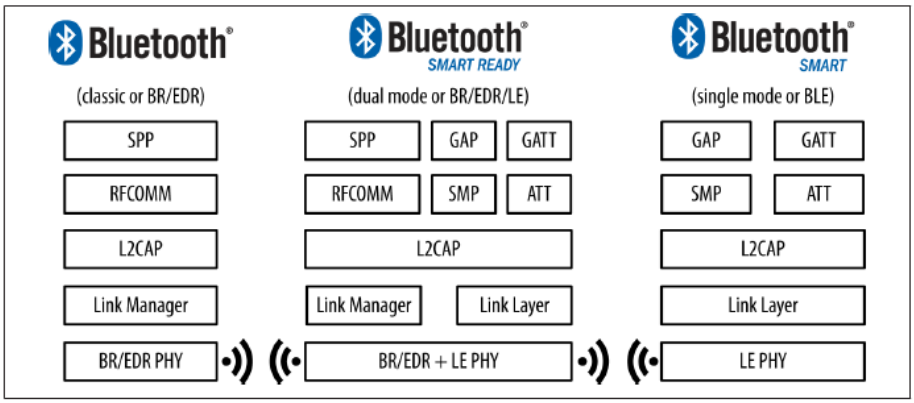
8 Bluetooth Low Energy

Bluetooth je standard za bežičnu komunikaciju kratkog dometa u ISM frekventnom opsegu ($2.4 - 2.48\ GHz$). Prva oficijelna verzija Bluetooth standarda (verzija 1.0) je objavljena 1994. godine od strane kompanije *Ericsson*, a njegova prvobitna namenjen je bila zamena za kablovsko povezivanje portabilnih i desktop uređaja (npr. miša i tastature sa desktop ili laptop računarom). Ubrzo nakon toga, ovaj standard je prešao u nadležnost organizacije *Bluetooth Special Interest Group* (SIG) i više puta je unapređivan uz očuvanje kompatabilnosti sa starijim verzijama. Počev od verzije 4.0, Bluetooth specifikacija obuhvata dve međusobno nekompatibilne bežične tehnologije: *Bluetooth Classic* (BR/EDR) i *Bluetooth Low Energy* (BLE). Standard BR/EDR je zadržao kompatabilnost sa starijim verzijama i koristi se u primenama koje zahtevaju veliki protok podataka (npr. bežični zvučnici i slušalice, medicinski uređaji, infotejment sistemi u automobilima i slično). S druge strane, BLE se odlikuje malom potrošnom energije, ali na račun manje brzine prenosa podataka i tipično se sreće kod baterijski napajanih uređaja kod kojih postoji potreba samo za sporadičnom komunikacijom radi prenosa male količine podataka (npr. senzorski uređaji).

Bluetooth varijante BR/EDR i BLE nisu međusobno kompatabilne uprkos tome što su obuhvaćene istim standardom i dele isti frekventni opseg. To znači da BR/EDR uređaj ne može direktno da komunicira sa BLE uređajem. Iz tog razloga, pojedini uređaji, poput savremenih mobilnih telefona, tablet i laptop računara, poseduju podršku za obe Bluetooth varijante (tzv. *Dual-mode Bluetooth* uređaji), što im omogućava komunikaciju sa oba tipa Bluetooth uređaja (Sl. 1).



Sl. 1 Bluetooth varijante i uređaji

Dakle, danas postoje dve Bluetooth varijante:

- BR/EDR (*Bluetooth Classic*) – bežični standard koji je evoluirao počev od Bluetooth specifikacije verzije 1.0. Klasični Bluetooth nije optimizovan za malu potrošnju, ali zato podržava veću brzinu prenosa podataka (maksimalno $3\ Mbps$ u poređenju sa $2\ Mbps$ kod BLE 5.0). Pogodan je za striming aplikacije, kao što su audio striming i prenos fajlova. (*Streaming* je reprodukcija tokom preuzimanja i odnosi se na mogućnost preuzimanja audio i videa sadržaja uz istovremenu reprodukciju. Sadržaj se preuzima preko Interneta (npr. *YouTube*, *Spotify*, *Netflix* i sl.), ili, što je karakteristično za Bluetooth, od drugog bliskog uređaja, npr. prenos audio sadržaja od pametnog telefona do bežičnih slušalica.)

- BLE (*Bluetooth Low Energy*) – bežični standard male potrošnje uveden u verziji 4.0 Bluetooth specifikacije.

i tri tipa Bluetooth uređaja:

- BR/EDR uređaj - klasični Bluetooth uređaj sa podrškom samo za BR/EDR.
- *Single-mode* (BLE) uređaj – podrška samo za BLE. Može da komunicira sa drugim BLE i *dual-mode* uređajima, ali ne i sa BR/EDR uređajima.
- *Dual-mode* (*BR/EDR/LE* ili *Bluetooth Smart Ready*) uređaj – podrška za BR/EDR i BLE. Može da komunicira sa Bluetooth uređajem bilo kog tipa.

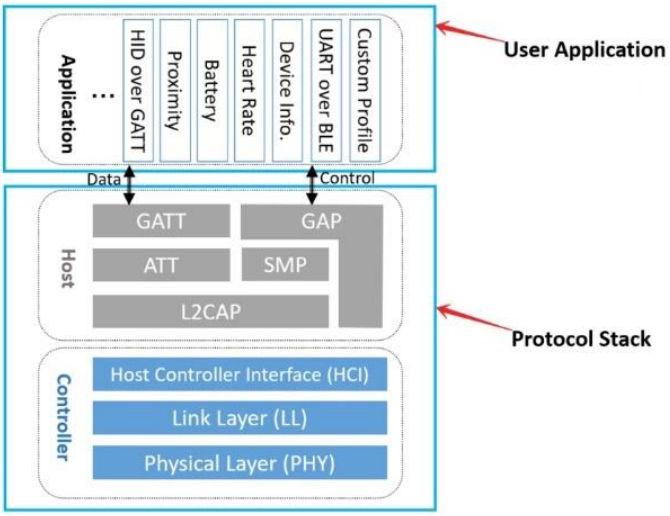
Naš fokus u ovom predmetu je BLE i *single-mode* BLE uređaji. Od 2014. godine, kada je uvedena Bluetooth verzija 4.0, do danas (2025), BLE je prošao kroz još nekoliko revizija. Godine 2016 objavljena je Bluetooth specifikacija 5.0, sa posebnim fokusom na dalje unapređenje BLE. Najnovija i trenutno aktuelna verzija nosi oznaku 5.4 (uvedena 2023. godine). Sve BLE verzije su međusobno kompatabilne, s tom što je, u tom slučaju, komunikacija ograničena na mogućnosti starije od dve verzije. Tako, BLE 5 uređaj može da komunicira sa BLE 4 uređajem, pri čemu će u toj komunikaciji biti omogućene samo one opcije koje podržava BLE 4 uređaj. *Single-mode* BLE senzori su danas široko rasprostranjeni i zajedno sa *dual-mode* Bluetooth uređajima predstavljaju osnovu za formiranje mnogih IoT sistema.

8.1 Uvod

8.1.1 Arhitektura

BLE, kao većina komunikacionih standarda, predstavlja kolekciju protokola razvrstanih u nekoliko „horizontalnih“ slojeva (Sl. 2). Kao što znamo, ovakva komunikaciona arhitektura se naziva protokol stek. Svaki sloj u protokol steku koristi usluge sloja ispod i pruža usluge sloju iznad. Sloj na dnu je odgovoran za fizički prenos podataka, dok sloj na vrhu obezbeđuje interfejs korisničkoj aplikaciji. Pojedinačni slojevi BLE protokol steka biće obrađenu u narednim poglavljima ovog dokumenta. Za sada, uvešćemo tri osnovna gradivna bloka svakog BLE uređaja:

- *Aplikacija*: korisnička aplikacija koja se, za potrebe komunikacije, oslanja na BLE protokol stek.
- *Host*: viši slojevi BLE protokol steka, koji se, po pravilu, realizuju u softveru.
- *Kontroler* (*Controller*): obuhvata najniže slojeve BLE protokol steka, uključujući i RF primopredajnik. Najvećim delom, realizuje se u hardveru.

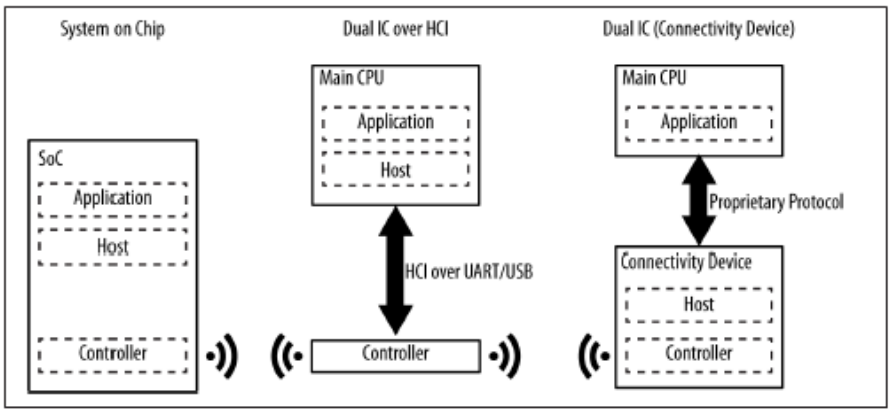


Sl. 2 BLE arhitektura

Sloj *Host Controller Interface* (HCI), koji je prikazan na vrhu dela kontroler zapravo predstavlja standardni komunikacioni protokol između hosta i kontrolera, koji omogućava interoperabilnost (tj. zajednički rad) hostova i kontrolera različitih proizvođača.

Tri gradivne celine BLE-a se mogu realizovati u okviru jednog integrisanog kola (IC), ili se mogu podeliti na nekoliko IC tipično povezanih standardnim serijskim komunikacionim interfejsom, poput I2C, SPI ili UART. Kod komercijalno dostupnih BLE uređaja sreću se sledeće tri konfiguracije (Sl. 3):

- *SoC* (*system on chip*) – aplikacija, host i kontroler su realizovani na jednom IC.
- *Dva IC sa HCI* – korisnička aplikacija i Bluetooth host se izvršavaju na jednom, a Bluetooth kontroler na drugom IC. Dva IC komuniciraju putem HCI. S obzirom da je HCI standardizovan, ovaj pristup omogućava da se host može kombinovati sa bilo kojim kontrolerom.
- *Dva IC sa nestandardnom komunikacijom* – aplikacija se izvršava na jednom IC, a Bluetooth host i kontroler na drugom. Pošto Bluetooth specifikacija ne definiše protokol za komunikaciju između aplikacije i hosta, aplikacija se mora prilagoditi protokolu proizvođača „host-kontroler“ IC-a.



Sl. 3 Bluetooth konfiguracije

SoC konfiguracija se, po pravilu, koristi kod jednostavnih, senzorskih uređaja, kako bi se smanjila veličina štampane ploče i cena uređaja. Razvojni sistem koji koristimo na laboratorijskim vežbama iz ovog predmeta zasnovan je upravo na ovom konceptu. Konfiguracija *dual*-IC sa HCI se najčešće sreće kod pametnih telefona i tablet računara, s obzirom da ovi uređaji sadrže moćan CPU, koji bez velikog opterećenja može da izvršava host deo Bluetooth protokol steka. Treća konfiguracija je pogodna za laku nadogradnju postojećih uređaje Bluetooth konekcijom. Komunikacija između uređaja i Bluetooth IC se ostvaruje putem standardnog UART, I2C ili USB porta.

8.1.2 Topologije

BLE uređaj može da komunicira sa okolnim svetom na dva načina: *emitovanje* (*broadcasting*) i *konekcija* (*connection*). Ova dva načina komunikacije ima svoje specifičnosti, prednosti i nedostatke.

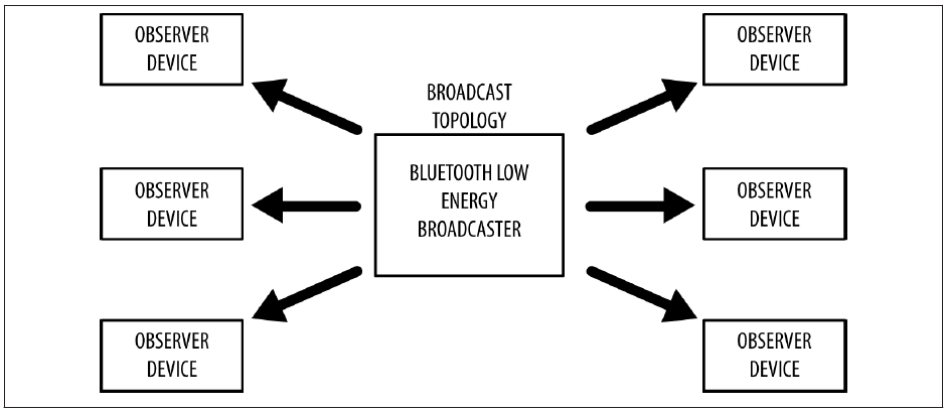
Emitovanje i osmatranje (*Broadcasting and Observing*)

Emitovanje se odnosi na mogućnost jednosmernog i neselektivnog slanja podataka ka svim uređajima ili prijemnicima u radio dometu (Sl. 4). U ovakvom načinu komunikacije svaki uređaj ima jednu od dve uloge:

- *Broadcaster*, ili *emiter*, koji periodično šalje pakete za oglašavanje (*advertising packets*) svim okolnim uređajima koji su voljni da ih prime.

- *Observer*, ili *osmatrač*, koji periodično skenira radio kanale na predodređenim frekvencijama da bi primio pakete za oglašavanje koji se trenutno emituju.

Emitovanje je jedini način kako BLE uređaj može istovremeno da šalje iste podatke ka više BLE uređaja. Takođe, emitovanje je beskonekcioni način komunikacije, jer ne zahteva prethodno usaglašavanje i uspostavljanje konekcije između učesnika u komunikaciji. Sve šta ima da kaže, emiter smešta u jedan paket koji emituje svima.



Sl. 4 *Broadcast* topologija

Standardni paket za oglašavanje sadrži 31 bajt. Jedan deo ovog paketa je rezervisan za podatke koji identifikuju emitera, dok preostali deo može da sadrži proizvoljnu informaciju, tj. onu koju emiter želi da saopšti drugim uređajima. Ukoliko standardni 31-bajtni paket za oglašavanje nije dovoljan za sve potrebne podatke, BLE predviđa mogućnost da uređaj-osmatrač, neposredno nakon prijema paketa za oglašavanje, zatraži od emitera sekundarne podatke o oglašavanju – tzv. *Scan Response*. *Scan Response* može da sadrži maksimalno 254 bajta.

Emitovanje je brz i jednostavan način za periodičnu distribuciju male količine podataka ka jednom ili više uređaja. Glavno ograničenje u primeni emitovanja jeste nedostatak podrške za sigurnost podataka i privatnost (svaki osmatrač može da primi emitovane podatke, a uz to podaci nisu ni šifrovani).

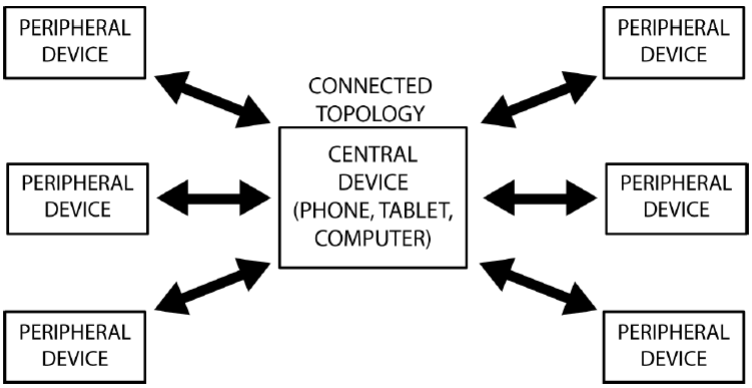
Konekcija (Connection)

Konekcija je neophodna onda kada se zahteva prenos podatke u oba smera ili kada treba preneti veću količinu podataka od one koja se može preneti mehanizmom oglašavanja. Konekcija je permanentna, periodična razmena paketa između *dva* uređaja. Komunikacija putem konekcije je inherentno privatna - podatke mogu da šalju i primaju samo dva uređaja koja su uključena u konekciju.

Dva uređaja koja učestvuju u konekciji imaju različite uloge, centrala i periferija:

- *Central*, *centrala* ili *centralni uređaj* skenira komunikacioni kanal na predodređenim frekvencijama u očekivanju prijema paketa za oglašavanje. Ukoliko paket za oglašavanje nosi indikaciju o tome da je pošiljalac paket dostupan za konektovanje, centralni uređaj može, ako želi, da inicira uspostavljanje konekcije. Nakon što je konekcija uspostavljena, centralni uređaj reguliše tajming konekcije i inicira periodičnu razmenu podataka sa perifernim uređajem.
- *Peripheral*, *periferija* ili *periferni uređaj* periodično šalje pakete za oglašavanje sa indikacijom o spremnosti za konektovanje. Nakon što primi i prihvati zahtev za konektovanje, periferni uređaj prati tajming centralnog uređaja i s njim razmenjuje podatke.

Dakle, da bi inicirao ekskluzivnu konekciju sa perifernim uređajem, centralni uređaj prvo mora da primi paket za oglašavanje od tog uređaja, a zatim da mu pošalje zahtev za uspostavljane konekcije. Nakon što je konekcije uspostavljena, periferni uređaj prestaje sa oglašavanjem i dva uređaja mogu da počnu sa razmenom podataka u oba smera.



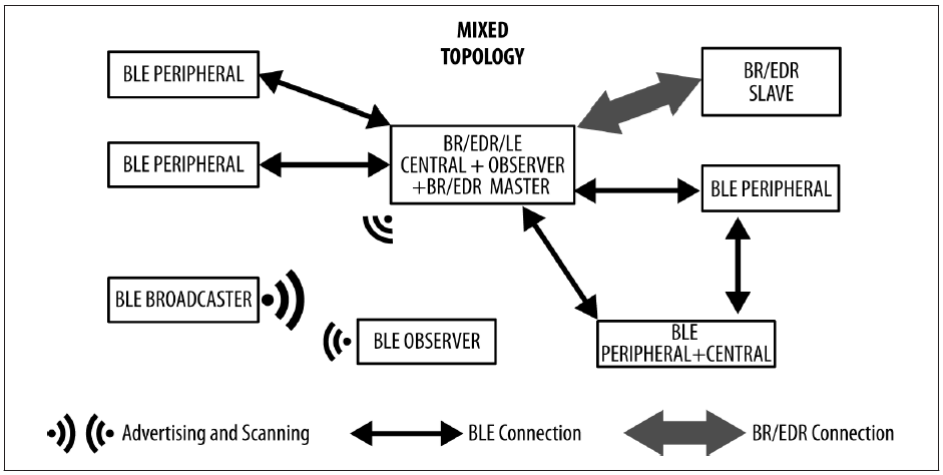
Sl. 5 Konekciona topologija

U suštini, konekcija nije ništa više od periodične razmene podataka između dva uređaja. Razmena podataka nije kontinualna, već je „sabijena“ u okviru kratkotrajnih, periodičnih vremenskih intervala, tzv. *connection events*. Uprkos tome što centralni uređaj upravlja uspostavljanjem konekcije, dva uređaja su potpuno ravnopravna tokom prenosa podataka – svaki uređaj nezavisno odlučuje da li će i kada slati podatke i u kom obimu. U tom pogledu ne postoje prioriteti niti restrikcije u propusnosti koja je na raspolaganju jednom i drugom uređaju. Takođe, postojanje konekcije između dva uređaja ne sprečava ni jedan od njih da uspostavlja konekcije sa drugim uređajima. Drugim rečima, BLE uređaj može istovremeno biti uključen u više konekcija u različitim ulogama, i to:

- Uređaj može istovremeno igrati ulogu centralnog i perifernog uređaja
- Centralni uređaj može uspostaviti konekciju sa više perifernih uređaja
- Periferni uređaj može biti konektovan sa više centralnih uređaja.

Glavna prednost konekcije, u odnosu na emitovanje, ogleda se u potpunijoj i finijoj kontroli prenosa podataka, što doprinosi uštedi energije koja se troši na komunikaciju i olakšava uvođenje dodatnih servisa, poput zaštite podataka i pregovaranja oko parametara konekcije. Na primer, periferni uređaj, kroz konekciju može da šalje podatke samo onda kada su nove vrednosti dostupne, za razliku od emitovanja gde uređaj mora neprekidno da šalje jedan te isti podatak čak iako se njegova vrednost ne menja (jer ne zna ko i koliko često sluša). Uz to, učesnici u konekciji su sinhronizovani, u smislu da svako od njih zna u kom trenutku počinje sledeći *connection event*, što im omogućava da isključe RF primopredajnike u vremenu između dva *connection event*-a i tako uštede energiju.

Konačno, dve BLE topologije, emitovanje i konekcija, se mogu kombinovati, kao što je ilustrovano na Sl. 6. *Dual-mode* uređaj može imati ulogu mosta između single-mode i BR/EDR uređaja. Praktično, broj kombinacija i učesnika u mreži ograničen je jedino ograničenjima RF primopredajnika i protokol-steka svakog pojedinačnog uređaja.



Sl. 6 Kombinovana topologija

8.1.3 Protokoli i profili

Bluetooth specifikacija pravi jasnu razliku između koncepta protokola i profila:

- *Protokol* je skup pravila kojima se reguliše razmena podataka između uređaja, sistema ili, generalno, entiteta koji komuniciraju. Bluetooth specifikacija definiše ne samo jedan već nekoliko protokola, koji su razvrstani po slojevima (Sl. 2) i od kojih svaki reguliše jedan specifičan aspekt komunikacije, poput formatiranja, rutiranja i multipleksiranja paketa, kodiranja i dekodiranja podatka.
- *Profili* su na višem nivou od protokola i definišu kako treba koristiti mogućnosti Bluetooth komunikacije i raspoložive protokole da bi se ostvarila konkretna funkcionalnost. Postoje generički i aplikaciono-specifični profili. Generički profili su definisani u samoj Bluetooth specifikaciji i odnose se na načine rada koje moraju da implementiraju svi Bluetooth uređaji (npr. *Generic Access Profile, Generic Attribute Profile*). Aplikaciono-specifični profili su prilagođeni specifičnim zahtevima različitih tipova uređaja i načinima njihovog korišćenja.

Bluetooth protokoli i profili su obrađeni u narednim sekcijama. U ovoj sekciji su ukratko pomenuti glavni generički i aplikaciono-specifični profili.

Generički profili

Glavni zadatak generičkih profila je da obezbede interoperabilnost BLE uređaja različitih proizvođača. Dva osnovna generička profila su:

- *Generic Access Profile* (*GAP*) - definiše kako treba koristiti BLE protokole niskog nivoa (kontroler) da bi se realizovale različite uloge uređaja (npr. *central/peripheral*), procedure i načini rada koji će omogućiti uređajima da emituju podatke, otkrivaju prisustvo drugih uređaja, uspostavljaju konekcije i pregovaraju oko nivoa sigurnosti podataka. GAP je, u osnovi, vršni upravljački nivo BLE-a. Ovaj profil je obavezan za sve BLE uređaje.
- *Generic Attribute Profile* (*GATT*) – reguliše razmenu podataka, odnosno definiše osnovne modele podataka i procedure koje omogućavaju uređajima da među sobom razmenjuju informacije o tome koje podatke poseduju, a onda i da razmenjuju, čitaju i upisuju elemente podataka. U osnovi, GATT je vršni sloj podataka u BLE.

Aplikaciono-specifični profili

Aplikaciono-specifični profili su proširenja GATT profila u cilju prilagođenja konkretnom slučaju upotrebe.

SIG profili

U ovu kategoriju spadaju aplikaciono-specifični profili koji su obuhvaćeni Bluetooth standardom. Njihov kompletan spisak i specifikacija je dostupna veb sajtu organizacije Bluetooth SIG. Ovde je dat skraćeni spisak ovih profila:

- *Find Me Profile* – omogućava uređajima da fizički lociraju jedni druge (koristimo pametni sat da bismo pronašli pametni telefon, ili obrnuto)
- *Proximity Profile* - detektuje prisustvo/odsustvo okolnih uređaja (zvučni signal na telefonu ako prilikom izlaska iz stana zaboravimo da ponesemo ključeve od automobila)
- *HID* (*Human Interface Device*) *over* GATT – prenos HID podataka preko BLE (HID je uređaj poput tastature, miša, daljinskog upravljača,...).
- *Health Thermometer Profile* – prenos očitavanja telesne temperature preko BLE
- *Glucose Profile* - prenos nivoa glukoze preko BLE
- *Cycling Speed and Cadence Profile* – omogućava prenos preko BLE očitavanja senzora za brzinu i broj obrtaja točka bicikle.

