

Modeliranje brezžičnih omrežij

Maja Podbevšek
Peter Benko (63090004)
Žiga Ham
Miha Zidar (63060317)

8. januar 2013

Kazalo

1	Zgledi	3
1.1	HandOver	3
1.2	HostToHost	3
1.3	MultiRadio	5
1.4	Synchronized	5
1.5	Throughput	7
2	Lan80211	9
2.1	Gradniki	9
2.1.1	Access Point	9
2.1.2	WirelessHost	11
2.1.3	ChannelControl	14
2.1.4	IPv4NetworkConfigurator	14
3	Naš model	14
3.1	Opis	14
3.2	Meritve	14

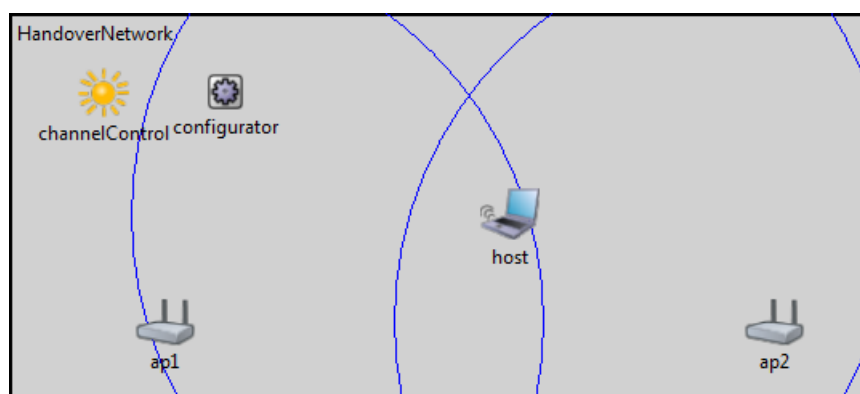
1 Zgledi

1.1 HandOver

V zgledu imamo naslednje module (Slika 1):

- dve dostopni točki
- enega odjemalca
- modul Configurator

Imamo le eno možno konfiguracijo. Odjemalec se premika po prostoru in na določeni točki zamenja dostopno točko s katero komunicira.



Slika 1: Zgled HandOver

1.2 HostToHost

V zgledu imamo naslednje module :

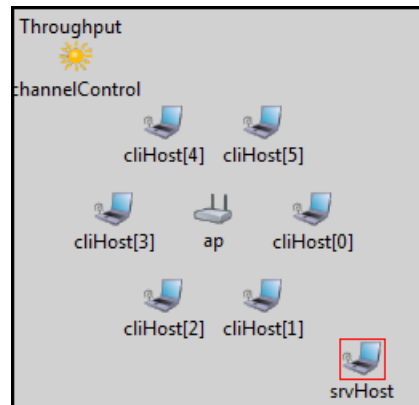
- nekaj odjemalcev, od katerih en deluje kot strežnik
- eno dostopno točko

Odjemalec in strežnik sta povezana z brezžično povezavo.

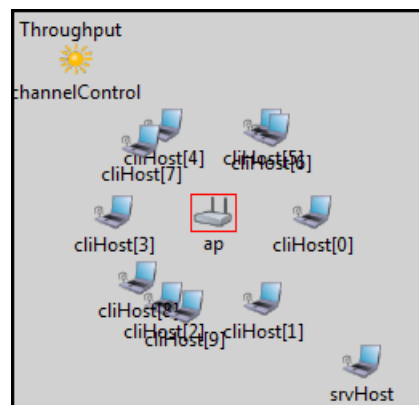
Na razpolago imamo dve možni konfiguraciji:

- Throughput1 (Slika 2) 6 odjemalcev in en strežnik skozi dostopno točko
- General (Slika 3) poljubno število odjemalcev in en strežnik skozi dostopno točko

Z zgledom merimo prepustnost omrežja pri komunikaciji odjemalca s strežnikom. Opazujemo tudi vpliv ostalih odjemalcev na komunikacijo med zgoraj omenjenim odjemalcem in strežnikom. Vsi akterji v omrežju povzročajo šum in s tem zmanjšujejo prepustnost.



