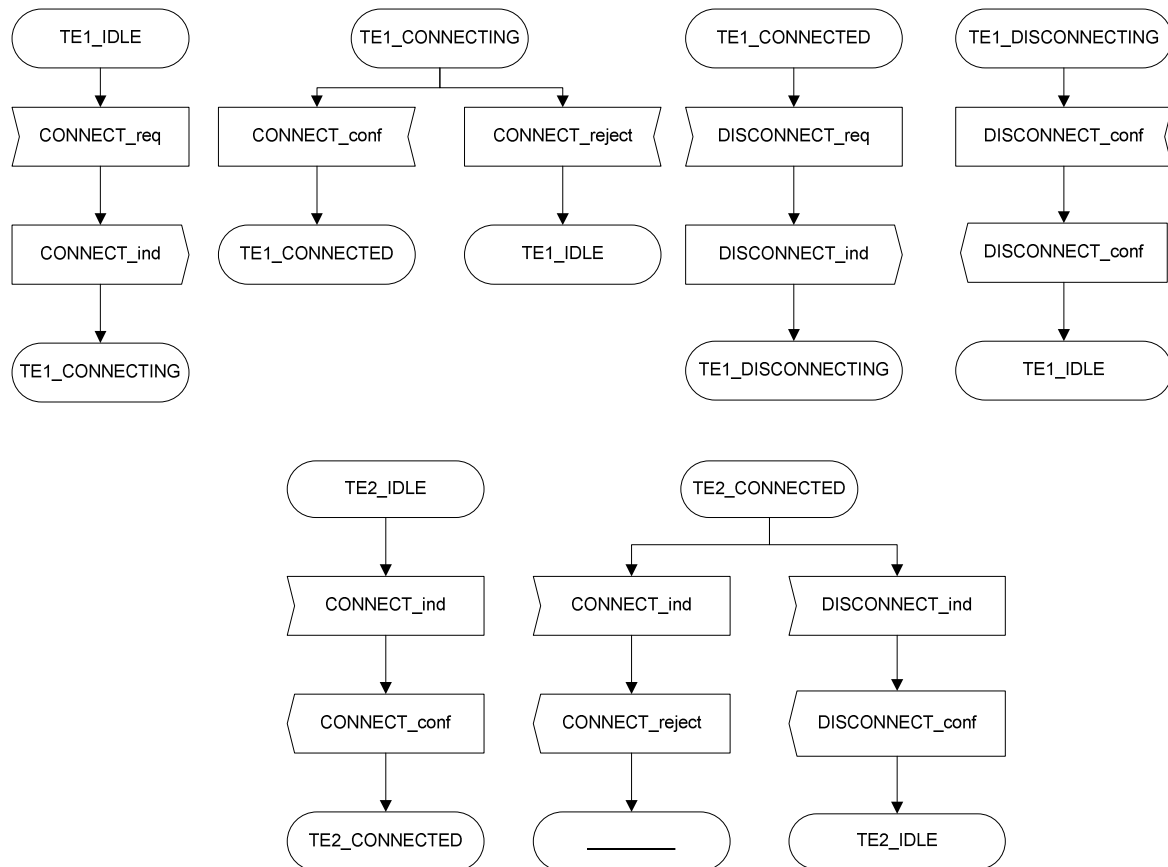


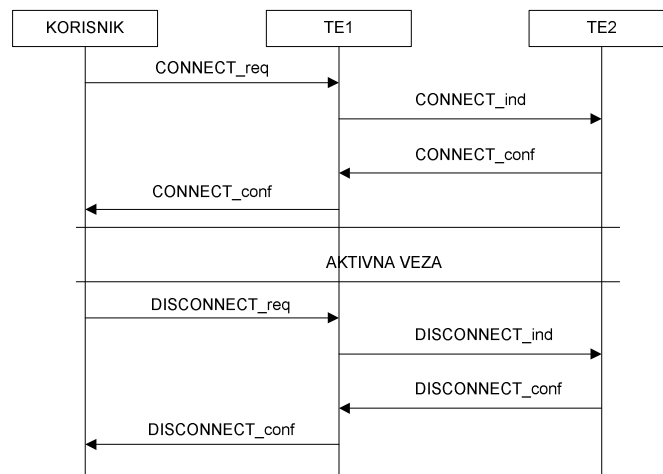
## Primer 1

Ovaj primer demonstrira proceduru za uspostavu i raskid veze između dva objekta. Veza se uspostavlja slanjem zahteva za uspostavom veze i prijemom portvrde. Veza se raskida slanjem zahteva za raskidom i prijemom potvrde. U nastavku (Slika 1.) su dati SDL dijagrami koji opisuju ponašanje objekata u komunikaciji.



Slika 1. SDL dijagrami objekata u komunikaciji

MSC dijagram na Slici 2 opisuje tok uspostave i raskida veze.



Slika 2. MSC dijagram uspostave i raskida veze

U narednim tabelama su navedeni primeri TTCN deklaracija, ograničenja i opisa samih testova koji predstavljaju deo TTCN skupa testova za dati primer.

Simple Type Definitions			
Type Name	Type Definition	Type Encoding	Comments
O_1	OCTETSTRING[1]		
O_2	OCTETSTRING[2]		
Detailed Comments:			

PDU Type Definition			
PDU Name : CONNECT_ind			
PCO Type :			
Encoding Rule Name :			
Encoding Variation :			
Comments : Primer PDU definicije			
Field Name	Field Type	Field Encoding	Comments
Source_address	O_1		
Destination_address	O_1		
User_data	O_2		
Detailed Comments:			

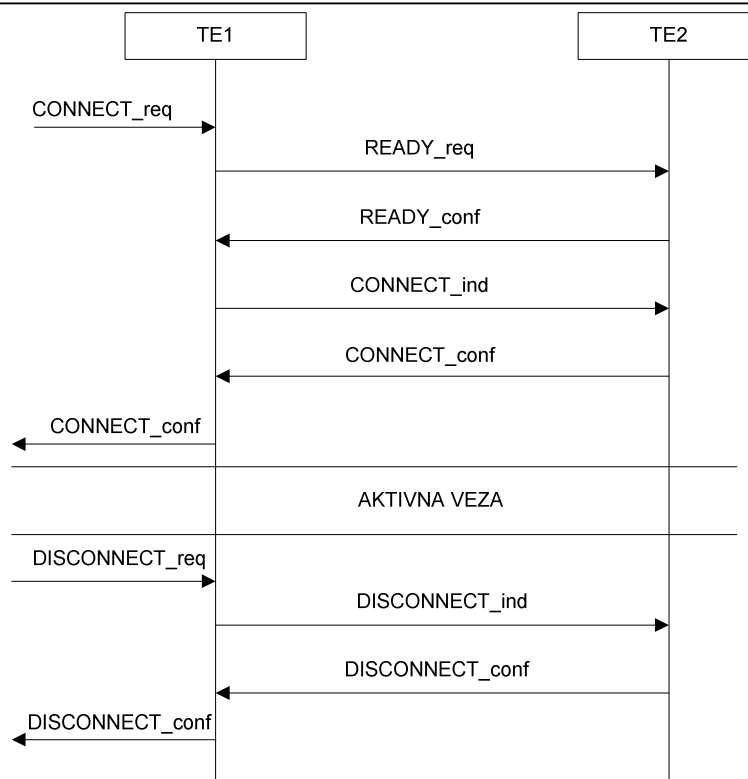
PDU Constraint Declaration			
Constraint Name : CONNECT_ind			
PDU Type : CONNECT_ind			
Derivation Path :			
Encoding Rule Name :			
Encoding Variation :			
Comments :			
Field Name	Field Value	Field Encoding	Comments
Source_address	–		
Destination_address	–		
User_data	–		
Detailed Comments:			

