

Komunikacijski protokol

Andrej Bagon
RSM

Komunikacijski protokol

- Protokol je nabor pravil in postopkov, ki urejajo, vodijo in upravljajo prenos informacij med ljudmi, napravami in procesi.
- Standard je eden ali več formalno ali neformalno sprejetih protokolov, ki jih razumejo vsi zainteresirani ljudje, naprave in procesi v želji po vzpostavitvi medsebojne komunikacije.

Komunikacijski protokol

- Kakor smo ugotovili že v prejšnjih poglavjih, je komunikacijski protokol poleg pristopne točke osnovni gradnik večplastne arhitekture. V bistvu smo se večine lastnosti N-protokola dotaknili že pri dosednji obravnavi, v tem poglavju pa bomo izločili in podrobno obdelali predvsem mehanizme, ki od plasti niso odvisni in ki jih srečamo pri vseh komunikacijskih protokolih. Sistem ima toliko vrst komunikacijskih protokolov, kolikor plasti vsebuje njegova arhitektura. Skupne lastnosti N-protokolov lahko razdelimo v dve skupini, ki zaznamujeta njegove lastnosti.

Komunikacijski protokol

- Prva skupina vsebuje mehanizme, ki so namenjeni odkrivanju in odpravljanju napak, ki jih pri prenosu PPE odkrije sprejemnik, druga skupina mehanizmov pa skrbi za kontrolo podatkov (PPE) med entitetama, kar lahko bistveno vpliva na izkoriščenost in uporabnost sistema. Brez kontrole pretoka podatkov lahko sprejemnika na primer poplavi velika količina podatkov, ki je ne more obdelati.

Komunikacijski protokol

- Najprej bomo obravnavali mehanizme odkrivanja in odpravljanja napak. Pri tem gre za to, da ob ugotavljeni napaki pri sprejemniku le-ta od oddajnika lahko zahteva odpravo določene napake – na primer tako, da pošlje sporočilo: "PPE x mi pošlji še enkrat".

Mehanizmi potrjevanja

- Če se strinjamo s tem, da je potrjevanje namenjeno odpravljanju napak, je prav, da povemo, kakšne napake se lahko zgodijo v sistemu. Glede na funkcionalno razmejenost plasti ni težko razumeti, da so tipi napak odvisni od protokolarne plasti, na kateri deluje določen N-protokol. Poglejmo si najbolj tipične napake, ki jih srečujemo na posameznih plasteh.

Mehanizmi potrjevanja

- Na **fizični plasti** lahko opazujemo stanje konektorja (vključen ali izključen) in o statusu fizične povezave obveščamo sistem in seveda tudi končnega uporabnika. Napake so največkrat posledica okvar ali motenj na strojni opremi.

Mehanizmi potrjevanja

- Na **povezavni plasti** odkrivamo napake, ki se zgodijo pri prenosu okvirja po neidealnem prenosnem mediju. Večina nas je že slišala za kontrolo parnosti (paritetno zaščito, parity check), ki omogoča sprejemniku identificirati PPE, v kateri se je na prenosnem mediju pojavila sprememba – napaka. Seveda bo sprejemnik v takem primeru ustrezno reagiral – zahteval bo na primer, da se določena PPE pošlje še enkrat.

Mehanizmi potrjevanja

- Na **omrežni** in **transportni** plasti se odkrivanje napak omeji predvsem na kontrolo zaporedja sprejetih PPE. To pomeni, da se ugotavlja, ali kak paket manjka, ali gre za večkrat sprejet isti paket, ali pa se je spremenil vrstni red PPE pri sprejemniku.
- Na višjih, **informacijskih plasteh** gre predvsem za odkrivanje vsebinskih, semantičnih nepravilnosti uporabniških podatkov.

Mehanizmi potrjevanja

- Ne glede na plast je očitno, da če se želi odpraviti odkrita napaka, morata biti o tem obveščeni oziroma pri tem sodelujeta obe entiteti. V dialogu entitetnega para je ena entiteta oddajnik druga pa sprejemnik. **Oddajnik** je entiteta, ki oddaja pakete – PPE in sprejema potrditve, **sprejemnik** pa je entiteta, ki sprejema PPE in oddaja potrditve. Tak način komuniciranja ustreza našemu opisu tako imenovane **povezane storitve**, kakor smo jo opredelili v prejšnjem poglavju.

Mehanizmi potrjevanja

- Le nekaj vrst napak je takih, da jih pod določenimi pogoji sprejemnik lahko odpravi avtonomno: večkratni sprejem iste PPE lahko sprejemnik reši tako, da take pakete identificira s pomočjo zaporedne številke in nato kopije zavrže, drug tipičen primer avtonomnega odpravljanja napak pa je urejanje vrstnega reda sprejetih PPE na osnovi zaporednih števil – seveda pa za to potrebuje vgrajeno sposobnost za sortiranje.

Mehanizmi potrjevanja

- Na nižjih plasteh gre po pravilu za odpravljanje napak na nivoju posameznega PPE (paketa), medtem ko na višjih nivojih, kjer je poleg sintakse pomembna predvsem vsebina aplikacijsko-uporabniških podatkov, napake odpravljamo na nivoju sporočila.

Mehanizmi potrjevanja

- **Sporočilo** je zaporedje protokolarnih enot – PPE, ki jih višje plasti pretvorijo v aplikacijske podatkovne strukture. Ne glede na nivo obravnavanja napak (PPE, sporočilo) govorimo o **pozitivni (ACK – acknowledge)** in **negativni potrditvi (NACK – not acknowledge)**. Ti dve sporočili sta namenjeni kontroli pravilnosti/nepravilnosti izvajanja določene povezane storitve.

Mehanizmi potrjevanja

- Sprejemnik pošlje sporočilo ACK, ko spozna sprejeti paket/sporočilo kot neoporečen, v nasprotnem primeru pa odgovori s sporočilom NACK. Na osnovi teh dveh sporočil (ACK, NACK) lahko **oddajnik** ustrezno ukrepa oziroma v sodelovanju s sprejemnikom odpravi napako. Zgornji pojmi so tudi **terminologija**, ki jo bomo uporabljali v naslednjih odstavkih, ko si bomo ogledali različne različice mehanizmov za odpravljanje napak.

Mehanizmi potrjevanja

- Seveda je smiselno tudi to, da mehanizmi potrjevanja zagotavljajo avtomatsko in za vsako plast neodvisno odpravljanje napak. To pomeni, da je odpravljanje napak za preostale plasti, predvsem pa za uporabnika, **transparentno**. Na osnovi povedanega lahko postavimo splošno definicijo mehanizma za potrjevanje:
- ***Mehanizem potrjevanja*** je opredeljen s pravili, ki določajo, kako oddajnik pošilja PPE in kako sprejemnik odgovarja s potrditvima ACK in NACK, kar zagotavlja idealen prenos PPE na nivoju posamezne plasti.

Mehanizmi potrjevanja

- Najprej si oglejmo grobo razčlenitev mehanizmov za potrjevanje, ki je zasnovana na dinamiki pošiljanja PPE. Ločimo dva osnovna načina komunikacije med oddajnikom in sprejemnikom, in sicer govorimo o sprotnem potrjevanju in o tekočem pošiljanju. V nadaljnji obravnavi bomo eksplicitno govorili zgolj o PPE, vendar pa vse ugotovitve veljajo tudi za sporočila.

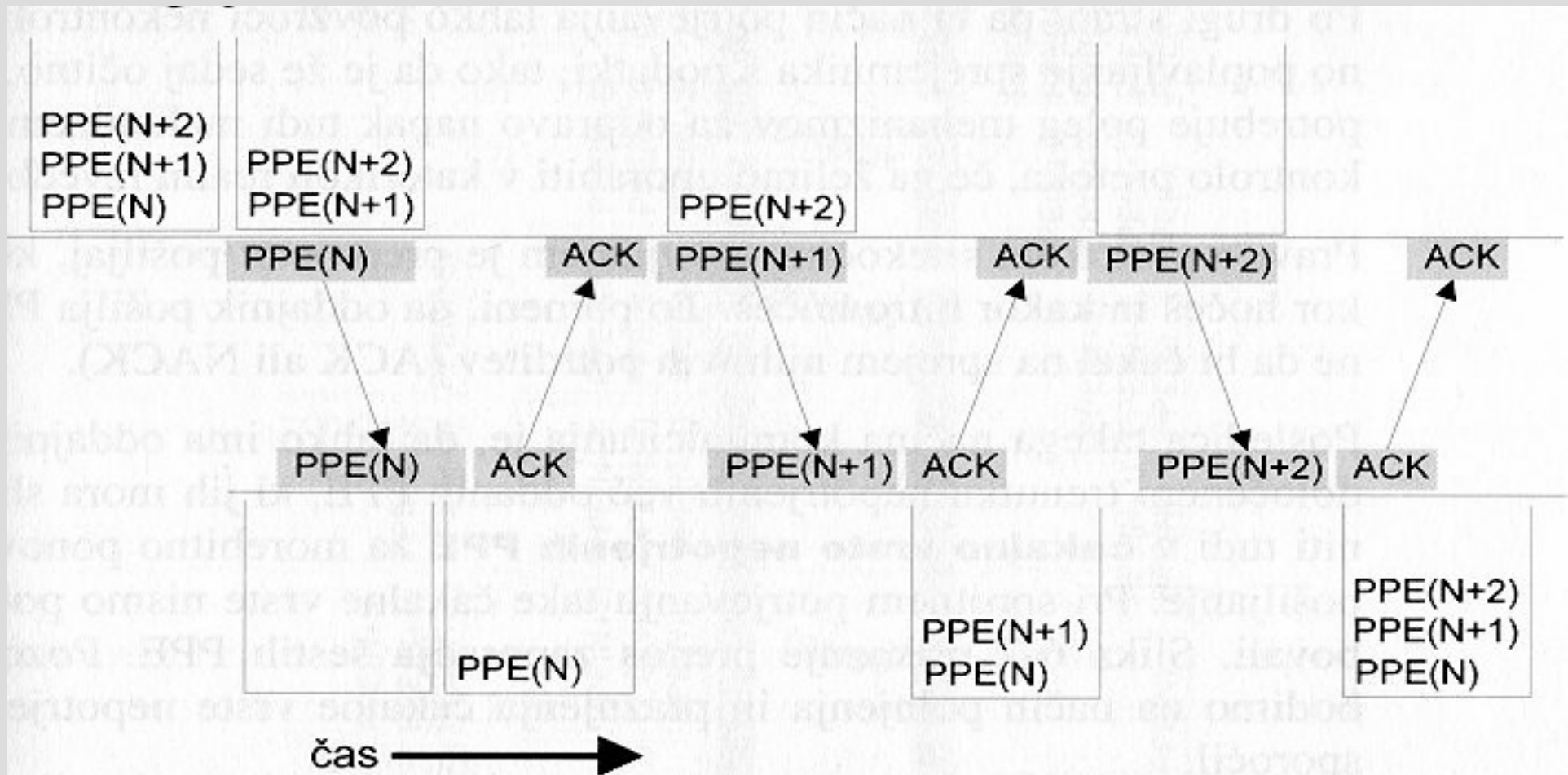
Sprotno potrjevanje

- **Sprotno potrjevanje** je povezana storitev, ko oddajnik odda naslednji paket PPE šele takrat, ko sprejme potrditev za predhodno poslano PPE. Zaradi take lastnosti tak način potrjevanja imenujemo tudi **pošlji in čakaj**.

