**Facultatea de Electronica Telecomunicații si Tehnologia Informației**

**DCCP-Protocol de Control al Congestiei TCP cu Datagrame**

**Studenți: Profesor coordonator:**

**Dună Robert Dr. Ing. Ștefan Stăncescu**

**Dunărențu Radu Florin**

**441A**

CUPRINS

**Introducere** **(Duna Robert) 3**

**MOD DE FUNCTIONARE**

 ***Sequence number* (*Dunarentu Radu*) 5-6**

 ***TCP sequence numbers* (Duna Robert) 6-7**

***DCCP sequence numbers* (*Dunarentu Radu*) 7-8**

***Sincronizarea*** **(Duna Robert) 8**

***Acknoledgment* (*Dunarentu Radu*) 8-9**

***Sequence number length*** **(Duna Robert) 9-10**

**PERFORMANTE**

***Controlul congestiei TCP* (*Dunarentu Radu*) 10-11**

***Controlul congestiei DCCP* (Duna Robert) 11**

***Testarea* (*Dunarentu Radu*) 11-12**

 ***Timpul vs Rata de bit.* (Duna Robert) 12-13**

***Delay Variabil* (*Dunarentu Radu*) 13-14**

 ***Variatia Pierderii* (Duna Robert) 14-17**

**CONCLUZII 17-18**

**BIBLIOGRAFIE 19**

**Introducere**

In ultimii ani s-a constatat o crestere a aplicatiilor cum ar fi : online multiplayer, telefonie prin internet, conferinte video.Aceste tipuri de aplicatii au in comun faptul ca prefera sa obtinerea mesajelor noi in favoarea retransmiterii mesajelor.Din pacate TCP si UDP nu sunt protocoale care sa se potriveasca acestor tipuri de aplicatii.TCP nu este un protocol potrivit pentru astfel de aplicatii deoarece este un protocol care prefera transmiterea fiabila a datelor , spre deosebire de protocolul dorit care trebuie sa transmita cat mai repede posibil datele.Problema UDP-ului este ca ii lipseste controlul congestiei.De aici rezulta nevoia unui nou protocol care sa poata face in acelasi timp doua lucruri:sa transmita pachetele foarte repede si sa realizeze si un control al congestiei.

Pe langa utilitatea in aplicatiile care prezinta constrangeri de timp , DCCP poate fi folosit ca mecanismul general de control al aplicatiilor de baza ale UDP.

Conexiunea DCCP contine si trafic ACK (acknoledgment) si trafic de date.ACK informeaza emitatorul daca datele au ajuns la destinatie.

**MOD DE FUNCTIONARE**

DCCP este un protocol unicast cu flux bidirectional de date.Conexiunile incep si se termina cu three-way handshakes.Acest lucru inseamna ca la inceperea unei conexiuni noi clientul trimite o cerere catre server ,server-ul trimite un raspuns clientului si clientul trimite un ACK serverului pentru a il asigura ca datele au ajuns.In cazul inchiderii conexiunii server-ul trimite primul o cerere de inchidere (CloseReq) clientul inchide conexiunea(Close) iar serverul trimite un semnal de Reset.Acest lucru este ilustrat in figura 1.



In figura 2 este reprezentat header-ul unui pachet DCCP.Acesta are o dimensiune de 16 octeti.



Offset-ul fiecarui pachet de date este masurat de Data Offset.Acesta are o dimensiune de 1 octet si contine 1024 de optiuni.Cu ajutorul campului Type obtinem informatii despre tipul pachetului.Antetului i se pot adauga campuri suplimentare numai in cazul in care acest lucru este necesar.Acesta este un aspect pozitiv pentru ca antetul nu este incarcat cu campuri care nu sunt frecvent folosite.Un alt mod de a nu incarca antetul este de a avea un nr variabil de ACK-uri in cazul in care transferul de date este unidirectional.

***Sequence number***

DCCP este format dintr-o parte principala care se ocupa de management-ul conexiunii,setare ,sincronizare etc si o parte ,formata din module,care se ocupa de controlul congestiei.

Partea principala este foarte simpla.Acest lucru are avantaje si dezavantaje.Un exemplu de dezavantaj este urmatorul:TCP poate simplifica unele aspecte in managament-ul conexiunii, dar acest lucru poate afecta fiabilitatea ,care este caracteristica de baza a TCP.

TCP realizeaza controlul fluxului prin sincronizarea continua a doua puncte.Fiabilitatea TCP este realizata prin descrierea starii fluxului de date cu un numar fix de ACK-uri.

Cu alte cuvinte functionarea TCP-ului este dependenta in egala masura de controlul fluxului si de fiabilitatea ,iar atunci cand unul din acesti factori nu functioneaza mecanismul nu functioneaza.

