PSTK ustni (1)

(1) Osnovni izrazi

→ Komunikacijski osebki ali entiteta = komunikacijska naprava
→ komunikacijsko sredstvo = mediji

→ Komunikacijski proces = dogajanje v komunikacijskem osebku oz. v celotnem sistemu

→ Tk omrežje = Sistem, ki ga sestavljajo komunikacijski osebki in kanali

→ t_p - čas propagacije ⇒ čas, ki ga potrebujemo za prenos informacije čez naš kanal. Zgornja meja prenosa informacije je $3 \cdot 10^8$ m/s...

→ kapaciteta kanala ⇒ MAX hitrost prenosa informacije v b/s pri kateri je še možna zanesljiva komunikacija

→ Pasovna širina ⇒ je širina frekvenčnega pasu, ki ga je možno prenašati skozi kanal.

→ ~~Hitrost~~ Čas potreben za oddajo sporočila oz. sprejem le tega:

$$t_t = \frac{L}{r}$$ L → dolžina sporočila, r → hitrost prenosa v b/s!

→ Zakasnitev v fizičnem kanalu $t_D = \frac{D}{v}$ D - fizična dolžina kanala, v - hitrost EM vala v kanalu

→ Pogostost napak ⇒ Razmerje pokvarjenih sporočil proti vsem ostalim.
⇒ Pogostost bitnih napak BER

→ Povezavno orientirana komunikacija ⇒ najprej se vzpostavi povezava nato se pošlje sporočilo in povezava se zapre... Zveza je lahko tudi stalna se pravi je povezava omogočena ves čas..
Sporočila pridejo na cilj v istem vrstnem redu kot so bila poslana

→ Nepovezavno orientirana komunikacija ⇒ Paketi samostojno krožijo pot do sprejemnika. Sporočila ne prihajajo v enakem vrstnem redu!

PSTK (10)

3) Intervalna potrditev
Potrjujemo določen interval sporočil ACK(m, n) (m – od, n – do)

→ Največ se uporablja komulativne

→ V primeru napake, oddajni osebek pošlje ponovno tisto sporočilo, ki se je pokvarilo + vsa ostala sporočila za njim, tudi če so ta pravilno bila sprejeta

→ Upoštevati je potrebno čas t_{DA} pri računanju časa do potrditve t_{rt}

$$t_{rt} = t_i + t_a + t_{da} + 2t_p$$

→ Pomemben je vrstni red

→ Širina sprejemnega okna je 1 ($W_r = 1$), širina oddajnega okna pa je večja od sprejemnega ($W_s > W_r$)

→ Modul štetja pa je potem enak $M \geq W_s + 1$

(20) Protokol s selektivnim ponavljanjem SRP

→ V primeru, da se pokvari kako sporočilo oddajnik pošlje samo tisto še 1x.

→ Na sprejemni strani potrebujemo več spomina za sprotno razvrščanje sporočil

→ Oddajno okno ter sprejemno okno sta enaka ($W_r = W_s > 1$) in sicer večja od 1.

→ Modul štetja je $M \geq W_r + W_s = 2W_r = 2W_s$ (večji ali enak)

(21) Natovarjanje (poglej še sliko na str 171)

→ V primerih dupleksne povezave oz pol-dupleksne ko se oddajnik in sprejemnik zamenjata vlogi...

→ Lahko potrditvena protokolna sporočila vključujemo kar v informacijska protokolna sporočila!

→ pri tem je $t_{rt} = 2t_i + t_{da} + 2t_p$ (čas prenosa; čas izteka časovnika v katerem mora sprejemnik oddati sporočilo potrditve oziroma sporočilo; čas razširjanja) $t_{da} > t_i$

(21) KRMILJENJE PRETOKA

PC...

-> To je vpliv protokolnih entitet na pretok informacije

-> npr. če je eden PC hitrejši od drugega da ne more tako hitro obdelovati podatke ter prazniti čakalno vrsto. Zato pride do nasičenja ter do izgeba paketov..

-> DEAD LOCK (ali mrtva točka) -> ko zaradi prenasičenja pade (opravljeni) prometni pretok na 0.