Sprotno potrjevanje

- Dobra lastnost takega načina potrjevanja je, da se pri pošiljanju ne more zgoditi **poplavljanje sprejemnika** s podatki, saj sprejemnik lahko podatek zavrne ali prekliče, lahko pa preprosto ne reče nič, s čimer se tok komuniciranja seveda ustavi. Slabost sprotnega potrjevanja pa je počasnost protokola in majhna izkoriščenost prenosnega kanala.

Sprotno potrjevanje



Slika 6-1: Sprotno potrjevanje

Sprotno potrjevanje

- Na sliki je lepo viden osnovni princip. Opazimo lahko tudi, da prenosni kanal v obeh smereh večji del trajanja komunikacije ni izkoriščen. Predstavljene so tudi vsebine oddajnih in sprejemnih čakalnih vrst ter trenutni, ko se spremeni njihova vsebina. Očitno je, da se PPE iz oddajne vrste izbriše šele takrat, ko se dobi pozitivna potrditev.

Tekoče pošiljanje

- **Tekoče pošiljanje** je mehanizem, pri katerem pošiljatelj oddaja PPE, ne da bi čakal na potrditev predhodno oddanih PPE. Kakor bomo videli na sliki, je dobra lastnost protokola s tekočim pošiljanjem boljša izkoriščenost prenosnega kanala, saj smo pri sprotnem potrjevanju videli, da oddajnik večji del časa miruje – ne oddaja, ker čaka na sprejem potrditve.

Tekoče pošiljanje

- Po drugi strani pa ta način potrjevanja lahko povzroči nekontrolirano poplavljanje sprejemnika s podatki, tako da je že sedaj očitno, da potrebuje poleg mehanizmov za odpravo napak tudi mehanizem za kontrolo pretoka, če ga želimo uporabiti v katerikoli realni izvedbi.
- Pravilo protokola s tekočim pošiljanjem je preprosto: pošiljaj, kolikor hočeš in kakor hitro hočeš. To pomeni, da oddajnik pošilja PPE, ne da bi čakal na sprejem njihovih potrditev (ACK ali NACK).

Tekoče pošiljanje

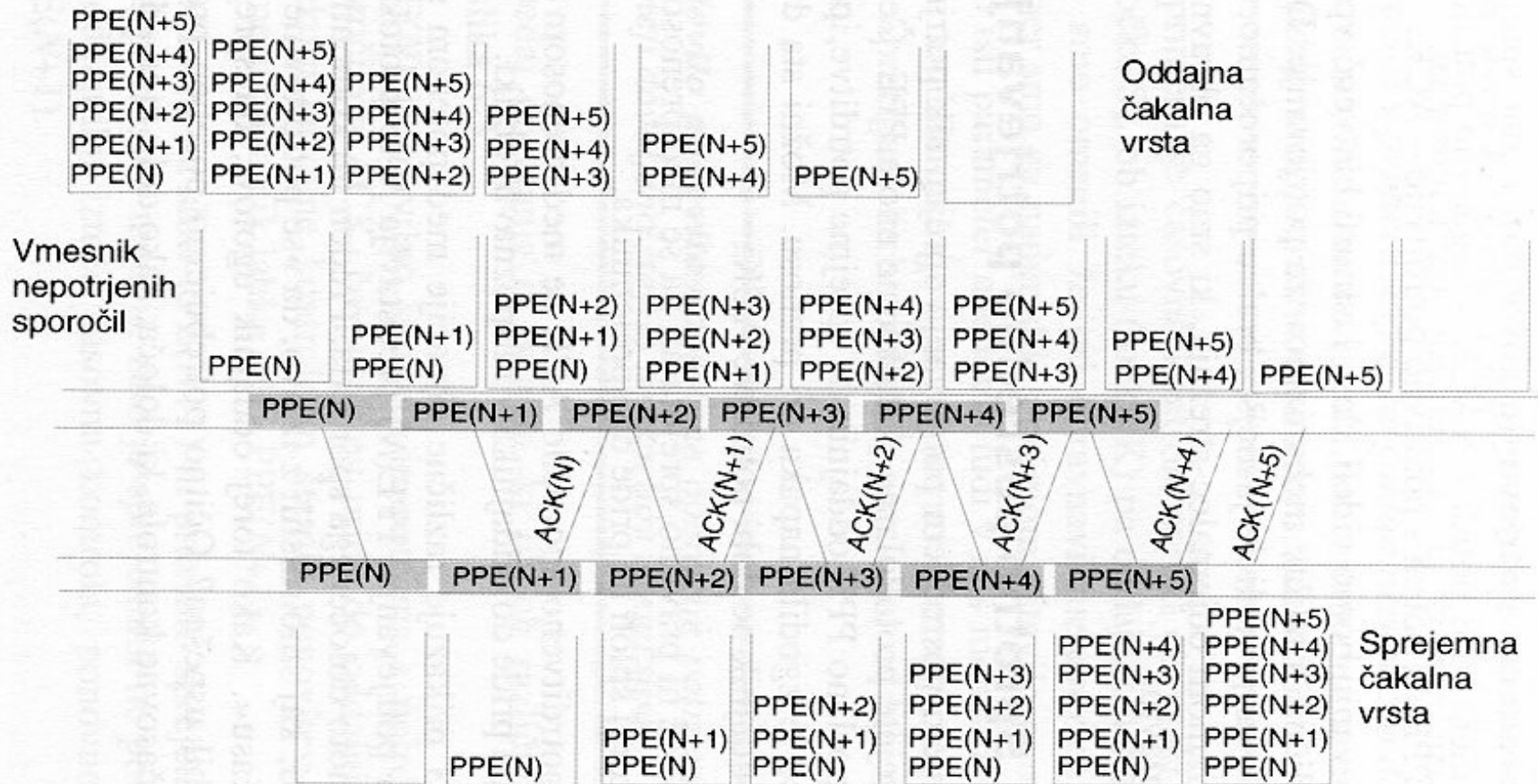
- Posledica takega načina komuniciranja je, da lahko ima oddajnik v določenem trenutku nepotrjenih več oddanih PPE, ki jih mora shraniti tudi v **čakalno vrsto nepotrjenih PPE** za morebitno ponovno pošiljanje. Pri sprotnem potrjevanju take čakalne vrste nismo potrebovali. Slika prikazuje prenos zaporedja šestih PPE. Pozorni bodimo na način polnenja in praznenja čakalne vrste nepotrjenih sporočil.

Tekoče pošiljanje

- Posledica tekočega pošiljanja PPE je tudi ta, da morajo potrditve vsebovati zaporedno številko paketa, na katerega se nanašajo. To iz navedenih primerov seveda ni neposredno jasno, ker so izbrane zgolj sekvence, kjer se niso pojavile napake. Ko bomo obravnavali tudi take sekvence, bomo lahko mehanizme potrjevanja razčlenili še nekoliko podrobneje: na posredno in neposredno potrjevanje, in sicer ne glede na način potrjevanja.

Tekoče pošiljanje

Slika 6-2: Tekoče pošiljanje



Mehanizmi potrjevanja

- O **posrednem potrjevanju** govorimo, če sprejemnik potrjuje zgolj pravilno sprejete PPE, na napake pa lahko sklepamo, kadar pozitivne potrditve ne dobimo.
- O **neposrednem potrjevanju** govorimo, kadar sprejemnik nepravilno sprejeto PPE potrdi z NACK – negativno potrditvijo, pravilno sprejeto PPE pa potrdi s pozitivno potrditvijo – ACK.

Sprotno posredno potrjevanje

- Pri tem protokolu potrjujemo samo pravilno sprejete PPE.
- Če v določenem času za poslano PPE oddajnik ne sprejme potrditve, posredno razume, da je prišlo do napake pri prenosu.
- Dve možni situaciji, ko oddajnik ne dobi potrditve PPE:
 - PPE ni pravilno sprejeta ali pa se med prenosom izgubi, torej sploh ne pride do sprejemnika
 - Potrditveno sporočilo ACK se med prenosom izgubi ali pa pride do oddajnika v neprepoznavni obliki.