Korisnički profili

Bluetooth specifikacija omogućava proizvođačima elektronskih uređaja da definišu svoje sopstvene profile za primene koje nisu obuhvaćene SIG profilima. Ovi profili mogu ostati privatni, ili se mogu publikovati kako bi postali dostupni i drugim proizvođačima. Primera radi, kompanije XY je razvila novi BLE senzor, u obliku prstena, za merenje, recimo, nivoa (psihičkog) stresa. Pošto ne postoji SIG profil za ovakav senzor, kompanija XY je razvila i objavila BLE profil za svoj senzor. To onda omogućava kompanijama YZ i ZY, koje se bave Android programiranjem, da, nezavisno jedna od druge i nezavisno od kompanije XY, razvijaju aplikacije za pametni telefon koje će omogućiti očitavanje, obradu i prikaz podataka o nivou stresa.

Protokoli

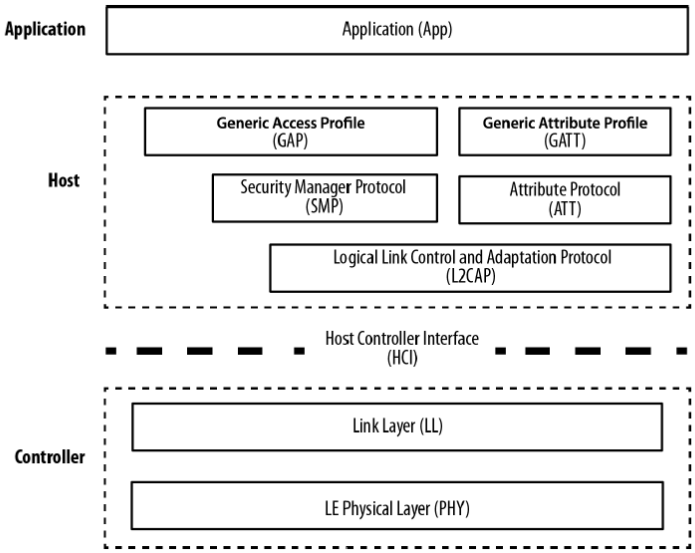
Na Sl. 7 je, ponovo, prikazana arhitektura kompletnog *single-mode* BLE uređaja, koja se sastoji iz tri celine: kontroler, host i aplikacija. Kontroler i host, od kojih svaki obuhvata nekoliko slojeva, zajedno čine BLE protokol stek, koji je delom realizovan u hardveru, a delom u softveru. Aplikacioni deo kreira embeded programer koji razvoja softver za konkretan BLE uređaj.

Host sadrži sledeće slojeve:

- *Generic Access Profile* (*GAP*)
- *Generic Attribute Profile* (*GATT*)
- *Logical Link Control and Adaptation Protocol* (*L2CAP*)
- *Attribute Protocol* (*ATT*)
- *Security Manager* (*SM*)
- *Host Controller Interface* (*HCI*), na strani hosta

Kontroler obuhvata slojeve:

- *Host Controller Interface* (*HCI*), na strani kontrolera
- *Link Layer* (*LL*)
- *Physical Layer* (*PHY*)



Sl. 7 BLE protokol stek

8.1.4 Prednosti, nedostaci i primene

Svaka tehnologija ima svoja ograničenja i u tom pogledu BLE nije izuzetak. Kao što je prethodno napomenuto, BLE je pre svega pogodan za komunikaciju na kratkom rastojanju sa sporadičnim prenosom male količine podataka.

Nedostaci

Propusnost. Propusnost BLE-a je ograničena brzinom prenosa podataka RF primopredajnika, koja zavisi od Bluetooth verzije[1]. Za verzije BLE 4.2 i niže, brzina prenosa podataka je fiksno $1\ Mbps$. Za verzije BLE 5 i kasnije, brzina prenosa zavisi od izabranog načina rad primopredajnika i može biti $1\ Mbps$ ili $2\ Mbps$. Za prenos na veća rastojanja, brzina prenosa se spušta na 500 ili $125\ Kbps$. Međutim, propusnost, odnosno brzina protoka podataka na nivou aplikacije i krajnjeg korisnika, mnogo je manja od brzine prenosa preko bežičnog medijuma. Primera radi, propusnost BLE sa brzinom prenosa podataka od $1\ Mbps$ je reda $125\ Kpbs$, tj. čak 8 puta manja od brzine protoka na fizičkom nivou. Smanjena propusnost je posledica:

- *Pauze između paketa*. Na fizičkom nivou, podaci se prenose u paketima. Bluetooth specifikacija nameće obaveznu pauzi u trajanju od $150\ us$ između svaka dva uzastopna paketa. Tokom ovih pauza, podaci se, naravno, ne prenose.
- *Režijski troškovi* (*packet overhead*). Svaki paket, pored korisnih podataka (onih koji potiči od aplikacije), sadrži obavezno zaglavlje i dodatne podatke koji se umeću u paket na slojevima protokol steka između aplikacionog i fizičkog. Zbog toga, ukupna količina podataka prenetih radio vezom je uvek veća od količine podataka poslatih od strane aplikacije.
- *Slejv paketi*. Nakon uspostavljene konekcije između dva BLE uređaja, jedan od njih igra ulogu mastera a drugi slejva. Obaveza je slejva da na svaki primljeni paket odgovori slanjem svog paketa, čak i u slučajevima kada nema svoje korisne podatke za slanje.

[1] *Propusnost* (*network throughput*) je količina podataka koja se u jedinici vremena može uspešno preneti između dve krajnje tačke u komunikacionoj mreži. Za razliku od toga, *bitski protok*, ili *brzina protoka*, se odnosi na brzinu prenosa podataka preko fizičkog komunikacionog medijuma. Propusnost i bitski protok se izražavaju u jedinicama bita-u-sekundi (*bps*, odnosno *Kbps*, *Mbps* i *Gbps*). Propusnost je uvek manja od bitskog protoka. Bitski protok je teorijski maksimalna, dok je propusnost stvarna, tj. ostvarena brzina prenosa podataka.

- *Retransmisija paketa.* U slučaju gubitka ili greške u prenosu paketa, pošiljalac paketa je u obavezi da taj paket ponovo pošalje.

Domet. BLE je projektovan za komunikaciju između uređaja na malom rastojanju i zbog toga je ograničen i radio-domet. Osnovni faktori koji dodatno ograničavaju domet BLE komunikacije su:

- Bluetooth komunikacija se najčešće ostvaruje u zatvorenom prostoru gde prisustvo zidova, metalnih objekata i vode (npr. one u ljudskom telu) značajno utiče na prostiranje radio signala u $2.4\ GHz$ ISP frekventnom opsegu. Na otvorenom prostoru, pod pretpostavkom optičke vidljivosti između uređaja, BLE radio domet dostiže i više od $100\ m$. U zatvorenom prostoru, radio domet je značajno manji i tipično iznosi oko $10\ m$.
- Performanse i dizajn antene BLE uređaja.
- Dimenzije, oblik i tip materijala fizičkog kućišta uređaja utiču na performanse antene naročito ako se koristi interna antena (smeštena u kućištu uređaja).
- Orijentacija uređaja, koja zapravo određuje međusobnu orijentaciju antena predajnog i prijemnog uređaja.

Gateway (mrežni prolaz). U mnogim primenama, potrebno je podatke koje generiše BLE uređaj preneti na Internet, ili je potrebno BLE uređaju isporučiti podatke koji su mu poslati preko Interneta. Ukoliko uređaj poseduje samo BLE, neophodan je još jedan uređaj (tzv. mrežni prolaz, ili *gateway*), koji će pored BLE posedovati i IP konekciju (WiFi ili Ethernet). Zadatak *gateway*-a je da podatke primljene preko BLE usmerava na Internet, odnosno da podatke sa Interneta prosleđuje BLE uređaju.

Prednosti

Uprkos navedenim nedostacima, BLE poseduje i značajne prednosti u odnosu na slične IoT radio tehnologije. Neke od prednosti su:

- *Mala potrošnja energije*. BLE je optimizovan za malu potrošnju, koja se postiže time što se preko BLE prenosi mala količina podataka relativno malom brzinom prenosa i pri tom se RF primopredajnik drži u isključenom stanju koliko god je to moguće, odnosno uključuje se samo onda kada je potrebno poslati ili primiti paket podataka.
- *Otvorena specifikacija*. BLE specifikacija je besplatna i dostupna za slobodno korišćenje (na veb sajtu organizacije Bluetooth SIG).
- *Niska cena BLE modula i IC-ova*.
- *Rasprostranjenost*. Većina savremenih pametnih telefona poseduje BLE. Ovo je možda i najveća prednost BLE u poređenju sa konkurentskim radio tehnologijama, kao što su *ZigBee*, *Z-Wave* i *Thread*.

Primene

Polazeći od navedenih prednosti i nedostataka, lako se mogu identifikovati oblasti kod kojih primena BLE ima najviše smisla:

- *Mali protok podataka*. BLE je optimalan izbor u slučajevima kada uređaj prenosi malu količinu podataka, npr. očitavanja senzora ili jednostavna kontrola aktuatora.
- *Konfiguracija uređaja*. Čak i u slučajevima kada propusnost BLE-a ne može da zadovolji komunikacione potrebe sistema, BLE se i dalje može koristiti kao sekundarni interfejs za konfiguraciju uređaja, pre nego što se uspostavi glavna bežična konekcija. Na primer, nakon reseta ili prvog uključena, aktivan je samo BLE, koji se koristi za podešavanje parametara WiFi konekcije (mreža i šifra). Nakon toga, BLE se isključuje i celokupan saobraćaj se usmerava na WiFi.

- *Pametni telefon kao grafički korisnički interfejs*. Mali uređaji obično ne poseduju displej, ili je njihov displej malih dimenzija i može da prikaže korisniku samo elementarne informacije. Alternativno rešenje je da se korisnički grafički interfejs malog uređaja realizuje na pametnom telefonu, pri čemu se za komunikaciju između uređaja i telefona koristi BLE. Dodatna beneficija korišćenja pametnog telefona je mogućnost da se podaci primljeni od uređaja prenesu na Internet.
- *Lični i nosivi uređaji*. Misli se na portabilne uređaje poput pametnog sata ili fitnes-trakera, koji se zajedno sa osobom koja ih nosi mogu naći u oblasti koja nije pokrivena stalnom bežičnom vezom (poput WiFi). U takvim slučajevima, BLE se može koristiti za komunikaciju uređaj – pametni telefon.
- *Uređaji samo za emitovanje* (*broadcast only devices*), ili *Beacon*-uređaji su jednostavni baterijski napajani uređaji koji periodično emituju svoje podatke, tako da drugi uređaji mogu da otkriju njihovo prisustvo i preuzmu poslate podatke. Podaci koje emituju *beacon* uređaji se tipično odnose na lokaciju uređaja ili senzorska očitavanja (npr. temperatura). Uprkos tome što se za ovu vrstu primene koriste se i druge radio-tehnologije, BLE je ipak najpopularniji izbor budući da većina ljudi sa sobom nosi pametne telefone koji danas, po pravilu, poseduju podršku za BLE.

Primena BLE nije pogodna u sledećim oblastima:

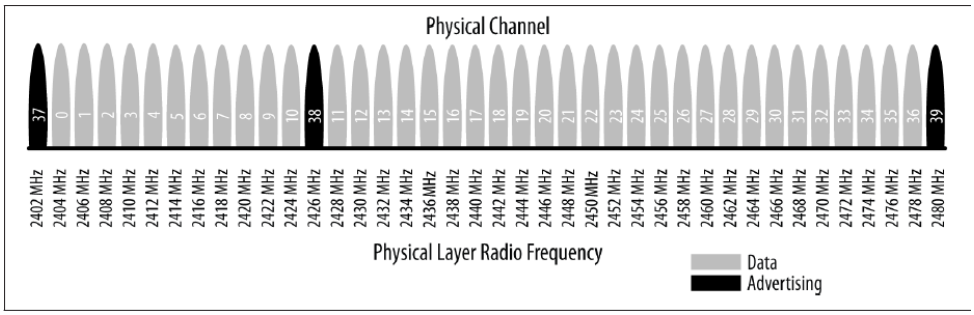
- Video striming
- Audio striming visokog kvaliteta
- Prenos velike količine podataka u dužem vremenskom periodu (zbog povećanog utroška energije, pa time i brzog pražnjenja baterije).

8.2 Fizički sloj (*Physical Layer - PHY*)

Fizički sloj BLE protokol steka je implementiran u RF primopredajniku, koji je zadužen za modulaciju/demodulaciju i predaju/prijem radio signala.

8.2.1 Frekventni opseg i kanali

BLE koristi $2.4\ GHz$ ISM frekventni opseg, odnosno opseg frekvencija od $2400\ MHz$ do $2483.5\ MHz$. BLE deli ovaj opseg u 40 kanala širine $1\ MHz$ sa razmakom od $2\ MHz$ između susednih kanala. Kanali su numerisani kako na Sl. 8. Kanali 37, 38 i 39 se koriste za oglašavanje, dok se preostalih 37 koriste za prenos podataka. Na kanalima za oglašavanje se prenose paketi za oglašavanje, kao i paketi za uspostavljanje konekcije. Tri kanala za oglašavanje su razmaknuti kako bi se ublažio problem interferencije sa WiFi i drugim radio tehnologijama koje koriste ISM frekventni opseg. Na primer, širina WiFi frekventnog kanala je 20 do $40\ MHz$, što znači da emitovanje bliske WiFi stanice može da se preklopi sa najviše dva od tri BLE kanala za oglašavanje.



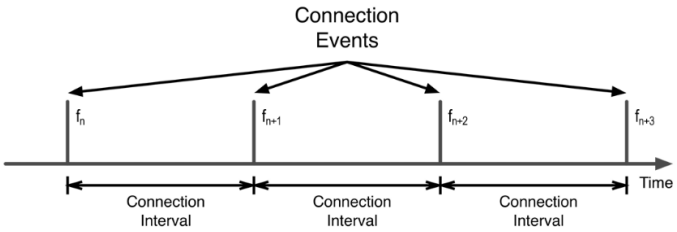
Sl. 8 Frekventni kanali

8.2.2 Snaga zračenja (*Transmit Power*)

- Maksimum: $100\ mW$ $(+20\ dBm)$ za verzije ≥ 5.0 i $10\ mW$ $(+10\ dBm)$, za verzije ≤ 4.2
- Minimum: $0.01\ mW$ $(-20\ dBm)$

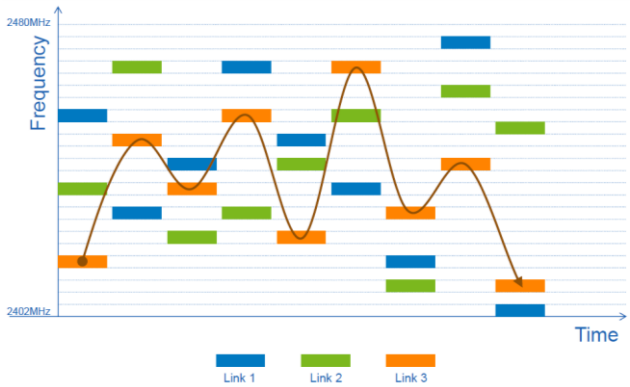
8.2.3 Frekventno skakanje (*Frequency Hopping - FH*)

Da bi se izbegla interferencija sa drugim uređajima iz ISM opsega, BLE koristi tehniku frekventnog skakanja (*frequency hopping*) , gde se RF signal prebacuje sa jedne na drugu frekvenciju nakon svake transmisije. Na taj način, s obzirom na to da BLE transmisija ne ostaje dugo na jednom kanalu, izbegava se trajna interferencija sa drugim uređajima. Da bi FH funkcionisao, neophodno je da su promene frekventnog kanala dva uređaja koji učestvuju u komunikaciji sinhronizovane, u smislu da se promene dešavaju u istim trenucima i da oba uređaja uvek biraju isti novi kanal. Konkretno, kod BLE, skakanje frekvencije se dešava na početku svakog *connection event*-a, pri čemu se novi frekventni kanal bira iz skupa kanala za prenos podataka (0,1,...,36) i to korišćenjem algoritma za izbor kanala (*channel selection algorithm*), Sl. 9. Ovaj algoritam funkcioniše na sledeći način: Neka je $f_n \in \{0,1,\dots,36\}$ frekventni kanal koji se koristi u tekućem *connection event*-u. Za sledeći *connection event* bira se kanal: $f_{n+1} = (f_n + hop)\ mod\ 37$, gde je vrednost numeričkog parametra hop usaglašena između dva uređaja prilikom uspostavljana konekcije i ostaje konstanta tokom trajanja konekcije. Svaka sledeća konekcija imaće svoje novo hop.



Sl. 9 *Connection event*-i i *connection interval*-i. Između konektovanih BLE uređaja, paketi se razmenjuju samo tokom kratkotrajnih *connection event*-a. *Connection interval* je perioda *connection event*-a i predstavlja parametar konekcije. U svakom *connection event*-u koristi se drugi frekventni kanal, određen algoritmom za izbor kanala

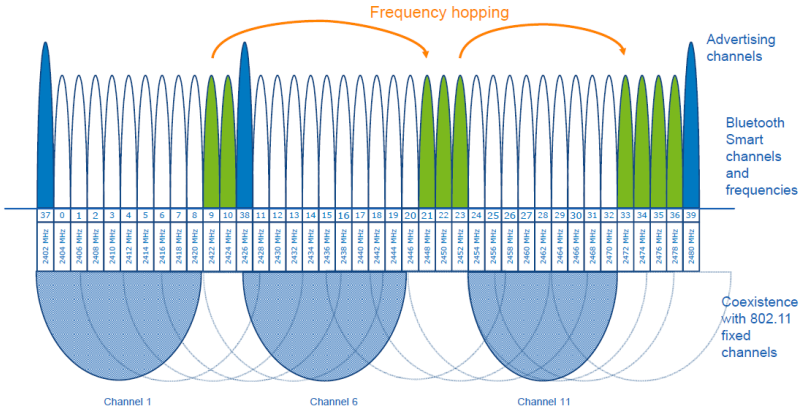
Na Sl. 10 je ilustrovan princip frekventnog skakanja na primeru tri istovremeno aktivne BLE konekcije. Sekvence frekventnih kanala svake konekcije su prikazane u različitim bojama. Frekventno skakanje na konekciji Link 3 je posebno izdvojeno. U suštini, frekventno skakanje smanjuje verovatnoću kolizije transmisija dva ili više BLE uređaja. Kolizija se dešava samo onda kada su *connection event*-i dve konekcije preklopljeni u vremenu i pri tom obe koriste isti frekventni kanal. Verovatnoća za tako nešto je veoma mala. Čak iako se kolizija desi, već u sledećem *connection event*-u, dve konekcije će preći na različite kanale (zbor različitog *hop*), što će omogućiti nastavak komunikacije.



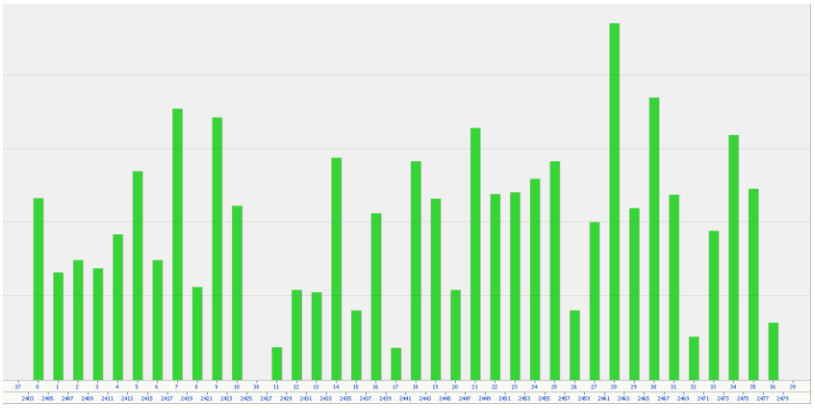
Sl. 10 Princip frekventnog skakanja

Adaptivno frekventno skakanje (*Adaptive Frequency Hopping*). U datom fizičkom okruženju, pojedini frekventni kanali mogu biti pogođeni interferencijom od okolnih uređaja. Zbog učestalih kolizija, komunikacija na ovim kanalima je nepouzdana. U isto vreme, neki drugi kanali mogu biti „čisti“ (nisu izloženi interferenciji) i zbog toga je komunikacija na ovim kanalima pouzdana. Vremenom, lista pouzdanih i nepouzdanih kanala se može menjati, zavisno od prisustva i aktivnosti okolnih bežičnih uređaja. Ovaj problem se prevazilazi mehanizmom adaptivnog frekventnog skakanja, koji je ilustrovan na Sl. 11.

Pretpostavimo, na primer, da su u okolini BLE uređaja aktivne tri WiFi mreže i to na WiFi kanalima 1, 6 i 11. WiFi kanali su širi, tako da se svaki WiFi kanal preklapa sa okvirno 9-10 BLE kanala. Adaptivno frekventno skeniranje omogućava BLE uređaju da označi kao „loše“ BLE kanale 0-8, 11-20 i 24-32. To znači da će dva BLE uređaja, tokom konekcije, za frekventno skakanje birati samo „dobre“ kanale (prikazani zelenom bojom na Sl. 11). Ako kasnije WiFi mreža na kanalu 6 prestane sa radom, BLE kanali iz opsega 11-20 postaće „zeleni“. Ovaj mehanizam se implementira tako što BLE uređaj održava tzv. mapu kanala u kojoj je svaki kanal označen kao „dobar“ ili „loš“. Kanali se karakteriše kao dobar/loš na osnovu statistika korišćenja kanala, npr. učestanost pojave kolizije na tom kanalu u prethodnom vremenskom periodu. Prilikom uspostavljanja konekcije, centralni BLE uređaj šalje svoju mapu kanala perifernom uređaju kako bi oba imala istu informaciju o tome koje kanale koristiti, a koje ne. Tokom konekcije, promena u mapi kanala bilo kog uređaja se saopštava drugom uređaju. Na Sl. 12 je prikazana statistika korišćenja kanala tokom konekcije dva BLE uređaja u nekom realnom okruženju. Bez adaptivnog frekventnog skakanja, sve zelene trake, od kojih svaka odgovara jednom BLE kanalu, bi bile iste visine (svi kanali se ravnopravno koriste). Sa adaptivnim frekventnim skakanjem, BLE komunikacija daje više prioritet kanalima bez interferencije, a izbegava kanale sa uočenom interferencijom.



Sl. 11 Princip adaptivnog frekventnog skakanja



Sl. 12 Adaptivno frekventno skakanje – distribucija korišćenih kanala

8.2.4 Modulacija

BLE koristi varijantu frekventne modulacije poznatu pod skraćenicom GFSK[2] (*Gaussian Frequency Shift*). Informacija (u obliku niza bita) se kodira promenom frekvencije RF signala. Neka je f_0 frekvencija nosećeg signala, tj. centralna frekvencija jednog od 40 kanala na koji je predajnik podešen. Prenos jednog bita informacije traje jedan bit-interval tokom kojeg se emituje signal frekvencije $f_0 - 185\ kHz$, ako se prenosi binarna '0', odnosno signal frekvencije $f_0 + 185\ kHz$, za prenos binarne '1'.

Trajanje bit-intervala iznosi $1us$, što odgovara brzini prenosa podataka od $1\ Mbps$. Brzinu od $1\ Mbps$ podržavaju sve BLE verzije. Kod novijih verzija, počev od BLE 5.0, postoji mogućnost izbora između 1 i $2\ Mbps$. Takođe, u BLE 5 uveden je još jedan način modulacije, *Coded PHY*, koja omogućava veći domet komunikacije. Dakle, tri postoje modulacione opcije (Tabela 1):

1M PHY ($1\ Mbps$), podržano u svim BLE verzijama

2M PHY ($2\ Mbps$), dostupno u BLE 5.0 i novijim verzijama. Za oglašavanje se uvek koristi brzina prenosa $1\ Mbps$, dok se prelazak na $2\ Mbps$ dešava kao rezultat pregovaranja između uređaja tokom uspostavljanja konekcije i to, naravno, u slučaju da oba uređaja podržavaju $2\ Mbps$ i oba žele da komuniciraju na većoj brzini. Pored dva puta veće brzine prenosa, 2M PHY donosi i sledeće prednosti u odnosu na 1M PHY:

- Manja potrošnja energije, pošto se ista količina podataka prenosi za kreće vreme, tj. RF primopredajnik provodi manje vremena na predaji/prijemu.
- Bolja „bežična koegzistencija“, odnosno mogućnost da veći broj uređaja deli bežični medijum (zbog kraćeg trajanja emitovanja).

