencija pogodnih za prenos radio talasima).

- Analogni postupci koriste se za modulisanje analognog signala
 - Amplitudna modulacija
 - Frekventna modulacija
 - Fazna modulacija
- Digitalni postupci koriste se za modulisanje digitalnog signala
 - Pulsna kodna modulacija
 - Delta modulacija

Agenda

- 1 Radio komunikacije
- 2 Celularna mreža
- 3 Telefonija
- 4 SMS

Razvoj mobilne telefonije

- 1991 Global System for Mobile Communications, originalno Groupe Spécial Mobile (GSM)
- 2000 General Packet Radio Service (GPRS)
- 2003 Enhanced Data rates for GSM Evolution (EDGE)
- 2002 Universal Mobile Telecommunications System (UMTS, 3G)
- 2008 High-Speed Downlink Packet Access (HSDPA) / High-Speed Uplink Packet Access (HSUPA)
- 2011 Long Term Evolution (LTE, 4G)

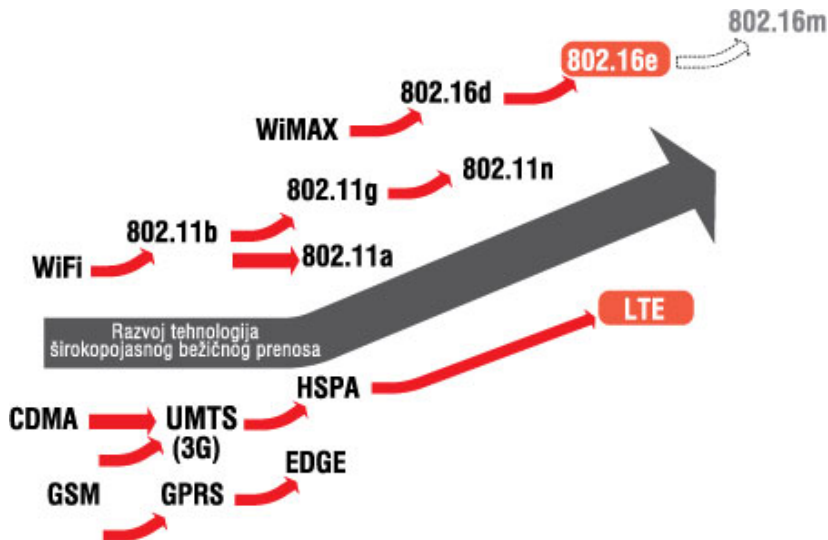


Figure 6: Razvoj mobilne telefonije [1].

Groupe Spécial Mobile (GSM)

- GSM je ćelijski sistem mobilne telefonije zasnovan na digitalnom prenosu podataka.
- Namenjen je pre svega za prenos govora, podaci se prenose standardnom modemsom vezom brzine brzinom od 9600 bit/s.
- Govor se koduje linearno prediktivnim koderom, a noseći signal se moduliše GMSK modulacijom na učestanostima od 900 do 1800 MHz.
- Mobilni telefoni pristupaju baznim stanicama po algoritmu TDMA.

General Packet Radio Service (GPRS)

- Oslanja se na postojeću komunikacionu infrastrukturu GSM-a.
- Podatke prenosi u paketima.
- Teorijska brzina je do 144 kbit/s, ali se praktično ostvaruje 56 kbit/s (dovoljno za e-mail i WAP).

Enhanced Data rates for GSM Evolution (EDGE)

- Oslanja se na postojeću komunikacionu infrastrukturu GSM-a i GPRS-a.
- Sistem pruža mogućnost adaptivnog protoka u zavisnosti od kvaliteta signala. Adaptivnost se postiže korišćenjem više tipova modulacija i kodovanja.
- Teorijska brzina je do 473.6 kbit/s, ali se praktično ostvaruje 236 kbit/s.

Code Division Multiple Access (CDMA)

- Razvio se paralelno sa razvojem GSM-a (posebno u Americi).
- CDMA podrazumeva čitavu familiju tehnika koje se zasnivaju na tehnici proširenog spektra.
- Standardna brzina za govor je 9600 bit/s, dok se za podatke nude varijabilne brzine od 1200 bit/s do 14400 bit/s.