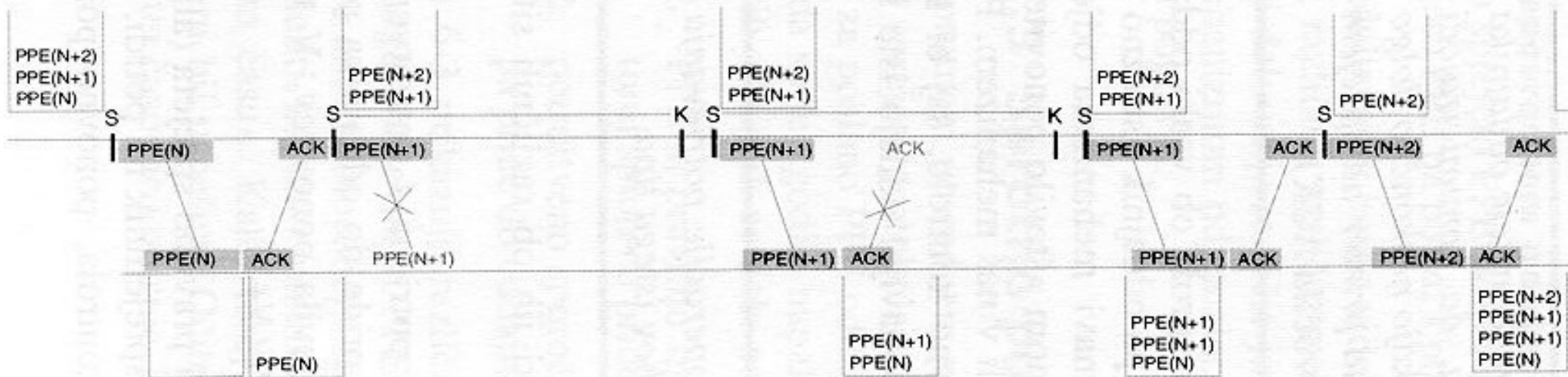
Sprotno posredno potrjevanje

- **Časovna kontrola** je interni mehanizem, ki zagotavlja, da komunicirajoči procesi v primeru napak ne čakajo neskončno dolgo na odziv, ki se ne bo zgodil (na primer na sprejem izgubljenega potrditvenega sporočila ACK).
- Časovna kontrola se sproži ob vsaki oddaji PPE (točka S), ob izteku (točka K) pa oddajnik ustrezno ukrepa.

Sprotno posredno potrjevanje

- Poleg mehanizma časovne kontrole oddajnik potrebuje tudi parameter **število potrditev** določene operacije, kar omeji dovoljeno število neuspehov istega tipa.
- Ta parameter je potreben iz enakih razlogov kot časovna kontrola, saj bi sicer na primer ob “pretrgani žici” oddajnik ponavljal oddajo istega PPE do onemoglosti.

Sprotno posredno potrjevanje



Sprotno posredno potrjevanje

- Opis dogajanja:
 - PPE(N) je pravilno sprejeta in potrjena s sporočilom ACK. Če potrditev pride do oddajnika pred iztekom časovne kontrole, lahko ta odda PPE(N+1).
 - Oddajnik pošlje PPE(N+1). PPE(N+1) ni pravilno sprejeta (ali pa se izgubi med prenosom), zato je sprejemnik ne potrdi. Ko poteče časovna kontrola, oddajnik ponovno pošlje PPE(N+1).
 - V drugem poizkusu je paket PPE(N+1) pravilno sprejet in ga zato sprejemnik potrdi s sporočilom ACK.

Sprotno posredno potrjevanje

- Ker se $ACK(N+1)$ med prenosom izgubi ali popači, ga oddajnik ne sprejme in po izteku časovne kontrole ponovno pošlje $PPE(N+1)$. To se lahko zgodi tolikokrat, kolikor neuspešnih poizkusov dovoljuje parameter ponavljanja. Glede na to, da lahko pride do večkratnega sprejema istega paketa, mora imeti sprejemnik sposobnost zavračanja kopij ali pa dovolj velik vmesni pomnilnik za njihovo shranjevanje. $PPE(N+1)$ je bil zato 2x pravilno sprejet.

Sprotno posredno potrjevanje

- Časovno kontrolo pa ne uporablja le oddajnik, ampak tudi sprejemnik. Če oddajnik preneha z oddajanjem, sprejemnik pa o tem ni obveščen, bi namreč v nasprotnem primeru lahko čakal na sprejem sporočila, ki ga nikoli ne bo dobil.

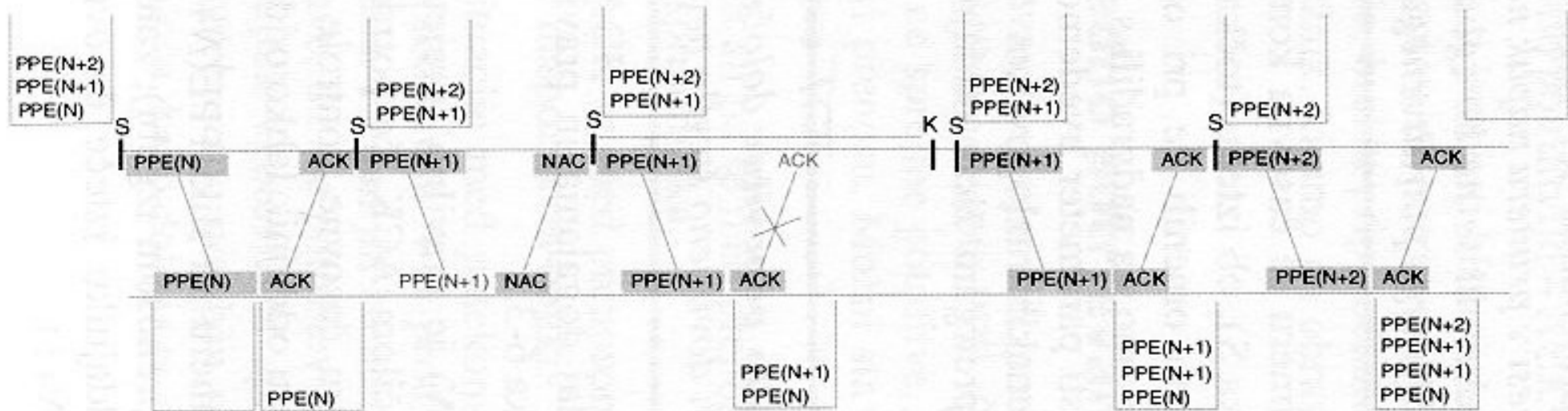
Sprotno posredno potrjevanje

- Protokol s posrednim sprotnim potrjevanjem za svoje delovanje potrebuje naslednje parametre:
 - časovno kontrolo oddajnika
 - število ponovnih oddaj podatkovnih paketov
 - časovno kontrolo sprejemnika ob izpadu oddajnika
 - število ponovnih oddaj potrditev
- Zaporedna številka podatkovnega paketa je potrebna zato, da lahko protokol zazna večkrat sprejete pakete. Tudi kontrolna sporočila bi lahko vsebovala zaporedno številko paketa, a to nima posebnega smisla.

Sprotno neposredno potrjevanje

- Protokol se od sprotnega posrednega potrjevanja razlikuje v toliko, da v primeru napake sprejemnik odda obvestilo – negativno potrditveno sporočilo – NACK. Tako oddajniku ni treba čakati do izteka časovne kontrole.

Sprotno neposredno potrjevanje



Sprotno neposredno potrjevanje

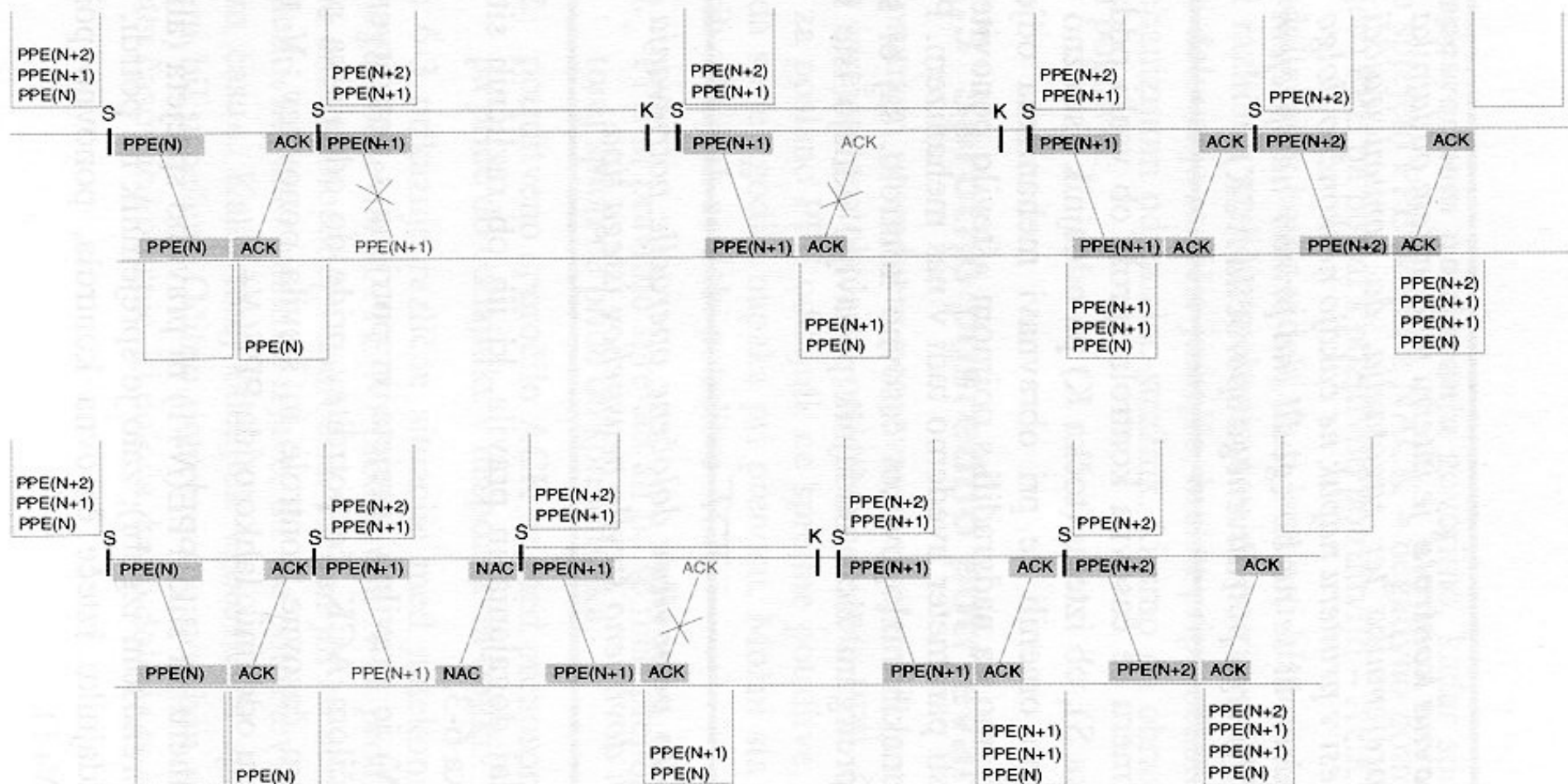
- Opis dogajanja:
 - Dokler ni napak, se protokol obnaša enako kot protokol s posrednim sprotnim potrjevanjem. Enako se obnaša tudi, če se izgubi PPE ali ACK.
 - Posredna izvedba ne razlikuje med primeroma, ko se PPE izgubi in ko se med prenosom popači. V neposredni izvedbi pa nastopi razlika v primeru popačenja PPE(N+1). Sprejemnik ne molči, temveč takoj pošlje oddajniku negativno potrditveno sporočilo NACK, s čimer pospeši dogajanje. Oddajniku v tem primeru ni potrebno čakati na iztek časovne kontrole, temveč nemudoma ponovno pošlje PPE(N+1)

Sprotno neposredno potrjevanje

- Če se ACK ali NACK izgubita med prenosom, oddajnik po izteku časovne kontrole ponovno pošlje PPE. V primeru izgubljene potrditve ACK pride kasneje do večkratnih sprejemov istega sporočila.
- Enako kot pri posrednem potrjevanju morajo biti določeni isti parametri oddajnika in sprejemnika, kar velja tudi za vse primere, ki jih bomo obravnavali v nadaljevanju.