Nedostatak korišćenja 2M PHY je potencijalno manji radio domet, jer prenos većom brzinom smanjuje radio-osetljivost na prijemnoj strani. Konkretno, kod 2M PHY trajanje bit-intervala je dva puta kraće ($0.5\ us$, prema $1\ us$ kod 1M PHY), ali su zato frekventni pomaci dva puta veći, tj. $\pm 370\ kHz$, prema $\pm 185\ kHz$ kod 1M PHY. Veća frekvencija promene povećava verovatnoću greške na prijemu, što efektivno smanjuje radio domet (na oko 80% dometa koji se postiže sa 1M PHY).

Coded PHY, podržano u BLE 5.0 i novijim verzijama. Pruža mogućnost da se radio domet poveća i do četiri puta u odnosu na 1M PHY. Veći domet se postiže primenom tehnike *Forward Error Correction* (FEC). Umesto da se nakon uočene greške u prenosu zahteva ponovno slanje istih podataka (kao kod 1M i 2M PHY), primena FEC omogućava prijemniku da koriguje greške i oporavi originalne podatke. FEC podrazumeva da postoji značajna redundansa u podacima koji se šalju. Tako, 1 bit informacije se sada kodira sa dva simbola (šema kodiranja $S = 2$), tj. dve promene frekvencije nosećeg signala, ili sa osam simbola ($S = 8$). Neizbežan kompromis je taj što je sada brzina prenosa podataka manja i iznosi $500\ kpbs$ (S=2), ili $125\ kbps$ (S=8). Manja brzina prenosa povlači za sobom i veći utrošak energije za predaju/prijem.

Tabela 1 Poređenje PHY varijanti

[2] *Gaussian Frequency Shift Keying* (GFSK) je vrsta frekventne modulacije (FSK) koja koristi Gausov filter da "smanji oštrinu" digitalnih impulsa pre modulacije, čime se smanjuje širina spektra i poboljšava efikasnost. Standardna FSK modulacija menja frekvenciju nosioca u zavisnosti od digitalnog signala (npr. "1" i "0" koriste različite frekvencije). Problem FSK je u naglim promenama frekvencije, što stvara širok spektar i smetnje u susednim frekventnim opsezima. GFSK rešava ovaj problem propuštanjem digitalnog signala kroz Gausov filter (nisko-propusni filter čija karakteristika prati Gausovu krivu). Ovaj filter "ublažava" oštre prelaze između simbola - frekventni skokovi su postepeni (kontinualni), što smanjuje širinu spektra u odnosu na standardnu FSK modulaciju.

	1M PHY	2M PHY	Coded PHY S=2	Coded PHY S=8
Symbol rate	$1\ Msps$	$2\ Msps$	$1\ Msps$	$1\ Msps$
Data rate	$1\ Mbps$	$2\ Mbps$	$500\ kbps$	$125\ kbps$
Error Correction	None	None	FEC	FEC
Range Multiplier	1	~ 0.8	~ 2	~ 4

8.3 Sloj veze (*Link Layer - LL*)

Sloj veze je deo protokol steka koji neposredno upravlja i interaguje sa fizičkim slojem (tj. RF primopredajnikom) i obično se realizuje u kombinaciji hardvera i softvera. Sloj veze je jedini sloj protokol steka koji je podložan čvrstim vremenskim ograničenjima (*hard real-time*) s obzirom da je obezbeđivanje poštovanja vremenskih zahteva definisanih BLE specifikacijom upravo u njegovoj nadležnosti. Iz tog razloga, sloj veze je po pravilu izolovan od viših slojeva protokol steka standardnim interfejsom (HCI - *Host-Controller Interface*) koji sakriva složenost i zahteve za rad u realnom vremenu od ostalih slojeva. Računski intenzivne funkcionalnosti sloja veze, koje se lako mogu automatizovati, realizuju se u hardveru. To obuhvata funkcije poput pakovanje i raspakovanje paketa, generisanje i verifikacije CRC-a, „izbeljivanje“ podataka, generisanje slučajnih brojeva i šifrovanje (kriptovanje) podataka.

Softverski deo sloja veze upravlja stanjem linka i sprovodi procedure za uspostavljanje konekcije između uređaja. Na nivou sloja veze, BLE uređaj može biti master ili slejv (ili i master i slejv), što zavisi od zahteva konkretne primene. Master je uređaj koji inicira konekciju, a slejv uređaj koji oglašava svoju spremnost da prihvati konekciju. U opštem slučaju, master se može povezati sa više slejvova i slejv se može povezati sa više mastera. Primera radi, uređaji poput pametnih telefona i tablet računara tipično igraju ulogu mastera, dok jednostavniji uređaji, poput senzora, igraju ulogu slejvova. Za BLE je karakteristična asimetričnost između master i slejv uređaja na nižim nivoima protokol steka, u smislu da se od mastera zahteva veće angažovanje uz veći utrošak resursa (CPU vreme, memorija i energija). Ovo nije slučajno jer se na taj način omogućava da se za realizaciju perifernih *single-mode* BLE uređaja koriste jednostavniji mikrokontroleri i RF primopredajnici.

Uloge uređaja

Sloj veze uvodi sledeće uloge BLE uređaja:

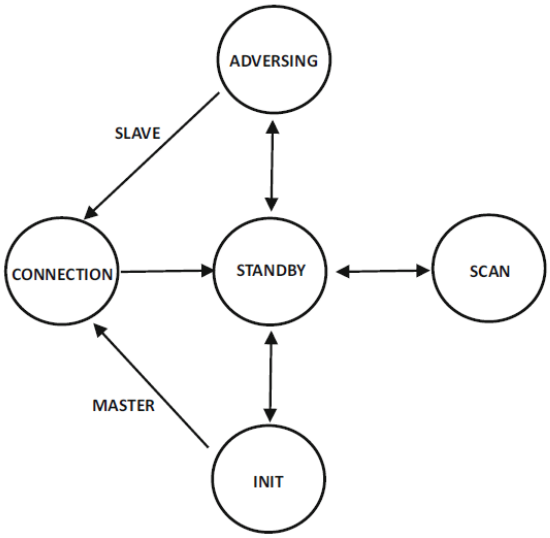
- *Oglašivač* (*Advertiser*) – uređaj koji emituje pakete za oglašavanje
- *Skener* (*Scanner*) – uređaj koji osluškuje kanale za oglašavanje u očekivanju paketa za oglašavanje
- *Master* (*Master*) – uređaj koji inicira uspostavljanje konekcije sa slejvom i upravlja uspostavljenom konekcijom
- *Slejv* (*Slave*) – uređaj koji prihvata zahtev za uspostavljanje konekcije sa masterom i sledi tajming uspostavljene konekciju.

Četiri uloge se mogu logički grupisati u dva para: oglašivač-skener (sve dok konekcija nije otvorena) i master-slejv (dok je konekcija otvorena). Izbor LL uloge uređaja je pod kontrolom viših slojeva protokol steka. Na primer, uređaj X može biti postavljen u ulogu skenera što bi odražavalo njegov interes za uspostavljanje konekcije sa nekim drugim uređajem. Kada uređaj X primi paket za oglašavanje od uređaja Y, startuje se procedura za uspostavljanje konekcije i X postaje master, a Y slejv.

Stanja

Na nivou sloja veze, BLE uređaj je uvek u jednom od pet stanja (Sl. 13). Pripravno stanje (*Standby*) je podrazumevano stanje u kome RF primopredajnik niti emituje niti prima podatke. Dok igra ulogu

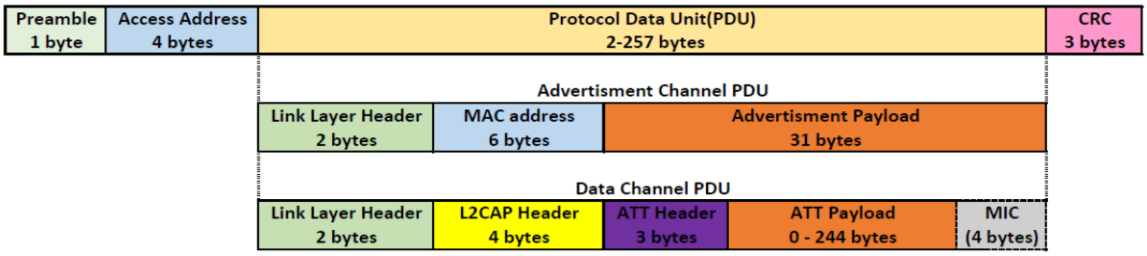
oglašivača, uređaj je u stanju *Advertising*, a dok je u ulozi skenera, njegovo stanje je *Scanning*. Uvek kada skener primi paket za oglašavanje, sloj veze obaveštava više slojeva o identitetu oglašivača. Kada viši slojevi odluče da uspostave konekciju sa nekim od uočenih oglašivača, uređaj se iz stanja *Scanning* prvo prebacuje u stanje *Standby,* a onda u stanje *Initiating* u kome se inicira uspostavljanje konekcije sa izabranim oglašivačem. Nakon uspostavljene konekcije, uređaj prelazi u stanje *Connected* u kome igra ulogu mastera. S druge strane, oglašivač (koji je u stanju *Advertising*), nakon prijema zahteva za uspostavljanje konekcije prelazi u stanje *Connected* u kome igra ulogu slejva. Nakon raskida konekcije, uređaj se vraća u stanje *Standby*.



Sl. 13 LL dijagram stanja

Format paketa

Na sloju veze koristi se jedinstveni format paketa, koji se sastoji iz četiri polja (Sl. 14). Prvo se prenosi preambula, čija je dužina 1 bajt (za 1M PHY) ili 2 bajta (2M PHY), zatim polje *Access-Address* dužine 4 bajta, a onda i korisni podaci (polje PDU), dužine od 2 – 257 bajta. Paket se završava poljem za proveru (CRC) dužine 3 bajta.



Sl. 14 Format BLE paketa (**uskladiti sa 5.0 – preambula 1 ili 2 bajta**)

Preamble (*Preambula*): dužine 1 bajta (za 1M PHY) ili 2 bajta (2M PHY). Preambula je niz naizmeničnih 0 i 1 (npr. 01010101, ili 10101010_10101010), pri čemu, prvi bit u nizu mora biti podudaran sa prvim bitom polja *Access_Address*. Preambula se koristi za „treniranje“ prijemnika. Tokom prijema preambule, prijemnik se frekventno i fazno sinhroniše sa prijemnim signalom i podešava pojačanje ulaznog pojačavača. Sve ovo je neophodno kako bi prijemnik pouzdano dekodirao sadržaj preostalog dela paketa.

Access_Address (*Adresa pristupa*): 4-bajtno polje koje sadrži adresu pristupa. Adresa pristupa nije adresa uređaja, već identifikuje oglašavanje ili konkretnu konekciju. Adresa pristupa svih paketa za oglašavanje je fiksna, oblika $0x8E89BED6$. Svaka konekcija između bilo koja dva

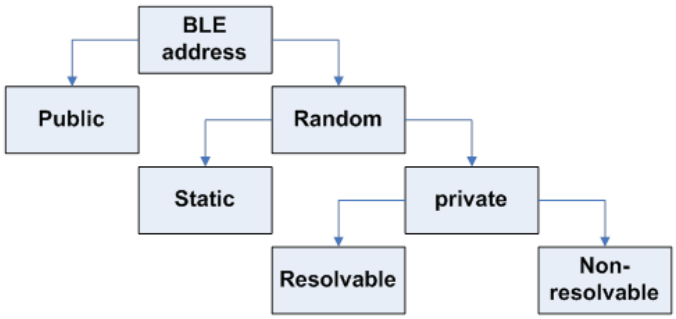
uređaja ima svoju posebnu adresu pristupa, koja se generiše od strane mastera prilikom uspostavljanja konekcije se slejvom, a zatim se koristi u svim paketima koji se razmenjuju tokom konekcije. Generisana adresa je u osnovi slučajna uz poštovanje određenih ograničenja, kako bi se izbeglo preklapanje sa adresom pristupa za oglašavanje i obezbedio uravnotežen broj 0 i 1 u adresi. Na strani prijema, adresa pristupa se koristi za filtriranje paketa, jer će sloj veze odbaciti svaki primljeni paket sa adresom pristupa koju prijemnik trenutno ne očekuje.

PDU (korisni podaci paketa): ovo polje sadrži podatke koji potiči iz slojeva višeg nivoa, a njegov format zavisi od toga da li se paket prenosi na kanalu za oglašavanje ili na kanalu za prenos podatka (Sl. 14). U oba slučaja, PDU počinje 2-bajtnim zaglavljem (*Header*), čiji sadržaj koje ukazuje na tip podataka koji slede. Deo za podatke (*Payload*) je maksimalne dužine 37 bajta, za slučaj paketa za oglašavanje, odnosno 251 bajta u paketima za podatke. 4-bajtno polje MIC (*Message Integrity Check*) je opciono i postoji samo onda kada su podaci u delu *Payload* šifrovani.

CRC (Polje za zaštitu): 3-bajtno polje čiji se sadržaj određuje (tj. izračunava) na predajnoj strani na osnovu sadržaja polja PDU. Na prijemu se ponavlja isto izračunavanje nad primljenim podacima iz polja PDU i dobijeni rezultat se poredi sa sadržajem CRC primljenog paketa. U slučaju nepodudarnosti, paket se odbacuje kao neispravan. CRC se izračunava korišćenjem pomeračkog registra sa linearnom povratnom spregom (LFSR) za karakteristični polinom: $x^{24} + x^{10} + x^{9} + x^{6} + x^{4} + x^{3} + x + 1$

BLE adrese uređaja

BLE adresa je 48-bitni (6 bajta) identifikator BLE uređaja, slično MAC adresi kod Ethernet ili WiFi. Koristi se za jedinstvenu (jednoznačnu) identifikaciju uređaja u BLE mreži. Postoje dva tipa BLE adresa uređaja, javne i slučajne, pri čemu u kategoriji slučajnih postoji još nekoliko pod tipova, kao što se može videti na Sl. 15.



Sl. 15 Tipovi BLE adresa uređaja

- *Javna adresa uređaja* (*Public device address*) – mora biti registrovana kod organizacije IEEE, što garantuje globalu jedinstvenost. Javna adresa se čuva u permanentnoj memoriji uređaja i ne menja se tokom celokupnog životnog veka uređaja.
- *Slučajna adresa uređaja* (*Random device address*) – slučajno generisana adresa koja ne zahteva registraciju kod IEEE. Može se jedanput generisati i poput javne adrese čuvati u permanentnoj memoriji. Takođe, slučajna adresa se može iznova generisati pri svakom ponovnom uključenju uređaja, ili se može menjati, s vremena na vreme, tokom rada uređaja. Slučajne adrese mogu biti statičke i privatne.
    - *Statička adresa uređaja* – koristi se kao zamena za javnu adresu uređaja. Čuva se u permanentnoj memoriji i nikada se ne menja, ili se iznova generiše uvek kada se uređaj uključi.

- *Privatna adresa* – slučajna privremena adresa koja se povremeno menja tokom normalnog rada uređaja. Koristi se za zaštitu privatnosti. Činjenica da svaki paket sadrži BLE adresu uređaja otvara mogućnost neovlašćenog praćenja uređaja (gde se konkretan uređaj kretao i kada je komunicirao). Ovakav način zloupotrebe se može predupreditи povremenom promenom adrese uređaja. Postoje dve vrste privatnih adresa uređaja:
  - *Nerazrešiva privatna adresa* (*Non-resolvable Private Address*) – uređaj X može da promeni svoju nerazrešivu adresu u bilo kom trenutku. Međutim, ne postoji način da uređaji s kojima je X do tog trenutka bio u komunikaciji povežu novu adresu sa uređajem X. Efekat je isti kao da je uređaj X naprasno nestao, a pojavio se novi. Retko se koristi.
  - *Razrešiva privatna adresa* (*Resolvable Private Address*) – privatna adresa sa dodatkom mehanizma koji omogućava okolnim (prijateljskim) uređajima da povežu novu privatnu adresu uređaja X sa njegovom stvarnom adresom. U ovom scenariju, uređaju X je dodeljena javna ili statička adresu, a privatne adrese, koje zapravo koristi u komunikaciji, generiše na osnovu ključa (IRK - *Identity Resolving Key*) i slučajno generisanog broja. Svaki drugi uređaj koji poseduje informaciju o IRK uređaja X može da transformiše svaku njegovu privatnu adresu u stvarnu i da na taj način bude siguran da komunicira sa uređajem X. Za uređaje koji ne poseduju IRK uređaja X, efekat promene adrese je isti kao da uređaj X koristi nerazrešive privatne adrese.

8.3.1 Skeniranje i oglašavanje

Kao što je već rečeno, BLE uređaj u ulozi oglašivača, tj. u stanju *Advertising*, periodično šalje pakete za oglašavanje na kanalima za oglašavanje. Svrha ovih paketa je dvojaka:

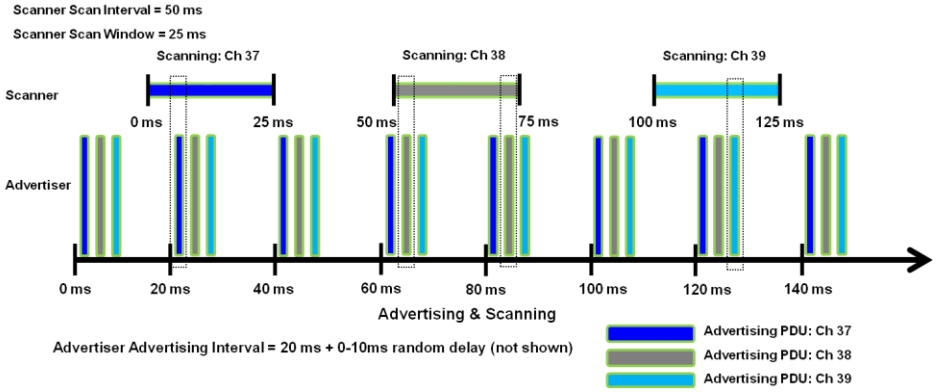
- BLE uređaj može da koristi pakete za oglašavanje za distribuciju svojih podataka okolnim BLE uređajima bez potrebe za uspostavljanjem konekcije. Ovo je pogodno za primene koje ne zahtevaju dvosmernu komunikaciju kao ni prenos velike količine podataka
- Slanjem paketa za oglašavanje, uređaj otkriva svoje prisustvo okolnim uređajima, koji, nakon toga mogu da iniciraju uspostavljanje konekcije.

U paket za oglašavanje može da stane do 31 bajt podataka, zajedno sa elementarnim zaglavljem i BLE adresom oglašivača (Sl. 14). Takve pakete oglašivač šalje periodično i „na slepo“, bez prethodnog znanja o tome da li u okolini uopšte postoji skener. Perioda slanja je regulisana parametrom *interval oglašavanja* (*Advertising Interval*), koji može da se podešava u opsegu od $20\ ms$ do $10.24\ s$. Na svaki interval oglašavanja, oglašivač uključuje svoj RF predajnik i šalje uzastopno zapravo tri identična paketa za oglašavanje, po jedan na svakom od tri kanala za oglašavanje.

S druge strane, skener, radi uštede energije, ali i činjenice da ne može istovremeno da prati više od jednog frekventnog kanala, takođe ne osluškuje neprekidno kanale za oglašavanje. Tajming rada skenera se reguliše sa dva konfigurabilna parametra: *prozor skeniranja* (*Scan Window*), koji definiše trajanje skeniranja i *interval skeniranja* (*Scan Interval*), koji definiše koliko često se skeniranje ponavlja. U svakom sledećem prozoru skeniranja, skener osluškuje različiti kanal za oglašavanje.

Na Sl. 16 je ilustrovan princip oglašavanja/skeniranja i uticaj vremenskih parametara. Da bi skener primio paket za oglašavanje, neophodno je da se prozor skeniranja skenera preklopi sa emitovanjem paketa za oglašavanje, kako u vremenu tako i po kanalu. Kraći interval oglašavanja znači veću frekvenciju emitovanja paketa za oglašavanje, pa time i veću verovatnoću da će skener primiti ove pakete, ali ujedno povlači i veću potrošnju energije oglašivača. Pošto se vremenski parametri rada

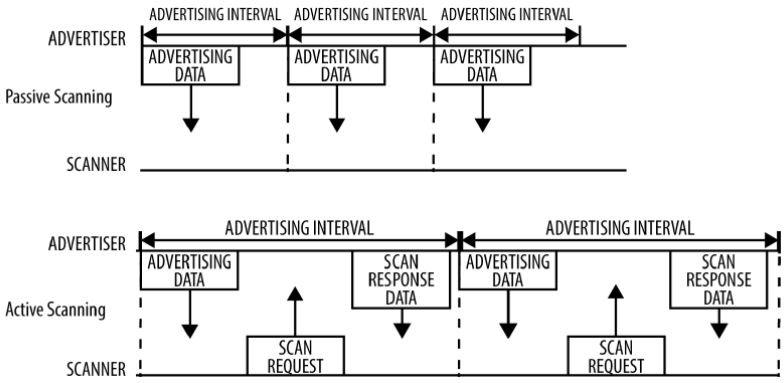
oglašivača i skenera nezavisno podešavaju, u principu, može se desiti da uvek kada je skener na prijemu, oglašivač pravi pauzu. Da bi se ovakve situacije predupredile, svaki intervala oglašavanja se automatski proširuje slučajno izabranim dodatnim kašnjenjem u intervalu $0 - 10\,ms$. Na taj način, izvesno je da će se skeniranje i oglašavanje kad-tad preklopiti.



Sl. 16 Vremenski parametri oglašavanja i skeniranja

BLE specifikacija pravi razliku između dva tipa skeniranja (Sl. 17):

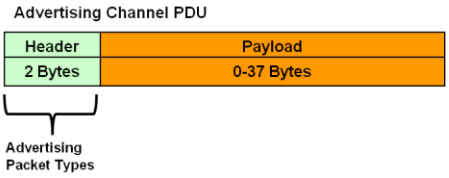
- *Pasivno skeniranja* – skener, jednostavno, osluškuje kanale za oglašavanje u očekivanju paketa za oglašavanje. Pri tom, oglašivač nema povratnu informaciju o tome da li je i kada neki skener primio emitovan paket.
- *Aktivno skeniranje* – nakon prijema paketa za oglašavanje, skener šalje oglašivaču paket tipa *Scan Request*. Kao odgovor na prijem *Scan Request* paketa, oglašivač šalje skeneru paket tipa *Scan Response* sa dodatnim podacima o oglašavanju. U *Scan Response* paketu može da stane do 254 bajta podataka, što značajno povećava količinu podataka koja se oglašavanjem može preneti skeneru. Međutim, bitno je napomenuti da je komunikacija i dalje jednosmerna, jer ni u ovom slučaju skener nema mogućnost da u paketu *Scan Request* pošalje oglašivaču bilo kakve korisne podatke.



Sl. 17 Pasivno i aktivno skeniranje

Tipovi paketa za oglašavanje

Paket za oglašavanje se prenosi u polju PDU BLE paketa (vidi Sl. 14). Format paketa za oglašavanje je prikazan na Sl. 18. Na početku paketa je 2-bajtno zaglavlje (*Header*), nakon kojeg sledi maksimalno 37 bajta korisnih podataka (*Payload*).