Universal Mobile Telecommunications System (UMTS, 3G)

- Napravljen je kao potpuno nov sistem (ne oslanja se na postojeću GSM infrastrukturu).
- U osnovi je širokopojasni CDMA.
- Najavljene su multimedijalne usluge (kao što je video poziv). Zbog male brzine najčešće su neupotrebljive.
- Doživeo je komercijalni neuspeh.
- Najavljene su brzine do 14 Mbit/s, ali se u praksi ostalo na 384 kbit/s.

High-Speed Downlink/Uplink Packet Access (HSDPA/HSUPA)

- Komercijalni i tehnički neuspeh 3G sistema uslovio je intenzivan rad na njegovom unapređenju.
- Može se reći da predstavlja konačan domet CDMA tehnike. Dobre performanse ostvaruju se velikim brojem novih tehničkih rešenja (fast packet scheduling, adaptivnost modulacija i kodovanja, HARQ, itd).
- Omogućava veliki protok (maksimalno 7,2 Mbit/s, a tipično do 1,4 Mbit/s do 42 Mbit/s za downlink).

Long Term Evolution (LTE, 4G)

- Odgovor na WiMAX i vrlo mu je sličan.
- Od WiMAX-a su preuzeti principi spektralne prilagodljivost i potpunog oslonca na IP transport.
- Prednost LTE-a je u tome što se oslonja na postojeću GSM, GPRS i 3G komunikacionu infrastrukturu. Omogućena je dosledna podrška mobilnosti u heterogenim mrežama (GSM/3G, GPRS/EDGE, HSPA).
- Visok protok (do 100 Mbit/s) i malo kašnjenje (< 10 ms).

GSM spektar

GSM koristi nekoliko frekventnih opsega:

- 850 MHz
- 900 MHz
- 1800 MHz
- 1900 MHz

Ćelijski sistem

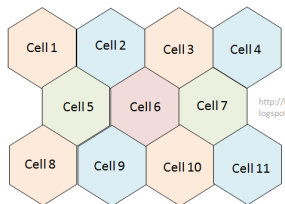


Figure 7: Ćelijski sistem.

- Određena teritorija je podeljena na ćelije.
- Svakoj ćeliji je dodeljen skup frekvencija koje su izabrane tako da minimizuju interferenciju sa susednim ćelijama.
- Skup frekvencija može se koristiti i u drugim ćelijama, sve dok te ćelije nisu susedne.
- Kada mobilni telefon pređe iz jedne ćelije u drugu ćeliju dok je poziv u toku, mreža će izdati naredbu mobilnom telefonu da promeni kanal (frekvenciju) i u isto vreme preusmeriti poziv na novi kanal.

Usluge GSM-a

- Telefonija (prenos govora)
- Short Message Service (SMS)
- Multimedia Messaging Service (MMS)
- Prenos podataka

Arhitektura GMS-a

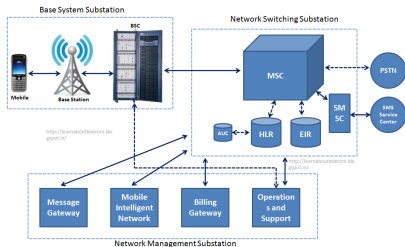


Figure 8: Arhitektura GMS-a.

- Base Station Subsystem (BSS)
- Network Switching Subsystem (NSS, Core)
- Network Management Subsystem (NMS)

Base Station Subsystem (BSS)

- Base Station Subsystem (BSS) je podsistem koji vodi računa o radio komunikaciji između mobilnih telefona i mreže.
- Sastoji se iz base transceiver station (BTS) i base station controller (BSC).
- BTS sadrži uređaje za slanje i prijem radio signala (radio prijemnik, radio predajnik, antena, izvor napajanja, itd.).
- BSC omogućava baznim stanicama da uspostave, predaju i prekinu poziv. Takođe konfiguriše radio primopredajnike i frekvencije svake ćelije.

Network Switching Subsystem (NSS, Core)

- Network Switching Subsystem (NSS) je podsistem koji vodi računa o uspostavljanju, autentifikaciji, šifrovanju i nadgledanju poziva.
- Glavni element je Mobile Station Controller (MSC) koji rukuje sa više BSC.
- MSC upravlja mrežom (rukuje uspostavljanjem, predajom i prekidom poziva i vodi računa o naplati) i omogućava povezivanje sa javnom telefonskom mrežom.
- Gateway Mobile Station Controller (GMSC) je MSC koji povezan na javnu telefonsku mrežu.