Sprotno posredno in neposredno potrjevanje

Slika 6-3: Sprotno potrjevanje (posredno in neposredno)

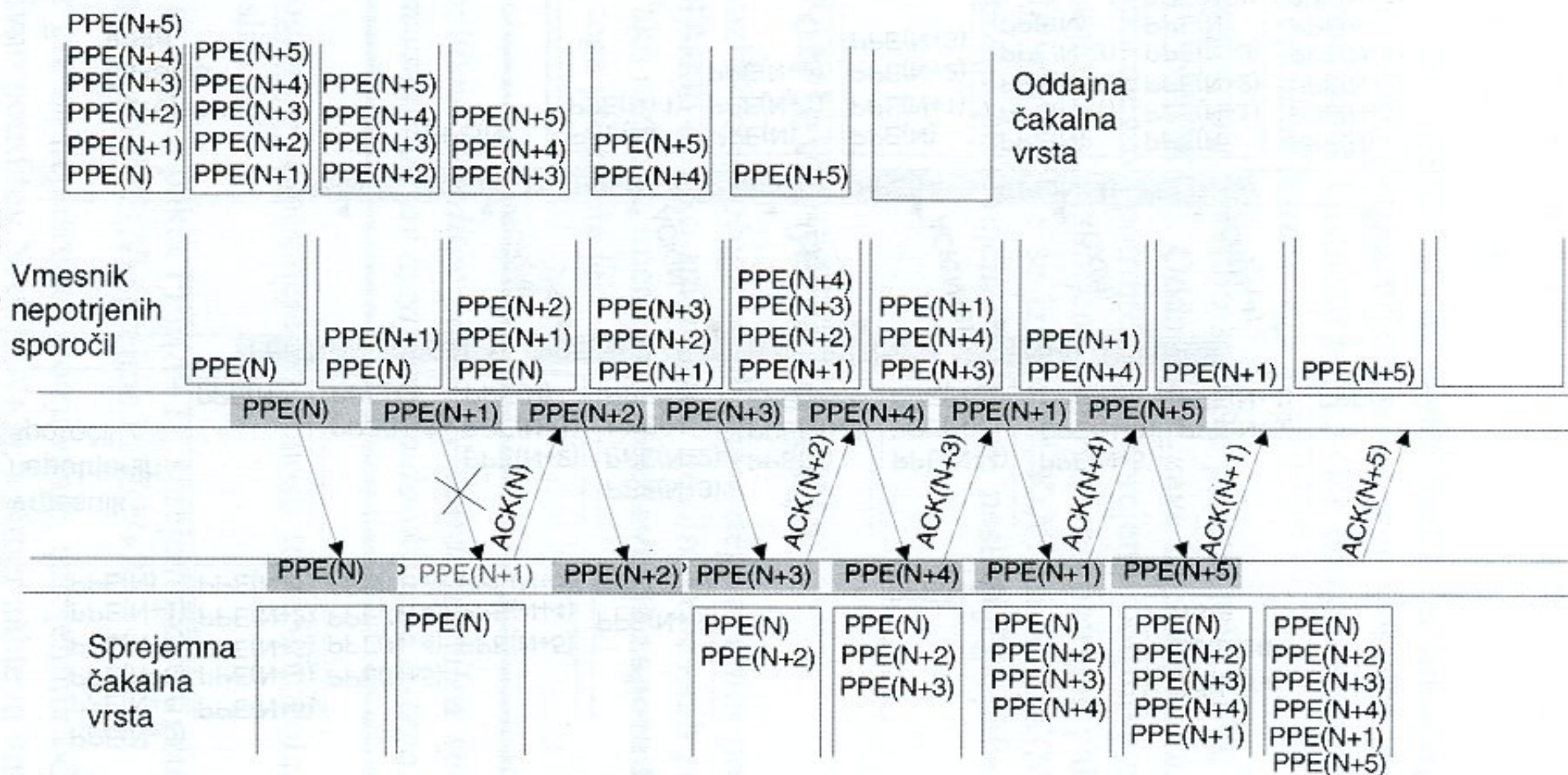


Tekoče pošiljanje – posredno potrjevanje

- Podobno kot mehanizem sprotnega posrednega potrjevanja tudi mehanizem posrednega potrjevanja s tekočim pošiljanjem ne loči med izgubo in nepravilnim sprejemom PPE ali ACK.

Tekoče pošiljanje – posredno potrjevanje

Slika 6-4: Izgubljeni paket



Tekoče pošiljanje – posredno potrjevanje

- Če sprejemnik sprejme popačeno $PPE(N+1)$ ali pa se ta med procesom izgubi, oddajnik iz sekvence potrditvenih sporočil posredno ugotovi, da se je zgodila napaka pri paketu $PPE(N+1)$. V našem primeru potrditvi $ACK(N)$ namreč ne sledi paket $ACK(N+1)$, temveč kar $ACK(N+2)$ in $ACK(N+3)$.
- Ko oddajnik namesto pričakovanega $ACK(N+1)$ sprejme potrditev $ACK(N+2)$, mu je jasno, da $PPE(N+1)$ ni bil sprejet. Zato ponovno pošlje $PPE(N+1)$ – v primeru na sliki namesto $PPE(N+5)$.

Tekoče pošiljanje – posredno potrjevanje

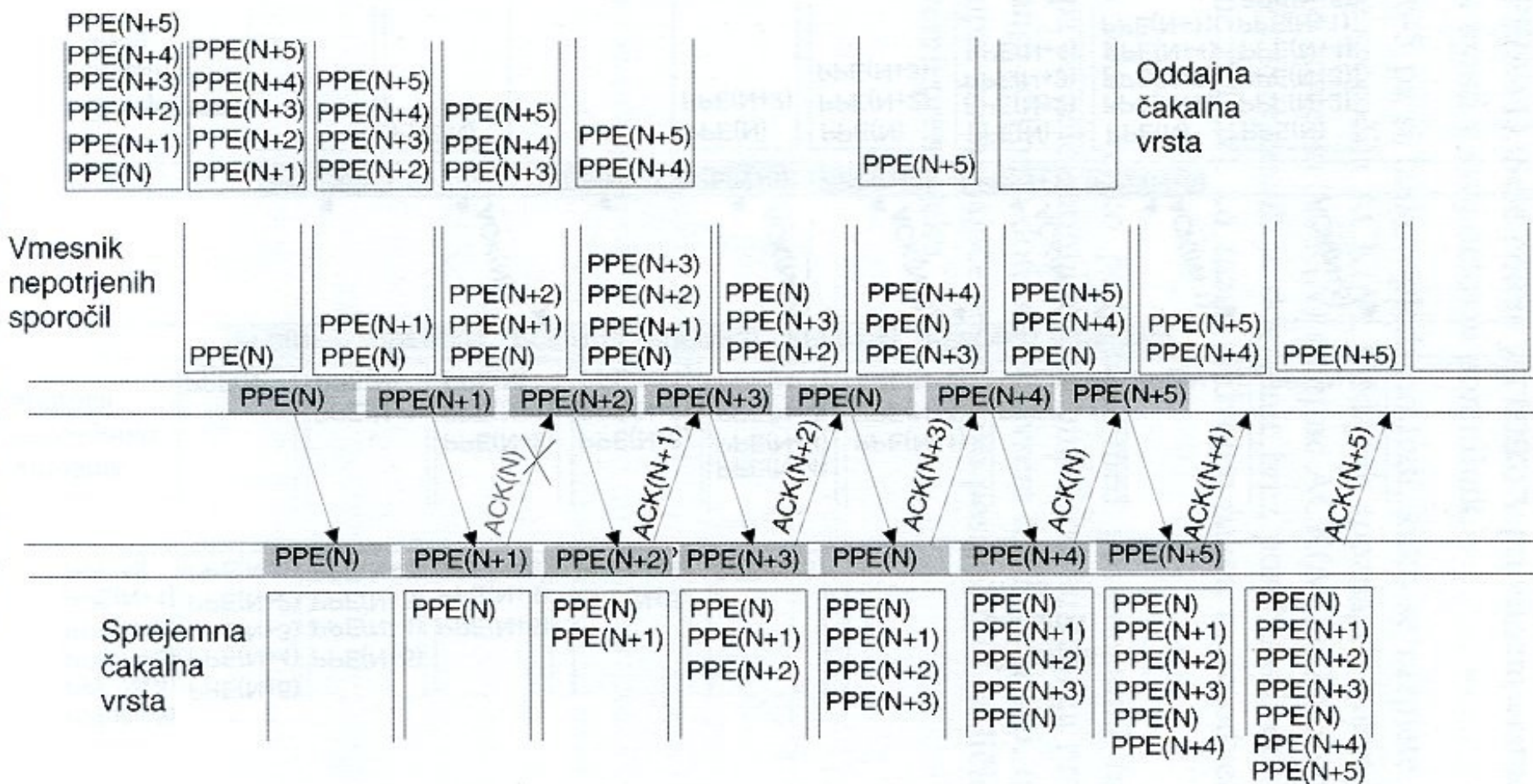
- Sprejemnik ves čas shranjuje pravilno sprejete pakete. Naj opozorimo na nepravilen vrstni red sprejetih paketov PPE, kar se zgodi zaradi napačnega sprejema PPE($N+1$). Vrstni red PPE v vmesnem pomnilniku je: N , $N+2$, $N+3$, $N+4$, $N+1$, $N+5$.
- Sprejemnik mora sporočila obdržati v začasni čakalni vrsti vsaj do trenutka, ko pravilno sprejme manjkajočo PPE($N+1$). Šele potem jih lahko v pravilnem vrstnem redu uvrsti v svoj delovni pomnilnik.