Sequence number este raspunzator de gestionarea conexiunii DCCP.Trebuie avut in vedere ca modificarile in protocolul de stari pot fi determinate si de schimbari sensibile ale semanticii fluxului de date.

***TCP sequence numbers***

Dimensiunea sequence number al TCP este de 32 de biti,iar fiecare pachet contine cate un seqno si akcno.

Un ackno cumulativ se comporta ca un istoric al conexiunii numarand cate seqno au fost primite de respectivul ackno.Ackno nu contine informatii despre momentul in care datele sunt receptionate.Din acest motiv pot aparea transmisii redundante ale datelor.Pentru a evita astfel de transmisii redundante se folosesc unii algoritmi de centralizare care presupun ca data a fost receptionata.Exemple de alg de centralizare(fast recovery,NewReno,limited trasmit).In cazul in care transmitatorului i s-ar da un feedback cu privire la faptul ca data a fost receptionata ar ajuta la evitarea acestor transmisii redundante si la evitarea folosirii alg de centralizare.

Sequence number ,in cazul TCP,poate contine valorile ferestrelor de congestie sau receptie sau poate contine octeti ai aplicatiilor de date.In cazul in care seqno contine valori ale ferestrei de congestie sau de receptie se pot indica incercari de subminare ale controlului congestie , iar in cazul in care seqno contine octeti ai aplicatiilor de date un endpoint poate recunoaste o parte din continutul pachetului.

 Caracteristicile conexiunilor TCP care trebuiesc cunoscute sunt configurarea conexiunii,inchiderea conexiunii ,rapoartele ECN etc.Fiecare caracteristica in parte trebuie sa aiba propriul mecanism ack.In cazul configurarii si deconfigurarii conexiunii mecanismul ack este gestionat de bitii SYN si FIN.

***DCCP sequence numbers***

 Spre deosebire de seqno al TCP seqno al DCCP se ocupa cu masurarea datatagramelor nu a octetilor.Acest lucru se intampla deoarece in protocoalele nefiabile sunt analizate datagrame si nu octeti.Folosirea datagramelor in loc de octeti are si avantajul ca usureaza munca algoritmilor de congestie.

 DCCP are printre altele si urmatoarele obiective:configurarea si intreruperea conexiunii ,controlul congestiei ack si controlul congestiei in banda.Obiectivul principal este detectarea ack-urilor piredute,iar cel secundar recunoasterea negocierilor caracteristice.S-a incercat atingerea obiectivelor prin includerea SYN si FIN in sequence space insa aceasta alegere a avut efecte secundare si anume un singur sequence space contine si pachetele de date si ack-urile.In cele mai multe cazuri acestea ar trebui sa fie separate.DCCP ar trebui sa nu reduca rata emitatorului cand un ack este pierdut asa cum face TCP.In cazul pachetelor care nu contin date ar trebui introdus un sequence number secundar.

 DCCP ,fiind un protocol de tratare a datagramelor de fiabile,nu are un ACK cumulativ ca in cazul TCP.De aici rezulta ca nu avem un istoric al conexiunilor.In DCCP ACK semnalizeaza ultimul pachet receptionat si nu primul nereceptionat.Totodata in pachetele DCCP se gasesc un ack si un numar de ordine.Algoritmii DCCP vor avea de rezolvat si problema punerii in ordine a pachetelor in cazul in care acestea nu au fost trimise in ordine.

***Sincronizarea***

Este o problema delicata in cazul DCCP din urmatorul motiv.In cazul unei caderi a retelei fiecare pachet care este trimis foloseste un sequence number nou.Problema este ca atunci cand reteaua redevine functionala endpoint-urile se desicronizeaza deoarece ele asteptau un anumit sequence number iar datagramele vin cu sequence number diferit fata de ce se astepta.TCP are un avantaj in cazul unei caderi a retelei pentru ca retransmite pachetele si foloseste sequence number asteptate.

In cazul DCCP solutia este sincronizarea explicita.Endpointu-ul care a primit un sequence no neasteptat trimite catre celelalt endpoint un semnal de sincronizare cerandui partenerului sa valideze seqno neasteptat.Dupa ce al doilea endpoint proceseaza semnalul primit acesta raspunde cu un pachet SyncAck.Cand endpointul original primeste semnalul ambele terminale isi updateaza ferestrele cu seqno asteptat pe baza SyncAck.Se poate vedea acest lucru si in figura3.



***Acknoledgment***

In cazul TCP avem nevoie de ACK cumulativ pentru a avea un instoric al conexiunii.In cazul DCCP nu se asteapta ACK din partea destinatarului pentru ca este un protool cu constrangeri de timp in care datele noi sunt preferate celor vechi.