Slika 2: Zgled HostToHost, s konfiguracijo Throughput1



Slika 3: Zgled HostToHost, s konfiguracijo General

1.3 MultiRadio

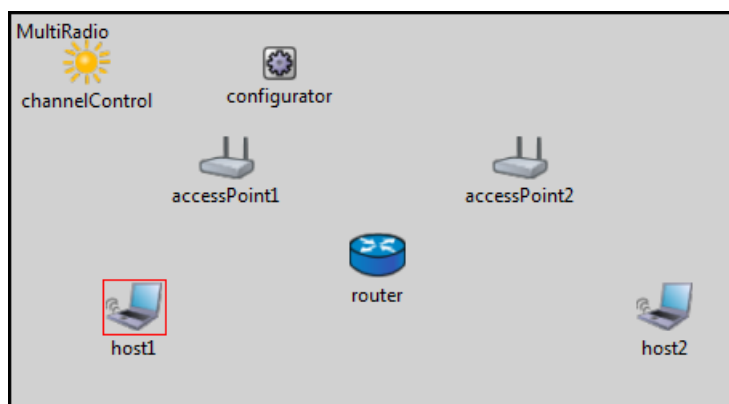
V zgledu imamo naslednje module:

- dva odjemalca
- dve dostopni točki
- en usmerjevalnik
- modul Configurator

Imamo tri konfiguracije:

- Switched Wlans (Slika 5) povezuje dostopnih točk preko ethernet stikala
- Routed Wlans (Slika 6) dva WLAN-a povezana preko usmerjevalnika z dvema brezžičnima mrežnima karticama
- Independent Wlans (Slika 7) dva neodvisna WLAN-a na različnih radijskih kanalih
- General (Slika 1.3) dva neodvisna WLAN-a preko usmerjevalnika

Zgled prikazuje uporabo večih brezžičnih vmesnikov na enem usmerjevalniku, tako da imamo več brezžičnih omrežij na različnih kanalih.

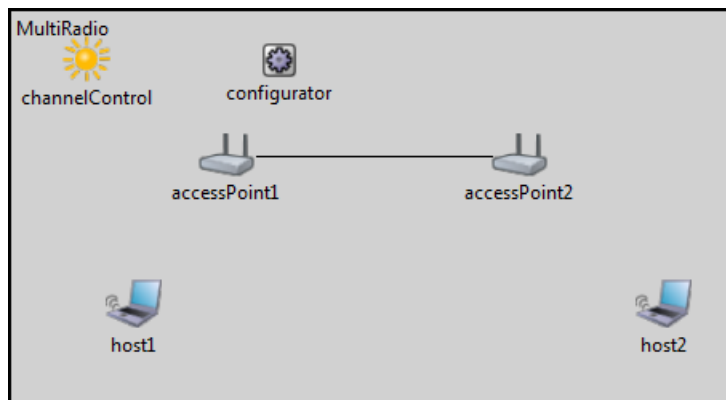


Slika 4: Zgled MultiRadio, s konfiguracijo General

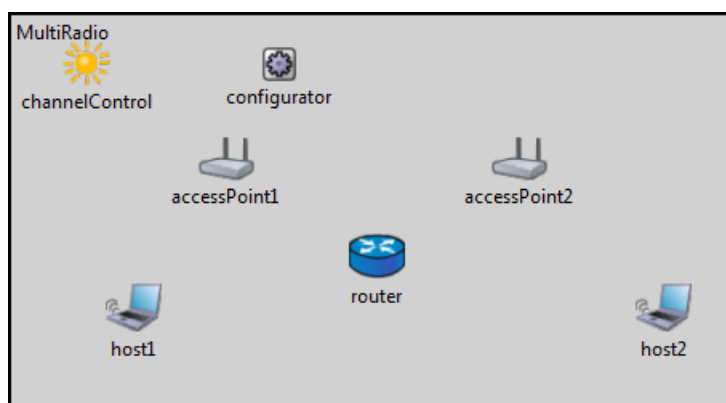
1.4 Synchronized

V zgledu imamo naslednje module:

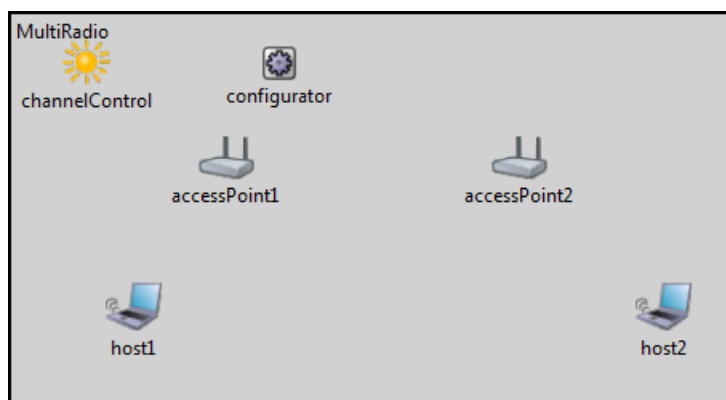
- AdhocHost
- modul Configurator



Slika 5: Zgled MultiRadio, s konfiguracijo Switched Wlans



Slika 6: Zgled MultiRadio, s konfiguracijo Routed Wlans



Slika 7: Zgled MultiRadio, s konfiguracijo Independent Wlans

Imamo dve konfiguraciji (Slika 8):

-

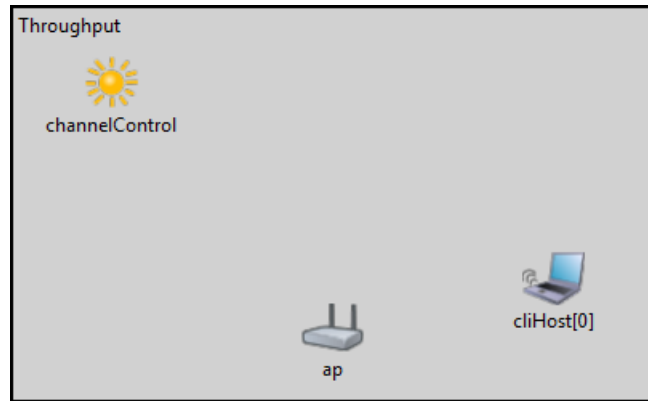
1.5 Throughput

- nekaj mobilnih odjemalcev
- eno dostopno točko

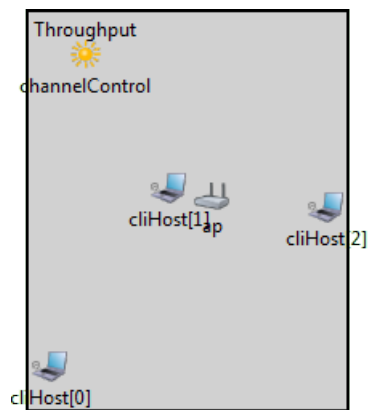
Imamo dva načina konfiguracije:

- Prepustnost merimo s “Sink” podmodulom dostopne točke.