Tekoče pošiljanje – posredno potrjevanje

- Na naslednji sliki pa si lahko ogledamo tudi, kako se razplete izguba ali popačenje sporočila $ACK(N)$. Oddajnik spozna, da paket $PPE(N)$ ni potrjen, ko namesto $ACK(N)$ sprejme $ACK(N+1)$. Zato vnovič odda $PPE(N)$. Za sprejemnika je dogajanje brez posebnosti, dokler vnovič ne sprejme $PPE(N)$: ko zazna podvojen paket, mora odvečne kopije zavreči.

Tekoče pošiljanje – posredno potrjevanje

Slika 6-5: Izgubljena potrditev



Tekoče pošiljanje – neposredno potrjevanje

- Ta protokol imenujemo tudi osnovna različica tekočega pošiljanja z neposrednim potrjevanjem, saj je najpogostejša izvedba potrditvenega protokola. Lastnosti izpeljimo iz do sedaj že obravnavanih tem.
- Sprejemnik potrjuje pravilno sprejete pakete z ACK(N), nepravilno sprejete pa z NACK(N). Oddajnik se zave, izgube paketa, ko ugotovi, da manjka njegova potrditev – ob sprejemu potrditve naslednjega paketa iz zaporedja.

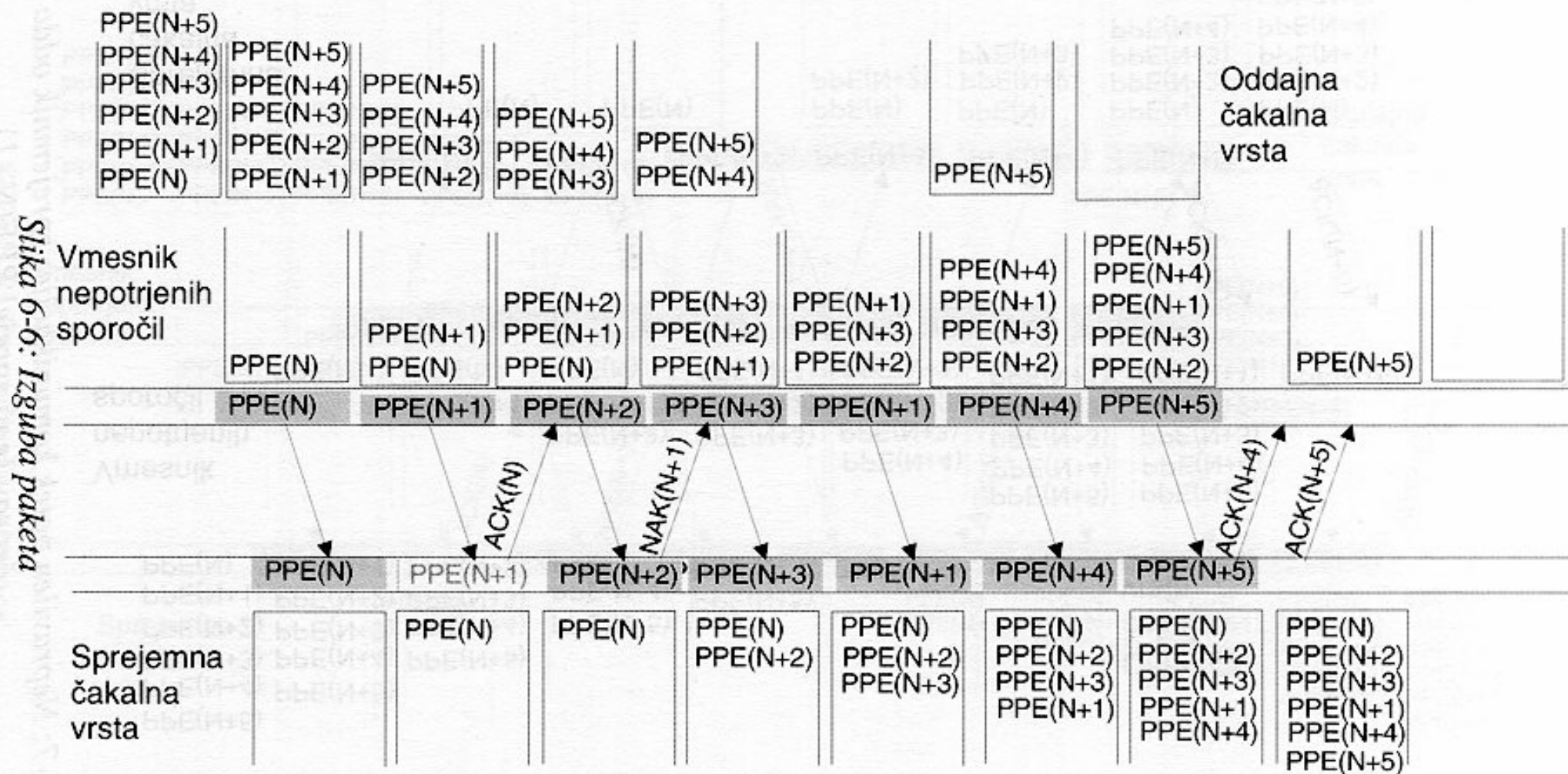
Tekoče pošiljanje – neposredno potrjevanje

- Izgubljene in negativno potrjene pakete oddajnik odpošlje vnovič. Če se izgubi ACK(N) ali NACK(N), se paket vnovič odpošlje po sprejemu potrditve naslednjega paketa iz zaporedja.

Tekoče pošiljanje – potrjevanje zaporedja

- Osnovno različico tekočega pošiljanja z neposrenim potrjevanjem lahko izboljšamo in tako zmanjšamo število prenesenih potrditvenih paketov. Tako različico imenujemo potrjevanje zaporedja.
- Osnovno pravilo tega protokola je: *Pozitivna potrditev paketa N potrdi pravilen sprejem vseh paketov iz zaporedja do vključno N-tega.*

Tekoče pošiljanje – potrjevanje zaporedja



Tekoče pošiljanje – potrjevanje zaporedja

- Če se izgubi paket $PPE(N+1)$, sprejemnik ugotovi, ko za paketom $PPE(N)$ sprejme $PPE(N+2)$
- Ker gre za neposredno potrjevanje, sprejemnik odda negativno potrditev $NACK(N+1)$, s čimer zahteva vnovično pošiljanje manjkajočega paketa. Medtem oddajnik odda $PPE(N+3)$.
- Ko oddajnik sprejme $NACK(N+1)$, vnovič odda $PPE(N+1)$, nakar nadaljuje s $PPE(N+4)$

Tekoče pošiljanje – potrjevanje zaporedja

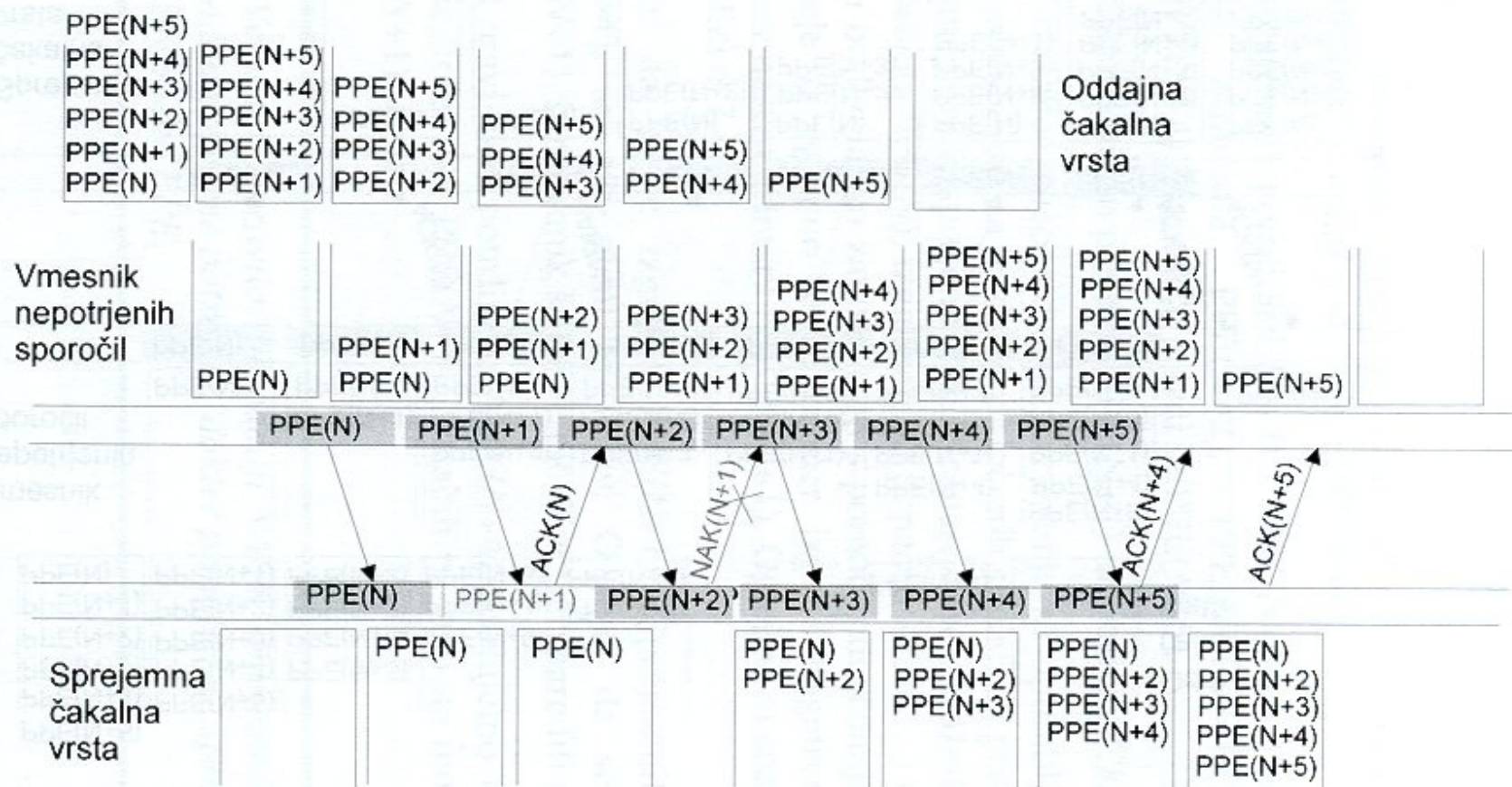
- Sprejemnik potrdi $PPE(N+4)$ z $ACK(N+4)$ in s tem celotno zaporedje $PPE(N+2)$, $PPE(N+3)$, $PPE(N+1)$, $PPE(N+4)$. Nazadnje potrdi še $PPE(N+5)$.
- Zakaj takoj po prejemu $PPE(N+1)$ ne pošlje $ACK(N+3)$

Tekoče pošiljanje – potrjevanje zaporedja

- Različica potrjevanje zaporedja se uporablja dokaj pogosto, njena slabost pa je, da ob določenih situacijah pride do napačnega vrstnega reda sprejetih PPE.
- Primer izgube negativne potrditve NACK – NACK(N+1) v primeru.

