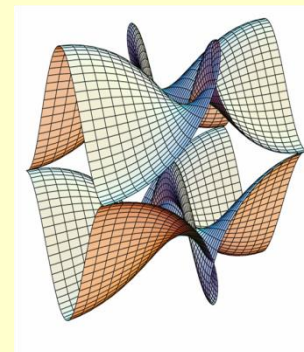




Sveučilište u Zagrebu
PMF – Matematički odsjek

MREŽE RAČUNALA
Predavanja 2022/2023



Poglavlje 5: Paketi, okviri, otkrivanje grešaka

Sastavio: Robert Manger; Prilagodio: Zvonimir Bujanović
06.10.2014, Matej Mihelčić 17.10.2023

Pojam paketa

- U većini računalnih mreža poruka se ne prenosi kao jedan kontinuirani niz bitova.
- Umjesto toga, poruka se dijeli u male dijelove koji se zovu *paketi* i koji se šalju zasebno.
- Zbog upotrebe paketa, mreže računala često se nazivaju *mreže s prospajanjem paketa* (packet switching networks).
- Telefonske mreže su prije VoIP-a radile na drugačijem principu, nazivale su se *mreže s prospajanjem linija* (circuit switching networks).

Prednosti upotrebe paketa (1)

- Efikasnije i pravednije korištenje zajedničkih resursa.
 - Kad bi se kroz zajednički resurs slale kontinuirane poruke, tada bi jedan par računala mogao zauzeti resurs, a drugi bi morali dugo čekati da dođu na red.
 - Razbijanjem poruka u pakete postiže se vremensko dijeljenje zajedničkog resursa. Dakle računala naizmjenično šalju pakete kroz resurs, ni jedno računalo ne osjeća dugi zastoј.

Prednosti upotrebe paketa (2)



- Mogućnost da paketi paralelno putuju različitim putovima kroz mrežu, čime se ubrzava prijenos podataka.
- Lakše ispravljanje grešaka u prijenosu podataka. Ako se otkrije greška, tada treba ponovo prenijeti samo jedan paket, a ne cijelu poruku.

Mane upotrebe paketa

- Određeni slojevi protokola moraju se baviti dijeljenjem poruka u pakete, te kasnijim sortiranjem i ponovnim sastavljanjem paketa u poruke.
- Nije moguće garantirati propusnost veze između dva računala. Budući da veza nije ekskluzivno rezervirana za jednu poruku, prijenos podataka može se usporiti zbog dijeljenja vremena s drugim porukama.

Pojam okvira (1)

- Svaka mrežna tehnologija definira u detalje kako izgledaju paketi koji se mogu prenositi kroz tu vrstu mreže.
- Da bi razlikovali općenitu ideju paketnog prijenosa od njene konkretne realizacije, uvodimo pojam okvira.
- Dakle *okvir* (frame) je paket s precizno definiranim formatom koji se koristi unutar određenog tipa mreže.

Pojam okvira (2)

- Na primjer, neka mrežna tehnologija mogla bi koristiti okvire varijabilne duljine koji se sastoje od ASCII znakova.
- Posebni znakovi `soh` odnosno `eot` služe za označavanje početka odnosno kraja okvira.
- Vidimo da se okvir sastoji od stvarnih podataka koje treba prenijeti, te od kontrolnih podataka.



Nadjevanje byte-ova (1)

- Prethodni format okvira s rezerviranim kontrolnim znakovima obično se ne može izravno primijeniti.
- Naime byte-ovi jednaki kontrolnim znakovima se mogu slučajno pojaviti unutar podataka koji se prenose.
- Kad bi blok s podacima sadržavao znak **eot**, primatelj bi ga pogrešno protumačio kao kraj okvira.
- Slično, znak **soh** unutar podataka pogrešno bi se tumačio kao početak novog okvira.
- Problem razlikovanja stvarnih podataka od kontrolne informacije može se riješiti tehnikom *nadjevanja byte-ova* (byte stuffing).

