



Obligatorisk uppgift 2

Chattapplikationsprotokoll

1 Introduktion

Detta chattapplikationsprotokoll används för att starta, använda och avsluta en chatttjänst där chattklienter kan skicka meddelanden till varandra via en chattserver. Kommunikationen i chattsessionen är av typen ”många-till-många”. När en chattklient startar kan den antingen direkt kontakta en chatserver eller en namnserver. För att en chattklient ska hitta en chatserver kan en namnserver användas. När en chattserver startar registrerar den sig hos en namnserver och så länge den är igång informerar den namnservern regelbundet att den lever. När chattklienten kontaktar en namnserver får den en lista på chattserverar som den kan ansluta sig till. När förändringar sker i deltagandet av chattsessionen så meddelar chattservern det till alla deltagande chattklienter.

Protokollet använder sig av tjänster från transportprotokollen UDP och TCP.

I avsnittet *Översikt* definieras termer som används i detta dokument samt visas en modell för kommunikation mellan enheter inom protokollet samt vilka tjänster protokollet tillhandahåller. Därefter följer två avsnitt som beskriver varsitt gränssnitt. Det första avsnittet, *”Peer-to-peer”-gränssnitt*, beskriver och definierar kommunikationen mellan enheter inom protokollet och det andra avsnittet, *Servicegränssnitt*, beskriver de tjänster som protokollet tillhandahåller.

2 Översikt

I detta avsnitt definieras ett antal termer som används i dokumentet. Vidare visas en övergripande modell för ingående delar samt vilka Protocol Data Units (PDU:er) som skickas mellan vilka delar. Syntax och semantik för dessa PDU:er samt regler för när och hur dessa PDU:er skickas beskrivs senare i avsnittet *”Peer-to-peer”-gränssnitt*. I den övergripande modellen visas också vilka tjänster som tillhandahålls av detta protokoll. Dessa tjänster beskrivs mer ingående senare i avsnittet *Servicegränssnitt*.

2.1 Terminologi

Här nedan följer ett antal begrepp och förkortningar som används i dokumentet.

Klient	En chattklient som ingår i en chattsession.
Server	En chattserver som är den server som administrerar en chattsession.
Namnserver	Namnservern erbjuder en uppslagningstjänst där servrar kan registrera sig och där klienter kan hitta servrar.
Session	En chattsession. Administreras av en server. I en session kan ett antal klienter ingå.
Användare	En användare nyttjar protokollets tjänster.



PDU	Protokolldataenhet (Protocol Data Unit på engelska). Ett paket (meddelande) med information som skickas mellan två enheter inom protokollet. En PDU skickas från en enhet till en annan enhet via underliggande lager. I en PDU finns applikationsdata samt ytterligare information som beskriver hur bifogad applikationsdata ska behandlas. Sektionen för applikationsdata kallas payload eller body. Den ytterligare informationen ligger oftast i en sektion som kallas header, kan även ligga efter data och kallas då footer eller trailer.
Operation	En händelse som sker inom protokollet. Innebär i de flesta fall att en PDU skickas till en annan enhet.
OP	Operationskod. En siffra som anger vilken operation som avses. Vilka operationer som finns, deras operationsnummer och en kort förklaring i protokollet framgår av tabell 1.

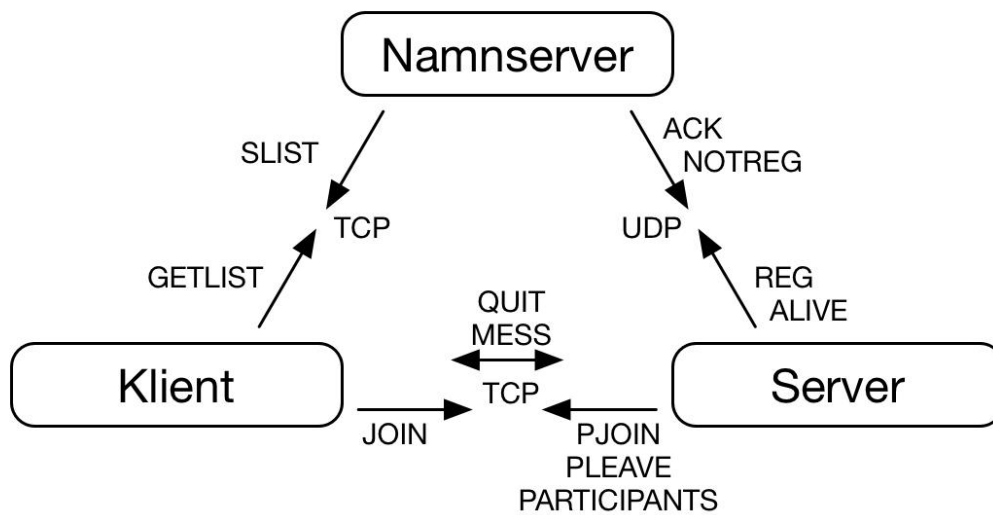


Tabell 1: Kort beskrivning av operationer i protokollet.