↳ * Eksplicitna metoda krmiljenja pretoka

Ko se protokolnemu osebku začne nabirati prevelika čakalna vrsta ter mu grozi nasičenje more eksplicitno povedati pošiljatelju, da naj preneha pošiljati dokler mu spet ne dovoli.

XOFF - XON metoda ... to sta ASCII znaka CtrlS in CtrlQ

-> V ARQ protokolih uporabljamo WACK (wait for acknowlage) Potem ga oddajnik oddaja sporočila ENQ use dokler ne dobi ACK od sprejemnika

* Implicitna metoda

-> Zraven funkcije krmiljenja pretoka opravljajo še ostala zadeve npr. popravljanje napak ARQ ...

-> ARQ => Zavlačevanje s potrditvami ...

-> Kreditni mehanizem (Sprejemnik zraven potrdil pošilja še informacijo koliko sporočil lahko še sprejme)

npr. TCP protokol uporablja kreditni mehanizem

PSTK (12)

(22) Mrtve duše

→ Ko recimo oddajnik pošlje sporočilo, to sporočilo se nekje zgubi, se izteče časovnik ponovno pošlje sporočilo ... sprejemnik dobi sporočilo hkrati pa se iz nekje spet pojavi prejšnje izgubljeno sporočilo ter nam naredi lahko zmedo.

→ Zadevo rešimo, da sporočilom dodamo življenjsko dobo (time-to-live), ko se izteče se tako sporočilo zavrže.

(23) Preberi si še str. 179 ter še 1k @ SOC

PSTK (2)

2 Standardi v TK

de iure - pravno veljavni standard

de facto - V praksi uveljavljen standard brez pravne veljave

=> ORGANIZACIJE -> Mednarodne -> ISO, ITU...

Nacionalne -> ANSI

Strokovna združenja -> IEEE, W3C...

(3) PRIMITIVI

-> Imenujemo jih tudi osnovni tipi interakcije (interakcije med uporabniki ter izvajalci storitev)

POZNAMO: ~~(Parametre)~~

-> request (req) ali zahteva
-> indication (ind) ali obvestilo
-> ~~od~~ response (resp) ali odgovor
-> confirm (conf) ali potrditev

generira uporabnik namenjena izvajalcu storitve

-> generira izvajalec s namenjeno uporabniku

-> Uporabljamo jih v paketu s specifikacijo!

-> Parametri nosijo s seboj dodatne informacije..

(4) Protokol

=> Množica pravil, ki določajo medsebojno izmenjavo sporočil ter njihove vsebine in formate imenujemo protokol!

-> Predstavlja implementacijo storitve, kako se neka storitev izvrši.

-> PDU - protokolne podatkovne enote (protocol data unit)
protokolna sporočila
↳ generirajo jih protokolni objekti ali entitete

uporabniška sporočila
↳ generirajo uporabniki

(5) Režija protokola

→ Nadzorna protokolna informacija PCI (to je rep + glava) predstavlja režijo protokola!

→ Nadzorna protokolna sporočila vsebujejo samo režijo!

(6) Mrtva točka (deadlock)

→ Je posledica protokola brez logične pravilnosti. Se pravi, da pravila v protokolu imajo napake, katere privedejo protokol v mrtvo točko. To se na primer zgodi, ko se protokolno sporočilo izgubi ter protokol se ustavi na mrtvi točki.

(7) Robustnost

Pomeni, da protokol predvideva tudi možnost nastopa nenormalnih situacij.
Se pravi, da zna reagirati ter sistem ponovno spraviti v pravilno delovanje

(8) Krmiljenje pretoka

Protokoli morejo skrbeti za usklajevanje pretoka podatkov skozi omrežje, da ne pride kje do nastajanja ...

(9) Sloji v OSI modelu

Sloj		
7	APLIKACIJSKI SLOJ	neodvisni od omrežja ter definirajo le neposredno komunikacijo med obema končnima terminaloma
6	PRIKAZNI SLOJ	
5	SEJNI SLOJ	
4	TRANSPORTNI SLOJ	
3	OMREŽNI SLOJ	odvisni od TK omrežja definirajo komunikacijo med 2 sosednjima TK napravama
2	KANALNI (POVEZAVNI) SLOJ	
1	FIZIČNI SLOJ	

Aleta

PSTK (4)

→ Fizični sloj

Naloga je prenos binarnih vrednosti s pomočjo el. mag. valovanja ali optičnega signala med dvema napravama!
Določa tudi mehanske lastnosti kot so konektorji ...
Procedura za vzpostavitev povezave ...