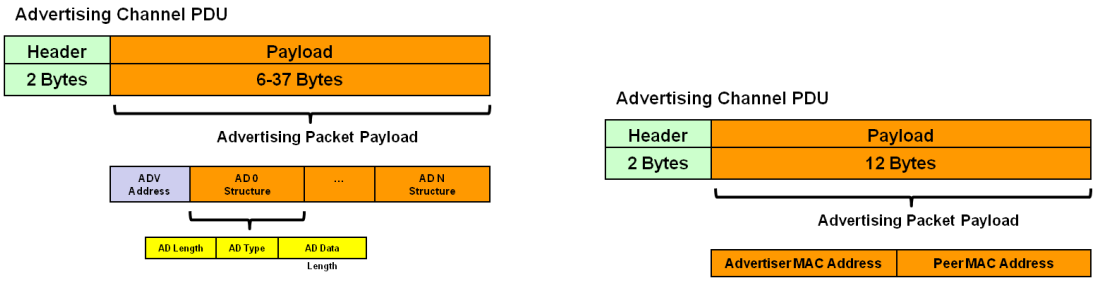
Sl. 18 Format paketa za oglašavanje

Zaglavlje paketa za oglašavanje, između ostalog, sadrži indikaciju o tipa paketa. Postoji nekoliko tipova paketa za oglašavanje, od kojih svaki odražava spremnost oglašivača da uspostavi specifičan način interakcije sa skenerom. Tipovi paketa za oglašavanje se mogu klasifikovati na osnovu tri kriterijuma. Prvi kriterijum je spremnost oglašivača da bude konektovan od strane skenera. Po ovom kriterijumu, oglašivač može biti „konektabilan“, tj. dostupan za konekciju ili „nekonektabilan“, što bi značilo da oglašivač ne prihvata zahteve za konekciju, odnosno da je paket za oglašavanje isključivo namenjen emitovanju podataka oglašivača. Drugi kriterijum se odnosi na to da li oglašivač podržava *scan request*, odnosno, da li skener može, nakon prijema paketa za oglašavanje, da pošalje oglašivaču paket *scan request* kako bi od oglašivača zatražio dodatne podatke o oglašavanju (*scan response*). Treći kriterijum se odnosi na „usmerenost“ konekcije. Jedna mogućnost je da oglašivač dozvoljava konektovanje samo jednom konkretnom skeneru, čija je BLE adresa navedena u samom paketu za oglašavanje. Druga mogućnost je da skener dozvoljava konektovanje bilo kom skeneru.

Tabela 2 Tipovi paketa za oglašavanje

Advertising Packet Type	*Description*
ADV_IND	Connectable Undirected Advertising
ADV_DIRECT_IND	Connectable Directed Advertising
ADV_NONCONN_IND	Non-connectable Undirected Advertising
ADV_SCAN_IND	Scannable Undirected Advertising

U tabli Tabela 1 su navedena četiri tipa paketa za oglašavanje. *ADV_IND* je „*connectable-undirected*“ (tj. konektabilan i neusmeren) tip, odnosno, slanjem paketa za oglašavanje ovog tipa, oglašivač saopštava da je spreman da prihvati konekciju od bilo kog skenera. Format paketa tipa *ADV_IND* je prikazan na Sl. 19(a). Polje za korisne podatke počinje BLE adresom oglašivača (*ADV Address*), dok u nastavku sledi ni jedno, jedno ili više polja za podatke koji se emituju (*AD_0,..., AD_N*). Svako polje za podatak sadrži dužinu (*AD Length*), tip (*AD Type*) i vrednost (*AD Data*) podatka.



(a) (b)

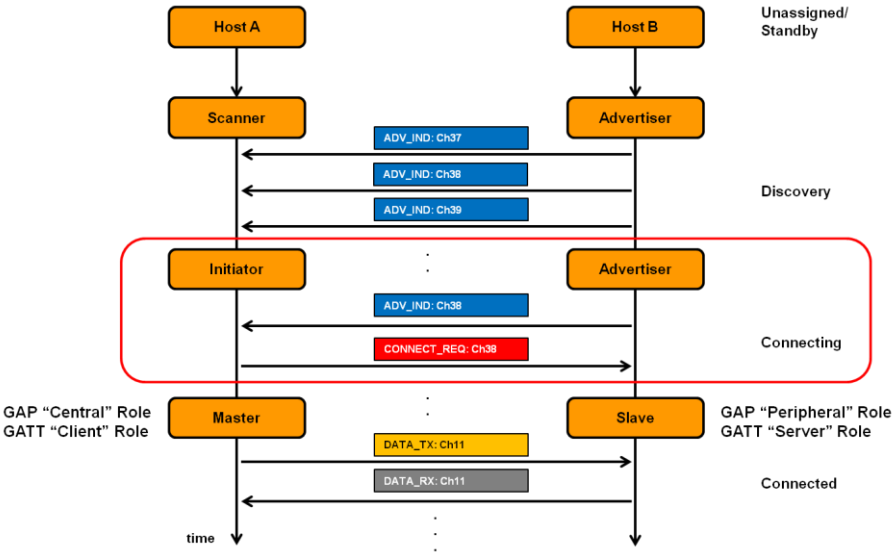
Sl. 19 Primeri formata paketa za oglašavanje: (a) ADV_IND; (b) ADV_DIRECT_IND

Tip paketa za oglašavanje *ADV_DIRECT_IND* je „*connectable-directed*“ (tj. konektabilan i usmeren) i njegov format je prikazan na Sl. 19(b). U polju za korisne podatke, pored BLE adrese oglašivača (*Advertiser MAC Address*), sadržana je i BLE adresa skenera kome je paket namenjen (*Peer MAC Address*). Smisao ovog paketa je taj da je oglašivač spreman da prihvati konekciju samo od skenera sa navedenom BLE adresom. Paket tipa *ADV_DIRECT_IND* ne može da sadrži podatke za emitovanje.

Paket tipa *ADV_NONCONN_IND* je *„non-connectable - undirected“*, što znači da oglašivač nije dostupan za konektovanje od strane bilo kog skenera. Paketi ovog tipa se koriste samo za emitovanje podataka. Konačno, paket za oglašavanje tipa *ADV_SCAN_IND* je *„scannable undirected“*, što znači da oglašivač, pored podataka koje emituje u ovom paketu, ima i dodatne podatke za emitovanje koje će poslati kao odgovor na *scan request*. Pri tom, oglašivač je dostupan za konektovanje. Osim navedenih, BLE specifikacija predviđa još nekoliko specifičnih tipova paketa za oglašavanje, koji ovde nisu navedeni.

8.3.2 Konekcija

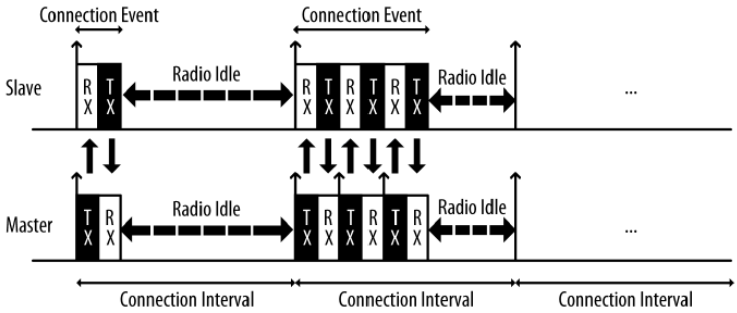
Procedura za uspostavljanje konekcije između dva BLE uređaja je ilustrovana dijagramom na Sl. 20. BLE uređaj koji želi da uspostavi konekciju sa nekim drugim uređajem mora biti prvo postavljen u režim skeniranja, kako bi otkrio oglašivače koji su trenutno spremni za konektovanje. Pri tom, uređaj (*Host A* na Sl. 20), koji je sada u ulozi skenera, može da filtrira primljene pakete za oglašavanje, npr. prema BLE adresi oglašivača ili na osnovu drugih podataka sadržanih u ovim paketima. Kada detektuje i izabere odgovarajućeg oglašivača (*Host B* na Sl. 20), uređaj (*Host A*) prelazi u stanje *Initiating* i započinje proceduru uspostavljanja konekcije. U stanju *Initiating Host A* čeka da ponovo primi paket za oglašavanje od uređaja *Host B*, a zatim šalje ovom oglašivaču konekcioni zahtev, odnosno paket tipa *CONNECT_REQ*. Konekcioni zahtev, između ostalog, sadrži informaciju o hop-inkrementu koji će koristiti oba uređaja prilikom frekventnog skakanja, kao i informaciju o vrednostima nekoliko drugih parametara koji se koriste za regulisanje tajminga i toka konekcije. Pod uslovom da je *Host B* prihvatio ovaj zahtev, konekcija između uređaja *Host A* i *Host B* je uspostavljena. *Host A* preuzima ulogu mastera, a *Host B* ulogu slejva. Sa otvorenom konekcijom, dva uređaji mogu da razmenjuju pakete podataka (*DATA_TX* i *DATA_RX*). Treba napomenuti da pre prelaska u fazu prenosa podataka kroz upravo otvorenu konekciju, dva uređaja prolaze kroz proceduru međusobnog usaglašavanja, što obuhvata prenos nekoliko paketa radi npr. razmene informacije o BLE verziji koju uređaji podržavaju, podržanim mogućnostima i atributima i slično.



Sl. 20 Procedura uspostavljanja konekcije

Konekcija je, u suštini, niz uzastopnih razmena paketa između mastera i slejva u tačno definisanim vremenskim intervalima (Sl. 21). Tokom vremenskog perioda, koji se zove *connection event*, master i slejv, alternativno šalju jedan drugom pakete sve dok obe strane ne pošalju sve podatke koje su spremile za slanje. Zatim, oba uređaja isključuju svoje RF primopredajnike, a komunikacija se nastavlja

u sledećem *connection event*-u. Vremenski period između dva uzastopna *connection event*-a je konstantan i iznosi *connection interval*.



Sl. 21 *Connection event* i *connection interval*

Za komunikaciju tokom *connection event*-a važe sledeća pravila:

- *Connection event*-i se dešavaju periodični i neprekidno sve dok se konekcija ne zatvori ili prekine.
- Svaki *connection event* sadrži bar jedan paket, koji šalje master.
- Slejv ima obavezu da na svaki primljeni paket odgovori slanjem svog paketa masteru, čak iako više nema podataka za slanje (u tom slučaju, poslaće „prazan“ paket, tj. paket bez korisnih podataka).
- Paket koji slejv šalje masteru je ujedno i potvrda prijema paketa koji je master prethodno poslao slejvu. Ukoliko odgovor slejva izostane, master prekida slanje svojih paketa – slanje će nastaviti u sledećem *connection event*-u.

Da bi razmena paketa bila moguća, maser i slejv moraju, tokom uspostavljanja konekcije, među sobom usaglasiti nekoliko parametara:

- *Connection Interval* – vrednost ovog parametra se može podešavati u opsegu između $7.5\ ms$ i $4\ s$ u inkrementima od $1.25\ ms$. Trajanje *connection interval*-a podešava master i vrednost ovog parametra je sadržana u paketu *CONNECT_IND*.
- *Slave Latency* – ovaj parametar omogućava slejvu da preskoči („prespava“) jedan ili više uzastopnih *connection event*-a, a da to ne izazove prekid konekcije. Motivacija za preskakanje *connection event*-a je ušteda energije, jer će tokom *connection event*-a koje preskače primopredajnik slejva ostati isključen. Parametar *slave latency* definiše broj *connection event*-a koje slave može bezbedno da preskoči. Na primer, ako je vrednost ovog parametra 3, slejv može da preskoči tri uzastopna *connection event*-a, ali se onda mora „probuditi“ u četvrtom, kako bi primio i poslao paket. U suprotnom, master će prekinuti konekciju. S druge strane, master se budi u svakom *connection event*-u i šalje svoj paket. Ako ne primi odziv (jer slejv preskače taj *connection event*), master će odustati od daljeg slanja i pokušaće ponovo u sledećem *connection event*-u.
- *Supervision Timeout* – vremenski parametar koji se koristi za detekciju prekinute konekcije. Prekid konekcije je situacija kada se jedna strane, bez ikakve indikacije drugoj strani, isključi iz dalje komunikacije. *Supervision Timeout* se definiše kao maksimalno dopušteno vreme između dva prijema paketa. Uređaj detektuje prekid konekcije ukoliko po isteku vremena *supervision timeout* nakon prijema paketa ne primi sledeći paket. Vrednost parametra *Supervision Timeout* je u opsegu $100\ ms$ do $32\ s$ u inkrementima od $10\ ms$. Pri tom, vrednost ovog parametra mora biti usaglašena sa parametrima *slave latency* i *connection interval* i to na sledeći način:

$$Supervision\ Timeout > (1 + Slave\ Latency) \times ConnectionInterval \times 2$$

- *Data Length Extension* – ovo je binarni parametar, koji se može uključiti ili isključiti. Ukoliko je uključen, omogućava da se u paketu može preneti veća količina podataka, do maksimalno 251 bajt. Kada je isključen, paketi podataka mogu prenositi najviše po 27 bajta. Podrška za pakete podataka veće dužine uvedena u je BLE verziji 4.2. Verzije 4.0 i 4.1 podržavaju samo prenos kratkih paketa, do 27 bajta u delu za podatke.

8.3.3 L2CAP

L2CAP (*Logical Link Control and Adaptation Protocol*) je najniži sloj host dela BLE protokol steka, koji deluje kao spona između protokola višeg nivoa i sloja veze. Prvo, L2CAP deluje poput multipleksera protokola, tako što podatke koji potiču iz viših slojeva inkapsulira u standardne BLE pakete i obrnuto. Drugo, na ovom sloju se obavlja fragmentacija i rekonstrukcija podataka, odnosno proces tokom kojeg se, na predajnoj strani, veći blokovi podataka, koji potiču iz slojeva višeg nivoa, rasparčavaju na segmente koji se mogu upakovati u jedan BLE paket (27 ili 251 bajt), što bi odgovaralo fragmentaciji. Na prijemnoj strani, fragmenti, koji sižu posebnim BLE paketima, objedinjuju se u prvobitni blok podatka, što bi odgovaralo rekonstrukciji. U suštini, zadatak sloja L2CAP je da omogući protokolima višeg nivoa, od kojih svaki ima svoje specifične zahteve, da koriste zajednički fizički link.

Konkretno, kod BLE, L2CAP je podrška protokolima *Attribute Protocol* (*ATT*) i *Security Manager Protocol* (*SMP*). Protokol ATT pruža osnovnu podršku za razmenu podataka između BLE aplikacija dva konektovana uređaja, dok je SMP zadužen za sigurnost, odnosno generisane i distribuciju kriptografskih ključeva.

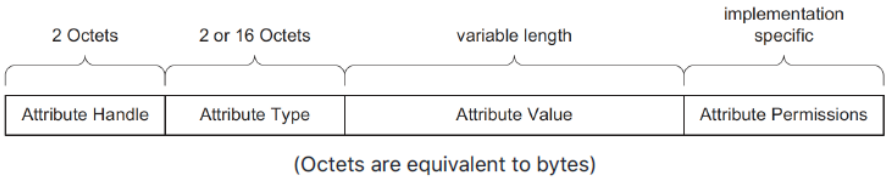
8.4 ATT (*Attribute Protocol*)

Protokol ATT (*Attribute Protocol*) je jednostavan klijent/server protokol za razmenu podataka između BLE uređaja. Pojam „razmena podataka“ iz prethodne rečenice se ne odnosi samo fizički prenos podataka, već na to kako server izlaže (predstavlja, čini dostupnim) svoje podatke klijentu i kako su ti podaci strukturirani (formatirani). U kontekstu protokola ATT, svaki BLE uređaj može biti klijent, server ili oba, bez obzira na to da li u tekućoj konekciji igra ulogu mastera ili slejva.

Server je uređaj koji *izlaže* (*exposes*) neke svoje podatka tako da su oni dostupni drugim uređajima (klijentima). Sever prihvata zahteve za pristup izloženim podacima od klijenta i šalje klijentima odzive, notifikacije i indikacije. Klijent može da pristupa podacima servera radi čitanja ili upisa. Notifikacije i indikacije su obaveštenja o promeni vrednosti podataka. Primera radi, termometar (kao BLE uređaj), u ulozi servera, može svom okruženju (drugim BLE uređajima) da izloži podatke kao što su: izmerena temperatura, nivo baterije, perioda merenja temperature i sl., i da uz to šalje notifikacije o tome da je temperatura prekoračila graničnu vrednost.

Klijent je uređaj koji je u vezi sa serverom radi pristupa njegovim izloženim podacima. Klijent šalje zahteve za pristup podacima i prima odzive, notifikacije i indikacije. Na primer, mobilni telefon koji koristi BLE radi očitavanja trenutne temperature BLE termometra igra ulogu klijenta.

Protokol ATT zahteva da izloženi podaci servera budu strukturirani na specifičan način, u formi tzv. atributa. Atribut je opšti pojam koji se odnosi na podatak bilo kog tipa koji je izložen od strane servera. Atribut nije samo vrednost konkretnog podatka, već obuhvata i dodatne informacije o tipu i načinu pristupa podatku (tzv. meta-podatke). Konkretno, svaki atribut sadrži sledeće elemente (Sl. 22):



Sl. 22 Logička reprezentacija atributa

Attribute type (*Universally Unique Identifier - UUID*) je 16-bitni ili 128-bitni univerzalno-jedinstveni identifikator koji specificira tip i prirodu podatka sadržanog u atributu. Pojedini tipovi atributa su definisani u samom BLE standardu i njihova dužina je 16 bita. Na primer, po BLE standard, za tip podataka „temperatura“ koristi se UUID vrednosti $0x2A1C$. Dakle, ako je nekom atributu pridružen ovaj UUID, tada znamo da je u polju *Attribute Value* tog atributa sadržana izmerena temperatura. Projektant može da uvodi i nove UUID-ove, za tipove podataka koji nisu obuhvaćeni standardom. Međutim, dužina takvih, tzv. korisničkih UUID-ova je 128 bita. Više o UUID u poglavlju GATT.

Attribute Handle je 16-bitna vrednost, u opsegu $0x0001 - 0xFFFF$, koju server pridružuje svakom svom atributu (nešto poput adrese atributa u memoriji servera). Svaki atribut u okviru servera mora imati svoj jedinstveni *handle*. Klijent navodi *handle* u svojim zahtevima da bi zatražio pristup određenom atributa.

Attribute Permissions je niz bita koji definišu pravo pristupa atributu, odnosno, da li je dozvoljeno atributu pristupati radi čitanja ili upisa, i da li su dopuštene notifikacije i indikacije, kao i to koji nivo sigurnosti se zahteva prilikom pristupa atributu.

Attribute Value je oblast atributa u kojoj je sadržana vrednost podatka koji je izložen atributom.

Klijent koji želi da pročita vrednost nekog atributa na strani servera, šalje zahtev za čitanje serveru. Ovaj zahtev sadrži *handle* atributa. Po prijemu zahteva, server prvo proverava pravo pristupa atributu, a zatim, ukoliko je čitanje traženog atributa dozvoljeno, odgovara klijentu slanjem vrednosti atributa. Na klijentu je da parsira (protumači) dobijenu vrednost u skladu sa tipom podatka koji je sadržan u tom atributu (što je definisano UUID-om atributa). U slučaju upisa, klijent šalje serveru zahtev za upis koji sadrži *handle* i novu vrednost atributa. Obaveza je klijenta da vrednost koju šalje uskladi sa tipom podataka konkretnog atributa. U suprotnom, ukoliko poslata vrednost nije usaglašena sa tipom atributa, server će odbiti takav zahtev. Na primer, server će odbiti zahtev za upis u atribut čija je vrednost 16-bitna ako je klijent u zahtevu dostavio 8-bitni podatak. Protokol ATT uvodi striktno sekvenciranje zahteva/odziva, u smislu da klijent ne sme da pošalje sledeći zahtev sve dok ne primi odgovor na prethodno poslati zahtev. Ovo ograničenje se primenjuje nezavisno u oba smera komunikacije u slučajevima kada dva uređaja povezana konekcijom istovremeno igraju uloge klijenta i servera.

Rad protokola ATT je u mnogome kompleksniji od onoga kako je to opisano u prethodnom pasusu. Ilustracije radi, u nastavku su navedene osnovne kategorije operacije podržane ovim protokolom:

Error Handling. Operacije koje koristi server kao reakciju na zahteve klijenta u kojima je pronađena greška (npr. neodgovarajuće pravo pristupa). Po pravilu, u slučaju greške u zahtevu, server šalje klijentu poruku obaveštenja o grešci.

Server Configuration. Omogućava postavljanje konfiguracionih parametara samog protokola ATT, kao npr. maksimalna veličina paketa koja se može razmenjivati između klijenta i servera.

Find Information. Omogućava klijentu da dobije informacije o atributima servera. Slanjem odgovarajućeg zahteva, klijent može da dobije spisak svih atributa servera u određenom opsegu *handle* vrednosti, ili da dobije *handle* vrednosti atributa datog UUID-a.

Read Operations. Omogućava klijentu da dobije vrednosti jednog ili više atributa, što uključuje:

- *Read by Type Request/Response* – čitanje vrednosti jednog ili više atributa na osnovu zadatog UUID.
- *Read Request/Response* – čitanje vrednosti atributa na osnovu *handle*-a.
- *Read Blob Request/Response* – čitanje jednog dela vrednosti atributa velike dužine na osnovu *handle*-a.
- *Read Multiple Request/Response* – čitanje vrednosti jednog ili više atributa korišćenjem više *handle*-ova.

Write Operations. Koriste se od strane klijenta radi upisa vrednosti jednog ili više atributa:

- *Write Request/Response*. Upis vrednosti atributa uz očekivanje odziva (potvrde) od strane servera.
- *Write Command*. Upis vrednosti atributa bez očekivanja bilo kakvog odziva ili potvrde servera.
- *Queued Writes*. Omogućava klijentu da upiše vrednost atributa, čija je dužina veća od maksimalne dužine dela za podatke BLE paketa. Vrednost za upis se šalje u više uzastopnih paketa.

Server Initiated. U ovu kategoriju spadaju operacije koje (asinhrono) inicira sam server, tj. bez prethodno poslatog zahteva od strane klijenta.

- *Handle Value Indication/Conformation*. Omogućava da server obavesti klijenta da se vrednost određenog atributa promenila. Server šalje klijentu novu vrednost i *handle* promenjenog atributa. Od klijenta se očekuje da potvrdi prijem.
- *Handle Value Notification*. Isto kao prethodno ali bez potvrde klijenta.

8.5 GAP (*Advertising and Connections Profile*)

GAP (*Advertising and Connections Profile*) je sloj Host dela protokol steka, koji omogućava BLE uređajima da otkrivaju (pronalaze) jedni druge, razmenjuju podatke, uspostavljaju sigurnu konekciju i izvršavaju brojne druge operacije na standardni i univerzalni način. GAP se oslanja na sloj veze i dodatno, kroz nove uloge uređaja, režime rada i procedure, proširuje njegovu funkcionalnost.

GAP-om su obuhvaćeni sledeći aspekti interakcije između BLE uređaja:

Roles (*Uloge*). Svaki uređaj može da radi u jednoj ili više uloga u isto vreme. Svaka uloga nameće izvesne restrikcije i zahteve u pogledu funkcionisanja uređaja. Pojedine kombinacije uloga omogućavaju uređajima da međusobno komuniciraju, a GAP specificira precizna pravila za interakciju između tih uloga. Uloga uređaja (*broadcaster, observer, central* i *peripheral*) po pravilu je određena njegovim tipom i namenom i ne menja se tokom životnog veka uređaja.

Modes (*Režimi, Načini rada*). Načini rada je stanje u koje uređaj prelazi na izvesno vreme kako bi ostvario određeni cilj ili omogućio drugom uređaju da obavi određenu aktivnost. Promena načina rada može biti inicirana od strane korisnika, putem korisničkog interfejsa ili automatski, po potrebi. Po pravilu, uređaji češće menjaju način rada nego što menjaju svoje uloge.

Procedures (*Procedure*). Procedura je niz akcija (obično sekvenca aktivnosti na nivou sloja veze, ili sekvenca razmene paketa) koja omogućava uređaju da ostvari neki cilj. Procedura je tipično pridružena određenom načinu rada uređaja na drugom kraju konekcije.

Security (*Sigurnost*). GAP koristi usluge *Security Manager* sloja radi definisanja sigurnosnih načina rada i procedura koje omogućavaju povezanim uređajima da prvo uspostave željeni nivo sigurnosti, a zatim i da sprovede odgovarajuće sigurnosne procedure.

8.5.1 Uloge

GAP uvodi četiri uloge koje definišu četiri načina na koji BLE uređaj može da se pridružio BLE mreži:

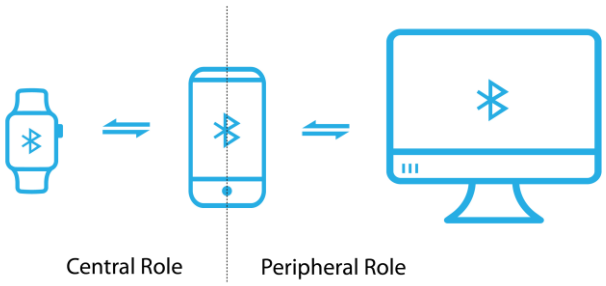
Broadcaster (*Emiter*). Ova uloga je optimizovana za primene kod kojih postoji potreba samo za periodičnim emitovanjem podataka u formi paketa za oglašavanje. U principu, za implementaciju uloge emitera, dovoljno je da uređaj poseduje radio predajnik (ne i prijemnik). Međutim, u praksi, većina uređaja u ulozi emiter ipak poseduje kompletan RF primopredajnik. Javni termometar, koji periodično emituje podatak o temperaturi zainteresovanim okolnim uređajima je tipičan primer emitera. Emiter šalje podatke u paketima za oglašavanje, a ne putem konekcije, što omogućava bilo kom uređaju koji osluškuje BLE kanale za oglašavanje da primi emitovane podatke. Za implementaciju uloge emitera koristi se funkcija oglašavanja iz sloja veze.

