

BModem – modemski protokol za prenos fajlova

Novi protokol izložen u ovom radu predstavlja unapređenje postojećih protokola za prenos podataka serijskom vezom u uslovima nepouzdanе veze. Unapređenje je postignuto lociranjem greške u paketu koji je sadrži, čime se smanjuje količina podataka koja se ponovo šalje kroz komunikacioni kanal. Postupak lociranja greške zasnovan je na uzastopnom deljenju paketa na potpakete, čiji se integritet proverava, a zatim se nastavlja deljenje onih potpaketa u kojima je greška otkrivena. Na taj način ukupna količina podataka koja se prenosi kroz komunikacioni kanal u slučaju greške može se značajno umanjiti.

Uvod

Serijska veza dva računara, uz upotrebu telefonske linije kao medijuma za prenos podataka, predstavlja veoma rasprostanjen vid računarskih komunikacija, jeftin i lak za održavanje. Međutim, osnovne mane su mu mala brzina prenosa i velika podložnost greškama usled šuma. Obe proističu iz osobina telefonske linije: dok se prva prevazilazi izradom boljih uređaja za kodiranje i dekodiranje, druga se može ublažiti samo izradom odgovarajućih protokola (hardverskih i softverskih) koji otkrivaju pojavu greške u komunikaciji.

Implementacije softverskih protokola podrazumevaju dva procesa: pošiljaoca i primaoca. Procesi mogu komunicirati međusobno samo korišćenjem dvosmernog komunikacionog kanala koji nije pouzdan, tj. koji ne obezbeđuje očuvanje integriteta podataka koji se preko njega prenose. Pošiljalac šalje podatke primaocu po nekom utvrđenom redosledu (koji je definisan protokolom), a oba procesa učestvuju u detektovanju greške. Najrasprostranjeniji softverski protokoli (Kermit, XModem, YModem, ZModem itd.) generalno ovo rade na taj način što podatke dele i šalju u paketima, tj. blokovima uglavnom fiksne dužine, kojima se pridružuju kon-

*Zoran Rilak (1978),
Bor, Devete brigade
11/7, student I
godine Fakulteta
organizacionih
nauka u Beogradu
bruce@afrodi ta.*

trolni kodovi na osnovu kojih primalac može da ustanovi da li je u prenosu došlo do greške izazvane šumom. Kontrolni kodovi se računaju na obe strane, primačevnoj i pošiljačevnoj. Ukoliko se kod koji je pošiljalac poslao uz blok podataka razlikuje od koda izračunatog na osnovu tog bloka na primačevnoj strani, primalac od pošiljaoca zahteva ponavljanje defektnog paketa (slanjem neke unapred određene sekvence).

Poželjno je da algoritam za računanje kontrolnih kodova bude iterativan; jedan od najpopularnijih je CRC (Cyclic Redundancy Check) koji nije komplikovan za računanje i dovoljno je pouzdan.

Opis protokola

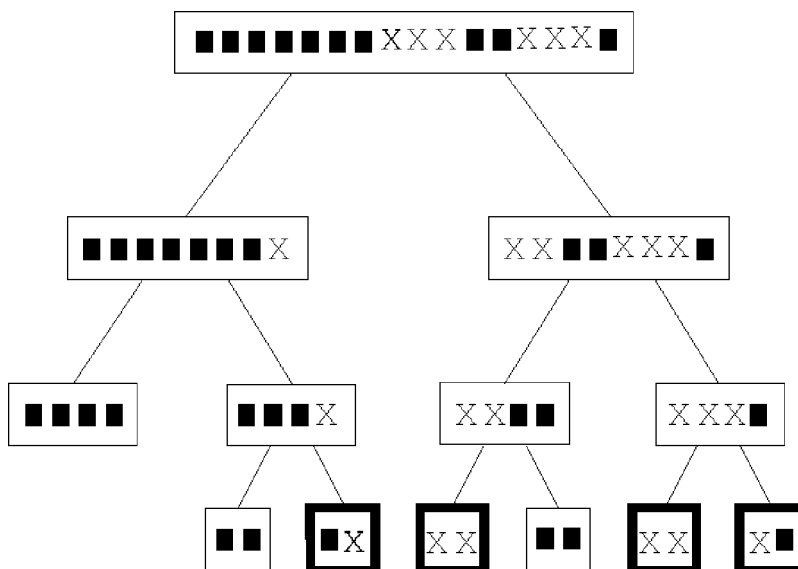
Za razliku od većine popularnih protokola koji detekciju greške ograničavaju samo na jedan paket, novi protokol (u daljem tekstu BModem – zbog zgodne asocijacije na osnovnu strukturu podataka korišćenu u izradi protokola, binarno stablo) korišćenjem jednostavnog algoritma i uz malu količinu razmenjenih informacija preko komunikacionog kanala umnogome sužava oblast primljenog paketa u kojoj se greška nalazi, skraćujući time i vreme provedeno u ispravljanju greške.

