

# Opis komunikacijskega sistema na osnovi protokola ABP

## Telekomunikacijski sistem in storitev

Telekomunikacijski sistem omogoča zanesljiv prenos zaporedja znakov med uporabnikoma U1 in U2, čeprav se v kanalu lahko protokolna sporočila izgubijo.

Dva uporabnika (U1 in U2) komunicirata tako, da uporabnik U1 pošilja uporabniku U2 uporabniška sporočila v obliki znakov (enega za drugim); na ta način naj uporabnik U1 pošlje uporabniku U2 besedilo »protokoli« (znak za znakom). Uporabnik U2 naj sestavi zaporedje znakov v besedilo in ga po končanem prenosu (sprostitvi zveze) v obliki znakovnega niza pošlje v okolje komunikacijskega sistema.

Navidezni kanal med uporabnikoma implementira izvajalec storitve. Izvajalca storitve sestavljata protokolna osebka P1 in P2 ter kanal, ki povezuje protokolna osebka in skozi katerega se lahko informacijsko ali nadzorno protokolno sporočilo prenese brez napake ali se v njem izgubi. Uspešno prenešeno informacijsko protokolno sporočilo se v kanalu zakasni za 8 časovnih enot, nadzorno pa za 3. Izvajalec storitve nudi uporabnikoma storitev zanesljivega prenosa sporočil med U1 in U2. Eno uporabniško sporočilo predstavlja en sam ASCII znak.

Navidezni kanal med uporabnikoma predstavlja zveza med protokolnima osebkom, saj lahko le povezavno orientirano komuniciranje omogoča zanesljiv prenos podatkov preko izgubnega kanala. Zahtevo po vzpostavitvi ali sprostitvi zveze lahko postavi le uporabnik U1, o izidu vzpostavitve oziroma sprostitve zveze pa sta obveščena oba uporabnika. Uporabnik U2 lahko ponujeno vzpostavitev navideznega kanala sprejme ali zavrne, ob sprostitvi zveze pa te možnosti nima.

## Interakcija med uporabnikoma in izvajalcem storitve

Vzpostavitev zveze za prenos uporabniških sporočil od U1 proti U2 lahko zahteva le uporabnik U1 s primitivom `ConReq`; o predlogu za vzpostavitev zveze je U2 obveščen s primitivom `ConInd`, na kar lahko odgovori s primitivom `ConResp( 'Yes' )`, če se z vzpostavitvijo strinja, oziroma s primitivom `ConResp( 'No' )`, če se z vzpostavitvijo ne strinja. U1 je o izidu vzpostavitve obveščen s primitivom `ConConf( 'Yes' )` oziroma `ConConf( 'No' )`. O napaki pri postopku vzpostavljanja zveze je U1 obveščen s primitivom `ConConf( 'Err' )`.

Ko je zveza vzpostavljena za prenos uporabniških sporočil, lahko U1 s primitivom `DiscReq` zahteva prekinitev zveze; U2 je o tem obveščen s primitivom `DiscInd`, U1 pa o sprostitvi s primitivom `DiscConf`. Pri sproščanju zveze uporabniku U2 ni dana možnost, da bi se sprostitev strinjal ali ne, zato primitivi `DiscReq`, `DiscInd` in `DiscConf` ne nosijo nikakršnih parametrov.

V stanju vzpostavljene zveze lahko uporabnik U1 pošilja uporabniku U2 uporabniška sporočila. Uporabnik U1 zahteva prenos uporabniškega sporočila s primitivom `DataReq` (z uporabniškim

sporočilom kot parametrom). Uporabnik U2 je obveščen o prihodu uporabniškega sporočila s primitivom `DataInd` (s sporočilom kot parametrom). V primeru, da izvajalec storitve ni več zmožen prenašati uporabniških sporočil, o tem obvesti uporabnika U1 s primitivom `DiscInd`, nakar uporabnik U1 šteje zvezo za porušeno.

### **Protokol**

*Vzpostavitev zveze:* Postopek vzpostavitve zveze začne protokolni osebek P1, potem ko je od uporabnika U1 prejel primitiv `ConReq`, tako da pošlje osebk P2 nadzorno protokolno sporočilo `Conn` in hkrati sproži časovnik. Ko P2 sprejme sporočilo `Conn`, o tem obvesti uporabnika U2. Če le-ta storitev sprejme, osebek P2 pošlje svojemu partnerskemu osebk P1 nadzorno protokolno sporočilo `Accept` in preide v stanje vzpostavljene zveze. Če U2 zveze ne sprejme, P2 ne preide v stanje vzpostavljene zveze in pošlje partnerskemu osebk P1 sporočilo `Reject`. Če osebek P1 sprejme `Accept`, obvesti uporabnika U1 o uspešni vzpostavitvi zveze, ustavi časovnik in preide v stanje vzpostavljene zveze. Če sprejme `Reject`, ustavi časovnik in obvesti uporabnika U1 o neuspehu s primitivom `ConConf` s parametrom 'No'. Če se časovnik izteče, P1 ponovno pošlje sporočilo `Conn`. Če se osebk P1 petkrat zapovrstjo izteče časovnik, ker ni prejel odgovora na sporočilo `Conn`, obvesti uporabnika U1 s primitivom `ConConf` s parametrom 'Err', da je pri vzpostavljanju zveze prišlo do napake, in ne pošilja več sporočila `Conn`. Če P2 v stanju vzpostavljene zveze sprejme sporočilo `Conn`, nanj odgovori s protokolnim sporočilom `Accept`.