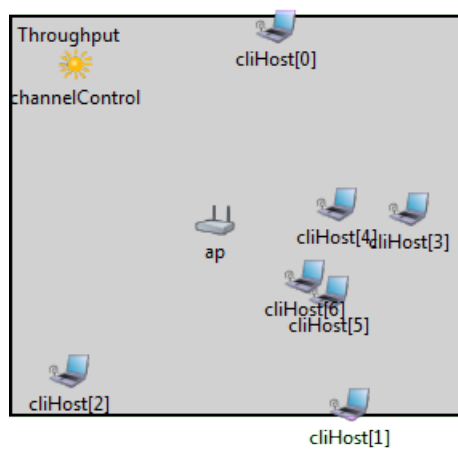
7



Slika 9: Zgled Throughput, s konfiguracijo Throughput1



Slika 10: Zgled Throughput, s konfiguracijo Throughput2



Slika 11: Zgled Throughput, s konfiguracijo General

2 Lan80211

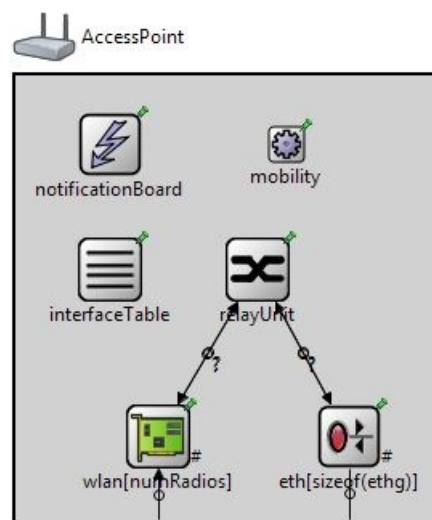


Slika 12: Shema modela 802.11

2.1 Gradniki

2.1.1 Access Point

oziroma dostopna točka je enota, ki lahko sprejema in oddaja brezžične signale. Hrani tudi tabelo vseh enot, ki so trenutno povezana na njo. Ima ena žična vrata v svetovni splet, ter na drugi strani anteno za brezžično komunikacijo. Vmes podatke posreduje relacijska enota.



Slika 13: Shema dostopne točke

2.1.1.1 interfaceTable je objekt preprostega modula InterfaceTable. Nima nikakešnih vrat in posledično tudi ne obdelave sporočil. Uporablja se zgolj z uporabo funkcij. Deluje kot

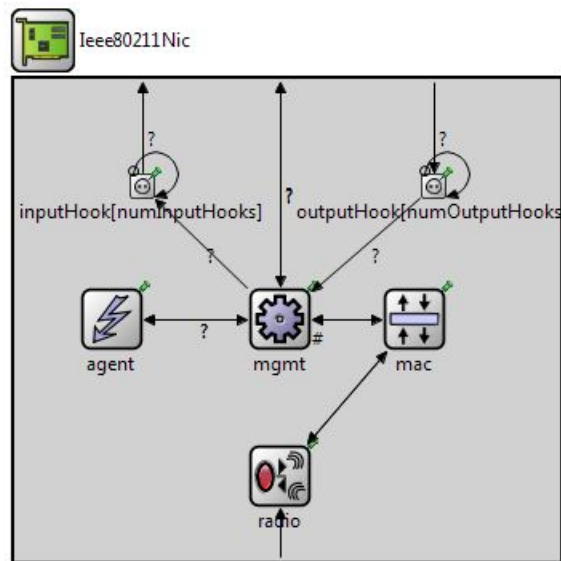
slovar za prevajanje, ali še bole, kot tabela za prevajanje vseh priključenih naprav. Obdeluje se dinamično, ob priklopu se nova naprava registrira v tabelo in poskrbi za vnose v routing tabele (IRoutingTable in IRoutingTable6). V interfaceTable se vnašajo InterfaceEntry-ji, ki vsebujejo podatke, kot so: interfaceId(unikadtna identifikacijska številka, ki zaznamuje vnos), id vhodnih in izhodnih vrat naslovnika, MTU, pasovno širino, MAC naslov, trenutno stanje (up/down) ter podatke kaj podpira (broadcast, multicast, point to point, loopback).

2.1.1.2 notificationBoard je objekt preprostega modula "notificationBoard", s katerim realiziramo vmesnik za sprejemanje in zazpečevanje obvestil. ob prijavi v tem modulu vsak odjemalec "prijavi" na kategorije, ki se jih tičejo. če želi katerikoli od odjemalcev poslati sporočilo mora zahtevo najprej predati notificationBoard-u, ki potem razpošlje to sporočilo vsem primernim odjemalcem, torej takim, ki so naročeni na specifično "kategorijo".