De exemplu in cazul streaming-ului media receptorul prefera datele noi celor vechi deoarece daca acesta se blocheaza pentru o vreme va avea in coada o cantitate foarte mare de pachete pe care nu le va putea gestiona.Acesta prefera ca la deblocare sa ignore pachetele vechi si sa le receptioneze pe cele noi.

***Sequence number length***

Marimea sequence number influienteaza direct latimea de banda si starea terminalelor.De exemplu cu cat avem sequence number mai mici cu atat avem latime de banda .In cazul TCP se observa ca ,datorita antetului de 32 bti,are probleme la viteze de ordinul gigabitilor.DCCP a inceput cu un antet de 24 biti insa acesta nu facut fata deoarece se trimiteau foarte putine pachete.De exemplu la un flux de 10gb/s a cate 1500 de bytes /pachet se trimit doar 224 de pachete pe secunde.Solutia gasita a fost dublarea dimensiunii antetului.In acest caz o conexiune care foloseste pachete de dimensiune 1500 de octeti ar avea un flux de date mai mare de 14 pentabiti pe secunda.

Cu toate ca DCCP are antetul de 48 de biti exista si pachete care pot avea un sequence number de 24 de biti insa receptorul le interpreteaza ca fiind pe 48 de biti.[1]

**PERFORMANTE**

***Controlul congestiei TCP***

In figura 4 se observa ca la formarea unei noi conexiuni TCP fereastra de congestie(CongWin) este initializata cu 1MSS.Rata de transmisie initiala este 1MSS/RTT urmand ca la fiecare RTT emitatorul sa dubleze valoarea ferestrei de congestie.In urma dublarii valorii ferestrei de congestie rata de transmisie capata o caracteristica exponentiala.Aceasta caracteristica nu creste la infinit,algoritmul TCP comportandu-se diferit in cazul unui timeout fata de cazul a 3 ack-uri duplicat.In cazul unui timeout fereastra de congestie (CongWin) este readusa la valoarea 1MSS.Aceasta resetare se mai numeste si faza startului lent.In cazul in care se receptioneaza un triplet de ack-uri valoarea la care a ajuns CongWin este redusa la jumatate iar caracteristica devine una liniara.Aceasta faza in care CongWin este redusa la jumate urmand ca apoi sa creasca liniar se numeste fast recovery.



***Controlul congestiei DCCP***

La DCCP mecanismul de control al congestiei este ales de catre utilizator.Mecanismul de control ales de utilizator si implementat de DCCP se stabileste intre cele doua terminale in timpul initializarii conexiunii.Octetul care defineste mecanismul se numeste CCID(Congestion Control Identifier).Dintre CCID-uri cel mai bine definite sunt CCID2 si CCID3.

CCID2 se aseamana cu mecanismul de control al TCP.Acesta se preteaza pentru aplicatiile care necesita latime mare de banda si care se pot adapta schimbarilor ferestrei de congestie.Caracteristicile CCID2 sunt urmatoarele:

 1.Ack-urile duplicat indica pierderea unor pachete de date.

 2.Transmitatorul are optiunea de timeout.Acesta calculeaza timpul dus intors al unei ferestre.Pentru a mentine acest timp calculat se foloseste alg TCP-ului.

In cazul in care apare congestia fereastra de congestie este redusa la jumatate.

CCID3 este un mecanism in care emitatorul mentine rata de transmisie in functie de cantitatea de date nereceptionate de aici rezultand o rata de transmisie mare .Emitatorul DCCP calculeaza rata de transmisie dupa ecuatia:



Unde:

T= rata de transmisie in octeti/secunda.

s = marimea pachetului in octeti

R= timpul dus intors in secunde

b = numarul de pachete confirmate de un singur TCP ack.

p= rata pierderilor

In aceasta lucrare au fost comparate performantele TCP si CCID2 CCID3 ale DCCP bazate pe transfer.Performanata a fost comparata prin transmisia de pachete fixe pentru ca aplicatiile video/audio transmit pachete fixe.Au fost variate ratele de intarziere si de pierdere.

***Testarea***

 Pentru aceste experimente a fost compilat kernelul linux si au folosite tool-uri ca iperf ,GnuPlot si Netem.



Configurarea a doua masini este ilustrata in figura de mai sus.Una dintre masini functioneaza ca serverul DCCP si cealalta masina functioneaza ca client DCCP.Masina client este este folosita pentru a emula schimbarile din retea.Serverul actioneaza ca o ‘chiuveta’.Aceste conditii ale retelei sunt aplicate prin interfata masinii client folosind functionalitatile Netem.