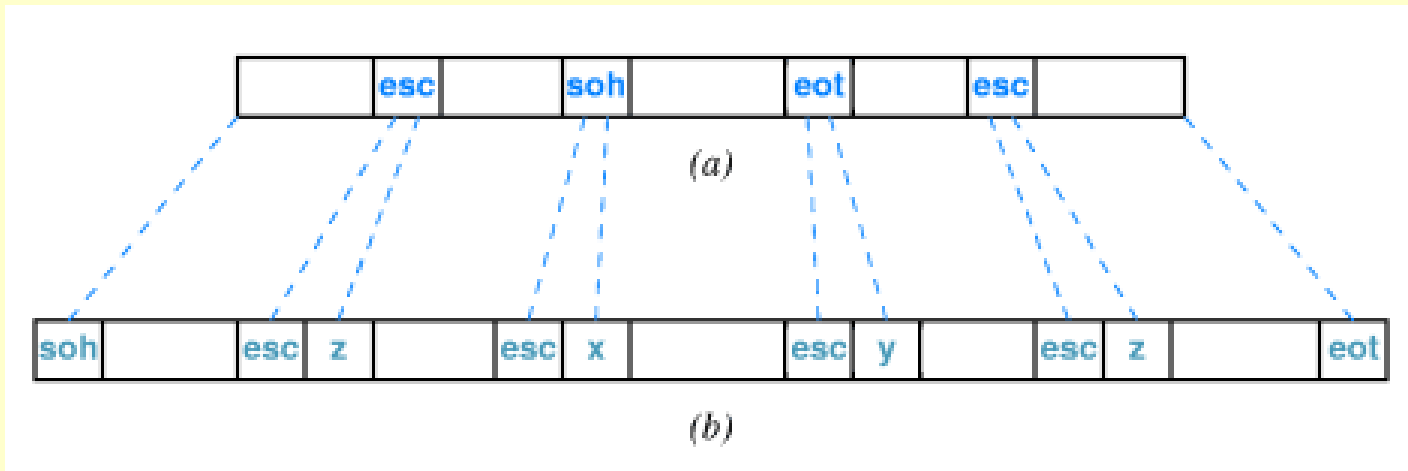
Nadjevanje byte-ova (2)

- Podaci se lagano modificiraju prije slanja, te vraćaju u polazno stanje nakon slanja.
- Za naš primjer okvira s dva rezervirana znaka **soh** i **eot**, nadjevanje byte-ova zahtijeva da uvedemo i treći rezervirani znak, na primjer **esc**.
- Prije slanja, pošiljatelj prolazi kroz podatke i zamjenjuje pojavu bilo kojeg rezerviranog znaka s kombinacijom od dva znaka prema tablici.

Character In Data	Characters Sent
soh	esc x
eot	esc y
esc	esc z

Nadjevanje byte-ova (3)

- Nakon ove zamjene, unutar dijela okvira s podacima više se ne pojavljuju ni **soh** ni **eot**. Primatelj zato može ispravno odrediti početak i kraj okvira i izdvojiti podatke.



- Da bi reproducirao originalne podatke, primatelj u dijelu okvira s podacima obavlja inverznu zamjenu znakova prema tablici.

Otkrivanje grešaka u prijenosu (1)

- Mediji za prijenos podataka podložni su smetnjama. Dešava se da podaci koji putuju mrežom budu izmijenjeni, oštećeni ili izgubljeni.
- Računalne mreže koriste razne mehanizme za otkrivanje grešaka u prijenosu, koji se svode na slanje neke dodatne informacije zajedno s podacima unutar istog okvira. Detaljnije:
 - Pošiljatelj računa vrijednost dodatne informacije iz originalnih podataka i umeće je u okvir.
 - Primateelj obavlja isto računanje na osnovu primljenih podataka.
 - Ako se dvije izračunate vrijednosti razlikuju, očito je došlo do greške u prijenosu.