→ Kanalni sloj oz. Povezovalni sloj

→ Izdelava okvirja
→ Osnovna naloga je detekcija napak v okvirjih, do katere pride v fizičnem sloju
→ Izvaja tudi krmiljenje pretoka podatkov med napravama ...

→ Omrežni sloj

-> Ponuja višje ležečemu sloju (Transportnemu) storitev vzpostavitve ter vzdrževanja zvez med terminalom.
→ Poskrbi tudi za usmerjanje (routing) oz. iskanje poti skozi omrežje

→ Transportni sloj

→ Ukvarja se s prenosom sporočil med osebki višjega sloja! ne glede na to kakšno omrežje se uporablja ...
→ Pravi vrstni red sprejemanja sporočil, ter prihod vseh sporočil
→ Podpira drobljenje (Fragmentiranje) sporočil (na oddajni strani zdrobi nato pa zlepi skupaj)
→ Čim manj kvalitetno je TK-omrežje bolj kompleksen more biti transportni sloj

→ Sejni sloj

-> Ta sloj skrbi za vzpostavitev, vzdrževanje in zaključek dialoga med aplikacijskimi osebki
-> Poroča o izjemnih stanjih višjim slojem

→ Prikazni sloj
→ Skrbi za format podatkov, ki se jih izmenjujeta aplikaciji in za pretvorbo med formati.
→ skrbi s kompresijo ter šifriranjem podatkov

→ Aplikacijski sloj
Nudi aplikacijam splošno uporabne storitve kot so prenos datotek, upravljanje z njimi, elektronska pošta, terminalski dostop...

(10) TCP/IP model (protokolni sklad TCP/IP)

→ Internet oz paketni način komunikacije preko usmerjevalnikov ter z IP naslavljanjem

SLOJI:

→ Sloj za dostop do omrežja
Omogoča dostop končnega računalnika ali usmerjevalnika do omrežja!
Vsebuje dejansko prve tri sloje OSI → fizični, povezovalni ter omrežni!

→ Internetni sloj
Usmerja informacije skozi omrežje (od PC-ja do PC-ja)...
Uporablja se protokol IP za naslavljanje entitet...
Dodeljuje IP številke ali naslove...

→ Transportni sloj
Skrbi za prenos informacije neposredno med dvema končnima uporabnika!
TCP - povezavno orientiran ter UDP - nepovezavno orientiran
Naslov točke dostopa do storitve transportnega sloja imenujemo vrata!

→ Aplikacijski sloj

Se uporablja vrsto protokolov, ki nudijo neposredno podporo porazdeljenim aplikacijam na Internetu.