Network Switching Subsystem (NSS, Core)

- Home Location Register (HLR) je baza podataka koja sadrži podatke o pretplatnicima koji su autorizovani da koriste GSM mrežu (podatke o SIM karticama koje je izdao operater, prvenstveno IMSI).
- Visitor Location Register (VLR) je baza podataka koja sadrži podatke o roaming pretplatnicima.
- Equipment Identity Register (EIR) je baza podataka koja sadrži podatke o identitetu mobilnih uređaja i omogućava sprečavanje poziva sa ukradenih ili neautorizovanih mobilnih uređaja.
- Authentication Centre (AUC) autentifikuje pretplatnike i omogućava šifrovanje poziva.

Network Switching Subsystem (NSS, Core)

- Short Message Service Centre (SMSC) omogućava slanje i primanje tekstualnih poruka.
- Multimedia Messaging Service Centre (MMSC) omogućava slanje i primanje multimedijalnih poruka.
- Voice Mail Centre (VMC) snima i skladišti govorne poruke.
- Billing Centre (BC) na osnovu podataka o naplati primljenih od HLR-a i VLR-a vodi evidenciju o računu svakog pretplatnika.

Network Management Subsystem (NMS)

- Network Management Subsystem (NMS) služi za upravljanje celim sistemom.

SIM

- **Subscriber Identity Module (SIM)** je pametna kartica koja omogućava mobilnom telefonu da promeni pretplatnika i pretplatniku da promeni mobilni telefon.
- Implementira Java Card specifikaciju da bi omogućio interoperabilnost aplikacija.
- Pruža sigurno skladištenje:
 - Integrated Circuit Card Identifier (ICCID) - identifikator kartice
 - International Mobile Subscriber Identity (IMSI) - identifikator pretplatnika
 - i kriptografskog ključa koji se koristi za autentifikaciju pretplatnika.

SIM

- Takođe skladišti:
 - Personal Identification Number (PIN) - lozinku za uobičajenu upotrebu
 - Personal Unblocking Code (PUK) - lozinku za otključavanje PIN-a
 - Service Provider Name (SPN)
 - Local Area Identity (LAI)
 - broj SMSC-a
 - broj za hitne slučajeve
 - spisak uluga kojima pretplatnik može da pristupi
 - itd.
- Nudi i dodatne funkcije kao što je skladištenje telefonskog imenika i tekstualnih poruka.

Mobilni telefon

Mobilni telefoni mogu se identifikovati pomoću:

- Mobile Equipment Identifier (MEID) je globalno jedinstven broj koji identifikuje CDMA mobilne telefone.
- International Mobile Station Equipment Identity (IMEI) je globalno jedinstven broj koji indentifikuje GSM, UMTS i LTE mobilne telefone.

Svaka lokacije mobilne mreže identifikovana je jedinstvenim identifikatorom poznatim kao Location Area Identity (LAI) koji se sastoji iz:

- Mobile Country Code (MCC) - identifikator zemlje, 3 cifre
- Mobile Network Code (MNC) - identifikator mreže u zemlji, 2-3 cifre
- Location Area Code (LAC) - identifikator lokacije u mreži, 16 cifara

BTS ili sektor BTS-a identifikovan je sa Cell ID (CID).

Agenda

- 1 Radio komunikacije
- 2 Celularna mreža
- 3 Telefonija**
- 4 SMS

Telefonija

- Osnovna usluga GSM-a je mobilna telefonija (prenos govora).
- Međutim, iz bezbednosnih razloga nije moguće napraviti "in call" aktivnost.
- **Android API omogućava:**
 - korišćenje podrazumevane "in call" aktivnosti za obavljanje telefonskih poziva
 - pristup podacima o telefonu (tip, identifikator, verzija softvera, telefonski broj)
 - pristup podacima o SIM kartici (stanje, država i ime operatora, serijski broj)
 - pristup podacima o mreži (država, identifikator operatora, ime mreže, tip mreže)
 - pristup podacima o prenosu podataka (stanje i trenutna aktivnost)
 - reagovanje na promenu stanja mreže, poziva, lokacija ćelije, snage signala, aktivnosti i stanja prenosa podataka, itd.