Observer (*Osmatrač*). Ova uloga je optimizovana za aplikacije koje žele da prikupljaju podatke koje emituju emiteri. Uređaj u ulozi osmatrača, osluškuje radio medijum i prima pakete za oglašavanje. Na primer, uređaj koji poseduje displej je često u ulozi osmatrača – npr. tablet računar koji na svom displeju prikazuje informaciju o temperaturi koju je dobio od BLE termometra u ulozi emitera. Za implementaciju ove uloge koristi se funkcija skeniranja iz sloja veze.

Central (*Centralni uređaj* ili *centrala*). Uloga centrale odgovara masteru iz sloja veze. Zadatak centralnog uređaja je da otkriva prisustvo drugih uređaja i uspostavlja konekcije s njima. Ovu ulogu obično imaju uređaji poput mobilnih telefona ili tablet računara, koji poseduju CPU visokih performansi i obilje memorijskih resursa, što omogućava uspostavljanje konekcija sa više BLE uređaja u isto vreme. Centralni uređaj osluškuje radio medijum, prima pakete za oglašavanje i uspostavlja konekciju sa izabranim uređajem. Ovaj proces se može ponoviti kako bi se u jedinstvenu mrežu priključilo više uređaja.

Peripheral (*Periferni uređaj* ili *periferija*). Ova uloga odgovara slejvu iz sloja veze. Uređaj u ulozi periferije koristi pakete za oglašavanje kako bi omogućio centrali da ga pronađe i nakon toga uspostavi konekciju s njim. Uloga periferije je po pravilu dodeljena jednostavnim i jeftinim BLE uređajima, poput senzora.

BLE uređaj može raditi u jednoj ili više uloga istovremeno. U scenariju na Sl. 23, u vezi između fitnes trakera i mobilnog telefona, traker igra ulogu periferije, a mobilni telefon ulogu centrale. Istovremeno, mobilni telefon je u vezi i sa desktop računarom, ali sada u ulozi periferije.



Sl. 23 Uređaj sa dva GAP uloge

8.5.2 Načini rada i procedure

U Tabela 3 su navedeni GAP načini rada i procedure koje se mogu sprovoditi u svakom načinu rada. Na primer, uređaj u ulozi emitera (*Broadcaster*) ima na raspolaganju dva načina rada: *broadcast* i *non-connectable*. Razlika je u tome što prvi način rada dopušta konektovanje, a drugi ne. Uređaj u ulozi periferije može da bira između više načina rada, koji se uglavnom razlikuju po tome da je i u kojoj meri uređaj vidljiv drugim uređajima. Za dati način rada i ulogu, u trećoj koloni tabele je navedena procedura koju bi trebalo obavlja povezani (*peer*) uređaj. Na primer, dok je *broadcaster* u režimu *broadcast*, povezani uređaj bi trebalo da sprovodi proceduru *Observation*. Za svaki način rada u određenoj ulozi, u Tabela 4 su navedene procedure koje povezani uređaj treba da izvršava. Na primer, ako uređaj ulozi centrale sprovodi proceduru *Any connection establishment*, tada bi povezani uređaj trebalo da bude u režimu (načinu rada) *Any connectable.*

Tabela 3 Načini rada i odgovarajuće procedure[3]

Mode	Applicable Role(s)	Applicable Peer Procedure(s)
Broadcast	Broadcaster	Observation
Non-discoverable	Peripheral	N/A
Limited discoverable	Peripheral	Limited and General discovery
General discoverable	Peripheral	General discovery
Non-connectable	Peripheral, Broadcaster and Observer	N/A
Any connectable	Peripheral	Any connection establishment

Tabela 4 Procedure i odgovarajući načini rada

Procedure	Applicable Role(s)	Applicable Peer Mode(s)
Observation	Broadcaster	Broadcast
Limited discovery	Central	Limited discoverable
General discovery	Central	Limited & General discovery
Name discovery	Peripheral, Central	N/A
Any connection establishment	Central	Any connectable
Connection parameter update	Peripheral, Central	N/A
Terminate connection	Peripheral, Central	N/A

[3] Prevodi originalnih naziva načina rada u GAP:
Broadcast mode – režim emitovanja
Non-discoverable mode – režim nevidljivosti
Limited discoverable mode - režim ograničene vidljivosti
General discoverable mode – režim opšte vidljivosti
Non-connectable mode - režim bez mogućnosti konektovanja
Any connectable mode - režim koji dozvoljava konektovanja

8.5.3 Emitovanje i osmatranje

GAP način rada *broadcast* i procedura *observation* čine okvir za jednosmerni prenos podataka od uređaja u ulozi *Broadcaster*-a ka jednom ili više uređaja u ulozi *Observer*-a. Bitno je istaći da ne postoji način da emiter (*Broadcaster*) sazna da li su podaci koje je emitovao uopšte primljeni od strane jednog ili više osmatrača (*Observers*). Slično, zadatak osmatrača je da osluškuje radio medijum (privremeno ili neprekidno) u očekivanju emitovanih podataka, ali pri tom ne postoje garancije da će uopšte primiti bilo kakve podatke.

Paketi za oglašavanje koje šalje emiter sadrže stvarne, korisne podatke zajedno sa još nekoliko tzv. meta-podataka, kao što je BLE adresa uređaja, koje pridodaje sloj veze. Kao što je već rečeno u sekciji o sloju veze, pakte za oglašavanje sadrži do 31 bajt podatka, uz mogućnost korišćenja *scan request/scan response* transakcije kojom se količina emitovanih podataka povećava na maksimalno 251. Princip je da se najvažniji podaci šalju u okviru paketa za oglašavanje, a da se manje bitni podaci ostave za *scan response* paket.

Emiter, prosto, odašilje podatke u spoljašnji svet, tako da ih bilo koji osmatrač u okolini može primiti, bez obzira da je reč o jednom ili više osmatrača. Ovo čini značajnu razliku između emitera i periferije, s obzirom na to da periferija prestaje sa oglašavanjem odmah nakon što je konektovana i na taj način praktično postaje nevidljiva za druge centralne uređaje.

8.5.4 Otkrivanje uređaja (*Discovery*)

Otkrivanje uređaja se odnosi na mogućnost detektovanja prisustva BLE uređaja u okolini i prikupljanja osnovnih informacija o detektovanom uređaju. Otkrivanje uređaja ne podrazumeva obavezno i nameru konektovanja sa tim uređajem radi razmene podataka, iako je to u većini slučajeva prirodan tok interakcije. U nekim slučajevima, posebno ako je centralni uređaj opremljen displejom, otkrivanje se koristi da bi se korisniku vizuelno predočio spisak okolnih uređaja (koji rade u ulozi periferije) i prepustilo korisniku da iz tog spiska izabere uređaj (ili uređaje) za konektovanje.

***Režimi otkrivanja* (*Discoverability modes*)**

Režim otkrivanja uređaja (*discoverability mode*) se odnosi na to kako periferija oglašava svoje prisustvo drugim uređajima i šta ti uređaji mogu ili treba da urade sa informacijom iz paketa za oglašavanje. Način otkrivanja je određen sadržajem 1-bajtnog polja u paketu za oglašavanje. Različiti načini otkrivanja su optimizovani po različitim kriterijumima, kao što je trajanje baterije ili vreme konektovanja. Postoje tri režima otkrivanja:

Non-discoverable mode. Periferija koja je postavljena u režim otkrivanja „*non-discoverable*“ (tj. režim nevidljivosti), prosto, ne može biti otkrivena od strane centralnih uređaja. To ne znači da je periferiji zabranjeno da šalje pakete za oglašavanje, već znači da je drugim centralnim uređajima onemogućeno da nakon prijema paketa za oglašavanje takvu periferiju uvrste u spisak otkrivenih uređaja ili da na bilo koji način konstatuju njeno prisustvo. Periferni uređaj, po pravilu, prelazi u režim *Non-discoverable* nakon što je konektovan.

Limited discoverability mode. Ovaj način otkrivanja omogućava periferiji da bude dostupna za otkrivanje samo u ograničenom periodu vremena (npr. $30\ s$, $1\ min$ i sl.), ali i to da bi trebalo da bude tretirana sa višim prioritetom u odnosu na druge periferije kad bude otkrivena od strane centralnog uređaja. Uređaj u režimu „*Limited discoverability mode*“ (tj. režim ograničene vidljivosti) može biti otkriven samo od strane centrale koja obavlja posebnu proceduru za *limited discovery.* Ovaj način otkrivanja se retko koristi, osim npr. u slučaju rekonekcije, odnosno pokušaja ponovnog uspostavljanja prekinute konekcije. Kada detektuje

gubitak konekcije, periferije prelazi u „*Limited discoverability mode*“ u trajanju od recimo $30\ s$ i pri tom emituje pakete za oglašavanje tipa ADV_DIRECT_IND (videti sloj veze) sa BLE adresom centrale s kojom je bila u vezi. S druge strane, centrala, nakon uočenog gubitka konekcije, pokreće proceduru za *limited discovery*.

General discoverability mode. Omogućava uređaju da bude dostupan za otkrivanje za sve vreme dok je to potrebno. Dok je u ovom režimu, periferni uređaj emituje pakete za oglašavanje tipa ADV_IND iskazujući pri tom svoju „želju“ da bude otkriven i konektovan. Da bi otkrila periferije u režimu *General discoverability mode* (tj. režim opšte vidljivosti), centrala mora da izvršava istoimenu proceduru.

Pojedini periferni BLE uređaji, poput pametnog sata, obično se isporučuju od strane proizvođača inicijalno postavljeni u *General discoverability mode.* Nakon što se uveže sa centralnim uređajem (recimo, mobilnim telefonom vlasnika), BLE sat prelazi u *Non-discoverable mode*, u kome emituje pakete za oglašavanje tipa ADV_DIRECT_IND što mu omogućava da se ubuduće konektuje samo sa tim centralnim uređajem. Za sve ostale BLE uređaje (npr. druge mobilne telefone), sat više nije vidljiv. U slučaju reseta, sat se vraća na fabrička podešavanja, tj. u *General discoverability mode*.

***Procedure za otkrivanje* (*Discovery Procedures*)**

BLE specifikacija predviđa dve procedure za otkrivanje uređaja:

Limited discovery procedure. Centrali uređaj koji izvršava ovu proceduru, analizira svaki primljeni paket za oglašavanje i ukoliko je u paketu postavljen fleg *Limited* Discoverable, prijavljuje aplikaciji novo-otkriveni uređaj.

General discovery procedure. Centrali uređaj koji izvršava ovu proceduru, analizira svaki primljeni paket za oglašavanje i ukoliko je u paketu postavljen jedan od flegova *Limited Discoverable* ili *General Discoverable*, prijavljuje aplikaciji novo-otkriveni uređaj.

Ove dve procedure izvršavaju centralni uređaji. Centralni uređaj koji je zainteresovan sve moguće periferije, koristiće proceduru *General discovery*. Procedura *Limited discovery* omogućava otkrivanje samo onih periferija koje su u *Limited discovery* režimu.

8.5.5 Uspostavljanje konekcije (*Connection Establishment*)

Slično kao u slučaju otkrivanja, i za upravljanje konektovanjem postoji nekoliko režima i procedura.

Režimi uspostavljanja konekcije

Postoje tri periferna režima konektovanja, koji se razlikuju po tome koji tip paketa za oglašavanje periferija koristi da bi oglasila svoje prisustvo:

Non-connectable mode. U ovom režimu, periferni uređaj ili ne šalje pakete za oglašavanje ili šalje pakete za oglašavanje tipa ADV_NONCONN_IND ili ADV_SCAN_IND. U oba slučaja, uređaj nije raspoloživ za konektovanje. Centrala koja primi ovakav paket za oglašavanje, svesna je prisustva konkretne periferije, ali joj je zabranjeno da s njim uspostavlja konekciju.

Direct connectable mode. U ovom režimu, periferni uređaj šalje pakete za oglašavanje tipa ADV_DIRECT_IND (direktno oglašavanje). Kao što je to objašnjeno u sekciji o sloju veze, paketi za direktno oglašavanje ne sadrže druge podatke osim BLE adrese ciljnog centralnog uređaja, tj. periferija se direktno obraća konkretnoj centrali. Tipično, periferija ovakve pakete šalje sa većom učestanošću od regularnih paketa za oglašavanje, ali u kraćem vremenskom periodu. Jedna primena *Direct connectable mode* je tzv. brza rekonekcija, koja se dešava nakon gubitka konekcije. Nakon što detektuje gubitak konekcije, periferija može osnovano da pretpostavi da centrala želi da što pre ponovo uspostavi konekciju i zato kreće sa ubrzanim slanjem

ADV_DIRECT_IND paketa za oglašavanje. Samo centrala čija je BLE adresa sadržana u ADV_DIRECT_IND paketu će primiti paket, što će joj dati mogućnost da uspostavi konekciju sa periferijom.

Undirected connectable mode. U ovom režimu, periferni uređaj šalje pakete za oglašavanje tipa ADV_IND. Ovo je standardni režim uspostavljanja konekcije, u kome je periferija dostupna za konektovanje u dužem vremenskom periodu i to od strane bilo koje centrale.

Procedure za uspostavljanje konekcije

Tokom osmatranja, koje prethodi uspostavljanju konekcije sa periferijom, centrala ne može da utiče na to koji tip paketa za oglašavanje će primiti, da li ADV_IND ili ADV_DIREC_IND. S obzirom na to, razlike između nekoliko raspoloživih procedura za uspostavljanje konekcije ne potiču od tipa primljenog paketa za oglašavanje, već od vrste filtriranja koje centrala primenjuje na primljene pakete.

Auto connection establishment procedure. Tokom izvršenja ove procedure, host popunjava „belu listu“ nizom otkrivenih perifernih uređaja, a zatim izdaje komandu kontroleru da uspostavi konekciju sa prvo-otkrivenim uređajem iz liste. *Auto connection* procedura je korisna kada na strani centrale ne postoji nikakav kriterijum za izbor periferije s kojom bi se povezala već sa konektuje sa bilo kojim od potencijalno više perifernih uređaja u okolini.

General connection establishment procedure. Ova procedura se sastoji iz dva koraka i uobičajeno se koristi za konektovanje sa novo-otkrivenom, do tog trenutka nepoznatom periferijom. Centrala ulazi u osmatranje bez bele liste i prihvata sve pakete za oglašavanje. Za svaku otkrivenu periferiju, centrala mora da odluči da li da je konektuje ili ne. U tom cilju, host se obraća aplikaciji, a aplikacija korisniku od koga se očekuje da izabere periferiju za konektovanje. Alternativa je da aplikacija, na osnovu informacija iz paketa za oglašavanje, a na osnovu programske logike, samostalno izvrši odabir periferije. Nakon što je periferija izabrana, centrala konektuje periferiju korišćenjem *Direct connection establishment procedure.*

Selective connection establishment procedure. Ova procedura je identična *General connection establishment procedure*, sa tom razlikom da host filtrira primljene pakete za oglašavanje koristeći belu listu od ranije poznatih periferija. „*Selective connection*“ procedura je korisna u slučajevima kada korisnik treba da izabere za konektovanje jedan od ranije poznatih perifernih uređaja. Za razliku od „*Auto connection establishment procedure*“, ovde se zahteva da korisnik (tj. aplikacija) eksplicitno dozvoli uspostavljanja konekcije sa otkrivenim uređajem iz bele liste.

Direct connection establishment procedure. Ovo je standardna procedura za uspostavljanje konekcije sa konkretnim perifernim uređajem. Host izdaje komandu kontroleru da inicira konekciju sa periferijom zadate BLE adrese, čak i bez informacije o tome da li je ta periferija uopšte prisutna. Procedura uspostavljanje konekcije, naravno, može biti neuspešna ukoliko ciljana periferija nije dostupna ili je postavljena u *non-connectable* režim.

Korisno je još jedanput napomenuti da centrala ima na raspolaganju, generalno, dva načina za iniciranje konekcije. Prvi metod zahteva dva koraka: prvi je skeniranje, odnosno osmatranje, a drugi direktno konektovanje (na osnovu BLE adrese) sa jednim od perifernih uređaja otkrivenih tokom osmatranja. Drugi metod preskače korak osmatranja i umesto toga inicira konektovanje sa konkretnim perifernim uređajem, bez znanja o tome da li je taj uređaj uopšte prisutan u okolini.

8.5.6 Dodatne GAP procedure

Osim prethodno navedenih, GAP definiše još nekoliko procedura koje se mogu koristiti tek nakon što je konekcija uspostavljena.

Name discovery procedure. Paketi za oglašavanje mogu da sadrže brojne različite tipove korisnih podataka, poput *Device Name* – string koji sadrži čitljiv tekstualni opis uređaja (odgovara *hostname*-u kod računarskih mreža). Međutim, veličina paketa za oglašavanje je mala i uređaj može izabrati da u paket za oglašavanje ne uključi *Device Name*, već neki drugi podatak koji je bitniji u kontekstu konkretne primene. U takvim slučajevima, procedura *Name discovery* omogućava centrali (ali i periferiji) da sazna ime uređaja s kojim je uspostavila konekciju.

Connection parameter update procedure. Uspostavljanje konekcije obuhvata postavljanje brojnih parametara kojima se reguliše tok konekcije. Ove parametre, jednostrano, postavlja centrala. Parametri konekcije su ključni za balansiranje između propusnosti i potrošnje i njihove postavke se mogu promeniti naknadno, tokom konekcije, kao odgovor na promenjene zahteve u pogledu propusnosti ili potrošnje. Centrala je uvek zadužena za „fizičku“ promenu parametara konekcije i to može uraditi u bilo kom trenutku. Periferija ne može direktno da menja parametre, ali može da zahteva od centrale promenu izvesnih parametara. Nakon prijema takvog zahteva, centrala može da odbije ili odobri zahtev, a onda i da fizički izvrši promenu. Čak i da odobri zahtev, nove vrednosti parametar ne moraju biti identične onim koje je periferija tražila, već to mogu bilo koje vrednosti koje su, po mišljenju centrale, razumni i najbliži zahtevanim. Promena parametara na nove vrednosti moraju da sprovedu oba uređaja, što se postiže tako što centrala šalje odgovarajuću poruku periferiji.

Terminate connection procedure. Ovo procedura se koristi za zatvaranje konekcije i potpuno je simetrična: bilo centrala ili periferije može da zatvori konekciju u bilo kom trenutku. Pri tom, strana koja inicira zatvaranje, mora zajedno sa zahtevom da suprotnoj strani pošalje i kôd razloga zatvaranja.

8.5.7 Dodatne GAP definicije

Oslanjajući se na uloge, režime i procedure, GAP takođe uvodi i dva dodatna proširenja koja su relevantna prilikom razvoja BLE aplikacija.

Format podataka koji se oglašavaju

Prethodno je u više navrata rečeno da paketi za oglašavanje nose korisničke podatke, ali ne i u kom formatu si ti podaci zapakovani u ovom paketu. GAP definiše generički (uopšteni) kontejner za podatke koji se sastoji iz niza struktura podataka, od kojih svaka sadrži 1-bajtno polje, *AD Type*, koje ukazuje na tip podatka i stvarni podatak (promenljive dužine). Svaka ovakva struktura sadrži jedan elementarni podatak. Kompletan spisak dopuštenih AD tipova je dostupan u okviru BLE specifikacije. Najčešće korišćeni tipovi su navedeni u tabeli.

Table 3-3. AD Types

Name	Actual data length in bytes	Description
Flags	1 (extendable)	Used to set limited or general discovery mode, as described in "Discovery" on page 39
Local Name	variable	Partial or complete user-readable local name in UTF-8
Appearance	2	A 16-bit value describing the type of device sending the advertising packet
TX Power Level	1	The power level in dBm used to transmit the advertising packet, useful to calculate path loss at the observer or central end
Service UUID	variable	A complete or partial list of GATT services offered by the device sending the packet (as a GATT server)
Slave Connection Interval Range	4	A suggestion to the central about the connection interval range that best fits this peripheral
Service Solicitation	variable	A list of GATT services supported by the device sending the packet (as a GATT client)
Service Data	variable	A UUID representing a GATT service and its associated data
Manufacturer Specific Data	variable	Freely formattable data, to be used at the discretion of the implementation

Nisu svi AD tipovi korisni u svim situacijama. Uz to ograničenje u dužini paketa za oglašavanje na 31 bajt, pa čak i uz proširenje koje donosi *Scan Response*, ograničava i to koliko se AD tipova može koristiti. Zato je bitno razumeti koji tip informacije je dostupan i koja polja su relevantna u različitim situacijama. U nastavku ove sekcije ukratko je predstavljeno nekoliko bitnih AD tipova.

Service UUID. Servisi (tj. GATT servisi) su bitan koncept u Bluetooth, pa i BLE specifikaciji i biće detaljnije obrađeni u narednom poglavlju (GATT). Servisi objedinjuju logički srodne podatke u posebne celine. Na ovom nivou, BLE uređaji povezani konekcijom igraju uloge GATT servera (onaj koji omogućava pristup svojim servisima) i GATT klijenta (onaj koji pristupa servisima GATT servera). BLE specifikacija definiše brojne standardne (oficijelne) servise i svaki od ovih servisa ima svoj, standardom definisan, jedinstveni UUID. Uz to korisnik ima slobodu da kreira svoje nove servise i za njih koristi svoje UUID-ove. U suštini, servisi omogućavaju interoperabilnost između BLE uređaja. Na primer, bilo koji uređaj koji implemetira standardni *Battery Service* mora to uraditi na način kako je specificirano BLE standardom – mora se držati formata podataka i UUID-ova koji su propisani ovim servisom.

Paket za oglašavanje može da sadrži kompletan ili delimičan spisak servisa koje uređaj podržava. Ovaj spisak je zapravo niz UUID-ova odgovarajućih servisa. U paketu za oglašavanje, svaki od ovih UUID-ova se navodi sa tipom *Service UUID AD Type*. Ova informacija je korisna centrali (BLE uređaj koji prima paket za oglašavanje), jer preko nje saznaje koje servise podržava periferija bez potrebe da sa njom uspostavlja konekciju. Ukoliko su podržani servisi interesantni centrali, ona će uspostaviti konekciju sa periferijom; ukoliko nisu, neće.

Service Solicitation. Ovaj AD tip donekle suprotan tipu *Service UUID*, jer omogućava periferiji da u paketu za oglašavanje navede servise (tj. njihove UUID-ova) za koje očekuje da budu dostupni na strani centrale. Na GATT nivou, oba povezana uređaja, tj. centrala i periferija, mogu igrati bilo koju od dve uloge GATT server ili GATT klijent, ili čak obe. To znači da ne samo što centrala može da koristi servise periferije, već i periferija može da koristi servise centrale. Međutim, ukoliko na osnovu podatka tipa *Service Solicitation* iz paketa za oglašavanje vidi da periferija očekuje servis koji ona ne poseduje, centrala može odustati od konektovanja.

Service Data. Servisima se može pristupati samo preko otvorene konekcije. Ipak, pojedini podaci koji su obuhvaćeni servisom mogu se preneti i paketom za oglašavanje. Ovo je omogućeno AD tipom *Service Data*, uz navođenje UUID-a servisa i odgovarajućeg podatka. Na ovaj način,

podaci iz servisa se mogu distribuirati proizvoljnom broju slušalaca, bez potrebe uspostavljanja konekcije sa svakim od njih.

Manufacturer Specific Data. Ovo je generički AD tip, za proizvoljne podatke (po izboru proizvođača uređaja).

GAP servis (GAP Service)

Kao što je već rečeno, BLE uređaj sadrži set GATT servisa preko kojih izlaže okolnim uređajima svoje podatke i funkcionalnost. Jedan od obaveznih servisa svakog BLE uređaja je GAP servis. Ovaj servis je dostupan za pristup radi čitanja svakom konektovanom uređaju i sadrži tri karakteristike:

Device Name characteristic. Sadrži ime uređaja u obliku stringa (čitljiv tekst). Ova karakteristika može biti uključena i u paket za oglašavanje – AD tip *Device Name*. Takođe, može biti pročitana procedurom *Name discovery*.