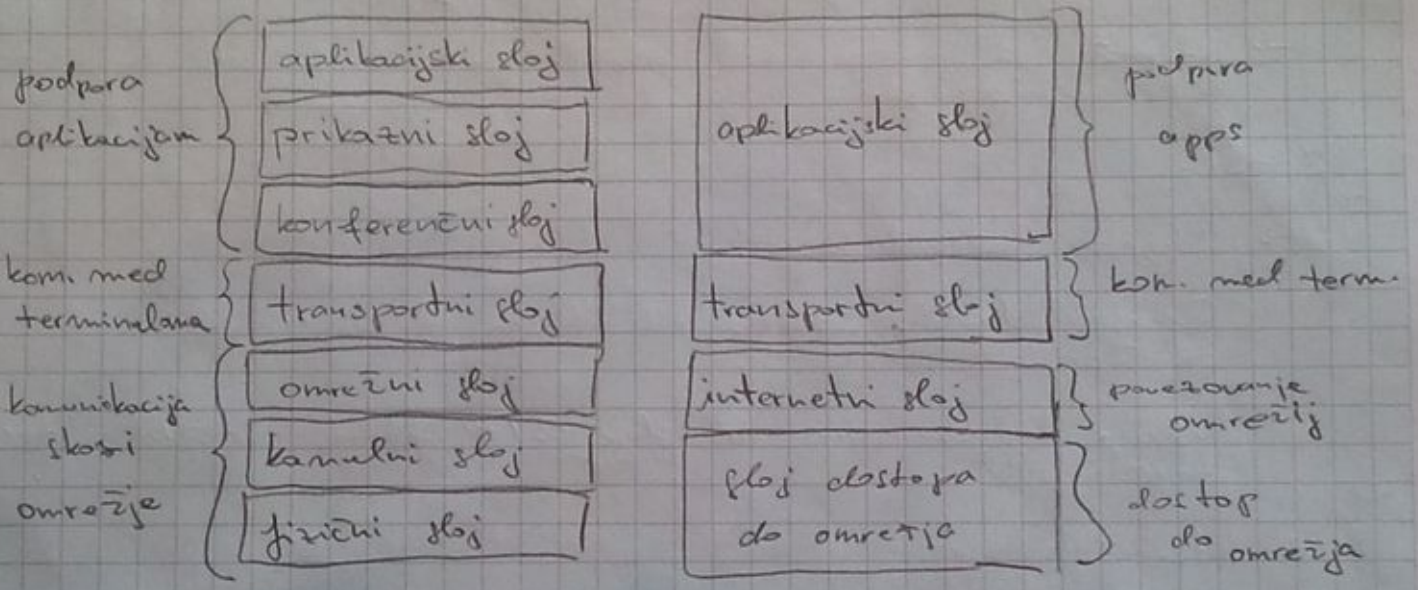
Npr. TELNET = omogoča delo na drugem R-ju ...

FTP, SMTP (pošiljanje emailov), POP3 (dostop do poštnega predala)

HTTP (HyperText Transfer protocol)

Vrata → FTP 21, SMTP 25, HTTP 80, POP3 110

(11) Primer med TCP/IP ter ISO/OSI (7 sl.)



(12) Učinkovitost protokola:

→ Faktor režije ⇒ Ker dejansko uporabnik ne vidi opravljenega prometa informacije, tudi ne učinkovitosti niti ne prometa.

Informacijska protokolna sporočila namreč poleg uporabniških sporočil pošiljajo tudi režijo. Zato je treba s stališča uporabnika vse te količine množiti še z faktorjem režije!

$$k_r = \frac{L_U}{L_P}$$

→ Z večanjem dolžine uporabniških sporočil manjša učinkovitost protokola

→ Kapaciteta kanala pri malo informacijah je veliko porabljena za vzpo... ...ie, usklajevanje in sprejemanje zvez.

PSTK (7)

(13) Paritetna metoda

→ Poznamo SODO ter LIHO metodo!

→ Besedilo zapišemo v binarni izvedbi s pomočjo ASCII kode
Za vsak znak izračunamo paritetni bit, to naredimo s seštevanjem po modulu 2. Liha izvedba je pa ravno njegova negacija.

→ Na sprejemni strani se zopet vse seštevje po MOD 2 ter če smo pošiljali SODE znake bo rezultat 0. Za Liho pa 1, če ni je prišlo do napake

(14) Metoda seštevanje (CHECKSUM)

→ Sporočilo razdelimo na delna zaporedja dolžine n bitov, ki jih imenujemo <u>besede</u>.

→ Besede seštejemo po modulu 2^n

→ ~~Dobimo~~ od dobljene vsote ohranimo le n-bitov z najmanjšo težo, to je tudi naša zaščitna beseda

→ Sprejemnik tudi računa vsoto ~~sprejetih~~ po istem modulu ter preveri če se ujema z zaščitno besedo!

(15) Hammingova razdalja (str. 173)

→ Uporaba pri ocenjevanju zmogljivosti nekega postopka kodiranja

→ d_H => Hammingova razdalja, ki je definirana kot število binarnih vrednosti, v katerih se med seboj razlikujeta poljubni dve kodni besedi.