AndroidManifest.java

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest ... >
3   <application ... >
4     <uses-permission android:name="android.permission.CALL_PHONE"/>
5   </application>
6 </manifest>
7
```

ExampleActivity.java

```
1 public void dialNumber() {  
2     Intent intent = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:1234567"))  
3     );  
4     startActivity(intent);  
5 }
```

AndroidManifest.java

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest ... >
3   <application ... >
4     <uses-permission android:name="android.permission.
      READ_PHONE_STATE"/>
5   </application>
6 </manifest>
7
```

ExampleActivity.java

```
1 TelephonyManager telephonyManager = null;  
2  
3 public void getTelephonyManager() {  
4     String serviceName = Context.TELEPHONY_SERVICE;  
5     telephonyManager = (TelephonyManager) getSystemService(serviceName);  
6 }  
7
```

ExampleActivity.java

```
1 public void readPhoneDetails() {
2     // Reads the phone type
3     int phoneType = telephonyManager.getPhoneType();
4     switch (phoneType) {
5         case (TelephonyManager.PHONE_TYPE_CDMA):
6             // Do something
7             break;
8         case (TelephonyManager.PHONE_TYPE_GSM):
9             // Do something
10            break;
11        case (TelephonyManager.PHONE_TYPE_NONE):
12            // Do something
13            break;
14    }
15
16    // Read the IMEI for GSM or MEID for CDMA
17    String deviceId = telephonyManager.getDeviceId();
18
19    // Reads the software version on the phone
20    String softwareVersion = telephonyManager.
21        getDeviceSoftwareVersion();
22
23    // Gets the phone number
24    String phoneNumber = telephonyManager.getLine1Number()
25        ;
26 }
```

ExampleActivity.java

```

1 public void readSIMDetails() {
2     int simState = telephonyManager.getSimState();
3     switch (simState) {
4         case (TelephonyManager.SIM_STATE_ABSENT):
5             // Do something
6             break;
7         case (TelephonyManager.SIM_STATE_NETWORK_LOCKED):
8             // Do something
9             break;
10        case (TelephonyManager.SIM_STATE_PIN_REQUIRED):
11            // Do something
12            break;
13        case (TelephonyManager.SIM_STATE_PUK_REQUIRED):
14            // Do something
15            break;
16        case (TelephonyManager.SIM_STATE_UNKNOWN):
17            // Do something
18            break;
19        case (TelephonyManager.SIM_STATE_READY): {
20            // Get the SIM country ISO code
21            String simCountry = telephonyManager.
22                getSimCountryIso();
23
24            // Get the operator code of the active SIM (MCC +
25            // MNC)
26            String simOperatorCode = telephonyManager.
27                getSimOperator();
28
29            // Get the name of the SIM operator
30            String simOperatorName = telephonyManager.
31                getSimOperatorName();
32
33            // Get the SIM's serial number
34            String simSerial = telephonyManager.
35                getSimSerialNumber();
36            break;
37        }
38    }
39 }

```


ExampleActivity.java

```

1 public void readNetworkDetails() {
2     // Gets connected network country ISO code
3     String networkCountry = telephonyManager.
        getNetworkCountryIso();
4
5     // Gets the connected network operator ID (MCC + MNC)
6     String networkOperatorId = telephonyManager.
        getNetworkOperator();
7
8     // Gets the connected network operator name
9     String networkName = telephonyManager.
        getNetworkOperatorName();
10
11     // Gets the type of network you are connected to
12     int networkType = telephonyManager.getNetworkType();
13     switch (networkType) {
14         case (TelephonyManager.NETWORK_TYPE_1xRTT): break;
15         case (TelephonyManager.NETWORK_TYPE_CDMA): break;
16         case (TelephonyManager.NETWORK_TYPE_EDGE): break;
17         case (TelephonyManager.NETWORK_TYPE_EVDO_0): break;
18         case (TelephonyManager.NETWORK_TYPE_EVDO_A): break;
19         case (TelephonyManager.NETWORK_TYPE_GPRS): break;
20         case (TelephonyManager.NETWORK_TYPE_HSDPA): break;
21         case (TelephonyManager.NETWORK_TYPE_HSPA): break;
22         case (TelephonyManager.NETWORK_TYPE_HSUPA): break;
23         case (TelephonyManager.NETWORK_TYPE_UMTS): break;
24         case (TelephonyManager.NETWORK_TYPE_UNKNOWN): break;
25     }
26 }
27