Appearance characteristic. Sadrži 16-bitnu vrednost koja ukazuje na generičku kategoriju BLE uređaja (telefon, računar, sat, senzor itd.). Uređaj sa druge strane konekcije, vrednost ove karakteristike može da za prikazivanje odgovarajuće ikone na svom displeju. Sadržaj *Appearance characteristic* takođe može biti uvršten i u paket za oglašavanje – AD tipa *Appearance*.

Peripheral Preferred Connection Parameters (*PPCP*) *characteristic*. Centrala, nakon uspostavljanja konekcije može da pročita vrednost ove karakteristike (mada to nije obavezno) i izvrši proceduru ažuriranja konekcionih parametara (kako bi ih uskladila sa „željama“ periferije).

8.6 GATT (*Generic Attribute Profile*)

GATT (*Generic Attribute Profile*) je profil koji do detalja specificira strukturu profilnih i korisničkih podataka, kako i procedure za razmenih tih podataka između konektovanih BLE uređaja. Za razliku od GAP, koji definiše interakcije niskog nivoa između BLE uređaja, GATT je isključivo fokusiran na procedure i formate za prenos podataka. Pojednostavljeno rečeno, svaki pojedinačni podatak koji je relevantan aplikaciji ili korisniku mora biti formatiran, upakovan i prenet između dva uređaja shodno pravilima koje definiše GATT.

GATT je takođe referentni okvir za sve izvedene, tzv. *GATT-based* profile koji pokrivaju specifične slučajeve primene i obezbeđuju interoperabilnost između uređaja različitih proizvođača. GATT je ključni deo BLE specifikacije, s obzirom na to da su svi standardizovani BLE profili zasnovani na GATT.

GATT koristi ATT kao transportni protokol za razmenu podataka između uređaja. Podaci su, shodno GATT-u, organizovani hijerarhijski, podeljeni u sekcije koje se nazivaju servisima. Servis grupiše konceptualno povezane delove korisničkih podataka, tzv. karakteristike.

8.6.1 Uloge

Kao svaki protokol ili profil u BLE specifikaciji, GATT uvodi uloge, koje se dodeljuju uređajima tokom međusobne interakcije. Kod GATT postoje samo dve uloge, klijent i server, koje su podudarne sa istoimenim ulogama u protokolu ATT:

Klijent (*Client*). GATT klijent je zapravo nadogradnja ATT klijenta. GATT klijent šalje zahteve za pristup podacima (atributima) GATT servera i prima odgovore od servera. GATT klijent unapred ne zna ništa o atributima servera i zbog toga je neophodno da prethodno pribavi informacije o tome koje atribute poseduje server i koja je njihova priroda. Ova aktivnost se zove *service*

discovery (*otkrivanje servisa*) i ostvaruje se razmenom niza upita/odgovora između klijenta I server, o čemu će više reči biti kasnije u ovoj sekciji. Nakon otkrivanja servisa, klijent može da čita i upisuje atribute servera, kao i da prima obaveštenja o promenama vrednosti atributa na strani servera.

Server (*Server*). GATT server je nadogradnja ATT servera. GATT server prima zahteve od GATT klijenta i šalje odgovore klijentu. Takođe, server može, ukoliko je tako konfigurisan, da „samoinicijativno“ (tj. asinhrono) šalje klijentu obaveštenja o promenama vrednosti izvesnih atributa. U suštini, GATT server obezbeđuje pristup GATT klijentu svim svojim izloženim podacima koji su u vidu atributa organizovani u servise. Svaki BLE uređaj mora da sadrži bar osnovni GATT server koji će odgovarati na zahteve klijenta, pa makar kao odgovor na zahtev uvek vraćao indikaciju o grešci.

Česta je greška da se uloge klijenta i servera (iz protokola ATT i profila GATT) poistovećuju sa ulogama centrale i periferije u kontekstu profila GAP, iako ova dva para uloga nisu ni u kakvoj vezi. Razmotrimo, na primer, fitnes traker koji je u vezi sa mobilnim telefonom. Fitnes traker igra ulogu GAP periferije i deluje kao GATT server onda kada na zahtev telefona isporučuje prikupljene podatke. Takođe, ponekada, traker može igrati ulogu GATT klijenta, npr. onda kada od telefona traži informaciju o tačnom vremenu. GATT uloga klijent/server se određuje na osnovu smera u kome se prenose zahtevi za pristup podacima i sami podaci. S druge strane, GAP uloge periferija/centrala ostaju nepromenljive – traker je uvek periferija i telefon je uvek centrala.

8.6.2 UUID

UUID (*Universally Unique Identifier*) – univerzalno jedinstveni identifikator je 128-bitni (16-bajtni) broj koji je garantovano (ili bar sa veoma velikom verovatnoćom) globalno jedinstven. Osim kod BLE, UUID-ovi se koriste u mnogim drugim protokolima i aplikacijama, a njihov format, upotreba i generisanje je definisano posebnim standardom (ISO/IEC 9834-8:2005). Poenta je u tome da u svim tim protokolima i aplikacijama ne postoje dva identična UUID.

Iz razloga efikasnosti, kao i zbog toga što bi 16-bajtni UUID-ovi zauzimali veliki deo kratkih BLE paketa, BLE specifikacija uvodi dva dodatna UUID formata: 16-bitni i 32-bitni UUID. Ovi skraćeni formati se mogu koristiti samo za UUID-ove koji su definisani u BLE specifikaciji. UUID-ovi koje uvodi korisnik moraju biti pune dužine.

Puni, 128-bitni UUID se rekonstruiše na osnovu skraćene verzije umetanjem 16- ili 32-bitnog kratkog UUID u osnovni BLE UUID (na poziciji x-ova):

xxxxxxxx-0000-1000-8000-00805F9B34FB

Prethodna linija teksta je standardni, heksadecimalni UUID zapis. Svaka hex-cifra predstavlja 4 bita.

BLE specifikacija propisuje UUID-ove za sve tipove, servise i profile koji su definisani specifikacijom. Međutim, ukoliko su korisničkoj aplikaciji potrebni dodatni UUID-ovi (npr. korisnik uvodi svoj, novi atribut ili profil), oni moraju imati pun formati i moraju se generisati na veb sajtu organizacije ITU (da bi se garantovala jedinstvenost).

8.6.3 Atributi

GATT (kao i ATT) definiše atribut kao najmanju adresibilnu jedinicu podataka. Atributi sadrže relevantne korisničke podatke ili meta-podatke o strukturi i grupisanju različitih atributa u okviru servera. GATT i ATT rade samo sa atributima, tako da sve informacije koje su predviđene za razmenu između servera i klijenta moraju biti organizovane u ovoj formi.

Atributi su uvek locirani na serveru i njima pristupa (i, eventualno, ih modifikuje) klijent. GATT specifikacija definiše atribute na konceptualnom nivou i ne bavi se time kako se i gde atributi skladište. S obzirom da atributi sadrže kako statičke (nepromenljive) podatke, tako i podatke koji se vremenom menjaju, za njihovo smeštanje se koristi po potrebi permanentna memorija (*Flash*) ili RAM.

Svaki atribut sadrži informacije o samom sebi (meta-podatke), a onda i stvarni podatak (što je već poznato iz sekcije o ATT protokolu).

8.6.4 Adresa atributa (*Handle*)

Adresa atributa, ili *handle*, je jedinstveni 16-bitni identifikator svakog atributa u okviru konkretnog GATT servera. *Handle* je deo svakog atributa, koji omogućava pristup atributu. Garantovano je nepromenljiv tokom trajanja konekcije, ili čak između konekcija, u slučaju uvezanih uređaja. Broj raspoloživih *handle* vrednosti na svakom GATT severu je $2^{16} - 1$ (65535), mada je, u praksi, broj atributa na serveru tipično nekoliko desetina.

Pojam *handle range* (opseg *handle* vrednosti) se odnosi na sve atribute čiji su *handle*-ovi u datom opsegu. Na primer, *handle* opseg $0x0100 – 0x010A$ se odnosi na sve atribute čija je *handle* vrednost u opsegu $0x0100 – 0x010A$. Na strani servera, atributi su poređani u niz po rastućim vrednostima *handle*-ova, s tim što su šupljine u *handle* vrednosti između uzastopnih atributa dopuštene. To praktično znači da klijent ako zna *handle* nekog atributa, ne može da smatra da je *handle* sledećeg atributu u nizu za 1 veći. Klijent mora da koristi *service discovery* da bi saznao *handle*-ove atributa za koje je zainteresovan.

8.6.5 Tip atributa

Tip atributa nije ništa drugo nego UUID koji je pridružen atributu. To može biti 16-, 32- ili 128-bitni UUID, koji zauzima 2, 4 ili 16 bajta, respektivno, kao što je to već objašnjeno. Tip atributa određuje tip podatka čiju vrednost atribut nosi. Dostupni su mehanizmi za pronalaženje atributa na osnovu njihovog tipa (spada u *service discovery*).

8.6.6 Dozvole pristupa (*Permissions*)

Dozvole pristupa, ili permisije, su meta-podatak koji specificira koje ATT operacije su dopuštene nad datim atributom i koji nivo sigurnosti je potreban da bi se pristupilo atributu. ATT (i GATT) definiše sledeće permisije:

Access Permissions. Definiše da li klijent ima pravo čitanja ili upisa (ili oba) vrednosti atributa. Svaki atribut može imati jedna od sledećih prava pristupa:

- *None*. Klijentu nije dopušteno ni da čita ni da upisuje vrednost atributa.
- *Readable*. Klijentu je dopušteno da čita vrednost atributa.
- *Writable*. Klijentu je dopušteno da upisuje vrednost atributa.
- *Readable and writable*. Klijentu je dopušteno da čita i upisuje vrednost atributa.

Encryption. Određuje da li se za pristup atributu zahteva određeni nivo sigurnosti i koji. GATT definiše sledeće enkripcione permisije (više o sigurnosnim opcijama u poglavlju „Sigurnost“):

- *No encription required* (*Security Mode 1, Level 1*). Nije potrebna enkripcija za pristup atributu. Vrednost atributa se između servera i klijenta prenosu u obliku otvorenog teksta.
- *Unauthenticated encryption required* (*Security Mode 1*, *Level 2*). Atributu se može pristupati samo putem enkriptovane konekcije, s tim što enkripcioni ključevi ne moraju biti autentikovani (ali mogu).

- *Authenticated encryption required* (*Security Mode 1, Level 3*). Atributu se može pristupati samo putem enkriptovane konekcije sa autentikovanim ključevima.

Authorization. Određuje da li se za pristup atributu zahteva korisnička permisija (autorizacija – provera identiteta korisnika). Moguće su dve opcije:

- *No authorization required*. Atributu se može pristupati bez autorizacije.
- *Authorization required*. Pristup atributu zahteva autorizaciju.

Prethodne tri permisije su međusobno nezavisne i mogu se kombinovati na nivou pojedinačnih atributa.

8.6.7 Vrednost atributa

Vrednost atributa sadrži stvarni podatak koji atribut predstavlja. Za vrednost atributa ne postoji ograničenje u pogledu tipa podataka, s tim što je dužina vrednosti atributa ograničena na maksimalno 512 bajta. U zavisnosti od tipa atributa, vrednost može da sadrži sam podatak ili dodatne informacije o atributu. Vrednost atributa je jedini deo atributa kojem klijent može da pristupa radi čitanja ili upisa (uz odgovarajuće prava pristupa). Svi ostali delovi atributa čine strukturu atributa i ne mogu se modifikovati, niti im se može direktno pristupati od strane klijenta.

Set atributa koji je obuhvaćen GATT serverom može se zamisliti kao tabela (poput Tabela 5) u kojoj svaka vrsta predstavlja jedan atribut, a svaka kolona jedan konstitutivni deo atributa. Ovaj fiktivni GATT server sadrži samo šest atributa. GATT server tipičnog BLE uređaja sadrži mnogo više atributa. Kao što je prethodno rečeno, *handle*-ovi atributa ne moraju biti sukcesivni brojevi (dozvoljene su šupljine), ali atributi moraju biti poređani u rastući redosled po njihovim *handle* vrednostima. Kolona *Value* u tabeli je popunjena tako da ukaže na raznolikost formata podataka koji se mogu čuvati u atributima. U ovom primeru, pojedini atributi sadrže 16-bitne vrednosti (prva dva i poslednji). Atribut sa *hande*-om $0x0215$ sadrži string dužine 23 karaktera, dok atribut $0x030C$ sadrži niz od 4 bajta. Atribut $0x030D$ sadrži broj u pokretnom zarez vrednosti 36.43, a njegova dužina je 8 bajta. Značenje atributa, tj. koji tip podatka atribut sadrži (npr. ime uređaja, trenutna temperatura, nivo baterije,...) određeno je UUID-om. Vidimo da su u tabeli svi osim jednog UUID-a 16-bitni (što znači da je reč o standardnim UUID-ovama, koji su definisani u BLE specifikaciji. UUID atributa $0x030D$ je naveden u punoj dužini (128-bita), što znači da se radi od nestandardnom atributu koji je uveden od strane korisnika (tj. specifičan je na taj konkretan BLE uređaj). Takođe, u tabeli vidimo da se atributi razlikuju po pravu pristupa i nivou sigurnosti.

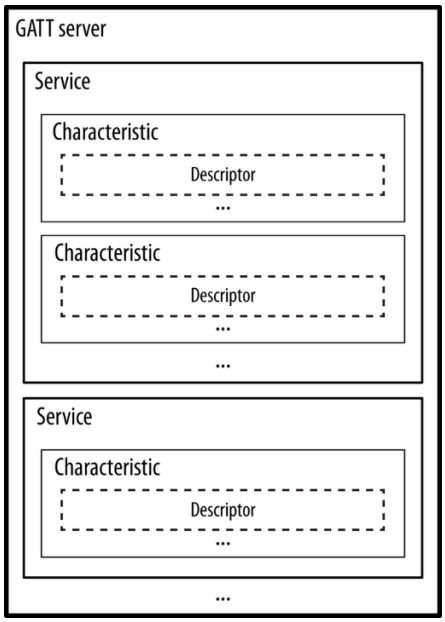
Tabela 5 Tabelarni prikaz atributa

Handle	Type	Permissions	Value	Value length
0x0201	$UUID_1$ (16-bit)	Read only, no security	`0x180A`	2
0x0202	$UUID_2$ (16-bit)	Read only, no security	`0x2A29`	2
0x0215	$UUID_3$ (16-bit)	Read/write, authorization required	"a readable UTF-8 string"	23
0x030C	$UUID_4$ (128-bit)	Write only, no security	`{0xFF, 0xFF, 0x00, 0x00}`	4
0x030D	$UUID_5$ (128-bit)	Read/write, authenticated encryption required	`36.43`	8
0x031A	$UUID_1$ (16-bit)	Read only, no security	`0x1801`	2

8.6.8 Atributi i hijerarhija podataka

Atributi su uvedeni na nivou ATT protokola. ATT, takođe, omogućava klijentu da pristupa atributima servera. GATT ide korak dalje i uspostavlja striktnu hijerarhiju u organizaciji atributa na način koji

omogućava da se razmena podataka između klijenta i servera može regulisati konciznim setom pravila. Sl. 24 ilustruje hijerarhiju podataka koju uvodi GATT.



Sl. 24 Hijerarhija podataka u profilu GATT

U okviru GATT servera, atributi su grupisani u servise (*Service*), od kojih svaki sadrži ni jednu ili više karakteristika (*Characteristic*). Karakteristike, dalje, mogu da sadrže ni jedan ili više deskriptora (*Descriptors*). Ovakav način strukturiranja podataka je obavezan za svaki BLE uređaj koji je usaglašen sa GATT profilom (praktično svi komercijalno dostupni BLE uređaji). To praktično znači da ja svaki atribut GATT servera, bez izuzetaka, obuhvaćen jednom od tri kategorije.

Za većinu tipova podataka u GATT hijerarhiji, važno je praviti razliku između *definicije* (kompletna grupa atributa koji čine dati deskriptor, karakteristiku ili servis) i *deklaracije*. Deklaracija je jedan atribut koji je uvek prvi (po *handle* vrednosti) u nizu atributa u okviru definicije, i koji sadrži većinu meta-podataka o podacima koji slede. Sve deklaracije imaju *read-only* permisiju i to bez sigurnosti, jer ne sadrže sigurnosno osetljive podatke. Deklaracije su, u suštini, strukturalni atributi koji omogućavaju klijentu da pronađe i otkrije raspored i prirodu atributa na serveru.

8.6.9 Servisi

GATT servis objedinjuje konceptualno povezane atribute u zajedničku sekciju atributa. U terminologiji BLE specifikacije, pojam definicija servisa (*service definition*) se odnosi na sve atribute obuhvaćene jednim servisom. Posmatrano na taj način, atributi samog GATT servera su zapravo servisi, odnosno niz definicija servisa, od kojih svaka počinje atributom koji označava početak servisa (tzv. deklaracija servisa – *service declaration*). Tip ovog atributa i njegova vrednost su definisani u GATT-u, kao u Tabela 6. U deklaraciji iz tabele, $\text{UUID}_{primary_service}$ $(0x2800)$ i $\text{UUID}_{secondary_service}$ $(0x2801)$ se odnose na standardne UUID-ove koji se koriste za označavanje početka servisa. Prvi UUID označava početak primarnog, a drugi sekundarnog servisa i oba su 16-bitni UUID-ovi (zato što su definisani u samoj BLE specifikaciji). Primarni servis obuhvata relevantne standardne podatke i funkcionalnosti uređaja koje su izložene kroz GATT server. Sekundarni servis je uvek dodatak nekom drugom, primarnom servisu i sam za sebe ne može da egzistira. Sekundarni servise se retko koriste. Vrednost deklaracije servisa (deo *Value* u tabeli) takođe sadrži UUID (*Service UUID*), koji identifikuje tip samog servisa. To može biti neki od tipova servisa koji su definisani specifikacijom, u kom slučaju je dužina *Value* sekcije 16 ili 32 bita, ili novi tip, uveden od strane korisnika, što zahteva UUID u punoj dužini (128 bita). Dakle, definicija

servisa uvek počinje deklaracijom servise, nakon koje sledi, po pravilu, mnoštvo drugih atributa, obično u formi karakteristika i deskriptora.

Tabela 6. Atribut deklaracije servisa (*Service Declaration attribute*)

Handle	Type	Permissions	Value	Value length
0xNNNN	$UUID_{primary_service}$ or $UUID_{secondary_service}$	Read Only	Service UUID	2, 4, or 16 bytes

Definicija servisa može da sadrži i reference na druge servise. Za referenciranje konkretnog servisa koristi se *include definition*, koja sadrži samo jedan atribut sa svim detaljima koji su neophodni kako bi klijent mogao da se obrati referenciranom servisu (Tabela 7). Referenciranje servisa može da pomogne u izbegavanju ponavljanja istih podatka u GATT serveru. Na taj način se smanjuje zauzeće memorije i pojednostavljuje struktura GATT servera. $UUID_{include}$ ($0x2802$) je definisan BLE specifikacijom i isključivo se koristi kao oznaka tip za *include* deklaraciju. Polja *Value* u ovom slučaju sadrži početni i završni *handle* servisa koji se referencira zajedno sa njegovim UUID-om.

Tabela 7. Atribut deklaracije za uključivanje servisa (*Include Declaration attribute*)

Handle	Type	Permissions	Value	Value length
0xNNNN	$UUID_{include}$	Read Only	Included service handle, end group handle, Included service UUID	6, 8, or 20 bytes

8.6.10 Karakteristike (*Characteristics*)

Karakteristika se može razumeti kao kontejner za korisničke podatke, koji uvek sadrži bar dva atributa: *deklaracija karakteristike* (*characteristic declaration*) i *vrednost karakteristike* (*characteristic value*). Deklaracija karakteristike je atribut koji sadrži meta-podatke o korisničkom podatku, dok je vrednost karakteristike atribut koji u svom polju za vrednost sadrži vrednost samo korisničkog podatka.

Dodatno, nakon atributa za vrednost karakteristike mogu da slede deskriptori, koji predstavljaju dopunu meta-podataka iz deklaracije karakteristike. Atributi za deklaraciju i vrednost karakteristike, zajedno sa deskriptorima čine *definiciju karakteristike* (*characteristic definition*) – set atributa koji zajedno čine jednu karakteristiku.

Tabela 8 prikazuje strukturu prva dva atributa svake karakteristike. GATT karakteristike su uvek deo servisa, odnosno, ne mogu egzistirati same za sebe i uvek su deo definicije servisa.

Tabela 8. Atributi za deklaraciju i vrednost karakteristike (*Characteristic declaration and characteristic value attributes*)

Handle	Type	Permissions	Value	Value length
0xNNNN	$UUID_{characteristic}$	Read Only	Properties, value handle (0xMMMM), characteristic UUID	5,7, or 19 bytes
0xMMMM	Characteristic UUID	Any	Actual value	Variable

***Atribut za deklaraciju karakteristike* (*Characteristic declaration attribute*)**

Tip atributa za deklaraciju karakteristike ($UUID_{characteristic}$ – prva vrsta u Tabela 8) je standardizovan i njegova vrednost je $0x2803$. Ovaj atribut označava početak karakteristike i poput sličnih deklaracija (npr. deklaracija servisa i *include* deklaracija) može samo da se čita, ali ne i modifikuje (*Read only* u polju *Permissions*).

Deo *Value* atributa za deklaraciju karakteristike sadrži tri nadovezana podatka koji bliže definišu karakteristiku (Tabela 9):

Tabela 9. Vrednost atributa za deklaraciju karakteristike (*Characteristic declaration attribute value*)

Name	Length in bytes	Description
Characteristic Properties	1	A bitfield listing the permitted operations on this characteristic
Characteristic Value Handle	2	The handle of the attribute containing the characteristic value
Characteristic UUID	2, 4, or 16	The UUID for this particular characteristic

Characteristic Properties. Ovo je 8-bitno bit-polje čiji sadržaj, zajedno sa još dva bita iz deskriptora za prošireni skup osobina (*Extended Properties Descriptor* – vidi ispod), ukazuje na to koje operacije i procedure su dopuštene sa ovom karakteristikom. Podržane opcije su navedene u Tabela 10. Čitanjem sadržaja ovog polja, klijent može da sazna koje operacije su dopuštene nad konkretnom karakteristikom. Ovo je posebno važno za *Notify* i *Indicate* osobine (biće objašnjeno kasnije), zato što ove operacije inicira server, ali da bi ih primio, klijent mora da ih dozvoli (biće objašnjeno kasnije).

Characteristic Value Handle. Ova dva bajta sadrže *handle* atributa koji sadrži stvarnu vrednost karakteristike.

Characteristic UUID. Ovo je UUID (tip) konkretne karakteristike i može biti bilo standardni (16- ili 32-bitni) UUID ili 128-bitni, korisnički UUID.

Tabela 10. *Characteristic properties*

Table 4-6. Characteristic properties

Property	Location	Description
Broadcast	Properties	If set, allows this characteristic value to be placed in advertising packets, using the Service Data AD Type (see "GATT Attribute Data in Advertising Packets")
Read	Properties	If set, allows clients to read this characteristic using any of the ATT read operations listed in "ATT operations"
Write without response	Properties	If set, allows clients to use the Write Command ATT operation on this characteristic (see "ATT operations")
Write	Properties	If set, allows clients to use the Write Request/Response ATT operation on this characteristic (see "ATT operations")
Notify	Properties	If set, allows the server to use the Handle Value Notification ATT operation on this characteristic (see "ATT operations")
Indicate	Properties	If set, allows the server to use the Handle Value Indication/Confirmation ATT operation on this characteristic (see "ATT operations")
Signed Write Command	Properties	If set, allows clients to use the Signed Write Command ATT operation on this characteristic (see "ATT operations")
Queued Write	Extended Properties	If set, allows clients to use the Queued Writes ATT operations on this characteristic (see "ATT operations")
Writable Auxiliaries	Extended Properties	If set, a client can write to the descriptor described in "Characteristic User Description Descriptor"

***Atribut za vrednost karakteristike* (*Characteristic value attribute*)**

Atribut za vrednost karakteristike sadrži stvarni korisnički podatak, koji klijent može da pročita ili upiše. Tip ovog atributa je identičan UUID-u iz polja *Value* atributa za deklaraciju karakteristike. Ovo je sada konkretan UUID, specifičan za tip podatka koje je sadržan u karakteristici (npr. temperatura u ^{0}C, brzina u km/h, kôd pritisnute dirke tastature i sl.)