Test Case Dynamic Behaviour					
<b>Test Case Name :</b> Osnovna uspostava veze TE1 <b>Group :</b> <b>Purpose :</b> Provera uspostave normalnog poziva. <b>Configuration :</b> <b>Default :</b> <b>Comments :</b>					
No.	Label	Behaviour Description	Constraint Ref	Verdict	Comments
1		L?CONNECT_req			
2		L!CONNECT_ind			
3		L? CONNECT_conf	CallEstablished	PASS	
4		L? CONNECT_reject	CallNotEstablished	INCONC	
Detailed Comments:					

Test Case Dynamic Behaviour					
<b>Test Case Name :</b> Osnovni raskid veze TE2 <b>Group :</b> <b>Purpose :</b> Provera raskida normalnog poziva. <b>Configuration :</b> <b>Default :</b> <b>Comments :</b>					
No.	Label	Behaviour Description	Constraint Ref	Verdict	Comments
1		L?CONNECT_req			
2		L!CONNECT_ind			
3		L?CONNECT_conf	CallEstablished		
4		L?DISCONNECT_req			
5		L!DISCONNECT_ind	Disconnect		
6		L?DISCONNECT_conf		PASS	
7		L? CONNECT_conf		FAIL	
8		L? CONNECT_reject		FAIL	
9		L?CONNECT_reject	CallNotEstablished	INCONC	
Detailed Comments:					

## Zadatak 1

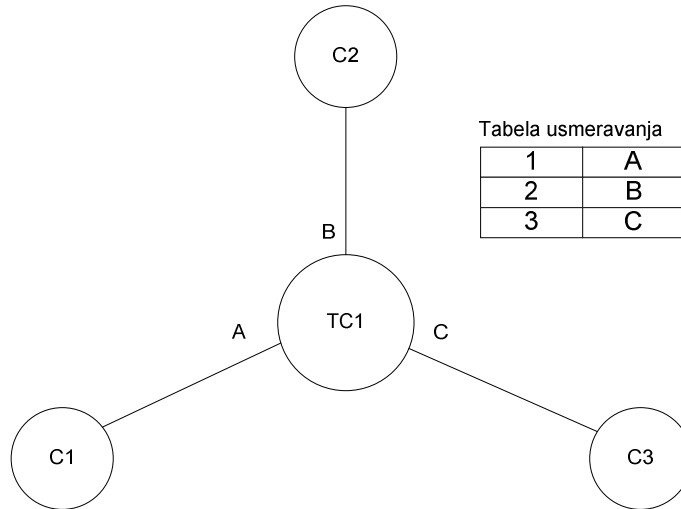
Ispraviti dati primer tako da uspostavi veze prethodi provera spremnosti TE2 objekta. MSC dijagram koji opisuje tok uspostave veze dat je na slici 3.



Slika 3. MSC dijagram uspostave i raskida veze

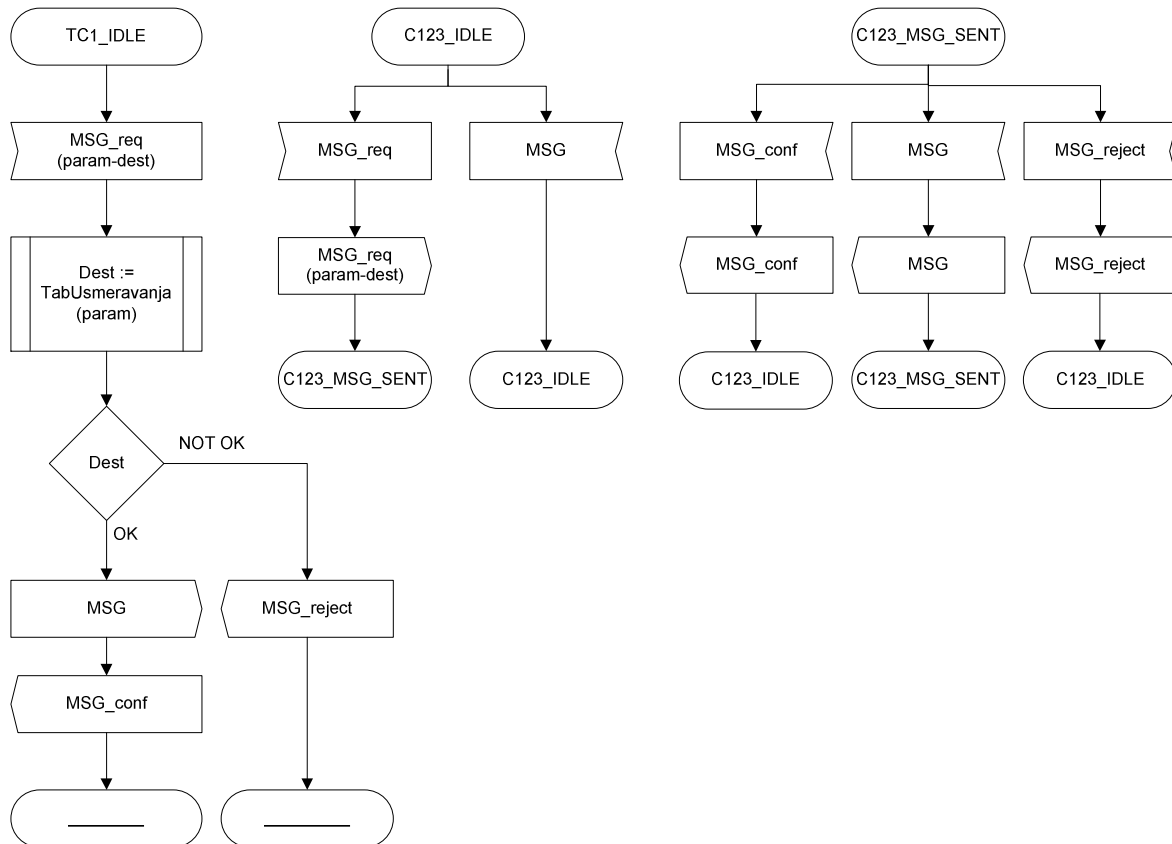
## Primer 2

Na slici 4 je prikazan način komunikacije između čvorova 1, 2 i 3. Na slici je prikazana i tabela usmeravanja. U zavisnosti od vrednosti parametra poruke, poruka se usmerava na određeni čvor mreže.