```

ExampleActivity.java

```

1 public void readDataConnectionStatus() {
2     int dataState = telephonyManager.getDataState();
3     switch (dataState) {
4         case TelephonyManager.DATA_CONNECTED:
5             // Connected
6             break;
7         case TelephonyManager.DATA_CONNECTING:
8             // Setting up data connection
9             break;
10        case TelephonyManager.DATA_DISCONNECTED:
11            // Disconnected
12            break;
13        case TelephonyManager.DATA_SUSPENDED:
14            // Suspended
15            break;
16    }
17
18    int dataActivity = telephonyManager.getDataActivity();
19    switch (dataActivity) {
20        case TelephonyManager.DATA_ACTIVITY_IN:
21            // Receiving IP PPP traffic
22            break;
23        case TelephonyManager.DATA_ACTIVITY_OUT:
24            // Sending IP PPP traffic
25            break;
26        case TelephonyManager.DATA_ACTIVITY_INOUT:
27            // Receiving and sending data traffic
28            break;
29        case TelephonyManager.DATA_ACTIVITY_NONE:
30            // No traffic
31            break;
32    }
33 }
34

```

ExampleActivity.java

```

1 public void registerPhoneStateListener() {
2     PhoneStateListener phoneStateListener = new
        PhoneStateListener() {
3         public void onCallForwardingIndicatorChanged(boolean
            cfi) {}
4         public void onCallStateChanged(int state, String
            incomingNumber) {}
5         public void onCellLocationChanged(CellLocation
            location) {}
6         public void onDataActivity(int direction) {}
7         public void onDataConnectionStateChanged(int state)
            {}
8         public void onMessageWaitingIndicatorChanged(boolean
            mwi) {}
9         public void onServiceStateChanged(ServiceState
            serviceState) {}
10        public void onSignalStrengthChanged(int asu) {}
11    };
12
13    telephonyManager.listen(
14        phoneStateListener,
15        PhoneStateListener.LISTEN_CALL_FORWARDING_INDICATOR
16        | PhoneStateListener.LISTEN_CALL_STATE |
17        PhoneStateListener.LISTEN_CELL_LOCATION |
18        PhoneStateListener.LISTEN_DATA_ACTIVITY |
19        PhoneStateListener.LISTEN_DATA_CONNECTION_STATE |
20        PhoneStateListener.LISTEN_MESSAGE_WAITING_INDICATOR
21        | PhoneStateListener.LISTEN_SERVICE_STATE |
22        PhoneStateListener.LISTEN_SIGNAL_STRENGTH);
23    }
24

```

ExampleActivity.java

```
1 public void onServiceStateChanged( ServiceState
    serviceState) {
2     int state = serviceState.getState();
3     switch (state) {
4     case ServiceState.STATE_IN_SERVICE:
5         // Normal operation condition
6         break;
7     case ServiceState.STATE_EMERGENCY_ONLY:
8         // Only emergency numbers are allowed
9         break;
10    case ServiceState.STATE_OUT_OF_SERVICE:
11        // No cell phone service is available
12        break;
13    case ServiceState.STATE_POWER_OFF:
14        // Radio of telephony is explicitly powered off
15        break;
16    }
17
18    // Gets current registered operator name in long
19    // alphanumeric format
20    String operator = serviceState.getOperatorAlphaLong();
21
22    // Gets current roaming indicator of phone
23    boolean roaming = serviceState.getRoaming();
24    };
```

ExampleActivity.java

```
1 public void onCallStateChanged(int state, String incomingNumber) {
2     switch (state) {
3         case TelephonyManager.CALL_STATE_IDLE:
4             // The phone is neither ringing nor in a call
5             break;
6         case TelephonyManager.CALL_STATE_RINGING:
7             // The phone is ringing
8             break;
9         case TelephonyManager.CALL_STATE_OFFHOOK:
10            // The phone is currently in a call
11            break;
12    }
13 }
14
```

ExampleActivity.java

```
1 public void onCellLocationChanged(CellLocation location) {  
2     GsmCellLocation gsmLocation = (GsmCellLocation) location;  
3     int cellId = gsmLocation.getCid();  
4     int locationAreaCode = gsmLocation.getLac();  
5 }  
6
```