Otkrivanje grešaka u prijenosu (2)



- Razmotrit ćemo tri takva mehanizma za otkrivanje grešaka:
 - *Bitovi za parnost* (parity bits)
 - *Kontrolni zbrojevi* (checksums)
 - *Cikličke provjere redundancije* (cyclic redundancy checks – CRC)
- Svi navedeni mehanizmi mogu otkriti neke vrste grešaka, no ne daju garanciju da greške nije bilo.

Bitovi za parnost

- Dodatna informacija dobiva se proširivanjem svakog byte-a iz originalnih podataka s još jednim bitom, tako da ukupan broj bitova-jedinica u proširenom byte-u bude paran („*even parity*”).
- Ista ideja koristila se u staroj 7-bitnoj verziji ASCII koda. Budući da se 7-bitni znak zapravo pohranjivao u jednom byte-u, postojao je dodatni osmi bit za provjeru parnosti.
- Ovaj mehanizam otkriva promjenu jednog bita unutar byte-a prilikom prijenosa, no ne otkriva promjenu dva bita.

Kontrolni zbroj (1)

- Podaci unutar okvira promatraju se kao niz cijelih binarnih brojeva određene duljine.
- Dodatna informacija dobiva se zbrajanjem tih cijelih brojeva i “normalizacijom” zbroja na neku određenu duljinu.
- U sljedećem primjeru, tekst se promatra kao niz 16-bitnih cijelih brojeva, tako da se ASCII kodovi od po dva susjedna znaka shvate kao jedan broj. Zbroj se normalizira na 16 bitova tako da se prijenos ponovo pribroji zbroju.

Kontrolni zbroj (2)

H	e	l	l	o	w	o	r	l	d	.	
48	65	6C	6C	6F	20	77	6F	72	6C	64	2E

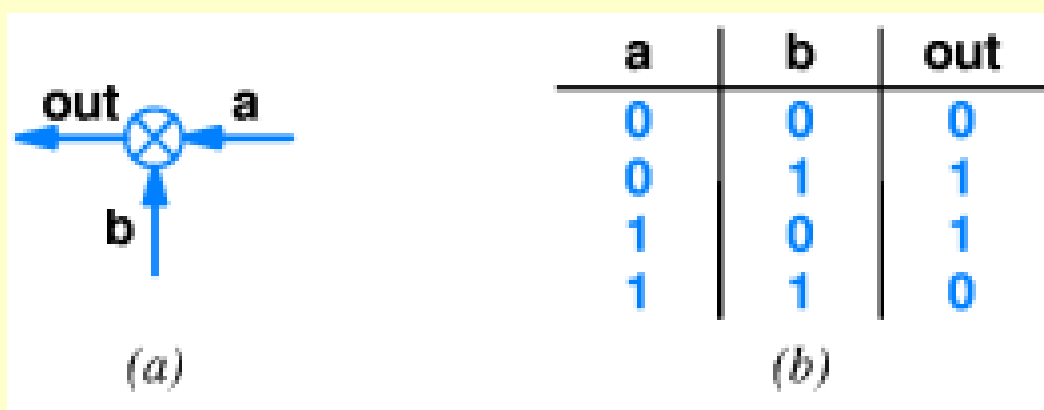
$$4865 + 6C6C + 6F20 + 776F + 726C + 642E + \text{carry} = 71FC$$

- Kontrolni zbroj je pouzdaniji mehanizam od bitova za parnost. Ipak, neke greške i dalje ostaju neotkrivene. U sljedećem primjeru su se promijenila 4 bita u podacima, a kontrolni zbroj je ostao isti.

Data Item In Binary	Checksum Value	Data Item In Binary	Checksum Value
0001	1	0011	3
0010	2	0000	0
0011	3	0001	1
0001	1	0011	3
totals	7		7