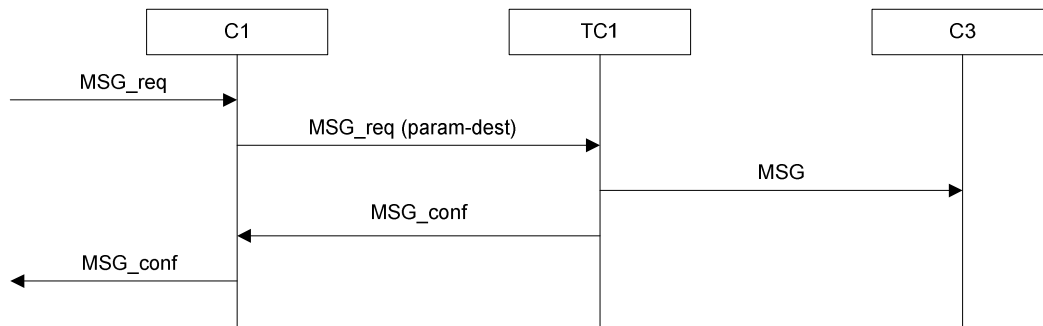
Slika 4. Komunikacija čvorova u sistemu

Naredni SDL dijagrami (slika 5) opisuju ponašanje pojedinih automata u sistemu.

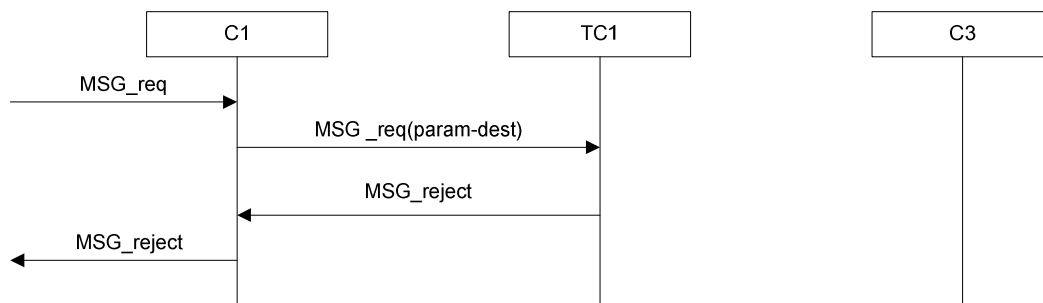


Slika 5. SDL dijagram MSC dijagram uspostave i raskida veze

Na narednim MSC dijagramima (slika 6) su ilustrovani primeri uspešnog (slika 6) i neuspešnog (slika 7) slanja poruke.



Slika 6. MSC dijagram uspešnog slanja poruke



Slika 7. MSC dijagram neuspešnog slanja poruke

U narednim tabelama su navedeni primeri TTCN testova.

PDU Type Definition			
PDU Name : MSG_req			
PCO Type :			
Encoding Rule Name :			
Encoding Variation :			
Comments : Primer PDU definicije			
Field Name	Field Type	Field Encoding	Comments
Destination_address	INTEGER		m
User_data	OCTETSTRING[2]		
Detailed Comments:			

PDU Constraint Declaration			
Constraint Name : MSG_req_destination_addr_ok			
PDU Type : MSG_req			
Derivation Path :			
Encoding Rule Name :			
Encoding Variation :			
Comments :			
Field Name	Field Value	Field Encoding	Comments
Destination_address	1		

User_data	*		
Detailed Comments:			

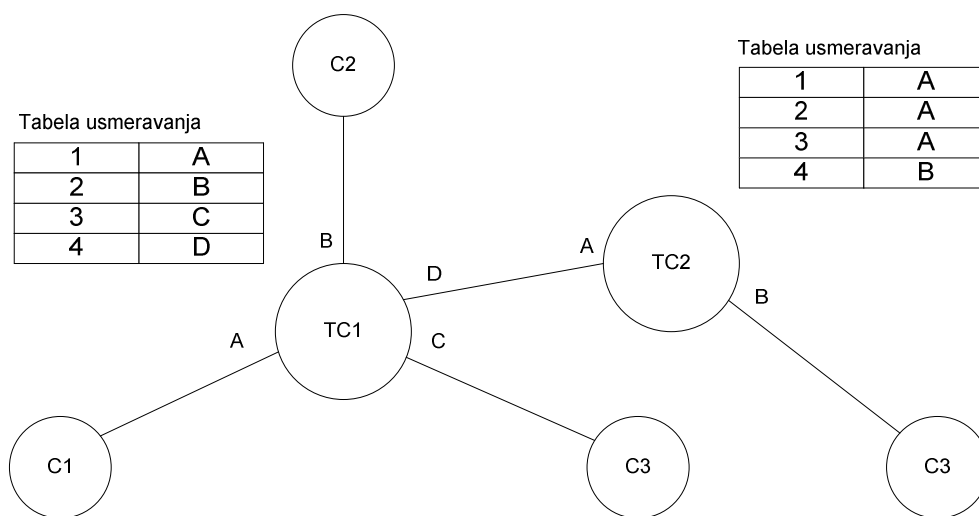
PDU Constraint Declaration			
Constraint Name	: MSG_req_destination_addr_not_ok		
PDU Type	: MSG_req		
Derivation Path	:		
Encoding Rule Name	:		
Encoding Variation	:		
Comments	:		
Field Name	Field Value	Field Encoding	Comments
Destination_address	-1		
User_data	*		
Detailed Comments:			

Test Case Dynamic Behaviour					
Test Case Name : Slanje poruke poznatom korisniku C1.					
Group :					
Purpose :					
Configuration :					
Default :					
Comments :					
No.	Label	Behaviour Description	Constraint Ref	Verdict	Comments
1		L?MSG_req			
2		L!MSG_req_destination_addr_ok			
3		L?MSG_conf	Poruka je uspešno poslata.	PASS	
4		L?MSG_reject		FAIL	
Detailed Comments:					

Test Case Dynamic Behaviour					
Test Case Name : Slanje poruke korisniku koji nije na raspolaganju.					
Group :					
Purpose :					
Configuration :					
Default :					
Comments :					
No.	Label	Behaviour Description	Constraint Ref	Verdict	Comments
1		L?MSG_req			
2		L!MSG_req_destination_addr_not_ok			
3		L?MSG_conf	Poruka je uspešno poslata.	FAIL	
4		L?MSG_reject		PASS	
Detailed Comments:					

## Zadatak 2

Na osnovu prethodnog primera nacrtati SDL i MSC dijagrame koji opisuju sistem prikazan na slici 8. Za dati sistem realizovati odgovarajuće TTCN testove.