ExampleActivity.java

```
1 public void onDataConnectionStateChanged(int state) {
2     switch (state) {
3         case TelephonyManager.DATA_CONNECTED:
4             // Connected
5             break;
6         case TelephonyManager.DATA_CONNECTING:
7             // Setting up data connection
8             break;
9         case TelephonyManager.DATA_DISCONNECTED:
10            // Disconnected
11            break;
12        case TelephonyManager.DATA_SUSPENDED:
13            // Suspended
14            break;
15    }
16 }
17
```

ExampleActivity.java

```
1 public void onDataActivity(int direction) {  
2     switch (direction) {  
3         case TelephonyManager.DATA_ACTIVITY_IN:  
4             // Receiving IP PPP traffic  
5             break;  
6         case TelephonyManager.DATA_ACTIVITY_OUT:  
7             // Sending IP PPP traffic  
8             break;  
9         case TelephonyManager.DATA_ACTIVITY_INOUT:  
10            // Receiving and sending data traffic  
11            break;  
12        case TelephonyManager.DATA_ACTIVITY_NONE:  
13            // No traffic  
14            break;  
15    }  
16 }  
17
```


Agenda

- 1 Radio komunikacije
- 2 Celularna mreža
- 3 Telefonija
- 4 SMS

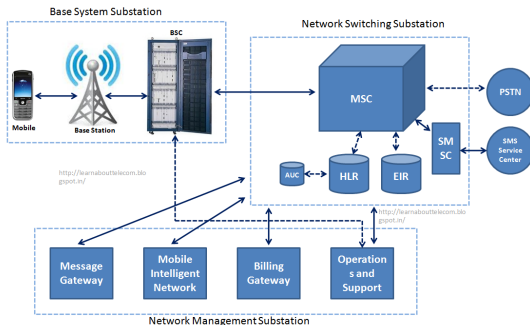
SMS

- Short Message Service (SMS) je tehnologija koja omogućava slanje i primanje kratkih poruka između mobilnih telefona.
- Podržavaju je (skoro) svi mobilni telefoni.
- Jedna SMS poruka može da sadrži najviše 140 bajtova (160 znakova ukoliko se koristi 7-bitno kodiranje, 70 znakova ukoliko se koristi 16-bitno kodiranje).
- Pored teksta, SMS poruke mogu da prenose i binarne podatke.

PDU mode SMS

- Konkatenirane SMS poruke ili PDU (protocol data unit) mode SMS poruke mogu da sadrže više od 140 bajtova.
- Mobilni telefon pošiljaoca deli dugačku poruku u manje delove i šalje svaki deo kao pojedinačnu SMS poruku.
- Mobilni telefon primaoca spaja pojedinačne SMS poruke u dugačku poruku.

Arhitektura SMS sistema



- Short Message Service Center (SMSC)
- SMS Gateway

Figure 9: Arhitektura SMS sistema.

SMSC

- Short Message Service Center (SMSC) upravlja SMS operacijama celularne mreže (njegova glavna funkcija je rutiranje SMS poruka).
- Kada mobilni telefon pošalje SMS poruku, ona stiže u SMS centar. SMS centar prosleđuje SMS poruku primaocu.
- Ako je primalac nedostupan (mobilni telefon je isključen ili nema domet), SMS centar skladišti poruku i prosleđuje je primaocu kada postane dostupan.

SMS Gateway

- Pre nego što stigne do odredišta, SMS poruka može da prođe kroz SMS gateway i druge SMS centre.
- SMS Gateway omogućava različitim operaterima mobilne telefonije da povežu SMS centre i razmenjuju SMS poruke.

Osnovni koncepti SMS-a

- Message validity period je period posle koga će SMS centar obrisati SMS poruku ako u međuvremenu ne bude prosleđena primaocu.
- Kada SMS centar primi SMS poruku od pošiljaoca, poslaće pošiljaocu message submission report i obavestiti ga da li je došlo do greške prilikom primanja poruke (zbog neodgovarajućeg formata SMS poruke, zauzetosti SMS centara, itd.).
- Mobilni telefon primaoca će poslati message delivery report SMS centru kada primi SMS poruku i obavestiti ga da li je došlo do greške prilikom primanja poruke (zbog nepodržanog formata SMS poruke, nedovoljno prostora za skladištenje poruke, itd.). Ako je pošiljaoc zatražio message status report, SMS centar će mu poslati message status report kada primi message delivery report od primaoca.