Kada primalac otkrije grešku u primljenom paketu na osnovu neslaganja primljenog i izračunatog kontrolnog koda, on o tome obaveštava pošiljaoca. I primalac i pošiljalac tada dele defektni paket na dva dela jednake dužine. Ti delovi se posmatraju kao zasebni paketi – *potpaketi* i oni čine čvorove binarnog stabla paketa na dubini 1. Koren tog stabla je polazni paket koji nalazi se na dubini 0.

Za oba potpaketa računaju se kontrolni kodovi na obe strane, primačevnoj i pošiljačevnoj. Pošiljalac šalje kontrolne kodove primaocu koji ih upoređuje sa odgovarajućim, izračunatim na osnovu podataka primljenih preko komunikacionog kanala.

U zavisnosti od toga koji par kontrolnih kodova nije jednak, primalac javlja pošiljaocu da jedan ili oba potpaketa sadrže grešku. Ovo zapravo stvara mogućnost sužavanja oblasti paketa za koju je poznato da sadrži grešku; dok je to ranije bio ceo paket, sada može biti samo jedna polovina paketa (ili ceo paket – ukoliko se oba kontrolna koda ne poklope). Time je završena obrada binarnog stabla paketa na dubini 1.

Prelazak na sledeći nivo stabla izvodi se ponavljanjem opisanog algoritma nad svakim potpaketom iz prethodnog nivoa u kome je otkrivena greška. potpaketi se dele na potpakete granajući binarno stablo paketa. Tako se na drugom nivou mogu stvoriti 4 potpaketa (ukoliko se greška/greške prostiru kroz svaku četvrtinu glavnog paketa) ili manje. Na ovaj način (ako je defektni deo paketa mali i lociran samo na jednom mestu) može se formirati najmanje jedan potpaket (slika 1).



Slika 1.
Šema stvaranja
potpaketa.

Figure 1.
Scheme of creation of
subpackets.

Grananje stabla paketa zaustavlja se na nekoj dubini D_{opt} koja je određena kao optimalna za date parametre komunikacije: početnu veličinu bloka (poželjno stepen broja 2, zbog načina na koji se generiše stablo paketa), dužinu kontrolnog koda i procenjenu verovatnoću pojavljivanja greške u prenosu. Potpaketi u kojima je ustanovljena greška na dubini D predstavljaju *listove* binarnog stabla paketa. Ti potpaketi se šalju primaocu kroz komunikacioni kanal.

Trenutno su razvijene dve strategije za ponavljanje podataka u kojima je ustanovljena greška (tzv. sakupljanje lišća). Prva se sastoji u slanju potpaketa jedan po jedan, zajedno sa odgovarajućim kontrolnim kodovima. Primalac obaveštava pošiljaoca o (ne)uspehu prenosa nakon svakog primljenog potpaketa, a ukoliko dođe do greške u prenosu ceo potpaket se ponavlja. Druga strategija je efikasnija kada je verovatnoća pojave greške manja, pa paket sadrži manje grešaka i listova binarnog stabla ima manje. Suština je da se svi potpaketi spoje u jedan, dužine $k \cdot 2^{b-D}$, gde je k – broj potpaketa, $b = \log_2 B$, B – početna dužina paketa. Taj paket se šalje zajedno sa svojim kontrolnim kodom i nad njim se u slučaju greške može ponovo primeniti opisani algoritam, stvarajući novo binarno stablo.

