

DCCP(Datagram Congestion Control Protocol)

Av Per Ivar Ardø

TDAT2004 Datakommunikasjon og nettverksprogrammering

Innledning

DCCP ligger i Transportlaget i OSI modellen og har som oppgave å transportere mellom to maskiner som om de var direkte tilkoblet. For å få til dette er det tre standarder for overføring av pakker de to mest kjente er TCP og UDP.

Disse er pålitelig(TCP) og upålitelig(UDP) måter og transportere pakker over nettet på. TCP vil sende pakker og be om at mottaker sender kvitteringer på at pakkene er motatt for å dobbelt sjekke at alt er sendt og motatt, mens UDP vil sende og ikke bry seg med å sjekke om alt kommer fram eller ikke. De siste årene har dataspill, og streaming blitt ett ganske stor del av livet til folk med tjenester og prgrammer som VLC mediaplayer, Youtube, Netflix og Twitch, og dataspill som Overwatch og Battlefield 1 foreksempel krever at mye data skal overføres mest mulig selv med begrenset data kapasitet. En må fremdeles kunne sende og motta nok til at det som kommer på mottakeren sin side fremdeles kan tolkes som ett bilde eller en god represnatsjon av hva motstaderen ser på andre enden.

Ønsketfunksjonalitet

Vi ønsker å kunne se video og livestreamer. Se filmer, serier eller spille spill over internet med best mulig kvalitet og presisjon uten for mange pakke tap. Unngå venting mellom sending og kvitering med dårlig forbindelser. Oppnå best mulig kvalitet på videoer eller få så små avvik i online spill som mulig. Her er DCCP løsningen, men DCCP er en kombinasjon av de to andre protokollene som vi skal se på nå.

Virkemåte

En av formålene bak å lage DCCP. Er at det ikke skal være noen diskusjon om å bytte fra UDP som er den tradisjonelle måten å sende og motta pakker for streaming og gaming. DCCP er lagt opp sånn at isteden for å bare sende og «håpe på det beste» eller gjøre en threeway handshake som TCP gjør ved at klient ber om å SYN, vente på SYN ACK fra tjener også sende ACK til slutt og sende kvitteringer mellom hver byte transfer. Om at hver byte er mottatt og tilslutt avslutte forbindelsen. DCCP kan derfor virke ganske dramatisk i sammenligning. DCCP Klient ber om respons fra Server. Etter at Server har svart, gjennomføres det data overføringen fra klient og server, men i overføringen blir det ikke bare gjennomført data overføring men både data, ACK og Data ACK in en og samme operasjonen og deretter avsluttes forbindelsen men det varierer om hvor mye det skal sendes ACK meldinger. Så DCCP kan gjøre sjekk i data overføringen i en og samme operasjonen.

DCCP blir dermed en kombinasjon av TCP og UDP som ikke bare funker for streaming, men også for online spill. Siden det som regel trengs så mye data trafikk som mulig uten å forhindre at andre kan bruke nettet på samme nettverk eller domene er DCCP ypperlig for disse formålene. Det at DCCP er såpass fleksibel at den kan justeres for tilgjengelig båndbredde gjør det til det åpenbare valget for scenarioene beskrevet. Det er viktig å merke seg at DCCP ikke er en trygg forbindelse såhvem som helst kan snike inn på en forbindelse. Så det er ikke alle bruksområder hvor DCCP lønner seg.

Pakkestruktur

DCCP pakke formatet består av Source Port, Destination Port, Data Offset, CCVal, CsCov,, Checksum, Res, Type, X, Sequece Number(high) og Sequence Number(low bits). Avhenging av en verdi som kalles X vil Generic Headeren være 16 bytes med 48 bits Sequece Number (hvis x er 1) og 12 bytes med 24 bits Sequece Number(hvis X er 0).

DCCP Pakkestruktur

Offsets	Octet	0							1								
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	Source port															
2	16	Destination port															
4	32	Data Offset							CCVal			CsCov					
6	48	Checksum															
8	64	Res	Type				X=1	Reserved									
10	80	Sequence Number (high bits)															
12	96	Sequence Number															
14	112	Sequence Number (low bits)															

Hvis X er lik 1

Offsets	Octet	0							1								
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	Source port															
2	16	Destination port															
4	32	Data Offset							CCVal			CsCov					
6	48	Checksum															
8	64	Res	Type				X=0	Sequence Number (high)									
10	80	Sequence Number (low bits)															

Hvis X er lik 0

I option har vi også feltet Reserved som avgjør hva pakken skal sende dette kan avhenging av bit verdi kan være:

0. DCCP-Request
1. DCCP-Response
2. DCCP-Data
3. DCCP-Ack
4. DCCP-DataAck
5. DCCP-CloseReq
6. DCCP-Close
7. DCCP-Reset
8. DCCP-Sync
9. DCCP-SyncAck
10. Reserved(dekker bit verdi 10 til 15 disse skal «ignoreres» av mottaker. Mottaker skal ikke sende tilbakemelding om at pakken er motatt hvis en motar en reserved pakke)

Det som også er mulig med DCCP er at det kan være halvforbindelser der det kan sendes data pakker fra en side samtidig som det sendes ACK andre vegen mellom DCCP klient A og B.

Per Ivar Ardø

Kilder:

https://en.wikipedia.org/wiki/Datagram_Congestion_Control_Protocol

<https://wiki.wireshark.org/DCCP>

<https://howdoesinternetwork.com/2013/dccp-datagram-congestion-control-protocol>

<https://tools.ietf.org/html/rfc4340>