Tekoče pošiljanje – potrjevanje zaporedja

Slika 6-7: Nepravilen potek komunikacije: sprejemnik odda ACK(N+4), čeprav še ni sprejel PPE(N+1).



Tekoče pošiljanje – potrjevanje zaporedja

- Izgubi se PPE(N+1). Sprejemnik je poslal ACK(N+4), saj pri njemu ni težav. O tem, da se je izgubil NACK(N+1), sprejemnik nič ne ve. Kje je napaka?
- Protokol pravi, da s potrditvijo ACK(N+4) potrjujemo vse predhodne pakete, torej tudi PPE(N+1). Pozabili pa smo na možnost, da se NACK(N+1) izgubi.
- Velja torej pravilo: po oddaji NACK sprejemnik ne odda pozitivne potrditve, dokler ne sprejme zavrnjene PPE.

Tekoče pošiljanje – ponavljanje zaporedja

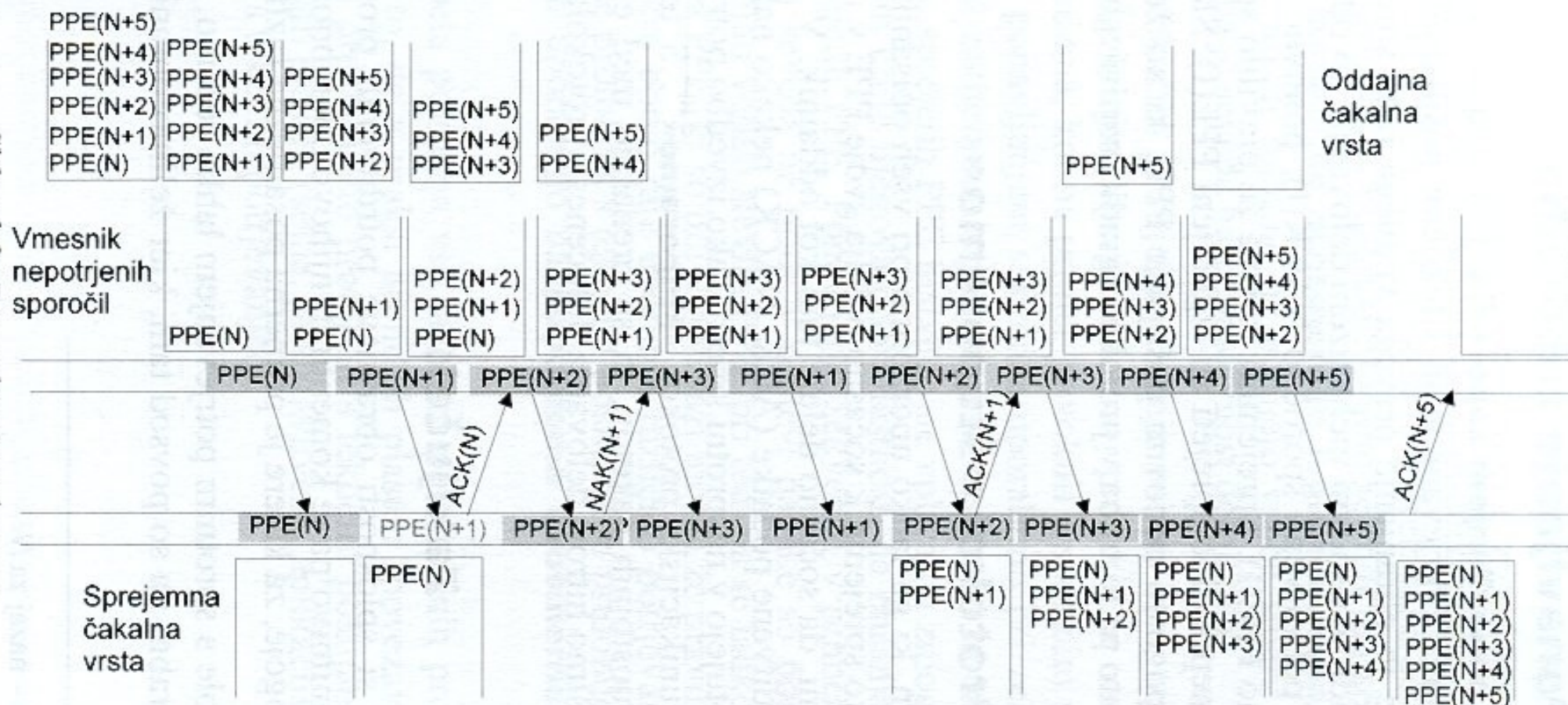
- Protokol s ponavljanjem zaporedja (go-back N) je še ena od različic protokola z neposrednim potrjevanem in s tekočim pošiljanjem.
- Dobra stran te različice je ohranjanje pravega vrstnega reda sprejetih PPE.
- Na sliki je prikazan osnovni mehanizem delovanja: ko oddajnik sprejme negativno potrditveno sporočilo NACK(N), ponovno odda vse PPE do vključno PPE(N).

Tekoče pošiljanje – ponavljanje zaporedja

- Sprejemnik zavrže že pravilno sprejete PPE, ki so sledile nepravilno sprejeti ali izgubljeni PPE, ki so že bile pravilno sprejete, zato pa ni treba v protokol vgraditi sortirnega algoritma.

Tekoče pošiljanje – ponavljanje zaporedja

Slika 6-8: Ponavljanje sekvence



Tekoče pošiljanje – protokol “štuporamo”

- Mehanizem, ki ga lahko uporabimo pri vseh opisanih osnovnih primerih je, ko sprejemnik sočasno pošilja svoje PPE v nasprotno smer – to pomeni, da sočasno deluje tudi kot oddajnik.
- V takem primeru lahko potrditvene podatke (ACK, NACK) nekako naložimo v glave PPE, ki potujejo v nasprotni smeri.
- Tako izvedbo potrjevanja imenujemo komunikacijski protokol “štuporamo” (piggy back acknowledgement).

Tekoče pošiljanje – protokol “štuporamo”

- S tem se zmanjša število samostojnih paketov, ki se prenašajo med entitetama in s tem pospešimo hitrost delovanja določenega protokola.

Zaključek – mehanizmi potrjevanja

- Za protokole s sprotnim potrjevanjem lahko rečemo, da so zelo počasni, uporabni pa so povsod tam, kjer želimo ohraniti **fazo oddajnika in sprejemnika**. Ohraniti fazo oddajnika in sprejemnika pomeni, da oddajnik ne prehiteva sprejemnika z oddajami. Posledično to pomeni, da se ne more zgoditi, da bi podatki poplavalili sprejemnik. Tak protokol ima torej vgrajeno kontrolo pretoka.

Zaključek – mehanizmi potrjevanja

- Tak protokol se zadovolji z **izmenično dvosmernim logičnim kanalom (half duplex)**, saj PPE in potrditve nikoli sočasno ne potujejo v obe smeri.
- Prednost teh protokolov pa je seveda tudi enostavnost delovanja.

Zaključek – mehanizmi potrjevanja

- Protokoli s tekočim pošiljanjem imajo vsi po vrsti težave s podatkovnim poplavljanjem sprejemnika, saj protokol ne vsebuje varnostnega mehanizma.

Zaključek – mehanizmi potrjevanja

- Različico s ponavljanjem sekvence uporabimo pri zelo zanesljivih kanalih med entitetoma, saj so napake redke in občasno ponovno pošiljanje že sprejetih PPE bistveno ne vpliva na zmogljivost protokola, medtem ko je njegova dobra lastnost, da ohranja pravilno zaporedje sprejetih PPE.

Zaključek – mehanizmi potrjevanja

- Ponavljanje sekvence v primeru nezanesljivega kanala seveda ne pride v poštev: napake so pogoste, zato je treba ponavljati veliko število PPE, pri tem pa se lahko pojavijo nove napake na PPE, ki so bili v prvi fazi morda že pravilno sprejeti.
- V takem primeru bomo raje uporabili različico potrjevanja sekvence.

Zaključek – mehanizmi potrjevanja

- Vse različice protokola s tekočim pošiljanjem pridobijo pri zmogljivosti, če jih uporabimo na **popolnoma dvosmernem logičnem kanalu (full duplex)**.

Zaključek – mehanizmi potrjevanja

- Sprotno potrjevanje:
 - posredno
 - neposredno
- Tekoče pošiljanje:
 - posredno
 - neposredno
 - osnovna različica
 - potrjevanje zaporedja
 - ponavljanje zaporedja

Zaključek – mehanizmi potrjevanja

- Pri vseh različicah potrebujemo protokolarne parametre:
 - časovno kontrolo oddajnika
 - časovno kontrolo sprejemnika
 - največje število ponovnih oddaj oddajnika
 - največje število ponovnih oddaj sprejemnika

Kontrola pretoka

- Poleg sposobnosti protokola, da obvlada težave pri nepravilno sprejetih ali izgubljenih PPE in potrditvah, je treba pri protokolih s tekočim pošiljanjem zagotoviti tudi **kontrola pretoka** (flow control) podatkov med procesi.

Kontrola pretoka

- Kontrola pretoka je potrebna iz različnih razlogov:
 - Iz potrebe po sinhronizaciji med sprejemnikom in oddajnikom, ker sprejemnik ne more “porabiti” podatkov tako hitro, kot mu jih oddajnik pošilja. Včasih govorimo tudi o faznosti dialoga, kar v bistvu pomeni, da oddajnik z oddajami ne prehiteva za več , kot mu dopušča N-protokol.