***Deskriptori karakteristike* (*Characteristic Descriptors*)**

Deskriptori karakteristike (ili samo deskriptori) se uglavnom koriste kako bi se klijentu predočili meta-podaci, odnosno dodatne informacije o karakteristici i njenoj vrednosti. Deskriptori su uvek smešteni unutar definicije karakteristike i to nakon atributa za vrednost karakteristike. Deskriptor je uvek obuhvaćen jednim atributom, tj. atributom za deklaraciju deskriptora karakteristike (*characteristic descriptor declaration*). UUID ovog atributa je tip deskriptora, a u njegovom polju za vrednost je sve ono što je specificirano tipom konkretnog deskriptora. Postoje dva tipa deskriptora:

GATT-defined descriptors. U ovu kategoriju spadaju osnovni, često korišćeni tipovi deskriptora koji proširuju definiciju karakteristike dodatnim meta-podacima o samoj karakteristici. U nastavku ove sekciju su predstavljeni neki od najčešće korišćenih GATT deskriptora.

Profile or vendor-specific descriptors. Bez obzira na to da li je reč o standardnom profilu (definisan u BLE specifikaciji) ili profilu proizvođača BLE uređaja, deskriptori iz ove kategorije mogu sadržati podatke bilo kog tipa koji pružaju dodatne informacije o vrednosti karakteristike, kao npr. tip kodiranja, ili bilo koja druga informacija koja je specifična za datu karakteristiku.

U nastavku su predstavljena tri tipa *GATT-defined* deskriptora:

Extended Properties Descriptor. Ovaj deskriptor, onda kad je prisutan u definiciji karakteristike, sadrži dva dodatna bita koji dopunjuju polje *Characteristic properties* u atributu za deklaraciju karakteristike.

Characteristic User Description Descriptor. Kao što se može naslutiti na osnovu naziva, ovaj deskriptor sadrži tekstualni (čitljiv) opis karakteristike u okviru čije definicije je smešten. Na primer, deskriptor može da sadrži string: „Temperatura u dnevnoj sobi“.

Client Characteristic Configuration Descriptor. Ovaj tip deskriptora (poznat i pod skraćenicom CCCD) je nesumnjivo najznačajniji i najčešće korišćeni deskriptor i od suštinske je važnosti za rad većine BLE periferija u brojnim primenama. Njegova funkcija je jednostavna: deluje kao prekidač koji dozvoljava ili zabranjuje serverske notifikacije (*server-initiated updates*).

CCCD sadrži samo dva bita, pri čemu jedan odgovara notifikaciji, a drugi indikaciji. Klijent može da setuje ili resetuje ove bitova u bilo kom trenutku. S druge strane, server proverava vrednost ovih bitova pri svakoj promeni vrednosti karakteristike i u zavisnosti od njihove vrednosti odlučuje da li da klijentu pošalje notifikaciju/indikaciju ili ne. Uvek kada klijent želi da dozvoli notifikacije ili indikacije neke konkretne karakteristike koja ima podršku za to, potrebno je pošalje *Write Request ATT* paket da bi postavio odgovarajući CCCD bit na 1. Server odgovara paketom *Write Response* i počinje sa slanjem odgovarajućih notifikacija/indikacija uvek kada poželi da upozori klijenta na promenu vrednosti karakteristike.

CCCD poseduje dve posebne osobine, po kojima se razlikuje od svih ostalih atributa:

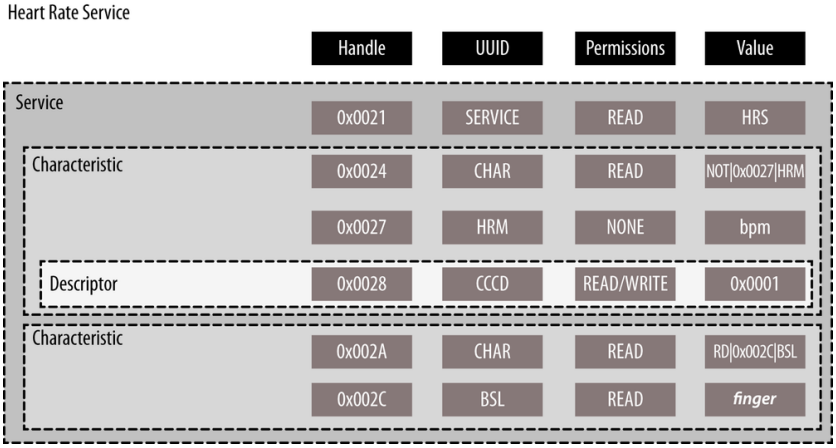
CCCD vrednosti su jedinstvene za konekciju. U scenarijima kad je centrala konektovana sa više periferija i pri tome igra ulogu GATT servera, svaka periferija će dobiti svoju kopiju CCCD vrednosti. Ovo omogućava da se notifikacije/indikacije šalju selektivno samo ka periferijama koje su ih dozvolile.

CCCD vrednosti se čuvaju između konekcija sa uvezanim uređajima. Ako klijent i server rade na uvezanim BLE uređajima, tada će poslednja CCCD vrednost koju je klijent postavio biti sačuvana na serveru, a onda i obnovljena nakon ponovne konekcije, bez obzira na to koliko je vremena u međuvremenu poteklo.

Characteristic presentation format descriptor. Onda kada je prisutan, ovaj deskriptor sadrži format vrednosti karakteristike. Spisak podržanih formata uključuje: *Boolean*, *string*, *integer*, *floating-point* itd.

8.6.11 Primer servisa

U ovoj sekciji je predstavljen primer jednog konkretnog servisa koji nalazi primenu u mnogim komercijalnim BLE uređajima. Reč je o HRS (*Heart Rate Service*) koji obezbeđuje pristup podatku o pulsu korisnika. Na Sl. 25 je prikazana instanca HRS-a na nekom fiktivnom serveru. HRS nije jedini servis ovog servera, tako da se prikaz na slici može smatrati jednim isečkom kompletnog seta atributa koji je dostupan klijentu. Primera radi, HRS može biti pokrenut na pametnom satu, čija je jedna od funkcija upravo merenje pulsa. Osim ove, sat ima i brojne druge funkcije, kao na primer, prebrojavanje koraka, praćenje lokacije i slično. Za svaku takvu funkciju, možemo pretpostaviti da postoji poseban servis, koji je deo istog servera.



Sl. 25 GATT *Heart Rate Service*

U nastavku sledi objašnjenje HRS servisa na Sl. 25, atribut po atribut (odnosno *handle* po *handle*).

Handle $0x0021$. Ovaj atribut sadrži deklaraciju servisa za HRS i sadrži sledeća polja:

UUID. U ovom polju je sadržan je standardni 16-bitni UUID za deklaraciju primarnog servisa, $\text{UUID}_{\text{primary_service}}$ ($0x2800$).

Value. U ovom polju upisan 16-bitni UUID *Heart Rate Service*, $0x180D$, koji je definisan BLE specifikacijom.

Handle $0x0024$. Ovaj atribut sadrži deklaraciju karakteristike za *Heart Rate Measurement* karakteristiku. Sadržaj pojedinačnih polja je sledeći:

UUID. Sadrži standardni 16-bitni UUID za deklaraciju karakteristike, $UUID_{characteristic}$ (0x2803).

Value. U ovom polju su navedene tri stavke: *Characteristic Properties* je postavljeno tako da je za ovu karakteristiku dozvoljena samo notifikacija. *Characteristic Value Handle* sadrži *handle* atributa za vrednost karakteristike ($0x0027$). Poslednja stavka je UUID za *Hart Rate Measurement* ($0x2A37$).

Handle $0x0027$. Ovaj atribut sadrži vrednost karakteristike. U ovom slučaju to je izmereni puls. Polja ovog atributa su:

UUID. Isti UUID kao u polju za vrednost atributa za deklaraciju karakteristike.

Permissions. Ovo polje je postavljeno tako da nije dozvoljeno ni čitanje ni upis, već samo notifikacije. To znači da klijent može da dobije vrednost ovog atributa samo putem notifikacija koje šalje server.

Value. U ovom polju se nalazi izmerena vrednost pulsa (npr. u jedinicama otkucaja u minuti).

Handle $0x0028$. Ovaj atribut sadrži CCCD (deskriptor). Tumačenje sadržanih polja je sledeće:

UUID. CCCD ima svoj, standardni, 16-bitni UUID ($0x2902$) .

Permissions. Čitanje i upis CCCD-a uvek mora biti dozvoljeno. Nivo sigurnosti za pristup CCCD-u je definisan profilom ili od strane aplikacije.

Value. CCCD je polje bita. Sadržaj ovog polja, u konkretnom slučaju je $0x0001$, što ukazuje na to da su za ovu karakteristiku dozvoljene notifikacije.

Handle $0x002A$. Ovaj atribut je deklaracija karakteristike, koja označava početak još jedne karakteristike. Sada je to *Body Sensor Location* karakteristika (gde je na telu korisnika postavljen senzor pulsa). Polja ovog atributa su:

UUID. Sadrži standardni 16-bitni UUID za deklaraciju karakteristike, $UUID_{characteristic}$ (0x2803).

Value. Sadrži tri stavke, za *properties*, *hande* i *value*, na osnovu čijeg sadržaja saznajemo da ovaj je ovoj karakteristici dozvoljen samo pristup radi čitanja, da je *handle* atributa za vrednost karakteristike $0x002C$ i da je UUID za *Body Sensor Location* karakteristiku $0x2A38$.

Handle $0x002C$. Ovaj atribut sadrži vrednost karakteristike *Body Sensor Location*. Njegova polja su:

UUID. Isti UUID kao u polju za vrednost atributa za deklaraciju karakteristike ($0x2A38$).

Permissions. Permisije ovog atributa su postavljene na *read-only*: klijent može samo da proveri gde se senzor nalazi, ali ne može da izmeni vrednost ovog podatka (to može samo server).

Value. U ovom polju se nalazi podatak o stvarnoj lokaciji senzora (npr. prst, nadlaktica, podlaktica, ...).

8.6.12 Keširanje atributa

GATT omogućava klijentu da korišćenjem *handle*-ova pristupa svim atributima servera. Međutim, otkrivanje liste dostupnih *handle*-ova i pribavljanje sadržaja odgovarajućih atributa može biti vremenski (i energetski) zahtevna operacija. U nastavku će biti objašnjeno šta sve klijent može da uradi kako bi izbegao potrebu za iščitavanjem atributa servera nakon svake rekonekcije.

U većini slučajeva, struktura atributa na strani servera je statička, tj. ne menja se često tokom životnog veka servera (ili se uopšte ne menja). Uprkos tome, BLE specifikacija daje potpunu slobodu serveru da u bilo kom trenutku rekonfiguriše strukturu atributa, da doda nove, izbaci ili zameni stare novim. Ovakve radikalne promene na strani servera su moguće npr. nakon ažuriranja firmvera BLE uređaja. Iz tog razloga, neophodna su izvesna pravila i ograničenja kako bi klijent mogao da se osloni na validnost prethodno pribavljenih *handle*-ova bez rizika da će se oni u međuvremenu promeniti na serveru (što bi onemogućilo dalju komunikaciju).

BLE specifikacija, kao opšte pravilo, preporučuje klijentu da keširа (tj. sačuva za naredne transakcije, pa čak i konekcije) *handle*-ove serverskih atributa za koje je zainteresovan. Ovo se naravno ne odnosi na vrednosti atributa, koje su po prirodi stvari, podložne čestim promenama, već pre svega na njihove *handle*-ove i meta-podatke. S druge strane, BLE specifikacija uvodi karakteristiku *Service Change*, koja omogućava serveru da obavesti klijenta o bilo kakvoj promeni koja je u vezi sa atributima. Ovo je

opciona karakteristika, tako da samo njeno prisustvo na serveru ukazuje na mogućnost promena u strukturi atributa.

Informaciju o tome da li bi trebalo ili ne da kešira atribute klijent može da dobije samo od servera i to na osnovu ispitivanje sledeća dva uslova:

Na serveru nije prisutna karakteristika Service Change. Ako je to tačno, klijent slobodno može trajno da kešira sve *handle*-ove koje je pronašao kod servera, jer server garantuje da se njegovi atributi neće menjati tokom celog njegovog životnog veka.

Na serveru jeste prisutna karakteristika Service Change. U ovom slučaju, potrebno je da se klijent pretplati na odgovarajuću seversku notifikaciju (upisom u deskriptor CCCD u okviru karakteristike *Service Change*). Ovo će omogućiti serveru da upozori klijenta na bilo kakvu strukturalnu promenu svojih servisa. Ukoliko su klijent i server uvezani, klijent će moći slobodno da čuva keširane *handle*-ove atributa i između konekcija i da pri tom očekuje da će oni neće promeniti. Ako klijent i server nisu uvezani, od klijenta se očekuje da izvrši otkrivanje atributa nakon svake rekonekcije sa serverom.

8.6.13 Korišćenje GATT atributa u paketima za oglašavanje

GATT je primarno namenjen razmeni podataka preko uspostavljene konekcije između centrale i periferije. Ipak, takođe je moguće da se deo informacije sadržane atributima servera distribuira paketima za oglašavanje i da time postane dostupan osmatračima i centralama koje su u režimu skeniranja.

U poglavlju o GAP profilu, napomenuto je da se za uključivanje u paket za oglašavanje podataka, tj. atributa, iz GATT servisa koristi AT tip Service Data. Kao što je to prikazano u Tabela 11, radi emitovanja podataka iz servisa, GATT server mora da uključi dva polja u sekciju *Service Data* u okviru paketa za oglašavanje. Prvo polje je UUID koji identifikuje podatak koji se šalje, dok drugo (druga vrsta u tabeli) sadrži kompletnoj ili delimičnoj vrednosti konkretne karakteristike ili deskriptora unutar odgovarajućeg servisa.

Tabela 11 AD tip *Service Data*

Field	Length in bytes	Description
UUID	2, 4, or 16	The actual UUID identifying the data
Service Data	Variable	The data associated with the service identified by the UUID

8.6.14 GATT procedure

GATT procedure omogućavaju razmenu podataka između klijent i servera. Sve one su zasnovane na operacijama ATT protokola.

***Otkrivanje servisa i karakteristika* (*Service and Characteristic Discovery*)**

Klijent, nakon što se prvi put poveže (konektuje) sa serverom, ne zna ništa o atributima koji su prisutni na GATT serveru. Zbog toga je neophodno da nakon uspostavljanje konekcije klijent obavi niz razmena paketa sa serverom kako bi došao do informacija o količini, lokaciji i prirodi svih serverskih atributa koji bi mogli da mu budu od interesa. Generalno, ova aktivnost se naziva otkrivanje (*discovery*) servisa i karakteristika.

Za otkrivanje primarnih servisa, GATT nudi klijentu sledeće dve opcije:

Otkrivanje svih primarnih servisa. Korišćenjem ove procedure, klijent može da prikupi kompletnu listu svih primarnih servisa udaljenog servera. Ova opcija se koristi u slučajevima kada je klijent

osposobljen za rad sa više različitih servisa i zbog toga želi da dobije potpunu informaciju o podržanim servisima na strani servera.

Otkrivanje primarnog servisa po UUID-u. Klijent koristi ovu proceduru onda kada unapred zna koji servis traži (obično, zato što podržava samo taj jedan servis).

Obe prethodne procedure vraćaju opseg *handle*-ova atributa obuhvaćenih svakim pronađenim servisom. U slučaju procedure za otkrivanje svih primarnih servisa, procedura, takođe, vraća i UUID-ove servisa.

Nakon što je od servera dobio informaciju o dostupnim servisima, klijent može da nastavi sa istraživanjem pojedinačnih servisa. Prvi korak u tom procesu je otkrivanje servisa koji su obuhvaćeni primarnim servisima, tzv. *relationship discovery*. Zatim, klijent prelazi na otkrivanje karakteristika u pojedinačnim servisima. Za tu namenu na raspolaganju su sledeće procedure:

Find included services (*pronalaženje obuhvaćenih servisa*). Ova procedura omogućava klijentu da postavi upit serveru o eventualnim servisima koji su obuhvaćeni primarnim servisom. Odgovor na upit je opseg *handle*-ova unutar postojećeg servisa koji omeđuje atribute obuhvaćenog servisa. Kao i prethodno, procedura dodatno vraća UUID obuhvaćenog servisa.

Discover all characteristics of a service (*otkrivanje svih karakteristika datog servisa*). Nakon što je pribavio opseg *handle*-ova servisa za koji je zainteresovan, klijent može da pribavi i pun spisak karakteristika ovog servisa. Ulazni argument procedure za otkrivanje karakteristika je opseg *handle*-ova. Kao odgovor, server, za svaku karakteristiku, vraća *handle* i vrednost atributa za deklaraciju karakteristike.

Discovery characteristics by UUID (*otkrivanje karakteristike po UUID*). Ova procedura je identična prethodnoj, s tim što klijent odbacuje sve karakteristike čiji se UUID razlikuje od traženog UUID-a.

Nakon što je došao u posed informacija o opsezima *handle*-ove svih karakteristika za koje je zainteresovan, klijent može da istrži deskriptore pojedinačnih karakteristika:

Discover all characteristic descriptors (*otkrivanje svih deskriptora*). Sa poznatim opsezima *handle*-ova i UUID-ovima nekih (ili svih) karakteristika sadržanih u servisu, klijent je u mogućnosti da, primenom ove procedure, istraži sve deskriptore konkretne karakteristike. Server odgovara listom UUID-ova i *handle*-ovima deskriptora date karakteristike.

Sve procedure iz ove sekcije, se mogu izvršavati preko otvorene konekcije koja ne mora biti sigurna, s obzirom da je otkrivanje servisa/karakteristika dozvoljeno svim klijentima bez ikakvih ograničenja.

***Čitanje karakteristika i deskriptora* (*Reading Characteristics and Descriptors*)**

Primenom prethodno navedenih GATT procedura, klijent je u mogućnosti da dođe u posed opsega *handle*-ova i UUID-ova karakteristika i deskriptora karakteristika, ali ne i njihovih vrednosti. Za pribavljanje vrednosti karakteristika i procedura koristi se poseban set procedura:

Reading characteristic value or descriptor (*Čitanje vrednosti karakteristika ili deskriptora*). Ova procedura omogućava klijentu da pročita sadržaj polja za vrednost karakteristike ili deskriptora na osnovu *handle*-a karakteristike/deskriptora. Ograničenje je da se ovom procedurom može pročitati samo mali broj bajtova vrednosti (zbog ograničenja u ATT dužini paketa).

Reading long characteristic value or descriptor (*„Dugačko“ čitanje vrednosti karakteristike ili deskriptora*). Ukoliko je vrednost karakteristike ili deskriptora previše velika da bi mogla da se

pročita prethodnom procedurom, potrebno je koristiti proceduru za „dugačko“ kod koje se u zahtevu za čitanje, pored *handle*, navodi i ofset (pomak), tako da se vrednost karakteristike/deskriptora može očitati razmenom više uzastopnih paketa.

Sledeće dve procedure se odnose samo na čitanje vrednosti karakteristika:

Read characteristic value using characteristic UUID (*Čitanje vrednosti karakteristike na osnovu UUID-a karakteristike*). Ukoliko klijentu nisu poznati *handle*-ovi karakteristika za koje je zainteresovan, on može da pošalje serveru zahtev za čitanje vrednosti svih karakteristika određenog tipa (tj. UUID-a). Argumenti ove procedure su opseg *handle*-ova i UUID. Procedura vraća niz vrednosti karakteristika obuhvaćenih zadatim opsegom.

Read multiple characteristic values (*Čitanje vrednosti više karakteristika*). Ova procedura omogućava klijentu, koji već poseduje *handle*-ove za više karakteristika, da pošalje objedinjen zahtev za čitanje i da nakon toga dobije vrednosti svih traženih karakteristika.

Treba napomenuti da je čitanje vrednosti karakteristika i deskriptora podložno sigurnosnim ograničenjima i da može biti odbijeno od strane servera ukoliko nivo sigurnosti, koji je uspostavljen između klijenta i servera, nije podudaran sa nivoom koji je pridružen karakteristici/deskriptoru.

***Upis karakteristika i deskriptora* (*Writing Characteristics and Descriptors*)**

Klijentu su na raspolaganju sledeće procedure za upis vrednosti karakteristika i deskriptora:

Write characteristic value or descriptor. Ova procedura se koristi za upis vrednosti karakteristike ili deskriptora. Klijent dostavlja serveru *handle* i novi sadržaj polja za vrednosti karakteristike/deskriptora. Na strani servera, dostavljeni sadržaj se upisuj na odgovarajuće mesto i konačno server potvrđuje upis slanjem odgovarajućeg paketa nazad klijentu. Pri tom, količina podataka koja se šalje na upis mora biti dovoljno mala kako se ne bi prekoračilo ograničenje u veličini BLE paketa.

Write long characteristic value or descriptor. Kao prethodna procedura, s tim što je sada omogućen upis vrednosti dužine od dužine BLE paketa. Slično proceduri za „dugačko“ čitanje, vrednost se dele na manje delove koji se jedan po jedan šalju serveru zajedno sa *handle* karakteristike/deskriptora i ofseta, odnosno pomaka dela u odnosu na početak polja za vrednost. Poslati delovi se gomilaju na strani servera, a sam upis se obavlja tek onda kad pristignu svi delovi vrednosti.

Sledeće dve procedure se odnose samo na upis vrednosti karakteristika:

Write without response (*Upis bez odziva*). Ova procedura je suprotna „inverznoj notifikaciji“. Procedura koristi paket tipa *Write Command* (protokol ATT) za dostavljanje serveru *handle*-a i novu vrednosti karakteristike, koji je karakterističan po tome što ne izazva odziv servera, bez obzira na ishod zahtevane operacije. Server ima slobodu da odbaci zahtev sadržan u paketu *Write Command* (npr. iz nekog razloga nije u mogućnosti da ga procesira, pravo pristupa atributu ne dozvoljava upis, pogrešan *handle* i sl.) i da o tome ne obavesti klijenta. To znači da klijent nikada neće znati da li je upis zaista izvršen. Jednini način kako to može da proveri je da nakon upisa, pročita vrednost iste karakteristike.

Reliable writes (*Pouzdani upis*). Slično proceduri za čitanje vrednosti više karakteristika, klijent ima mogućnost da dostavi serveru vrednosti više karakteristika. Server prihvata poslate podatke, ali ih ne upisu u polja za vrednost karakteristika sve dok od klijenta ne dobije zahtev za upis. Nakon dobijenog zahteva za upis, prethodno primljeni podaci se upisuju na svoja prava odredišta.

Kako i kod čitanja, upis vrednosti karakteristika i deskriptora je podložan sigurnosnim ograničenjima.

***Ažuriranja inicirana serverom* (*Server-Initiated Updates*)**

Ažuriranja inicirana od strane servera su jedini asinhroni paketi koji se mogu prenositi od servera ka klijentu. Asinhrono ovde znači da se paket ne šalje kao odziv na klijentov zahtev, već server, samoinicijativno, šalje paket klijentu. Ovim paketima server, po pravilu, šalje pravovremena upozorenja (obaveštenja) klijentu o promeni vrednosti određenih karakteristika, čime se eliminiše potreba da klijent neprekidno i periodično čita vrednost konkretne karakteristike. Na taj način se štedi na energiji i propusnom opsegu.

Postoje dva tipa serverskog ažuriranja: notifikacije i indikacije.

Characteristic Value Notification (*Notifikacija vrednosti karakteristike*). Notifikacije su paketi koji sadrže *handle* atributa za vrednost karakteristike i njegovu trenutnu vrednost. Klijent prima paket i shodno njegovom sadržaju može da preduzima određenu aktivnost, ali ne odgovara serveru slanjem bilo kakvog paketa potvrde prijema. Osim *write without response*, notifikacije su jedini mehanizam koji se ne uklapa u standardni zahtev/odziv način rada. Za prenos notifikacija koristi se *handle value notification* paketi iz protokola ATT.

Characteristic Value Indication. Indikacije, poput notifikacija, nose *handle* i vrednost karakteristike koja se promenila, ali, za razliko od notifikacija, zahtevaju eksplicitnu potvrdu prijema od strane klijenta. Drugim rečima, klijent je u obavezi da potvrdi serveru prijem indikacije. Za prenos indikacija i potvrda koriste se ATT paketi *handle value indication* i *handle value confirmation*.

Klijent mora da omogući slanje oba tipa serverskog ažuriranja, notifikacije i indikacije, upisom u odgovarajući CCCD.