*Prekinitev zveze:* Postopek prekinitve zveze začne protokolni osebek P1, potem ko je od uporabnika U1 prejel primitiv `DiscReq`, tako da pošlje osebk P2 nadzorno protokolno sporočilo `Disconnect` in hkrati sproži časovnik. Ko P2 sprejme sporočilo `Disconnect`, o tem obvesti uporabnika U2, protokolnemu osebk P1 pošlje nadzorno protokolno sporočilo `Disconnected` in preide v stanje brez vzpostavljene zveze. Ko osebek P1 sprejme `Disconnected`, obvesti uporabnika U1 o prekinitvi zveze, ustavi časovnik in preide v stanje brez vzpostavljene zveze. Če se časovnik izteče, P1 ponovno pošlje sporočilo `Disconnect`. Če se osebk P1 petkrat zapovrstjo izteče časovnik, ker ni prejel odgovora na sporočilo `Disconnect`, šteje zvezo za prekinjeno, o tem obvesti uporabnika U1 s primitivom `DiscConf` in ne pošilja več sporočila `Disconnect`. Če P2 v stanju prekinjene zveze sprejme sporočilo `Disconnect`, nanj odgovori s protokolnim sporočilom `Disconnected`.

*Prenos uporabniških sporočil:* V stanju vzpostavljene zveze se med osebkom P1 in P2 prenašajo informacijska protokolna sporočila in v obratni smeri potrditve po protokolu z alternirajočim bitom (ang. Alternating Bit Protocol - ABP).

Ko protokolni osebek P1 v stanju vzpostavljene zveze sprejme od uporabnika U1 primitiv `DataReq` z uporabniškim sporočilom kot parametrom, odpošlje partnerskemu osebk P2

informacijsko protokolno sporočilo  $Ipdu$  (ki s sabo nosi uporabniško sporočilo in sekvenčno število sporočila kot parametra), sproži časovnik in čaka na nadzorno protokolno sporočilo  $Ack$  (ki prinaša številko potrditve). Informacijska protokolna sporočila štejemo po modulu 2, štetje pa začnemo s sekvenčno številko 0 (torej 0, 1, 0, 1, 0, ...). Potrditev  $Ack$  s svojim parametrom pove protokolnemu osebk  $P1$ , sprejem katere sekvenčne številke osebek  $P2$  pričakuje; potrditev  $Ack(0)$  torej pove, da  $P2$  pričakuje sprejem informacijskega protokolnega sporočila s sekvenčno številko 0, s tem pa potrjuje pravilen sprejem informacijskega protokolnega sporočila s sekvenčno številko 1; seveda velja tudi obratno. Če je  $P1$  nazadnje uspešno oddal  $Ipdu$  s sekvenčno številko 0 (kar pomeni, da je tudi že prejel njegovo potrditev  $Ack(1)$ ), bo sedaj oddal  $Ipdu$  s sekvenčno številko 1, oziroma obratno. Prvi  $Ipdu$ , ki ga  $P1$  pošlje po vzpostavitvi zveze, ima sekvenčno številko 0. Če je protokolni osebek  $P1$  oddal  $Ipdu$  št. 0, čaka na  $Ack$  št. 1, če pa je oddal  $Ipdu$  št. 1, čaka na  $Ack$  št. 0. Če  $P1$  sprejme potrditev, ki je ne pričakuje, jo zavrže. Če protokolni osebek  $P1$  sprejme tisto potrditev, ki jo čaka, ustavi časovnik in je pripravljen na oddajo novega  $Ipdu$ -ja (torej čaka na primitiv  $DataReq$ ). Medtem, ko  $P1$  čaka na potrditev enega informacijskega protokolnega sporočila, ne more poslati naslednjega (v tem času prispeli primitivi  $DataReq$  morajo torej počakati v čakalni vrsti). Če se med čakanjem na potrditev časovnik izteče,  $P1$  ponovno odda isti  $Ipdu$  kot zadnjikrat (torej tistega, katerega potrditev čaka) z isto sekvenčno številko in sproži časovnik. Če se časovnik petkrat zapovrstjo izteče, osebek  $P1$  predpostavlja, da ne more več komunicirati z osebkom  $P2$ , šteje zvezo za prekinjeno in o tem obvesti uporabnika s primitivom  $DiscInd$ .

Protokolni osebek  $P2$  čaka na sprejem informacijskih protokolnih sporočil. Po vzpostavitvi zveze čaka na  $Ipdu$  št. 0. Po uspešnem sprejemu  $Ipdu$ -ja št. 0 čaka na  $Ipdu$  št. 1, po uspešnem sprejemu  $Ipdu$ -ja št. 1 čaka na  $Ipdu$  št. 0. Uspešen sprejem  $Ipdu$ -ja pomeni, da je sprejemnik brez napake sprejel tisti  $Ipdu$ , ki ga pričakuje. Po sprejemu  $Ipdu$ -ja, ki ne vsebuje napake (ne glede na to, ali je  $P2$  sprejem tega  $Ipdu$ -ja pričakoval ali ne), pošlje partnerskemu osebk potrditev  $Ack$  s številko informacijskega protokolnega sporočila, ki ga pričakuje. Če sprejemnik brez napake sprejme tisti  $Ipdu$ , ki ga pričakuje, s primitivom  $DataInd$  preda vsebovano uporabniško sporočilo uporabniku. Protokolni osebek  $P2$  zavrže  $Ipdu$ , ki ga sprejme, ko ga ne pričakuje.