MMS

- Multimedia Messaging Service (MMS) je proširenje SMS-a.
- MMS omogućava formatiranje teksta i slanje i primanje multimedijalnih poruka (poruka koje sadrže fotografije, audio i video).

SMS i Android

Android API do verzije 4.4 podržava slanje i primanje SMS poruka korišćenjem:

- podrazumevane SMS aplikacije ili
- SMSManager servisa koji omogućava zamenu podrazumevane SMS aplikacije za slanje i primanje SMS poruka ili korišćenje SMS protokola kao transportnog sloja.

ExampleActivity.java

```
1 public void sendSMS() {  
2     Intent smsIntent = new Intent(Intent.ACTION_SENDTO, Uri.parse("sms:+381641234567"))  
3     ;  
4     smsIntent.putExtra("sms_body", "Please call me as soon as possible");  
5     startActivity(smsIntent);  
6 }
```

ExampleActivity.java

```
1 public void sendMMS() {  
2     // Get the URI of a piece of media to attach.  
3     Uri attachedUri = Uri.parse("content://media/external/images/media  
4     /1");  
5     // Create a new MMS intent  
6     Intent mmsIntent = new Intent(Intent.ACTION_SEND, attachedUri);  
7     mmsIntent.putExtra("sms_body", "Please see the attached image");  
8     mmsIntent.putExtra("address", "+381641234567");  
9     mmsIntent.putExtra(Intent.EXTRA_STREAM, attachedUri);  
10    mmsIntent.setType("image/png");  
11    startActivity(mmsIntent);  
12 }
```

AndroidManifest.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest ... >
3   <application ... >
4     <uses-permission android:name="android.permission.SEND_SMS"/>
5   </application>
6 </manifest>
7
```

MMS

```
1 public sendSMS() {  
2     SmsManager smsManager = SmsManager.getDefault();  
3     String to = "+381641234567";  
4     String message = "Android supports programmatic SMS messaging!";  
5     smsManager.sendTextMessage(to, null, message, null, null);  
6 }  
7
```

SMS

| Parameters | Meaning |
|----------------------|---|
| destinationAddress | The address to send the message to |
| serviceCenterAddress | The service center address (or null to use the default SMSC). |
| text | The body of the message to send. |
| sentIntent | If not null this PendingIntent is broadcast when the message is successfully sent, or failed. |
| deliveryIntent | If not null this PendingIntent is broadcast when the message is delivered to the recipient. |

Table 1: Parametri sendTextMessage metode.

MMS/Data/Multipart Messages

- `sendDataMessage` (sends a data based SMS to a specific application port)
- `sendMultimediaMessage` (sends an MMS message)
- `sendMultipartTextMessage` (sends a multi-part text based SMS)

AndroidManifest.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest ... >
3   <application ... >
4
5     <uses-permission android:name="android.permission.RECEIVE_SMS"/>
6
7     <receiver android:name=".SMSReceiver">
8       <intent-filter>
9         <action name="android.provider.Telephony.SMS_RECEIVED" />
10      </intent-filter>
11    </receiver>
12
13  </application>
14 </manifest>
15
```


SMSReceiver.java

```

1 public class SMSReceiver extends BroadcastReceiver {
2
3     private static final String SMS_RECEIVED = "android.
        provider.Telephony.SMS_RECEIVED";
4
5     @Override
6     public void onReceive(Context context, Intent intent)
7     {
8         if (intent.getAction().equals(SMS_RECEIVED)) {
9             SmsManager smsManager = SmsManager.getDefault();
10            Bundle bundle = intent.getExtras();
11            if (bundle != null) {
12                Object[] pdus = (Object[]) bundle.get("pdus");
13                SmsMessage[] messages = new SmsMessage[pdus.
length];
14                for (int i = 0; i < pdus.length; i++) {
15                    messages[i] = SmsMessage.createFromPdu((byte
[]) pdus[i]);
16                }
17                for(SmsMessage message: messages) {
18                    String text = message.getMessageBody();
19                    String to = message.getOriginatingAddress();
20                    smsManager.sendTextMessage(to, null, text,
null, null);
21                }
22            }
23        }
24    }
25 }
26

```

SMS

- Android API od verzije 4.4 uvodi pojam podrazumevane SMS aplikacije.
- Samo podrazumevana SMS aplikacija može da pristupi Telephony dobavljaču sadržaja (koji, između ostalog, skladišti SMS poruke).
- Podrazumevana SMS aplikacija bi trebalo da deklariše prijemnike SMS i MMS poruka, omogući drugim aplikacijama da pomoću nje šalju SMS i MMS poruke i omogući odgovor na dolazne telefonske pozive SMS porukom.