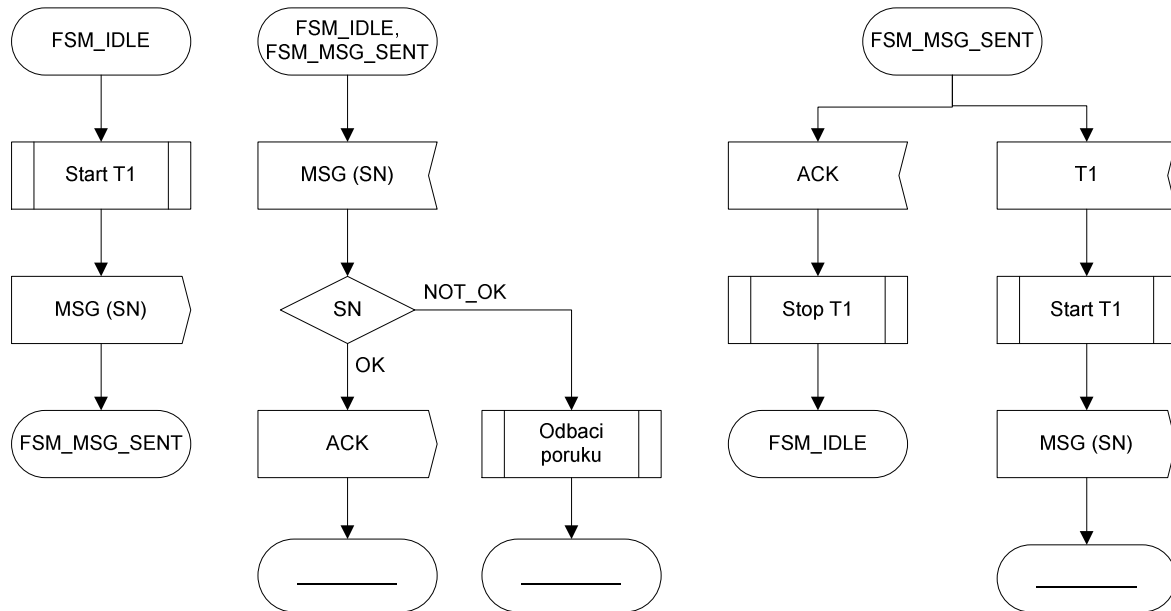


Slika 8. Sistem čvorova sa 2 čvora za usmeravanje

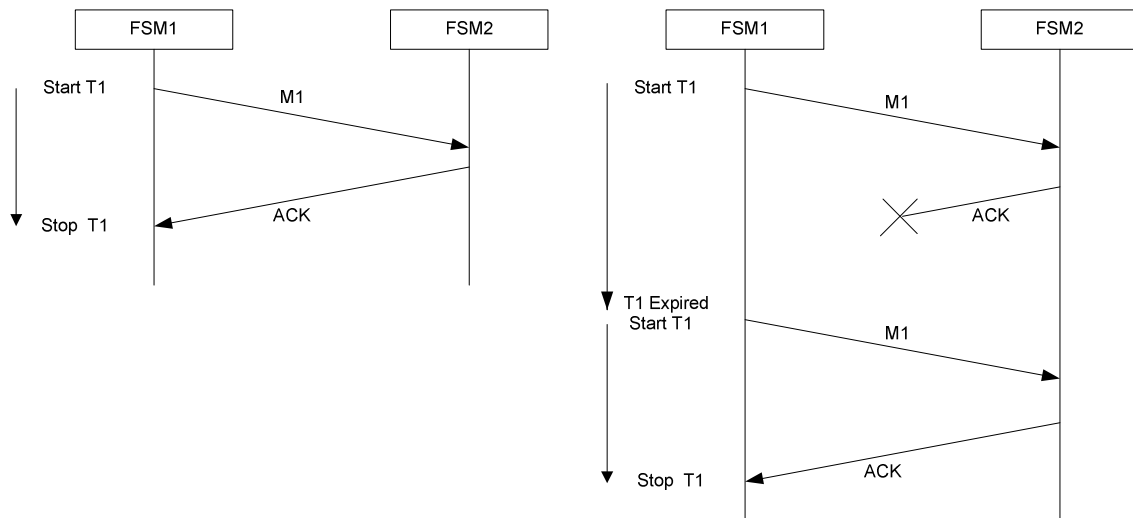


### Primer 3

Primer ilustruje mehanizam potvrde paketa. Svaki objekat u komunikaciji očekuje potvrdu prijema paketa koji je poslao. Ukoliko potvrda ne stigne u predviđenom vremenskom periodu paket se ponovo šalje. Naredni SDL (Slika 9) i MSC (Slika 10) dijagrami opisuju takav način komunikacije.



Slika 9. DSL dijagrami sistema sa potvrdom paketa



Slika 10. MSC dijagrami sistema sa potvrdom paketa

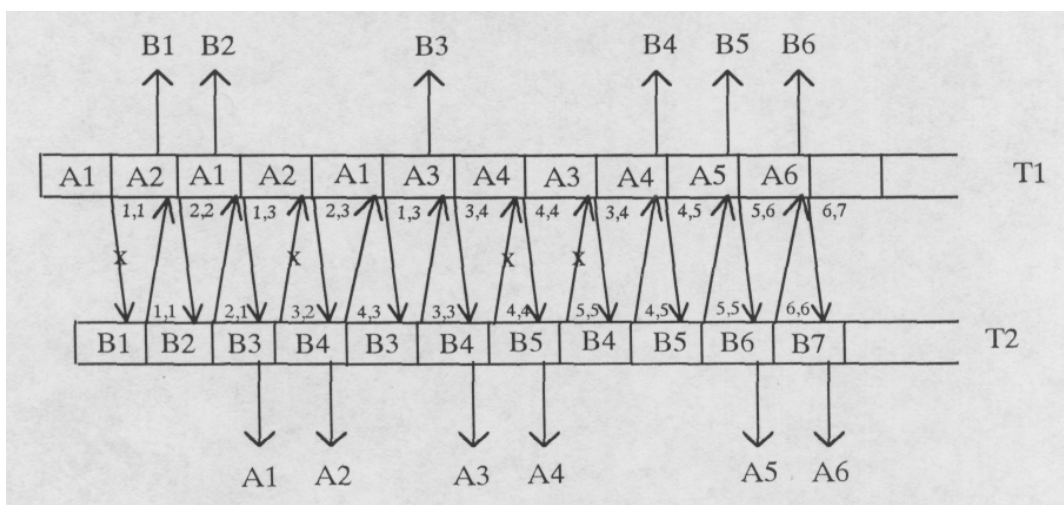
### Zadatak 3

Za prethodni primer realizovati TTCN skup testova.