Kontrola pretoka

- drugi razlog po uvedbi kontrole pretoka nastopi takrat, ko ima sprejemnik omejene pomnilniške kapacitete in preprosto nima več “prostora” za shranjevanje novih podatkov. V takem primeru mora prekiniti oddajnikovo pošiljanje PPE , da se ne izgubijo podatki ali da se ne zgodi prelitje (**overflow**) sprejemne čakalne vrste.
- Tako imenovane **aplikacijske in uporabniške omejitve** so pogojene predvsem z naravo aplikacij oziroma s sposobnostjo končnega uporabnika, ki je vključen v dialog.

Kontrola pretoka

- Problem kontrole pretoka je problem sprejemnika in njegovih omejitev. Kontrola pretoka se izvaja na več načinov:
 - neposredno opazovanje pomnilnika na strani sprejemnika
 - drugi način je povprečna obremenjenost pomnilnika in hitrost prenosa se uravnava glede na število trenutno nepotrjenih PPE

Kontrola pretoka

- Tretji način pa ni pogojen s performančnimi lastnostmi dialoga, temveč upošteva predvsem njegovo **vsebinsko naravo**. V principu to pomeni, da je oddajanje pogojeno s sposobnostjo sprejemnika, da sprejeta sporočila obdeluje sproti oziroma da je naslednja oddaja odvisna od rezultata obdelave prejšnjega sporočila.

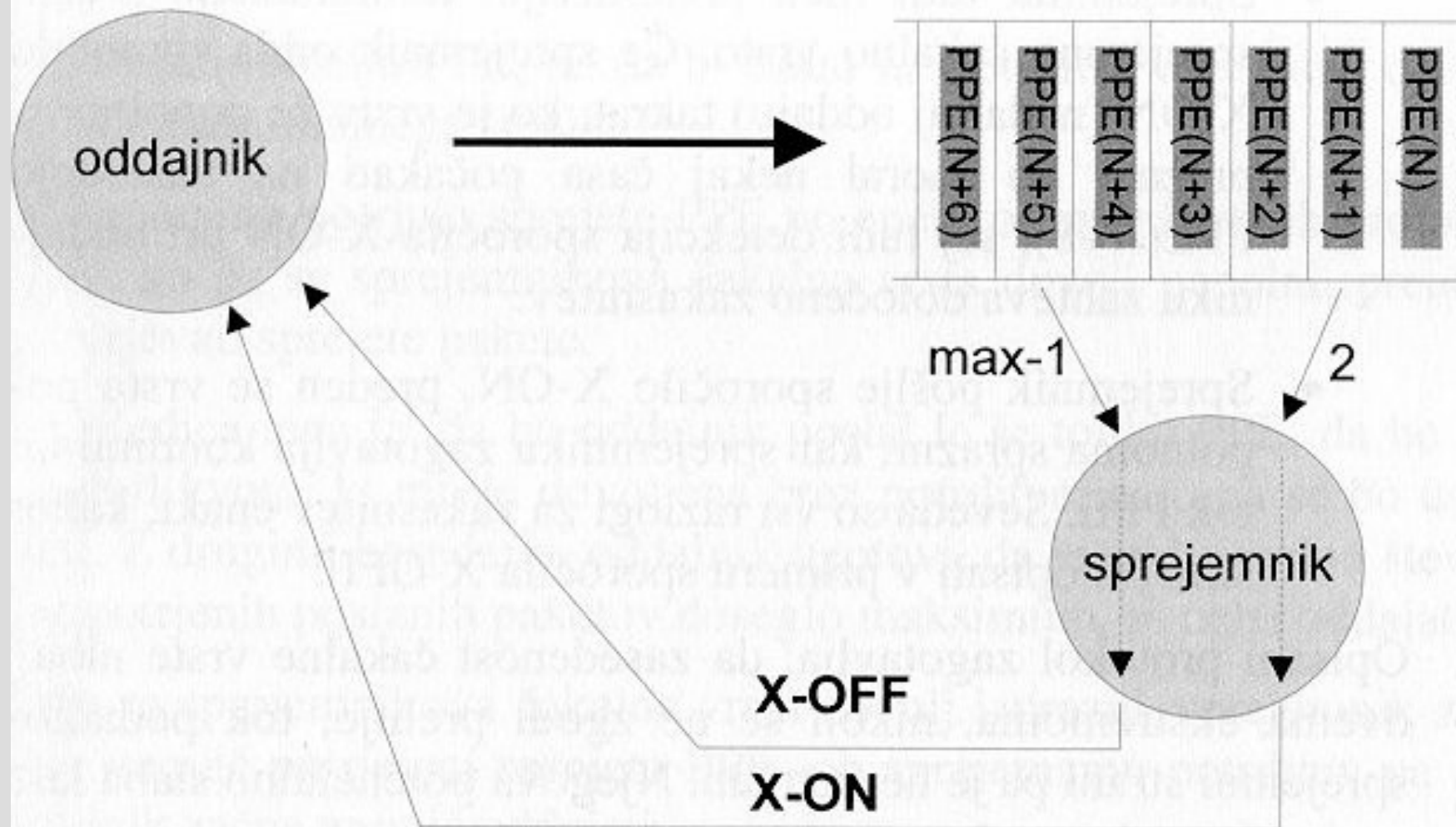
X-ON/X-OFF

- X-ON/X-OFF večkrat srečamo pri kontroli pretoka med računalnikom in terminalom.
- Protokol X-ON/X-OFF uravnava pretok tako, da neposredno nadzira zasedenost čakalne vrste. Uporablja dve protokolarni sporočili:
 - Protokolarno sporočilo X-OFF pošlje sprejemnik oddajniku, da se ustavi pošiljanje, ko je sprejemna čakalna vrsta polna
 - Ko pa se vrsta sprosti, sprejemnik pošlje protokolarno sporočilo X-ON, kar za oddajnik pomeni, da lahko nadaljuje z pošiljanjem.

X-ON/X-OFF

- Tako zasnovan protokol zagotavlja kontinuiran prenos PPE, hkrati pa preprečuje izgubo paketov, ko zmanjka sprejemnega pomnilnika.

X-ON/X-OFF



Slika 6-9: Protokol X-ON/X-OFF

X-ON/X-OFF

- Oddajnik pošilja $PPE(N)$, $PPE(N+1)$,... v sprejemnikovo čakalno vrsto, ki deluje po načelu FIFO (first in first out) – prvi noter, prvi ven.
- Oddajnik si lahko predstavljamo tudi kot proces v sistemu, ki komunicira s sprejemnim procesom.
- Sprejemnik oziroma sprejemni proces spremlja zasedenost sprejemne čakalne vrste, ki v našem primeru lahko sprejme sedem sporočil.

X-ON/X-OFF

- Zaradi zakasnitve pri prenosu X-OFF (nič v naravi se ne zgodi trenutno) mora sprejemnik oddati X-OFF, torej sporočiti oddajniku "nehaj pošiljati" preden je zasedena sedma celica.
- Čas zakasnitve pomeni čas, ki je potreben za analizo zasedenosti sprejemnikove čakalne vrste, čas, ki je potreben za prenos sporočila X-OFF od sprejemnika k oddajniku, ter čas, ki ga oddajnik potrebuje za analizo sprejetega sporočila X-OFF, ki mu sledi ustrezno ukrepanje.

X-ON/X-OFF

- Od realnih časov vseh teh zakasnitev je odvisno, koliko časa pred popolno zasedenostjo sprejemnikove čakalne vrste (7 podatkov) bo sprejemnik poslal X-OFF. Brez upoštevanja teh zakasnitev bi oddajnik ukaz X-OFF zaznal prepozno in pri oddaji paketa PPE(N+8) bi prišlo do preliva čakalne vrste in s tem do izgube PPE(N+8). V primeru na sliki je predvideno pošiljanje X-OFF sporočila takrat, ko se zasede šesta celica.

X-ON/X-OFF

- Sprejemnik tudi med prekinitvijo komunikacije prazni sprejemno čakalno vrsto. Če sprejemnik odda sporočilo X-ON (nadaljuje oddajo) takrat, ko je vrsta že popolnoma prazna, bo moral nekaj časa počakati na naslednjo PPE(N+8), saj tudi detekcija sporočila X-ON pri oddajniku zahteva določeno zakasnitev.

X-ON/X-OFF

- Sprejemnik pošlje sporočilo X-ON, preden se vrsta popolnoma sprazni, kar sprejemniku zagotavlja kontinuiran tok PPE. Seveda so vsi razlogi za zakasnitev enaki, kakor smo jih opisali v primeru sporočila X-OFF.

X-ON/X-OFF

- Opisani protokol zagotavlja, da zasedenost čakalne vrste niha med dvema ekstremoma, nikoli se ne zgodi prelitje, tok podatkov pa na sprejemnikovi strani pa nepretrgan. Njegova potencialno slaba lastnost pa je, da za kontrolo pretoka predvideva kontrolno povezanost oddajnega in sprejemnega procesa, kar v sistemu lahko povzroči kar nekaj težav oziroma zaplete izvajanje kontrole pretoka.

X-ON/X-OFF

- V splošnih omrežjih tak pristop ni izvedljiv, saj so zakasnitve odvisne od posamezne zveze, te pa se dinamično vzpostavljajo in rušijo. Za nižje plasti zato uporabljamo predvsem protokol z drsečim oknom, ki zagotavlja avtonomno kontrolo pretoka med oddajnikom in sprejemnikom. Za izvajanje kontrole ni potrebna medsebojna komunikacija.
- CTS/RTS?!

Protokol z drsečim oknom

- Tehnika drsečega okna (sliding window protocol) je naravnan bolj komunikacijsko. Pri tem mehanizmu ne kontroliramo zasedenosti čakalnih vrst, temveč varuje čakalne vrste pred pred napolnitvijo tako, da nadzira število oddanih PPE, ki jih sprejemnik še ni potrdil.
- Oddajnik pošilja PPE, ne da bi čakal na potrditev sprejema, do nekega maksimalnega števila.