AndroidManifest.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest ... >
3   <application ... >
4
5     <uses-permission android:name="android.permission.BROADCAST_SMS"
6       />
7
8     <!-- BroadcastReceiver that listens for incoming SMS messages -->
9     <receiver android:name=".SMSReceiver">
10       <intent-filter>
11         <action android:name="android.provider.Telephony.SMS_DELIVER"
12           />
13       </intent-filter>
14     </receiver>
15
16   </application>
17 </manifest>
```

AndroidManifest.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest ... >
3   <application ... >
4
5     <uses-permission android:name="android.permission.BROADCAST_WAP_PUSH"
6       />
7
8     <!-- BroadcastReceiver that listens for incoming MMS messages -->
9     <receiver android:name=".MMSReceiver">
10       <intent-filter>
11         <action android:name="android.provider.Telephony.WAP_PUSH_DELIVER"
12           />
13         <data android:mimeType="application/vnd.wap.mms-message" />
14       </intent-filter>
15     </receiver>
16
17   </application>
18 </manifest>
```

AndroidManifest.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest ... >
3   <application ... >
4
5     <activity android:name=".ComposeSMSActivity" >
6       <intent-filter>
7         <action android:name="android.intent.action.SEND
8           " />
9         <action android:name="android.intent.action.
10          SENDTO" />
11         <category android:name="android.intent.category.
12          DEFAULT" />
13         <category android:name="android.intent.category.
14          BROWSABLE" />
15         <data android:scheme="sms" />
16         <data android:scheme="smsto" />
17         <data android:scheme="mms" />
18         <data android:scheme="mmsto" />
19       </intent-filter>
20     </activity>
21
22   </application>
23 </manifest>
```

AndroidManifest.xml

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest ... >
3   <application ... >
4
5     <uses-permission android:name="android.permission.SEND_RESPOND_VIA_MESSAGE"
6     />
7
8     <!-- Service that delivers messages from the phone "quick response" -->
9     <service android:name=".HeadlessSMSSendService" android:exported="true" >
10       <intent-filter>
11         <action android:name="android.intent.action.RESPOND_VIA_MESSAGE" />
12         <category android:name="android.intent.category.DEFAULT" />
13         <data android:scheme="sms" />
14         <data android:scheme="smsto" />
15         <data android:scheme="mms" />
16         <data android:scheme="mmsto" />
17       </intent-filter>
18     </service>
19
20   </application>
21 </manifest>

```

Emulator Control

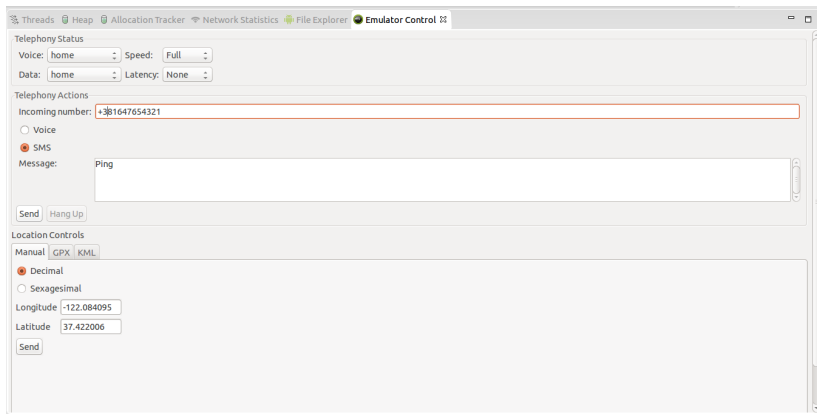


Figure 10: Emulator Control.

Reference

- [1] Kovačević, M. Đ., "Pregled aktuelnih tehnologija za mobilne i širokopojasne bežične komunikacije", Telekomunikacije
- [2] SMS and PDU format,
http://www.smartposition.nl/resources/sms_pdu.html