## Primer 4

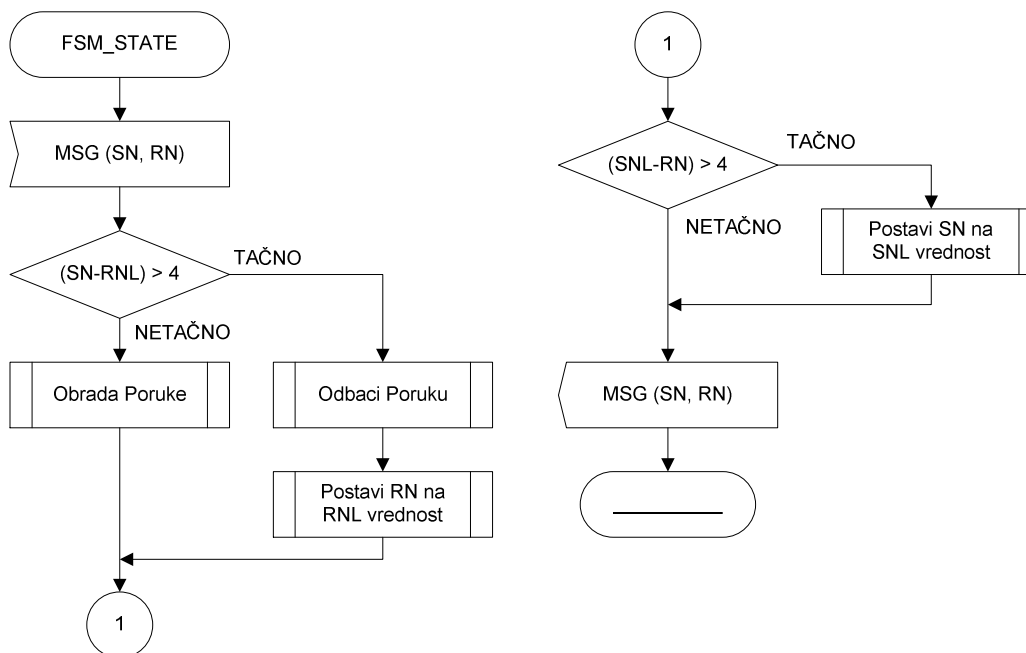
Naredni primer ilustruje mehanizam klizajućeg prozora. Prijemnik šalje predajniku broj sledećeg očekivanog okvira. U slučaju da predajnik ode suviše daleko u slanju okvira, on se vraća nazad i ponavlja predaju od prijemnikovog sledećeg očekivanog okvira. Za realizaciju ovog mehanizma potrebno je uvesti sledeća četiri brojača:

- SN – predajni broj redosleda (polje iz primljenog okvira)
- RN – prijemni broj redosleda (polje iz primljenog okvira)
- SNL – predajnik broji poruke (po modulu) iz izvora (lokalni brojač objekta u komunikaciji)
- RNL – broj sledećeg očekivanog okvira od strane prijemnika (lokalni brojač objekta u komunikaciji)

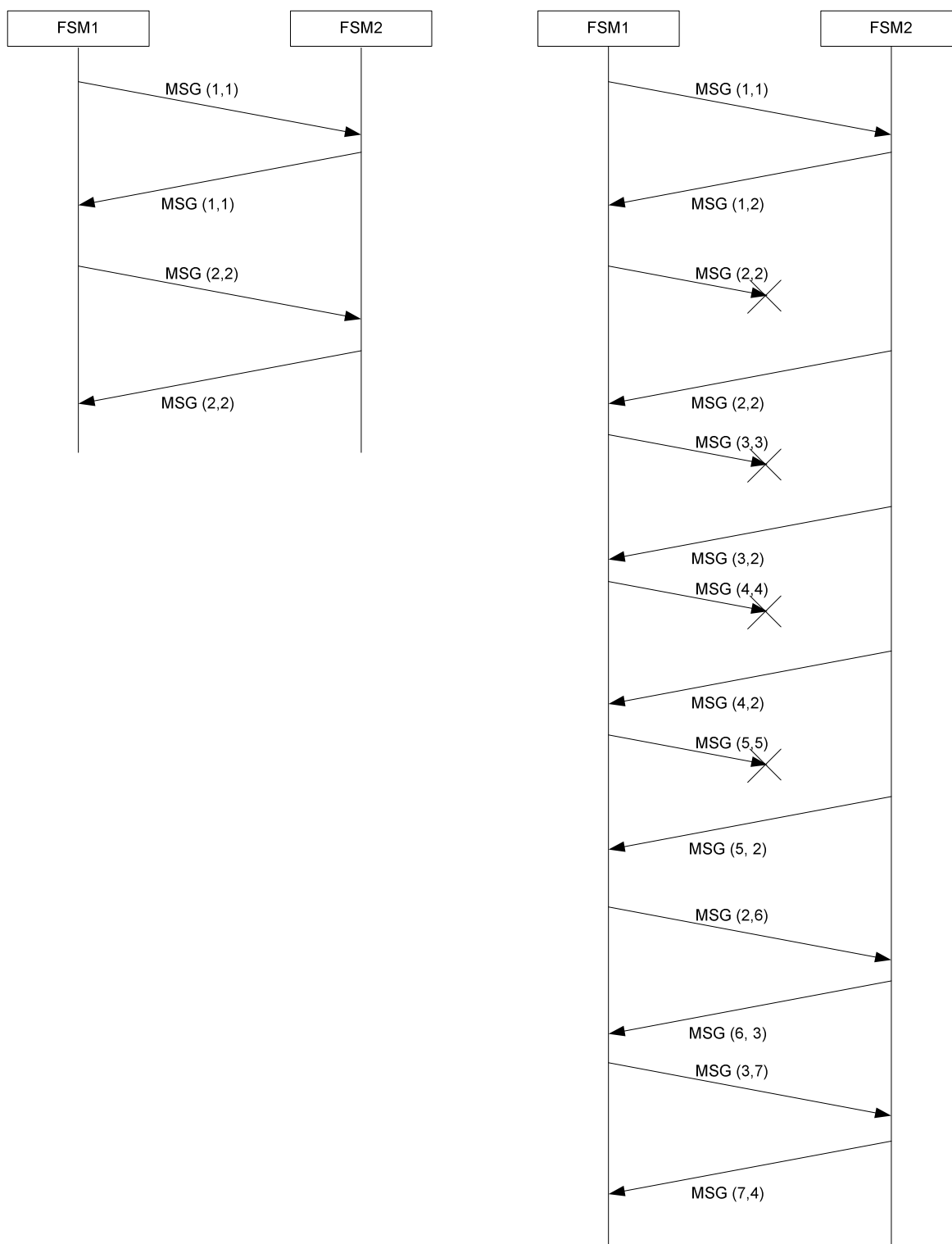


Slika 11. Primer korišćenja prozora širine 2.