2.1.1.3 wlan[] predstavlja brezžično mrežno kartico (IWirelessNic - Wireless Network Interface Controller, po protokolu implementiranem v Ieee80211Nic), ki je razširitev modula mrežnih kartic ('INic'). Gre za eno kartico, na kateri vsak vnos v tabelo wlan[] predstavlja eno anteno oz. oddajnik. Vsebovani moduli wlan[] enote so prikazani na Sliki 14. Modul Ieee80211Radio je fizični nivo wlan-a. Oddaja in sprejema brezžične signale ter jih posreduje enoti 'mac' (modul Ieee80211Mac). Mac za sprejeti paket skrbi, da se je pravilno sprejel, oz. da se bo pravilno poslal. Pri pošiljanju potrebuje vsa polja paketa izpolnjena. Sam zna zapolniti le izvorni MAC naslov paketka. Enota mgmt (modul Ieee80211Mgmt) posreduje pakete med nivoji ter dodaja funkcionalnost agenta. Agent (preprost modul Ieee80211AgentSTA) skrbi za skeniranje kanalov, autentikacijo ter povezovanje. Deluje v dveh načinih skeniranja: aktivno ter pasivno. Agentu (modul Ieee80211AgentSTA) lahko podamo spisek kanalov, ki naj jih preveri, zamik med skeniranjem, minimalni čakalni čas, ki ga mora porabiti na določenem kanalu med aktivnim skeniranjem, čas ki ga porabi na vsakem kanalu ob pasivnem skeniranju, avtentikacijski timeout ter timeout za povezovanje. Agentu določa tudi SSID, ki ga bo wlan[] oglaševal. Input in output zanki sta realizirani zgolj za simulacijo, saj sta zelo primerni za umetno ustvarjanje določenih napak na omrežju.

2.1.1.4 relayUnit je objekt modula IMACRelayUnit, ki skrbi za posredovanje in obdelavo prejetih okvirjev. Ima vhodna in izhodna vrata od/k nižjim plastem, skozi katera na eni strani komunicira z ožičenim omrežjem in na drugi strani z brezžičnim omrežjem. S pomočjo tabele naslovov usmerja pakete na pravilna vrata in oz. pravilne MAC naslove. Skrbi tudi za svežino tabele naslovov, saj ima omejeno velikost. Ko se le-ta zapolni, bo iz nje izbrisan najstarejši vnos.

2.1.1.5 eth[] sloni na preprostem modulu EtherMACFullDuplex, ki je preprostejša oblika IEtherMAC. Skrbi za ožičeno komunikacijo. Prav tako kot wlan[] ne izvaja nikakršne enkapsulacije/dekapsulacije, za to posreduje okvirje višjim plastem. Pri pošiljanju se vsi prejeti paketi (od višjih plasti) postavijo v vrsto, kjer počakajo, da je oddajnik prost. Ko je možno jih takoj



Slika 14: Shema dostopne točke

pošlje. V paketu morajo biti vsa polja že izpolnjena, le izvorni MAC se bo tu naknadko dodal, če je polje prazno. Pri sprejemanju paketov (iz omrežja) opravi najprej CRC check, pri čimer so paketi z napako zavrženi, in nato posreduje pakete naprej višji plasti.

2.1.1.6 mobility zagotavlja zgolj vizualno realizacijo gibljivih modelov na shemi omrežja - v tem primeru dostopne točke, gi pa je stacionarna, zato vrača lastnosti `getCurrentPosition` in `getCurrentSpeed` enaki 0.