Protokol z drsečim oknom

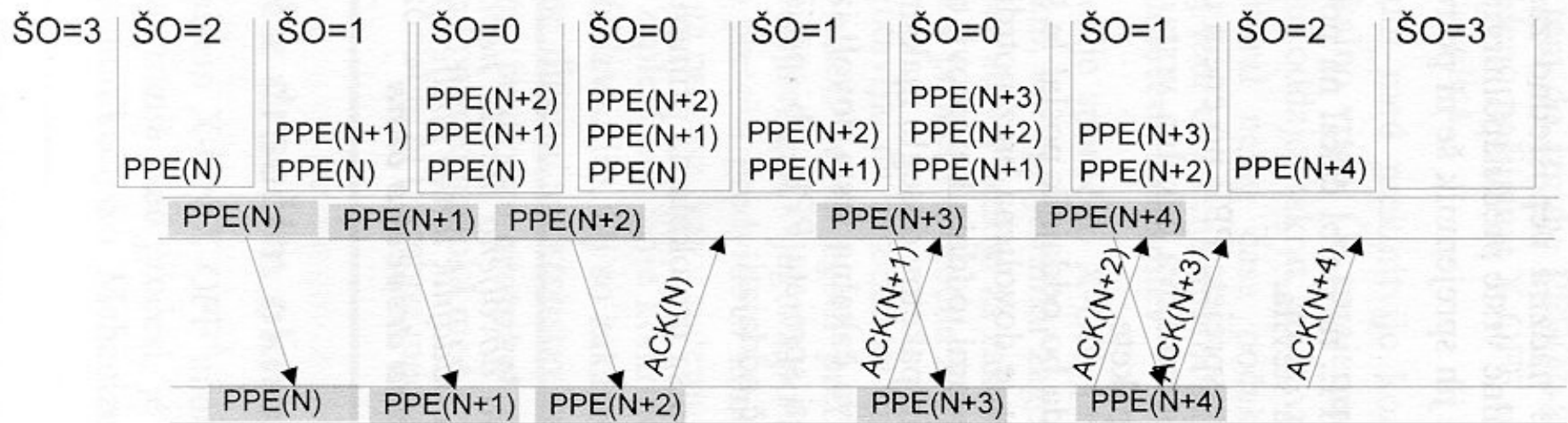
- Sprejemnik potrjuje spreteje PPE po enem od potrditvenih protokolov, ko pa se sprejemnikova čakalna vrsta dovolj napolni, preneha potrjevati sprejete pakete.
- Posledica tega je, da bo oddajnik poslal le še toliko PPE, da bo zapolnil kvoto, ki mu je dovoljena brez potrditve, nato pa se bo ustavil. Z drugimi besedami oddajnik ugotovi, da je maksimalno število nepotrjenih poslanih paketov doseglo maksimum in neha oddajati.

Protokol z drsečim oknom

- Ko se sprejemnikova čakalna vrsta dovolj izprazni, sprejemnik začne vnovič potrjevati PPE, ob sprejemanju potrditev pa začne oddajnik znova oddajati PPE.
- Pojem širine okna:
 - Maksimalno dovoljeno število nepotrjenih paketov PPE, ki jih oddajnik lahko pošlje sprejemniku, imenujemo širina drsečega okna (window size – širina okna (ŠO)).

Protokol z drsečim oknom

Slika 6-10: Protokol z drsečim oknom



Protokol z drsečim oknom

- Pred začetkom prenosa je okno na stežaj odprto. Primer na sliki $\check{S}O = 3$
- Ko oddajnik pošlje $PPE(N)$ se okno pripne $\check{S}O=2$
- Ko pošlje $PPE(N+1)$, se okno pripne še malo bolj $\check{S}O=1$
- Ob oddaji $PPE(N+2)$ se okno zapre , $\check{S}O=0$, zato oddajnik preneha pošiljati
- Približno takrat ko se sprejme $PPE(N+2)$ sprejemnik potrdi $PPE(N)$ in okno se vnovič odpre na $\check{S}O=1$
- Ob sprejemu potrditve oddajnik pošlje $PPE(N+3)$
- Po potrditvi $PPE(N+4)$, ko se dialog konča, je okno vnovič odprto, $\check{S}O=3$

Protokol z drsečim oknom

- S slike je tudi razvidno, da tok podatkov na sprejemni strani v našem primeru ni nepretrgan. Potrditev $ACK(N)$ je prišla prepozno. Če bi bila odposlana malo prej, se okno ne bi zaprlo in bi do oddaje $PPE(N+3)$ lahko prišlo takoj za $PPE(N+2)$. To težavo lahko odpravimo, če povečamo maksimalno širino okna. Za primer na sliki bi $\text{ŠO} = 5$ že zagotavljala kontinuiran pretok podatkov med oddajnikom in sprejemnikom.

Protokol z drsečim oknom

- Očitno je, da širina okna bistveno vpliva na prepustnost kanala oziroma da je širina okna parameter posamezne povezave, ki ga poskusimo nastaviti tako, da je tok sprejetih PPE kontinuiran.
- Kontrola pretoka je izvedljiva z vsakim od navedenih mehanizmov potrjevanja. Pri tem moramo opozoriti, da je mehanizem sprotnega potrjevanja pravzaprav poseben primer kontrole pretoka s $\text{ŠO} = 1$. Širino okna lahko dodamo v standardni nabor tipičnih protokolarnih parametrov.

Protokol z drsečim oknom

- Njegova bistvena prednost pred protokolom tipa X-ON/X-OFF je, da ne zahteva kontrolne komunikacije med procesi, zato ta mehanizem uporabljamo predvsem na nižjih plasteh.

Številčenje PPE in potrditev

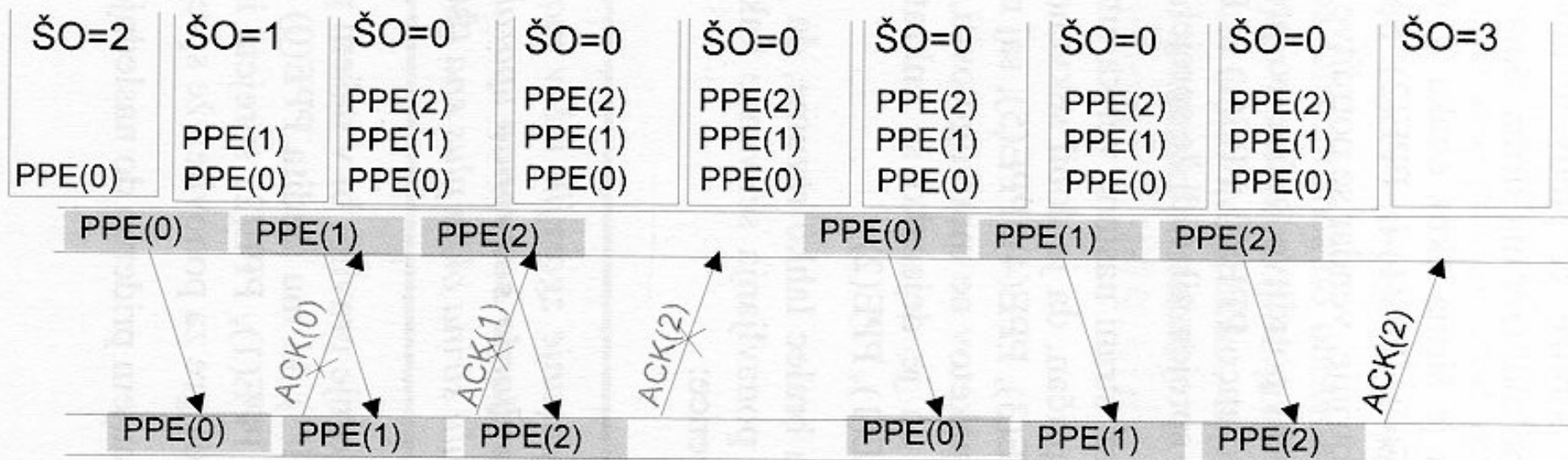
- Mehanizem kontrole pretoka z drsečim oknom omogoča tudi smiselno zaporedno označevanje PPE in potrditvenih sporočil. Ugotovili smo, da je sporočilo sestavljeno iz množice PPE, ki jih med seboj ločimo s **sekvenčno številko**. Ker je sporočilo lahko poljubno dolgo, se kaj hitro zgodi, da v glavi PPE predvidimo premalo bitov za visoke zaporedne številke, ali pa so polja, namenjena sekvenčni številki, pri kratkih sporočilih bolj ali manj neizkoriščena.

Številčenje PPE in potrditev

- Zaporedna številka PPE je za 1 povečana številka predhodnika. Mehanizem drsečega okna nam omogoča, da pakete označujemo ciklično (**Ciklično številčenje paketov**: krajša glava sporočila) glede na širino okna oziroma cikelj enega zapiranja in odpiranja okna. Tako zmanjšamo glavo PPE, saj ni treba rezervirati velikega polja za sekvenco, dolgo denimo nekaj tisoč paketov, temveč zadostuje velikost sekvenčne številke v odvisnosti od širine okna. Vendar moramo biti zelo previdni!

Številčenje PPE in potrditev

Slika 6-11: Napaka pri številčenju PPE



Številčenje PPE in potrditev

- Na primeru je širina okna enaka 3. Oddajnik pošlje tri pakete: PPE(0), PPE(1) in PPE(2). Sprejemnik jih sproti pozitivno potrjuje, vendar se potrditve izgubljajo.

Številčenje PPE in potrditev

- Ker oddajnik ne dobi potrditve, začne po preteku časovne kontrole vnovič pošiljati sekvenco PPE(0), PPE(1) in PPE(2), torej tisto, ki jo je že poslal in jo je sprejemnik tudi že sprejel.
- Na sprejemni strani nastane velika zmeda: sprejemnik je namreč prepričan, da je dobil sekvenco PPE(0), PPE(1), PPE(2), PPE(3), PPE(4), PPE(5), saj na podlagi zaporednih števil paketrov ne more ugotoviti, da je prišlo do ponovitve in da je dejansko še enkrat dobil že sprejete PPE(0), PPE(1) in PPE(2).

Številčenje PPE in potrditev

- Za preprečevanje težav morajo **protokoli brez ponavljanja sekvence** uporabljati številčenje po modulu okna plus ena ($\text{ŠO} + 1$).
- Pri protokolu s **ponavljanjem sekvence** mora biti modul oštevilčevanja paketov enak dvakratni širini okna ($2 * \text{ŠO}$).