Naredni SDL i MSC (slika 12,13) dijagrami ilustruju mehanizam klizajućeg prozora širine 4.



Slika 12. SDL dijagram mehanizma klizajućeg prozora širine 4



Slika 13. MSC dijagrami mehanizma klizajućeg prozora širine 4

#### Zadatak 4

Na osnovu prethodnog primera nacrtati SDL i MSC dijagrame koji opisuju mehanizam klizajućeg prozora širine 6.

## Primer 5

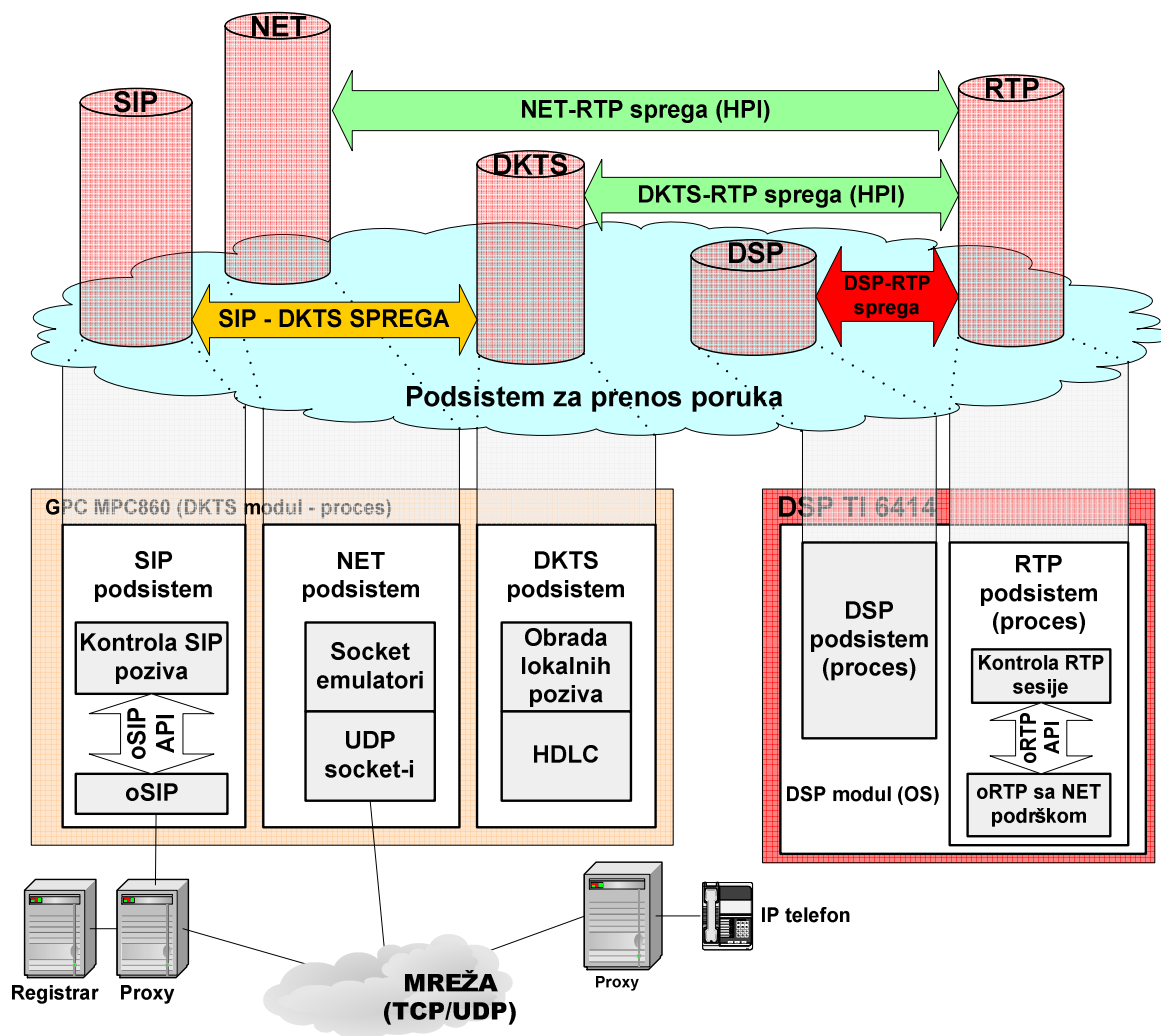
U ovom primeru prikazani su MSC i UML dijagrami 2 podsistema programske podrške MPU-352 uređaja.

Multifunkcionalni Pristupni Uređaj (MPU) je sistem koji omogućava prihvatanje 352 korisnika. MPU podržava lokalne veze, veze sa korisnicima preko E1 linka i veze sa Internet korisnicima preko *Ethernet* priključka. Osnovu fizičke arhitekture MPU sklopa čine:

- GPC (Glavni Pristupni Čvor) ploča;
- Jedna ili više učesničkih sekcija (US16).

Na GPC ploči se nalaze sledeći elementi (slika 2): mrežni procesor (*eng. Network Processor*) iz familije procesora Motorola MPC860, TDM (*eng. Time Division Multiplexing*) komutaciona matrica (SW), digitalni signalizacioni procesor (*eng. Digital Signal Processor - DSP*) TI 6414, četverostruka E1 sprega i linijski stepeni.