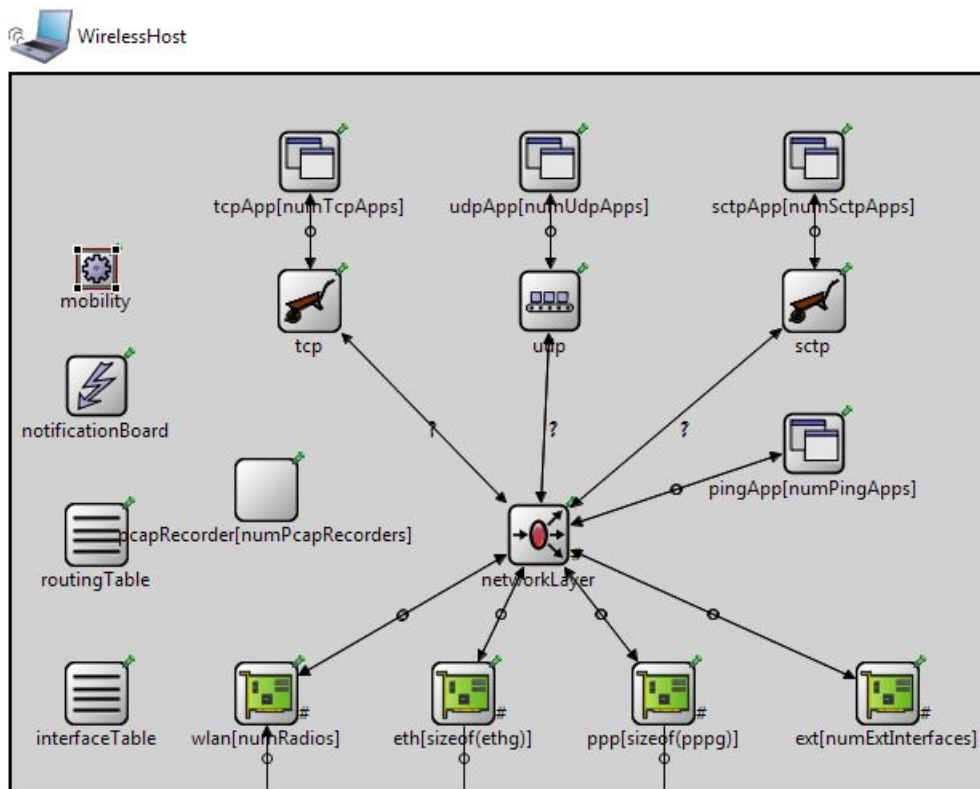
2.1.2 WirelessHost

2.1.2.1 Mrežne kartice

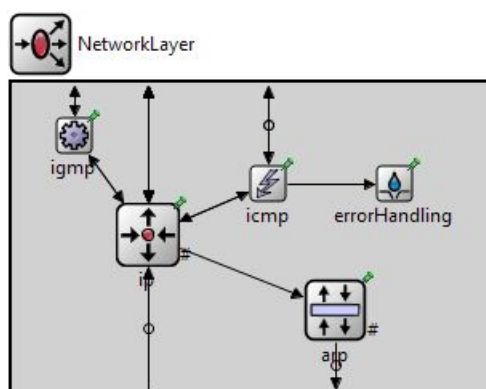
1. **wlan[]** zagotavlja brezžično povezljivost naprave, s pomočjo modula `Ieee80211Nic`, enako kot je opisano pod točko 2.1.1.3 - wlan[] dostopne točke.
2. **eth[]** zagotavlja ožičeno povezljivost s pomočjo modula `EthernetInterface`. Pri lan80211 se ne uporablja, saj vsa komunikacija poteka preko wlan[] povezave.
3. **ppp** zagotavlja point-to-point povezljivost, prav tako kot eth[] se pri lan80211 ne uporablja.
4. **ext** omogoča realizacijo zunanjih mrežnih kartic.

2.1.2.2 NetworkLayer

1. **ip** je implementiran s preprostim modulom IPv4. Izpolnjuje IPv4 en-/de-kapsulacijo paketa. Ko je sprejet paket sprejet, ga odpre in odstrani podatke predavišji plasti v obliki `IPv4ControlInfo`, izluščene podatke pa preda kot `cMessage`. V obratni smeri



Slika 15: Shema modula WirelessHost



Slika 16: Shema modula NetworkLayer

je pričakovano, da višja plast poleg sporočila poda tudi izpolnjen IPv4ControlInfo. Ob pošiljanji se preveri izhodni port v RoutingTable ter naslov naslednjega skoka paketa.