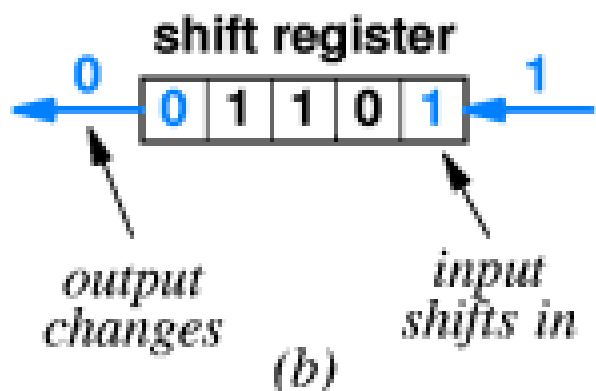
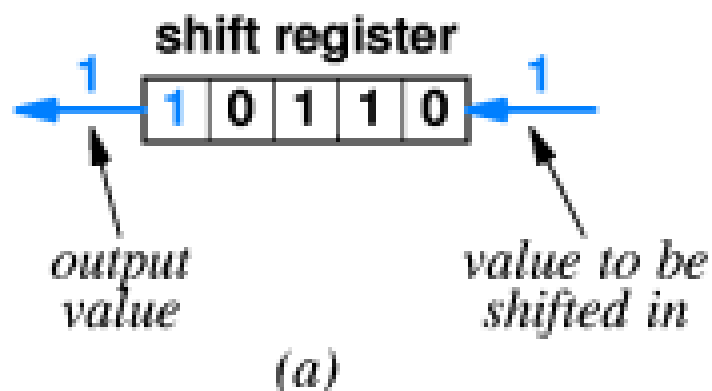
Cikličke provjere redundancije (1)

- Dodatna informacija dobiva se računanjem takozvanog CRC. Računanje se implementira u hardveru kombiniranjem logičkih sklopova za ekskluzivno-ili te shift-registara.
- Sklop za ekskluzivno-ili prima dva bita kao ulaz i daje jedan bit kao izlaz u skladu s tablicom.



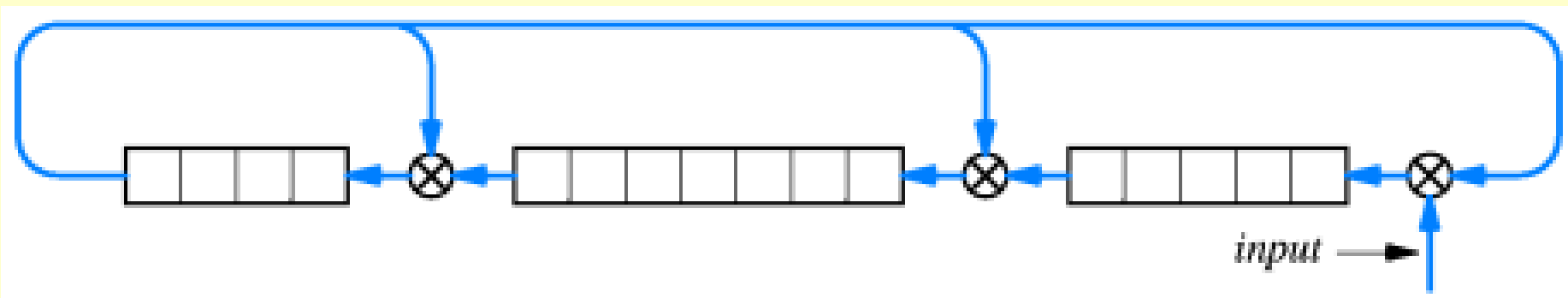
Cikličke provjere redundancije (2)

- Shift-registar pohranjuje niz bitova. Izvršavanjem operacije shifta novi bit zdesna ulazi u registar, svi bitovi u registru pomiču se za jedno mjesto ulijevo, a bit koji je do tada bio na krajnjem lijevom mjestu se gubi. Registar kao izlaz daje vrijednost trenutnog bita na lijevom kraju.



Cikličke provjere redundancije (3)

- Sljedeći sklop računa CRC od 16 bitova. Svi registri najprije postave na 0, a zatim se podaci iz okvira kao niz bitova uguravaju u sklop nizom shift operacija. Pritom svi registri simultano izvode svoje shiftove. Nakon što je cijeli niz bitova uguran u sklop, registri sadrže traženi CRC.



- Matematička analiza pokazuje da CRC otkriva više grešaka od kontrolnog zbroja.