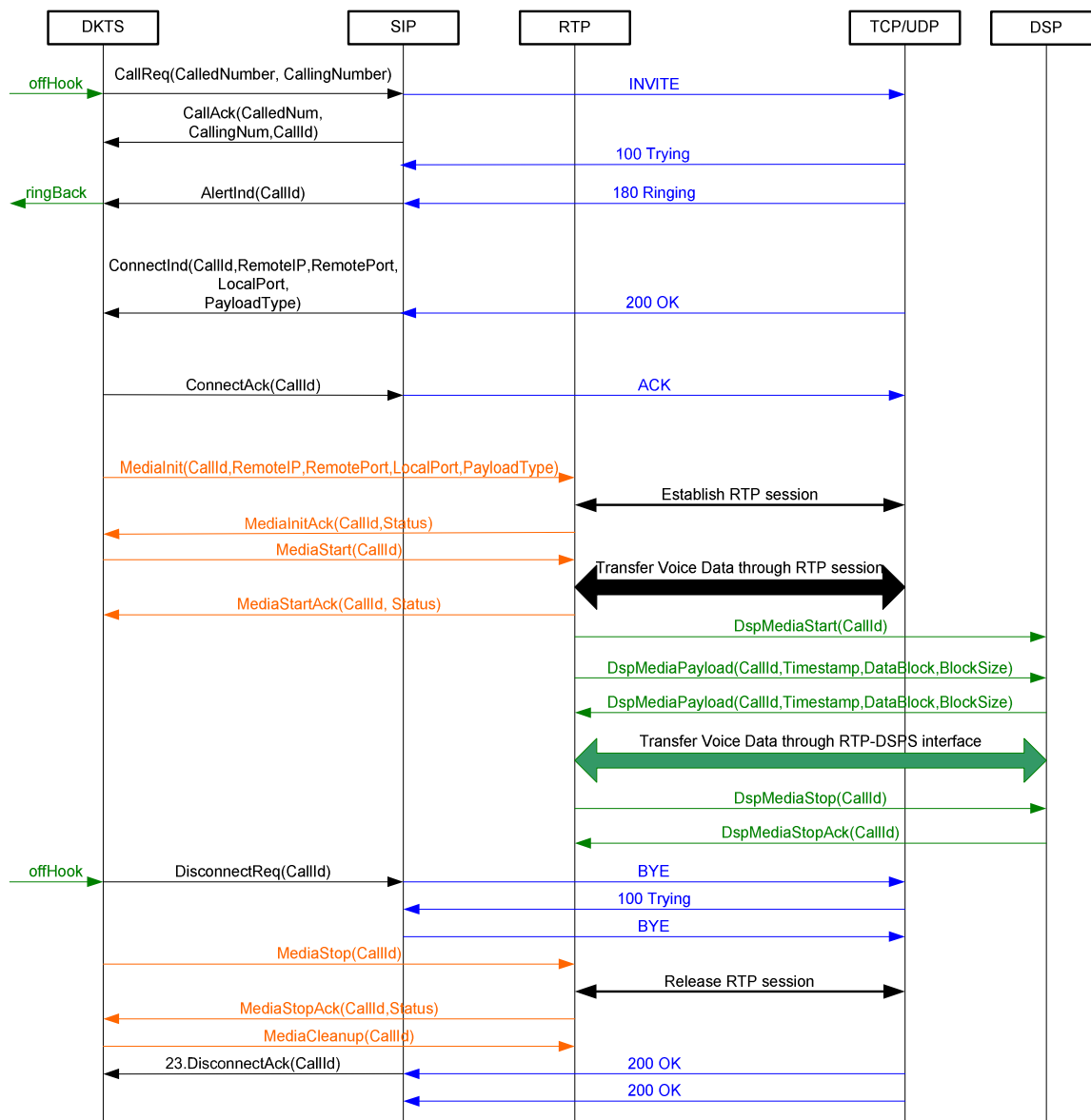
Programsku podršku MPU-352 uređaja čini određeni broj podsistema međusobno komuniciraju preko podsistema za prenos poruka. SIP, NET i DKTS podsistemi se nalaze u okviru DKTS modula, na procesoru opšte namene. DSP i RTP podsistemi se nalaze u okviru DSP modula, na DSP procesoru.



Slika 14. Arhitektura programske podrške MPU-352 uređaja

- SIP podsistem vrši signalizaciju mrežnih poziva preko SIP (eng. *Session Initiation Protocol*) protokola prema RFC 3261 standardu i obezbeđuje analizu i sintezu svih SIP i SDP poruka.
- DKTS podsistem vrši signalizaciju lokalnih poziva i predstavlja centar sistema jer nosi informacije o obe strane u pozivu.
- RTP podsistem vrši upravljanje prenosom medijskog sadržaja. U njemu je ostvaren prenos govora i video toka između dva korisnika, a vrši paketizaciju informacija prema RTP protokolu.
- NET podsistem je uslužni podsistem RTP podsistema. On omogućava RTP podsistemu pristup mreži preko procesora opšte namene.

Na slici 15. prikazan je MSC dijagram uspostave i raskida odlaznog poziva.