Analiza protokola

Na ponašanje algoritma najviše utiče parametar D (najveća dubina deljenja paketa) koji je od bitnog konceptualnog značaja, jer se za vrednost $D = 0$ kojom se deljenje paketa zabranjuje, BModem svodi na već poznate algoritme tipa X/Y/ZModem. Vrednost D treba biti pažljivo odabrana na

osnovu količine šuma na vezi: ako je šum veliki i greške učestale, povećanje efikasnosti se postiže smanjenjem ovog parametra. Obrnuto, ako se greška javlja retko, ima smisla povećati vrednost ovog parametra jer će se tako smanjiti ponavljanje ispravnih podataka zajedno sa onima koji sadrže grešku. Algoritam je dizajniran tako da omogući promenu parametra D u svakom trenutku prenosa, bilo umanjnjem ili uvećanjem, a na osnovu heuristika koje će biti postavljene tokom testiranja algoritma. Tako će se algoritam prilagođavati uslovima na vezi i, u cilju povećanja efikasnosti, podešavati dubinu deljenja i početnu dužinu paketa.

Početna dužina paketa je parametar koji takođe značajno utiče na vreme potrebno da se svi paketi razmene između pošiljaoca i primaoca. Što je početna dužina veća, veća je i verovatnoća da će u prenosu paketa doći do greške. Ukoliko na vezi nema puno šuma, početna dužina paketa se može uvećati, kako bi se smanjio broj prenetih kontrolnih kodova.

Procena efikasnosti algoritma zasnovana je na analizi njegovog ponašanja u najgorem i najboljem slučaju, tj. kada se greška prostire kroz ceo paket i kada se nalazi samo u jednom potpaketu na dubini D . Procenjena količina podataka u ova dva slučaja predstavljena je sledećim formulama:

$$w = B + 2(c + r)(2^{D+1} - 1)$$

$$b = \frac{B}{2^D} + 2(c + r)(2^D + 1)$$

gde je:

w – količina podataka preneti u najgorem slučaju

b – količina podataka preneti u najboljem slučaju

D – maksimalna dubina deljenja paketa

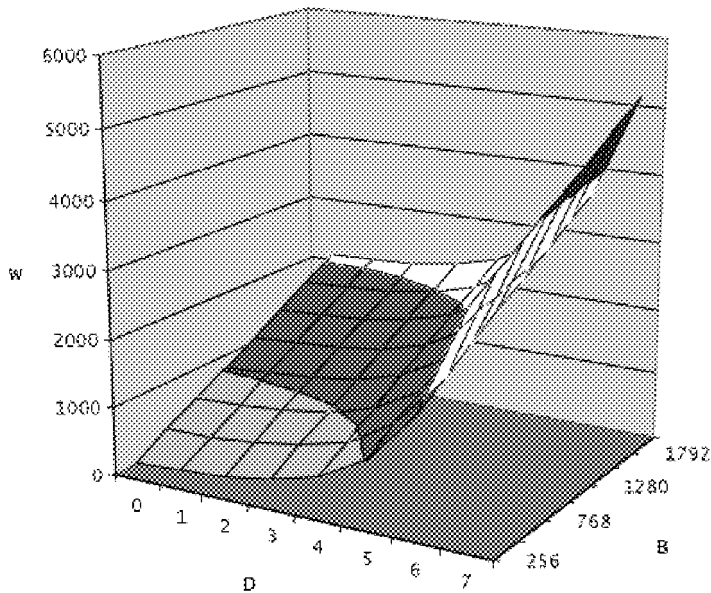
B – početna veličina bloka

c – dužina kontrolnog koda

r – dužina odgovora (feedback).

Ove formule predstavljaju procenu izloženog algoritma. On je neoptimizovan, kako u pogledu obrade podataka (računanja kontrolnih kodova), tako i u pogledu same razmene kontrolnih kodova koja se može sprovesti sa znatno manje razmenjenih informacija. Grafici ovih funkcija za $c = 4$, $r = 2$ i različite vrednosti B i D prikazani su na slici 2.

Količina prenetih podataka u najboljem slučaju, za različite vrednosti parametra D , je zbir dve funkcije: dužine potpaketa na dubini D i količine prenetih kontrolnih kodova do dubine D . Prva funkcija opada, a druga raste eksponencijalno. Kao posledica toga javlja se minimum prenete količine podataka za neku vrednost D_{opt} ; ta vrednost zavisi od zbira dužina kontrolnog koda i odgovora (feedback).