2. **icmp** skrbi za implementacijo ICMP-ja. Sprejema ICMP sporočila in jih procesira. Za prejete zahteve ustvari odgovore ter jih predatakoj nazaj v IPv4 enkapsulacijo. Opazimo da se zunanji 'ping' zahtevki obdelajo na mrežni plasti. če sprejme ICMP odgovor, ga izlušči iz ICMP okvirja, ter preda višji plasti. če sprejme ping zahtevek z višje plasti, ga enkapsulira v ICMP okvir ter preda IPv4 enoti.
3. **igmp** pošilja obvestila o pripadnosti multicast skupinam odjemalca, na strani routerja pa ta obvestila procesira.
4. **arp** implementira protokol ARP za IPv4 in 6-bajtni MAC.
5. **errorHandling** procesira sporočila o napakah od ostalih modulov.

2.1.2.3 Protokoli

1. **tcp** realizira TCP protokol (preko ITCP modula), preko katerega lahko aplikacije vzpostavljajo TCP povezave. Za vzpostavitev povezave mora aplikacija protokolni enoti najprej poslati zahtevek za vzpostavitev aktivne oz. pasivne povezave preko controlInfo lastnosti cMessage-a. Za pošiljanjepaketov mora aplikacija paketku nastaviti TCP_C_SEND kot vrsto sporočila ter priložiti TCPSendCommand kontrolne informacije. Za zaprtje povezave aplikacija pošlje TCP_C_CLOSE vrsto sporočila z TCPSend kontrolnimi informacijami, da se ve katero povezavo se zapira. Modula tudi obvešča aplikacijo o vseh velikih spremembah v povezavi (povezava vzpostavljena, prekinjena, zavrnjena, izgubljena...).
2. **udp** implementira UDP protokol (preko IUDP modula). Za pošiljanje aplikacija zkolj preda paket udp enoti s priloženim UDPControlInfo objektom. Za prejemanje mora aplikacija najprej 'rezervirati' določen port. To stori z sporočilom vrste UDP_C_BIND s priloženim UDPControlInfo, ki ima izpolnjeno srcPort polje.
3. **setp** vmesnik za realizacijo SCTP protokola.

2.1.2.4 Aplikacije So okvirni primeri aplikacij, ki imajo realizirana zgolj vrata, ključna za delovanje aplikacije s primernim protokolom.

2.1.2.5 Ostalo

1. **routingTable** hrani routingTable. V njej se hranijo podatki za usmerjanje paketkov. V programu se navezuje na dokument 'routingFile'.
2. **pcapRecorder** je implementiran za beleženje, katere okvirje pošiljajo ostali moduli znotraj iste enote. Izbiramo lahko katere module se beleži ter katerim signalom se sledi ter dodajamo.

3. **interfaceTable** opis v 2.1.1.1
4. **notificationBoard** opis v 2.1.1.2
5. **mobility** opis v 2.1.1.6

2.1.3 ChannelControl

Vsako omrežje z mobilnimi napravami ali brezžičnimi omrežji. Glede na lokacijo elementov določa kateri so v dosegu, da lahko komunicirajo med sabo oz. povzročajo inteferenco. Te podatke potem uporabijo brezžične enote, da lahko določajo s katerimi enotami lahko komunicirajo.

2.1.4 IPv4NetworkConfigurator

Ta modula dodeli IP naslove (lahko ročno ali avtomatično) in vzpostavi statično usmerjanje po IPv4 omrežju. Module, ki predstavljajo omrežne enote prepozna po @node lastnosti, zato jo morajo vse enote vsebovati. Prav tako je od njih pričakovano, da imajo vsaka svojo InterfaceTable, za routerje pa še RoutingTable.

3 Naš model

3.1 Opis

3.2 Meritve