Slika 15. MSC dijagram odlaznog poziva

Na slici 16. dat je UML dijagram klasa SIP podsistema.



```

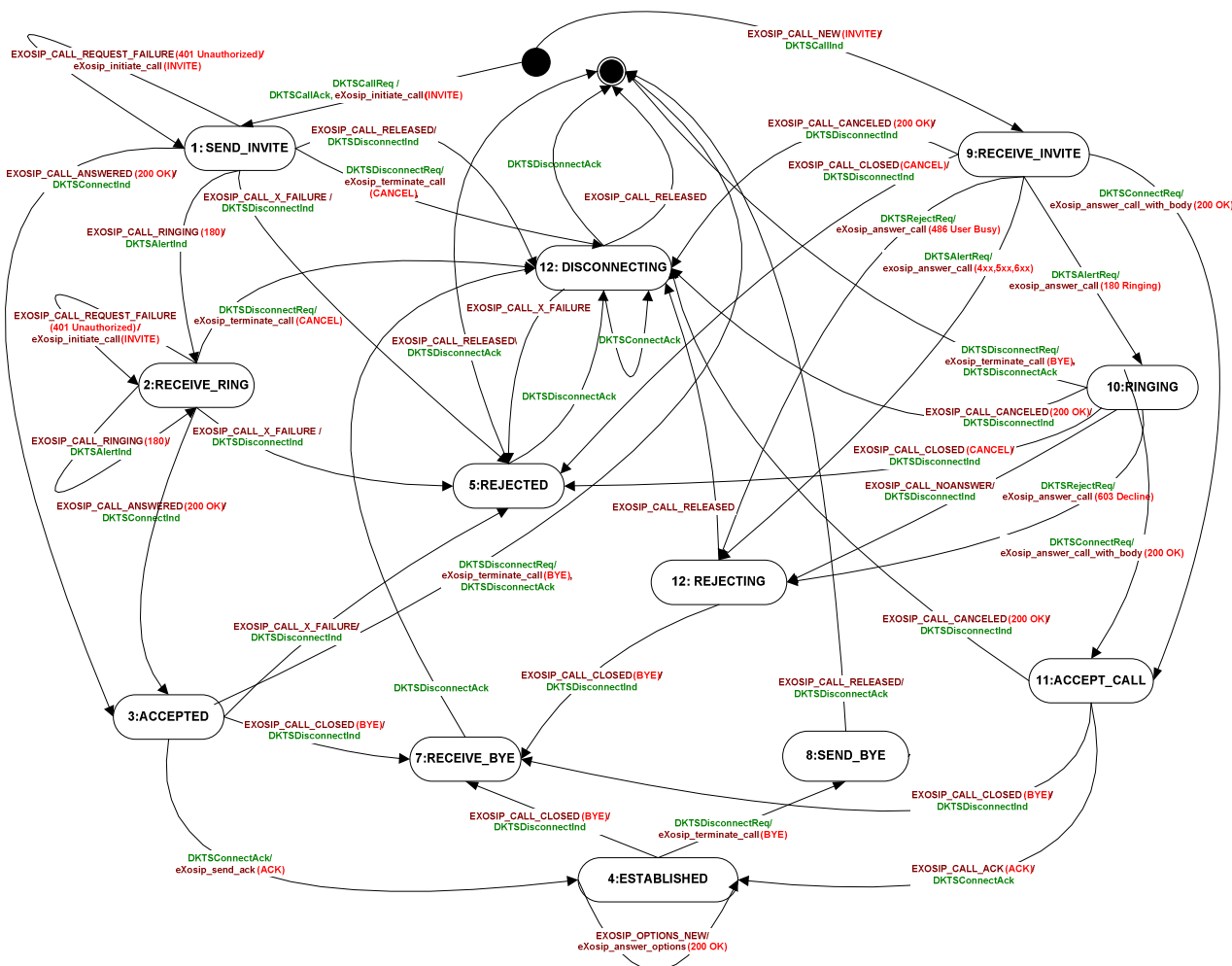
classDiagram
    class IPCMsg {
        +callID
        +localPort
        +remotePort
        -payloadType
        -localIP
        -remoteIP
        -timestamp
        -dataBlock : unsigned char
        -blockSize : unsigned int
    }
    class IPCReceiverTemplate {
        -id
        -idGen
        -*sl : SessionList
    }
    class ConcurrencySupplier {
    }
    class MsgRTP {
        +callID
        +localPort
        +remotePort
        -payloadType
        -localIP
        -remoteIP
        -timestamp
        -dataBlock : unsigned char
        -blockSize : unsigned int
    }
    class RTPImp_MsgSIP["RTPImp<MsgSIP>"] {
        -id
        -idGen
        -*sl : SessionList
    }
    class NETCtrlImp_MsgNETCtrl["NETCtrlImp<MsgNETCtrl>"] {
        -id
    }
    class SessionList {
        -*first : Session
        -count : unsigned int
        -*cs
        +putInList() : int
        +findSession() : Session
        +getFromList() : Session
    }
    class Session {
        -state : int
        -cid : int
        +localport : int
        +remoteport : int
        -lockCall() : void
        -unlockCall() : void
        +addCall() : void
        +removeCall() : void
    }
    class MsgNETCtrl {
        +sessionID
        +socketID
        +sockFamily
        -payloadType
        -addressFamily
        -errNet
        -errnoNet
        -dataBlock : unsigned char
        -remAddrLen : unsigned int
        -result
        -blockSize
        -addrlen
        -dataBlock
    }
    class RTPInt {
        -id
        +MediaInit() : void
        +MediaStart() : void
        +MediaStop() : void
        +MediaCleanup() : void
        +DspMediaStopAck() : void
        +DspMediaPayload() : void
    }
    class NETCtrlInt {
        -id
        +netSendtoRtpAck() : void
        +netSendToRtcpAck() : void
    }

    IPCMsg <|-- MsgRTP
    RTPReceiverTemplate <|-- RTPImp_MsgSIP
    MsgNETCtrl <|-- NETCtrlImp_MsgNETCtrl
    MsgRTP <|-- RTPInt
    SessionList *-- Session
    RTPImp_MsgSIP --> SessionList
    SessionList --> Session
    SessionList --> NETCtrlImp_MsgNETCtrl
    NETCtrlImp_MsgNETCtrl --> MsgNETCtrl
    NETCtrlImp_MsgNETCtrl <|-- NETCtrlInt
  
```

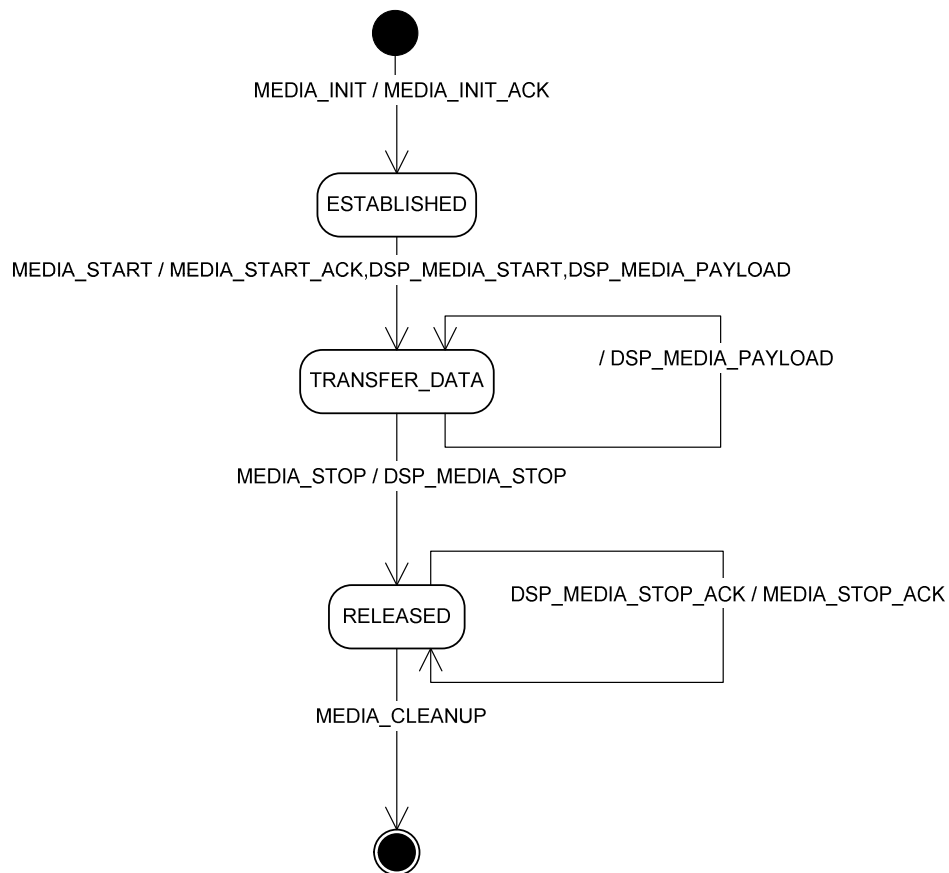
---

14

Na slici 18. i slici 19. prikazani su UML dijagrami stanja SIP i RTP podsistema, respektivno.



Slika 18. UML dijagram stanja SIP podsistema



Slika 19. UML dijagram stanja RTP podsistema