Slika 2.
 Grafici količine prenetih podataka u najgorem (w) i najboljem (b) slučaju u zavisnosti od početne dužine paketa B i maksimalne dubine deljenja D u slučaju kada dužina kontrolnog koda c iznosi 4, a dužina odgovora r 2. Uočava se da količina prenetih podataka u najgorem slučaju raste eksponencijalno sa D, dok u najboljem slučaju nakon minimuma raste i dostiže skoro istu vrednost, nezavisno od početne dužine paketa.

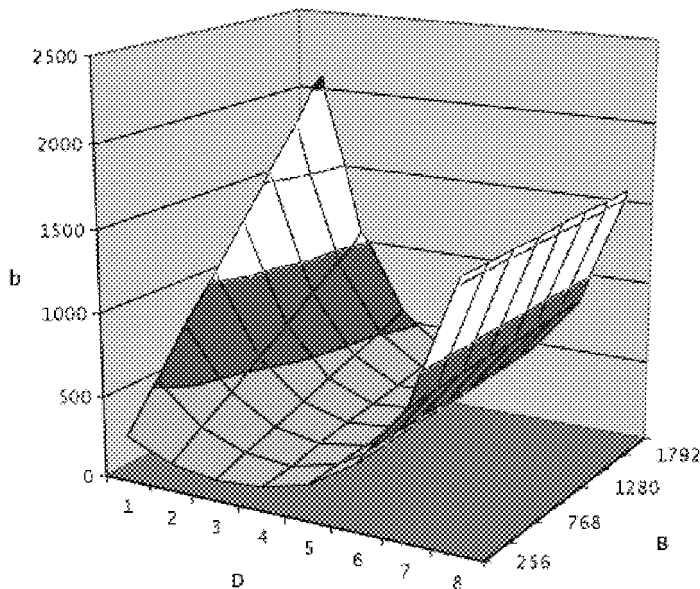


Figure 2.
 Graphs of quantity of transferred data in worst (w) and best (b) case in function of initial packet length B and maximal splitting depth D with control code length of 4, and feedback length of 2. Note that quantity of transferred data in the worst case grows exponentially with D, while in the best case after a local minimum it grows and reaches almost equal values, independently of initial packet length.

Na osnovu iznete procene očekuje se da se BModem pokaže efikas-niji od ostalih protokola u uslovima kada greška nije učestala. Čak i kada to nije slučaj, efikasnost neće opasti, ako algoritam bude sposoban da menja svoje parametre i time se uspešno prilagođava stanju na vezi.

Literatura

Banjanin, M.K. 1993. *Organizacione i instruktivne komunikacije (teorija i dizajn)*. Beograd: Naučna knjiga.

Wirth, N. 1991. *Programiranje na programskom jeziku Modula-2*. Beograd: Mikroknjiga.

Zoran Rilak

BModem – A New Modem Protocol for File Transfer

The protocol described in this paper represents a generalization of existing file transfer protocols over a modem connection. The advancement is achieved by splitting data into packets whose consistency is checked using common CRC algorithm. If data received and data sent proved to be inconsistent, the packet is split in half on both the receiver's and sender's side, thus creating two subpackets. Each one of these is checked against errors using the method described above. This recursive algorithm creates a binary tree consisting of subpackets whose length decreases with the increase of the tree depth. This splitting ends at some depth D , when smallest subpackets with errors are sent through the communication channel (Figure 1).

An algorithm analysis has been done, showing behaviour of the protocol with respect to different values of its parameters. Two functions were formed: one illustrating protocol's performance in worst case conditions (entire packet incorrectly transferred) and the other one representing performance in best case conditions (only one byte changed during transfer). These two functions show that for different values of other parameters there exist some optimal value of depth (D_{opt}) for which minimal quantity of data will be transferred in the best case, while in the worst case it will not be significantly greater than sole packet length (Figure 2). This implies that BModem can successfully replace other popular protocols, such as XModem, YModem, ZModem and Kermit, providing significant increase of performance in some cases, without reducing it otherwise.