→ $d(H_{MIN})$ => Minimalno število binarnih vrednosti v katerih se razlikujeta dve poljubni kodni besedi

□ → veljavna kodna beseda ◎ → neveljavne besede d - št napak

r → št. napak, ki jih lahko popravimo → - popravimo ? - le odkrijemo

Aleta

→ Če smo pri kodiranju dodali 1 paritetni bit bo d_H ter d_{Hmin} enaka 2! Če nismo dodali nič bo 1!.

→ Sprejemnik ob sprejemu napačnega sporočila vedno ugotovi, kolikšna je Hammingova razdalja sprejetega sporočila do najbližje veljavne kodne besede.
Če je ta razdalja manjša ali enaka neki v naprej določeni vrednosti ck, se na podlagi tega odloči ali bo napako popravil ali jo samo detektiral

$$\text{Če} \quad c = \frac{d_{Hmin} - 1}{2} \geq ck$$

(16) FEC in BEC

FEC ⇒ Popravljanje na sprejemni strani. Oddajnik sporočila zaščiti s kanalskim kodiranje z dovolj veliko minimalno Hammingovo razdaljo, ki sprejemniku samemu dovoli po odpravljanje napak, če ga teh ni preveč
(Forward error correction)

BEC ⇒ Popravljanje s ponovnim pošiljanjem. Oddajnik uporabi kanalno kodiranje z manjšo Hammingovo ~~kodo~~ razdaljo, kar omogoča le detekcijo napak ter s tem opozori ~~oddajni~~ sprejemnik oddajnika, da naj pošlje sporočilo še 1X.
(Backward error correction)

TAKI PROTOKOLI SO PROTOKOLI S PONAVLJANJEM ALI ARQ PROTOKOLI!

→ GLEDE NA ODKRIVANJE TER POPRAVLJAVE TOREJ ločimo:

→ napake ne odkrivamo in na popravljamo
→ napake odkrivamo ter pa sporočila zavržemo
→ FEC
→ BEC

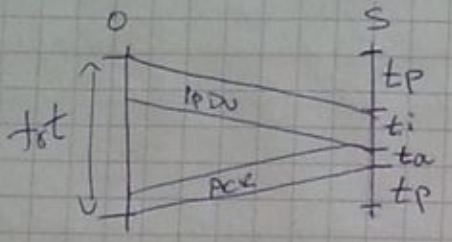
(17) ČASOVNIK PRI ARQ PROTOKOLIH (s ponavljanjem...)

$t_{rt} = t_i + t_a + 2t_p$

t_{rt} - čas do potrditve

t_i - čas IPDU (info protok. sporočilo (uskbuje režijo + uporabniško sporočilo)

t_p - čas propagacije signala skoz kanal



$t_i = \frac{L_i}{R}$ $t_p = \frac{D}{V}$ $t_a = \frac{L_a}{R}$

(18) Protokol s čakanjem ABP (alternirajoč bit)

→ Uporablja IPDU - pošiljanje info. prot. sporočila
Ack - sprejem pravilno potrjeno sporočilo
ter časovnik

→ Učinkovitost je $\eta = \frac{t_i}{t_{rt}}$ $\eta_U = \left(\frac{L_i - L_r}{L_i}\right) \cdot \frac{t_i}{t_{rt}}$ faktor režije

→ Širina oddajnega ter sprejemnega okna sta enaki oba 1 ($W_r = W_s = 1$) zato je pomemben vrsti red sporočil

→ Modul štetja je $M = 2 \Rightarrow W_r + W_s = 2$

(sprejemno okno (velikost) / oddajno okno; širina oddajnega in sprejemnega okna)

→ ~~Paket~~

(19) Kontinuirani protokoli s ponavljanjem GBN (Go Back N protokol)

(prvič v prvo se reče samo protokol s ponavljanjem)

→ Pošiljamo več sporočil hkrati, potrjujemo jih pa lahko na več načinov:

1) komulativna potrditev (potrjujemo od prvega pa do zadnjega, ki je v potrditvi specificiran) pač do tistega s zaporedno številko n ack(n).

2) Individualna potrditev
Potrjujemo enega samega IPDU-ja, ki je določen v potrditvi. Pri tem lahko imamo pomešan vrstni red sporočil

Aleta