***Timpul vs Rata de bit.***

 Rata de transmisie a TCP,CCID2 si CCID3 in intervalul de timp de o secunda si timpul total de transmisie de 10 secunde sunt aratate in graficul de mai jos:



Din grafic se trag urmatoarele concluzii:

CCID2 are variatii bruste ale ratei de bit in intervalele 1-2 si 8-9 ale axei timp.CCID3 si TCP nu prezinta variatii bruste ale ratei de bit.In cazul TCP se observa ca nu exista pierdere de date intre emitator si receptor.Acest lucru este indicat de faptul ca rata de bit a fost constanta in intervalul de 10 secunde.

***Delay Variabil.***

In figura urmatoare se arata comportamentul TCP –ului cu intarziere variabila.Pe axa x e timpul iar pe axa y rata de bit in Mbps.Graficul arata rata de bit cu delay de 50,100,150,200 de ms.

In grafic se observa ca atunci cand avem un delay mic nu avem schimbari mari in rata de bit si de aici caracteristica TCP ca o linie dreapta.Cu cat se creste delay-ul rata de bit scade. 

In fig 7 si 8 este reprezentat comportamentul cu delay pentru CCID2 si CCID3.Pe axa e x este timpul in secunde si pe axa y rata de bit in MBPS. Graficul arata rata de bit la delay-uri de 50,100,150,200 de ms .Aici este o mica diferenta intre graficul pentru TCP si graficul pentru CCID2 si CCID3.Rata de bit pentru CCID2 si CCID3 este mult mai mica la un delay de 200ms.



 ***Variatia pierderii***

Fig 9 arata comportamentul TCP atunci cand variem rata de pierderi.Axa X arata timpul in secunde si axa Y arata rata de bit in Mbps.Graficul arata rata de transfer a TCP cu rate de pierderi de 0 , 1, 3 , 5 , 10 si 15 %.Cand graficul inregistreaza o scadere se poate spune ca in acel moment a aparut o pierdere si transmitatorul TCP isi reduce rata de transmisie.Se poate vedea foarte clar pe grafic cand apare pierderea maxima.



Figura 10 arata comportamentul CCID2 in cazul unei rate de pierderi care variaza.



Figura 11 arata comportamentul CCID3 in cazul in care rata de pierderi variaza.Axa X arata timpul in secunde si axa Y arata rata de bit in Mbps.Graficul arata rata de bit in functie de urmatoarele valori ale ratei de pierdere 0,1,3,5,10,15 %.Caderea brusca a graficului este mai putin evident decat in cazul TCP.Dar daca comparam graficele pentru CCID2 si CCID3 putem observa ca CCID3 cade mail in decat CCID2.



Daca se integreaza graficele CCID2 si CCID3 vom avea graficul din figura urmatoare:



Rata de bit a graficului arata caracteristicile CCID2 si CCID3 cu pierderi 10%.Din acest grafic putem vedea ca graficul CCID2 este mai neregulat decat graficul CCID3 care este mai uniform.

Graficele care au fost prezentate au fost facute prin emularea retelei.In acelasi mediu si in aceleasi conditii ale retelei rata de bit se poate schimba si astfel graficul comportarii se poate schimba si el.[2]

**CONCLUZII**

Este prea devreme sa se spuna daca DCCP va avea succes in implementarea la scara larga.Chiar daca au aparut unele implementari in Linux ,inca nicio aplicatie nu il foloseste ca protocol principal de transport.

Se poate spune ca proiectarea unei alternative nefiabile a TCP este mai degraba un process simplu.Totusi controlul congestiei al TCP este atat de strans legat de semantica sa incat putine mecanisme TCP sunt aplicate direct fara schimbari substantiale.

TCP manageriaza un design integrat din doua motive.Primul ,abstractizarea bytestream-ului este foarte simpla.Cu exceptia pointerului urgent TCP nu are are nevoie sa ia in considerare semantic aplicatiilor.Al doilea,Tcp este capabil sa faca un bootstrap propriei sale fiabilitati.De exemplu ack cumulative in TCP serveste multor scopuri incluzand fiabilitatea ,controlul fluxului si controlul congestiei.Un protocol nefiabil nu isi permite acest lux si nu pare sa existe un mecanism echivalent cu ack cumulativ.

Totusi este posibil sa se proiecteze un protocol simplu care manageriaza robust controlul congestiei conexiunilor nefiabile.Sincronizarea explicita si noi formate de ack inca au cateva avantaje fata de echivalentele din TCP.Mecanismele de congestie modularizate fac posibila adaptarea controlului congestiei in cadrul unui protocol de retea in care constrangerile de retea si de aplicatie se schimba.[1]

**Bibliografie**

**[1]***Eddie Kohler ,Mark Handley and Sally Floyd***Designing DCCP: Congestion Control Without Reliability**

**[2]***Iffat Sharmin Chowdhury, Jutheka Lahiry, Kazi Chandrima Rahman and Syed Faisal*

*Hasan* **Performance Analysis of Datagram Congestion Control Protocol (DCCP*)***