8.6.15 GATT Servis

Shodno BLE specifikaciji, svaki GATT server mora da sadrži tzv. GATT servis. Ovaj servis sadrži samo jednu karakteristiku, *Service Change*, čija se vrednost može preneti klijentu samo putem indikacije (*characteristic value indication*). Kao sto je prikazano u tabeli, polje za vrednost karakteristike *Service Change* sadrži opseg *handle*-ova koji omeđuje oblast atributa na serveru koja je pogođena strukturalnim promenama i koju bi klijent trebalo ponovo da otkrije (*discover*). Klijent bi morao da obavi proceduru otkrivanja servisa i karakteristika u toj oblasti zato što atributi iz te oblasti možda više nisu validni. Od klijenta se takođe očekuje da dozvoli indikaciju za ovu karakteristiku u odgovarajućem CCCD-u.

Table 4-8. Service changed characteristic value

Handle	Type	Permissions	Value	Value length
0xNNNN	$UUID_{service\ changed}$	None	Affected handle range	4

Dakle, ukoliko na serveru dođe do bilo kakve promene u strukturi atributa, opseg *handle*-ova koji ukazuje na to gde se u nizu atributa desila promena upisuje se polje za vrednost karakteristike *Service Change*, a onda se, putem indikacije, vrednost ove karakteristike šalje klijentu. Pošto je ovde reč o indikaciji, server očekuje potvrdu od klijenta. Ukoliko su server i klijent uvezani i promena se desila u vremenu kada konekcija nije bila uspostavljena, server će poslati indikaciju odmah nakon uspostavljanje konekcije.

8.7 Sigurnost (*Security*)

Sigurnost (*security*) komunikacije je bitan zahtev kod svih komunikacionih sistema, pa tako i kod *Bluetooth*-a. Pre nego što se posvetimo podršci za sigurnost kod Bluetooth-a, u nastavku će biti objašnjeni neki osnovni oblici i koncepti sigurnosti.

Authentication (*Provera autentičnosti, ili autentifikacija*). Dokaz da je druga strana u komunikaciji upravo ona za koju se izdaje. Na primer, kada se konektujemo na neki BLE uređaja, želimo da budemo sigurni da se zaista konektujemo na „pravi“ uređaj, a neki zlonamerni, koji se samo „pretvara“ da je onaj koji očekujemo.

Integrity (*Integritet podataka*). Obezbeđuje da podaci koje smo primili nisu oštećeni ili namerno promenjeni od uređaja koji za tako nešto nema ovlašćenja.

Confidentiality (*Poverljivost*). Želimo da budemo sigurni da podaci koji nam se šalju nisu čitljivi neautorizovanim (neovlašćenim) korisnicima ili uređajima.

Privacy (*Privatnost*). Da li je komunikacija privatna, u smislu da li informacija o tome kada i gde smo komunicirali dostupna neautorizovanom uređaju ili korisniku? Privatnost je narušena ako je moguće pratiti uređaj, npr. po njegovoj BLE adresi.

Značaj prethodnih aspekata sigurnosti zavisi od zahteva konkretne primene. U nekim primenama je od primarne važnosti poverljivost, kod nekih drugih poverljivost i privatnost, a kod nekih trećih neka treća kombinacija autentičnosti, integriteta, poverljivosti i privatnosti.

Sigurnosni mehanizmi koji se implementiraju u komunikacionim sistemima treba da pruže zaštitu od različitih vrsta napada od strane zlonamernih (malicioznih) uređaja ili osoba. Tipične vrste napada su:

Passive Eavesdropping (*Pasivno prisluškivanje*). Zlonamerni uređaj osluškuje komunikaciju između dva uređaja i u mogućnosti je da razume o čemu oni „razgovaraju“ (npr. tako što je došao u posed kriptografskog ključa koji se u toj komunikaciji koristi za šifrovanje/dešifrovanje podataka).

Active Eavesdropping (*Aktivno prisluškivanje*). Takođe poznato i kao „*Man-In-The-Middle* (MITM)“ napad (čovek u sredini). Kod ove vrste napada, zlonamerni uređaj oponaša oba uređaja (npr. BLE centralu i periferiju). Prema centrali se ponaša kao periferija, a prema periferiji kao centrala. Presreće komunikaciju između dva uređaja, ali tako da dve strane nisu svesne da imaju neželjenog posrednika u komunikaciji. „Čovek u sredini“ može ne samo da pročita šta dve strane “pišu“, već i da izmeni podatke u paketima.

Privacy and Identity Tracking (*Privatnost i praćenje identiteta*). Kod ove vrste napada, uređaj ili korisnik se prati po svojoj BLE adresi. Pošto adresa otkriva identitet korisnika, na ovaj način je moguće pratiti lokaciju korisnika, kao i to s kim i kada je komunicirao.

Vratimo se sada na BLE. Sigurnosni mehanizmi su implementirani u različitim slojevima BLE protokol steka. Na primer, šifrovanje/dešifrovanje sadržaja paketa (kao zaštita poverljivosti) je funkcija sloja veza. Takođe funkcija sloja veze je i digitalno potpisivanje paketa, što predstavlja način za očuvanje integriteta podataka. Zaštita privatnosti se postiže korišćenjem razrešivih privatnih adresa.

8.7.1 Security Manager (SM)

Većina sigurnosnih mehanizama zahteva upotrebu kriptografskih ključeva. Kriptografski ključ je niz bita određene dužine (npr. 128 bita), koji se koristi za šifrovanje/dešifrovanje podataka, digitalno potpisivanje, razrešavanje privatnih adresa itd. Na primer, da bi prijemnik mogao pravilno da dešifruje sadržaj primljenog paketa, on mora da poseduje isti ključ kojim je paket šifrovan. Glavni izazov u

implementaciji sigurnosnih mehanizama u sigurnoj razmeni kriptografskog ključa. Upravo je generisanje i distribucija ključeva zadatak sloja *Security Manager* (*SM*). SM je ujedno protokol i set bezbednosnih algoritama za generisanje i razmenu kriptografskih ključeva između BLE uređaja. SM definiše sledećih pet sigurnosnih mehanizama:

Pairing (*Uparivanje*): proces kreiranja tajnih ključeva, deljivih između dva uređaja.

Bonding (*Uvezivanje*): proces kreiranja deljivih tajnih ključeva i njihovo čuvanje radi korišćenja u budućim konekcijama između dva uređaja.

Authentication (*Provera autentičnosti*): proces verifikacije da dva uređaja dele iste tajne ključeve.

Encryption (*Šifrovanje*): proces šifrovanja podataka koji se razmenjuju između dva uređaja. Kod BLE se koristi 128-bitni AES *Encryption* standard, koji pripada grupi kriptografskih algoritama sa simetričnim ključem (što znači da se isti ključ koristi za šifrovanje i dešifrovanje podataka).

Message Integrity (*Integritet poruka*): proces potpisivanja podataka i verifikacije potpisa na drugoj strani.

Korišćenjem sigurnosnih mehanizama, SM rešava probleme sigurnosti:

- Poverljivost – šifrovanjem
- Provera autentičnosti – uparivanjem i uvezivanjem
- Privatnost – korišćenjem razrešivih privatnih adresa
- Integritet – digitalnim potpisivanjem

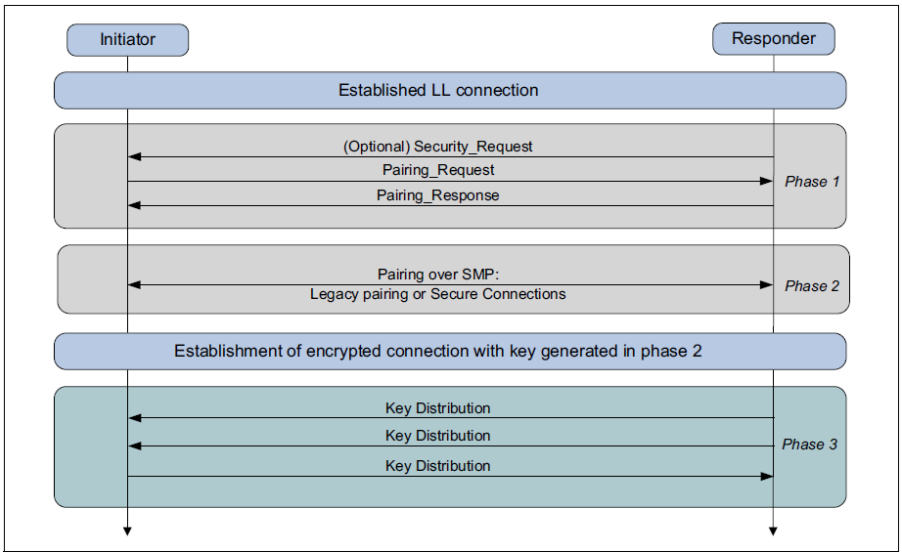
Svaki od ovih mehanizama se može koristiti nezavisno, ili se više mehanizama mogu kombinovati. Izbor sigurnosnih mehanizama koji će se koristiti u konkretnoj primeni, prepušta se aplikacionom programu.

Uloge uređaja

SM definiše dve uloge uređaja: *initiator* (*inicijator*) i *responder* (*risponder*). Inicijator je uređaj koji inicira izvršenje sigurnosnih procedura, a risponder je uređaj koji zajedno sa inicijatorom sprovodi sigurnosne procedure. Sigurnosne procedure se sprovode preko prethodno uspostavljena konekcije. Pri tom, inicijator je uvek master, a risponder slejv uređaj. Slanjem masteru odgovarajućeg zahteva, risponderu je dozvoljeno da traži startovanje sigurnosne procedure. Međutim, na masteru je ipak konačna odluka o tome da li će da prihvati zahtev i od onda, u slučaju da prihvati, oficijelno inicira proceduru.

Uparivanje i uvezivanje

Dijagram na Sl. 26 prikazuje faze u procesu uspostavljanje sigurne veze između inicijatora i rispondera. Uparivanje obuhvata faze 1 i 2, dok se uvezivanje proteže i na fazu 3. Uparivanje kreira sigurnu vezu koja traje samo tokom tekuće konekcije. Svaka buduća konekcija između istih uređaja zahteva novo uparivanje, što podrazumeva i generisanje novih tajnih ključeva. Za razliku od toga, uvezivanje kreira permanentnu sponu između dva uređaja, koja je omogućena generisanjem deljivih tajnih ključeva koji ostaju zapamćeni u permanentnoj memoriji oba uređaja i koji će se koristiti u svim budućim konekcijama. Ova spona traje sve dok jedan od dva uređaja ne odluči da obriše ključeve.



Sl. 26 Procedura uspostavljanje sigurne veze između dva BLE uređaja

Glavni zadatak uparivanja/uvezivanja je generisanje seta tajnih sigurnosnih ključeva. „Tajno“ ovde znači da su generisani ključevi poznati samo uređajima koji učestvuju u komunikaciji. Pretpostavimo da ključeve generiše master i da ih onda putom BLE konekcije šalje slejvu. Međutim, ovakvo rešenje nije bezbedno, zato što BLE konekcija još uvek nije osigurana, što malicioznom uređaju ostavlja mogućnost da, pasivno prisluškujući komunikaciju između mastera i slejva, dođe u posed ključeva. Rešenje ovog problema je u tome da se za interakciju mastera i slejva prilikom generisanja deljivih ključeva koristi neki drugi komunikacioni medijum, koji bi pružio veće nivo zaštite od neosigurane BLE konekcije. Princip je taj da se u fazi 1, master i slejv dogovore o tome koji alternativni medijum, odnosno metod uparivanja će koristiti, a da nakon toga, u fazi 2, korišćenjem tog sigurnog medijum prenese tzv. privremeni tajni ključ (TK). TK se ne koristi za osiguranje konekcije, već za generisanje tzv. kratkoročnog ključa (*STK – Short Term Key*) kojim će konekcija biti osigurana.

U fazi 1, master inicira proces uparivanja slanjem *Pairing Request* paketa slejvu, koji odgovara *Pairing Response* paketom. Svrha *pairing* paketa je razmena informacija o mogućnostima koje dva uređaja poseduju, kao i njihovim sigurnosnim zahtevima. Konkretno, u ovim paketima je sadržana informacije o tome da li uređaj poseduje displej, tastaturu, YES/NO tastere; da li podržava *Out-Of-Band* (*OOB*) metod i koji; da li uređaj zahteva proveru autentičnosti, uvezivanja i sl. Na osnovu razmenjenih informacija, utvrđuje se metod uparivanja koji će biti korišćen u fazi 2. Podržani metodi uparivanja su:

Just Works („Samo da radi“): Kod ovog metoda, TK je, prosto, postavljen na 0. $TK = 0$ se, zajedno sa ostalim podacima koji su razmenjeni između uređaja u *Pairing Request/Response* paketima, koristi za generisanje ključa STK. Iz očiglednih razloga, ovaj metod je najmanje siguran od svih metoda uparivanja jer je podložan MITM napadu.

Out of Band (OOB): Kod ovog metoda, TK se razmenjuje između dva uređaja, ali ne korišćenjem BLE već neke druge komunikacione tehnologije, kao npr. NFC (*Near Field Communication*) – master generiše TK, a onda se TK prenese slejvu putem NFC komunikacije. Ovaj metod je značajno sigurniji budući da NFC po svojoj prirodi (komunikacija veoma kratkog dometa – reda cm), onemogućava prisluškivanje.

Passkey: U ovom metodu, TK je 6-cifreni broj koji se prenosi između uređaja od strane krajnjeg korisnika. Na primer, master generiše TK i prikazuje ga na svom displeju; korisnik očitava TK i unosi ga u drugi uređaj preko tastature. Naravno, ovaj metod podrazumeva da uređaji poseduju odgovarajuće ulazno/izlazne mogućnosti (displej, tastatura..).

Nakon što su uparivanjem razmenili TK, uređaji nezavisno jedna od drugog, a korišćenjem TK i dodatnih razmenjenih podataka, generišu isti STK kojim osiguravaju konekciju.

U fazi 3 obavlja se proces uvezivanja uređaja (*bonding*). Ova faza je opciona i koristi se kako bi se izbeglo ponovno uparivanje prilikom uspostavljanje svake sledeće konekcije između dva uređaja. Rezultat uvezivanja je taj da svaki uređaj pamti set generisanih ključeva (tzv. dugoročni ključevi LTK – *Long Term Keys*) u svojoj permanentnoj kako bi mogao da iz iskoristi u budućim konekcijama. Razmena ključeva tokom uvezivanja se ostvaruje preko konekcije koja osigurana u fazi 2.

Sigurnosni ključevi

BLE koristi brojne ključeve tokom sprovođenja različitih sigurnosnih procedura. Najznačajniji su sledeći (neki od njih su već prethodno pomenuti):

TK (*Temporary Key – Privremeni ključ*): Privremeni ključ se generiše pri svakom uparivanju, a način generisanja zavisi od izabranog metoda uparivanja.

STK (*Short Term Key – Kratkoročni ključ*): Generiše se na osnovu TK koji je razmenjen između uređaja tokom uparivanja. STK se generiše pri svakom uparivanju i koristi za šifrovanje podataka samo tokom tekuće konekcije.

LTK (*Long Term Key – Dugoročni ključ*): Ovaj ključ se generiše u fazi 3 (uvezivanje) procedure uparivanja i uvezivanja. Oba uvezana uređaja čuvaju LTK i koriste ga u svojim narednim konekcijama.

IRK (*Identity Resolving Key – Ključ za razrešavanje identiteta*). Ovaj ključ se koristi za razrešavanje slučajnih privatnih adresa. IRK je jedinstven za svaki uređaj, tako da masterov IRK mora biti zapamćen na strani slejva, a slejvov IRK na strani mastera.

8.7.2 GAP sigurnost

Profil GAP, čiji je prevashodni zadatak upravljanje oglašavanjem i konekcijom, takođe definiše i set režima rada i procedura koje obezbeđuju standardni način za korišćenje sigurnosnih mehanizama obuhvaćenih SM-om.

Sigurnosni režimi

Svaka konekcija uspostavljena između BLE centrale i periferije radi u jednom konkretnom sigurnosnom režimu. Pojam režim se ovde odnosi na tekući nivo sigurnosti konekcije. Nivo sigurnosti može da se menja tokom trajanja konekcije u zavisnosti od sigurnosnih potreba jednog ili drugog uređaja. GAP upravo definiše režime i nivoe sigurnosti konekcije, kao i procedure za promenu nivoa sigurnosti.

Konkretno, GAP definiše dva sigurnosna režima, svaki sa po nekoliko nivoa sigurnosti:

Security mode 1. U ovom režimu, sigurnost se postiže šifrovanjem podataka. Režim sadrži tri nivoa sigurnosti:

- *Level* 1 (*No security*): Konekcija nije sigurna. Ne koristi se šifrovanje, niti je izvršena provera autentičnosti.
- *Level* 2: (*Unauthenticated encryption*): Šifrovanje, ali bez provere autentičnosti.
- *Level* 3: (*Authenticated encryption*): Šifrovanje sa proverom autentičnosti.

Security mode 2. U ovom režimu, sigurnost se postiže potpisivanjem podataka. Režim sadrži dva nivoa sigurnosti:

- *Level* 1: (*Unauthenticated data signing*): Potpisivanje podataka, ali bez provere autentičnosti.
- *Level* 2: (*Authenticated data signing*): Potpisivanje podataka sa proverom autentičnosti.

Konekcija se smatra autentičnom ili neautentičnom (sa ili bez proverene autentičnosti) na osnovu korišćenog metoda uparivanja. Na primer, korišćenje metoda uparivanja „*Just Works*“ nije dovoljno za potvrdu autentičnost dva uređaja. Za razliku od toga, uparivanje metodom OOB koji koristi NFC garantuje proveru autentičnosti.

Inicijalni sigurnosni režim novouspostavljene konekcije je *Security Mode* 1, *Level* 1, ali se kasnije, po potrebi, može promeniti u bilo koji drugi. Okidač za promenu nivoa sigurnosti može biti pristup karakteristici koja zahteva viši nivo sigurnosti od onog koji trenutno važi. Na primer, neka karakteristika zahteva nivo sigurnosti *Level* 1 za čitanje i *Level* 3 za upis. Konekcija može ostati na nivou *Level* 1 (bez sigurnosti) sve dok se obraća ovoj karakteristici samo radi čitanja. Onda kada klijent po prvi put ispostavi zahtev za upis karakteristike, biće pokrenuta sigurnosna procedura za promenu nivoa sigurnosti na *Level* 3, što će zahtevati i uparivanje.

8.7.3 GATT sigurnost

Profil GATT je fokusiran na procedure i formate za prenos podataka. Ovaj profil definiše hijerarhijsku organizaciju podataka upotrebom servise i karakteristika. Aspekt GATT koji se dotiče sigurnosti u vezi je sa permisijama koje definišu prava pristupa pojedinačnim atributima. Naime, okidač za promenu nivoa sigurnosti može biti pristup atributu koja zahteva viši nivo sigurnosti od onog koji trenutno važi. Na primer, neki atribut zahteva nivo sigurnosti *Level* 1 za čitanje i *Level* 3 za upis. Konekcija može ostati na nivou *Level* 1 (bez sigurnosti) sve dok se klijent obraća ovoj karakteristici samo radi čitanja. Onda kada klijent po prvi put ispostavi zahtev za upis karakteristike, biće pokrenuta sigurnosna procedura za promenu nivoa sigurnosti na *Level* 3, što će zahtevati i uparivanje.

Atributi koji su deklaracije, ne zahtevaju nikakav poseban nivo sigurnosti. (Ovo važi za deklaracije servisa i karakteristika, ali ne i za deklaracije deskriptora, kod kojih su relevantni podaci sadržani u samom deskriptoru umesto u posebnom atributu.) Otvoren pristup deklaracijama omogućava da klijent, koji se još uvek nije upario ili uvezao sa serverom, može da obavi osnovno otkrivanje servisa i karakteristika bez potrebe da osigurava konekciju. Struktura atributa i hijerarhija podataka na serveru se ne smatra senzitivnom informacijom i zbog toga je dostupna svim klijentima.

Međutim, prilikom pristupa vrednosti karakteristike ili deklaraciji deskriptora, klijent može, kao odgovor na zahtev, da primi *error response ATT* paket sa indikacijom da zahtev ne može biti izvršen zbog trenutno nedovoljno visokog nivoa sigurnosti. Sledeća dva kôda greške su najčešće sadržana u *error response* paketima:

Insufficient Authentication. Označava da konekcija nije šifrovana i da server na poseduje dugoročni ključ (LTK) za šifrovanje ili server poseduje LTK i konekcija je šifrovana, ali je LTK generisan bez provere autentičnosti (npr. korišćeno je „*Just Works*“ uparivanje), a da pri tom permisija atributa kome se pristupa zahteva šifrovanje sa proverom autentičnosti.

Insufficient Encryption. Ukazuje na to da konekcija nije šifrovana, mada je dostupan ogovarajući LTK.

Nakon pojave greške zbog nedovoljnog nivoa sigurnosti, a u zavisnosti od toga koja od dve strane u konekciji igra ulogu centrale, a koja periferije, na GATT klijentu ili GATT serveru je da inicira proceduru uparivanja, uvezivanja, ili šifrovanja radi podizanja nivoa sigurnosti konekcije. Nakon što uspostavljen nivo sigurnosti koji je propisan permisijama atribute kome se pristupa, klijent može ponovo da pošalje serveru zahtev za čitanje/upis atributa.

8.8 Vodič za učenje

Uvod

Dve varijante Bluetooth standarda i tri tipa Bluetooth uređaja

Tri konfiguracije Bluetooth uređaja (Sl. 3)

BLE topologije

Emitovanje/osmatranje i konekcija (Sl. 4 i 5). Uloge uređaja.

BLE: Prednosti, nedostaci i primene

Fizički sloj

Frekventni kanali. Koliko kanala, širina kanala i razmak između kanala. Kanali za oglašavanje i kanali za prenos podataka.

Frekventno skakanje kod BLE. Kada se dešava i kako se bira sledeća frekvencija.

Adaptivno frekventno skakanje. Problem interferencije sa WiFi i kako se rešava (Sl. 11)

BLE modulacije 1M PHY i 2M PHY. Bitske brzine, prednosti/nedostaci

BLE modulacija *Coded PHY*. Princip. Šta se dobija, a šta gubi u odnosu na 1M PHY i 2M PHY

Sloj veze

Uloge uređaja (oglašivač/skener, master/slejv)

Stanja uređaja (Sl. 13)

Format paketa. Samo prvi red na Sl. 14. Uloga adrese pristupa.

BLE adrese uređaja. Dužina. Klasifikacija adresa.

Skeniranje i oglašavanja. Princip (Sl. 16). Pasivno i aktivno skeniranje (Sl. 17). Tipovi paketa za oglašavanje (Ne treba pamtiti detalje formata, ali ako vam se da na uvid Tabela 2 trebalo bi da možete da objasniti svrhu svakog od 4 tipa paketa za oglašavanje).

Konekcija. Procedura uspostavljanja konekcije (Sl. 20, ne treba je pamtiti, ali bi trebalo da znate da je objasnite kad je vidite)

Connection event. Šta je to? Pravila za komunikaciju tokom *connection event*-a. Parametri konekcije (*connection interval, slave latency, supervision timeout i data Length extension* – na šta se odnosi svaki parametar).

ATT

Pojam i format atributa. Šta je to UUID? Dužine punog i skraćenih UUID, kao i namena jednih i drugih.

Uloga ATT klijenta i servera.

Sve ostalo u vezi ATT u ovom materijalu je informativnog karaktera.

GAP

(U ovom delu ima dosta preklapanja sa slojem veze (LL), zato što GAT koristi LL i proširuje ga u nekim aspektima.)

Uloge uređaja

Otkrivanje uređaja. Tri režima otkrivanja i dve procedure za otkrivanje

Uspostavljanje konekcije. Tri režima konektovanja (periferije) i četiri procedure za konektovanje (centrale).

GATT

(Postoji izvesno preklapanje sa ATT, jer GATT koristi ATT i dalje ga nadograđuje. Korisno je pročitati i pripremu za Vežbu 4.)

Hijerarhija podataka (Sl. 25). Servisi i karakteristike.

Servis. Atribut za deklaraciju servisa. Primerni i sekundarni servisi.

Karakteristika. Atributi za deklaraciju i vrednost karakteristike. Deskriptori karakteristike (samo CCCD)

GATT operacije (pogledati i na slajdovima)

Sve ostala u sekciji GATT je informativnog karaktera

Sigurnost

Informativnog karaktera

- ♦ ♦ ♦ -