Operation	Kod	Beskrivning
Server - namnserver		
REG	0	Skickas från en server när den vill registrera sig hos en namnserver
ACK	1	Skickas som svar på REG och ALIVE
ALIVE	2	Skickas från server till namnserver för att fortsätta vara registrerad
NOTREG	100	Skickas som svar på ALIVE om servern inte är registrerad
Klient - namnserver		
GETLIST	3	Skickas från klient till namnserver för att få en lista med serverar
SLIST	4	Svar på GETLIST, innehåller lista med serverar
Klient - server		
MESS	10	Meddelande mellan klient och server, kan skickas från både klient och server
QUIT	11	Skickas mellan klient och server för att indikera stängd anslutning, kan skickas från både klient och server
JOIN	12	Skickas från klient till server för att ansluta till sessionen
PJOIN	16	Skickas från server till klient för att indikera att en ny deltagare anslutit till sessionen
PLEAVE	17	Skickas från server till klient för att indikera att en deltagare lämnat sessionen
PARTICIPANTS	19	Skickas som svar på JOIN, innehåller lista med deltagare i sessionen

2.2 Modell

Chattjänsten består av tre huvuddelar: en klient, en server och en namnserver. I figur 1 framgår hur de tre delarna kommunicerar med varandra. I kommunikationen mellan server och namnserver används UDP, mellan klient och namnserver används TCP, samt mellan klient och server används TCP. I figur 1 framgår även vilka operationer som skickas mellan delarna.



Figur 1: Översikt över kommunikationen mellan klient, server och namnserver.

2.2.1 Server

När servern startar ska den registrera sig hos namnservern så att klienter kan hämta information om servern från namnservern. Servern har två huvuduppgifter: dels att hålla koll på vilka klienter som är uppkopplade mot servern (och därmed med i sessionen) inklusive deras identitet och adresser, samt dels att distribuera meddelanden från klienter till samtliga klienter i sessionen. Servern ska för varje klient hålla reda på identiteten och TCP-anslutningen för klienten. Under själva körningen väntar servern på att någon klient ska skicka ett meddelande. Beroende på typen av meddelande ska servern sedan skicka information om detta meddelande till alla klienter i sessionen.

2.2.2 Klient

När klienten startar kan den hämta en lista med servrar inklusive deras IP-adresser och portnummer från en namnserver. Sedan ska den ansluta till en av dessa servrar. Alternativt används inte namnservern utan klienten ansluter till en given server. När den är ansluten ska den kunna skicka och ta emot meddelanden, samt kunna lämna sessionen. Det finns inget krav på att kolla att en klientidentitet är unik i chattsessionen när en klient ansluter till chattsessionen.

2.2.3 Namnserver

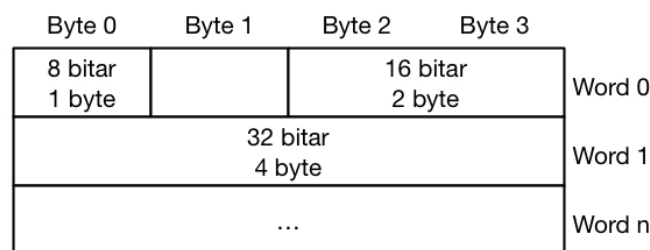
Alla servrar ska registrera sig hos en namnserver för att klienter ska kunna hitta dem. Sedan ska servrarna kontinuerligt meddela att de fortfarande är aktiva (ett s.k. heart-beat). Namnservern har en lista med aktiva servrar som uppdateras kontinuerligt. Klienter kan hämta listan med aktiva servrar från namnservern.

3 "Peer-to-peer"-gränssnitt

I detta avsnitt beskrivs hur kommunikation sker mellan enheter inom protokollet. Först beskrivs generellt PDU:er och begränsningar i protokollet. Därefter beskrivs de tre kommunikationsvägarna mellan klient, server och namnserver. För varje kommunikationsväg redogörs för de olika PDU:ernas funktionalitet och varje avsnitt avslutas med beskrivning av respektive PDU. Slutligen redogörs för felhantering och övriga krav.

3.1 Generellt om PDU:er

I all kommunikation är meddelandena indelade i PDU:er. Här nedan beskrivs samtliga PDU:er i detalj, deras syntax och semantik, samt när och hur de skickas. Det generella utseendet för dessa PDU:er framgår av figur 2.



Figur 2: Formatbeskrivning för en generell PDU.

Dessa konventioner och format enligt figur 2 gäller i samtliga PDU:er. PDU:er består av ett eller flera words där varje word består av 4 bytes

3.1.1 PDU:ers generella struktur

Värdet på den första byten i varje PDU bestämmer vilken typ av PDU det är (PDU:ns OP-kod), och utifrån denna byte kan formatet på PDU:n bestämmas. Alla OP-koder visas i Tabell 1.

Varje PDU består av flera fält som används i kommunikationen. Det finns två typer av fält. Den första typen består av ett förutbestämt antal bytes och en förutbestämd position i varje PDU, dessa fält ryms alltid inom ett enda och samma word.

Den andra typen är ett fält med obestämd längd som kan överstiga längden hos ett word, dessa fält börjar alltid i början på ett word och i slutet fylls de alltid ut med bytes med värdet 0 så att de alltid slutar i slutet på ett word. Deras längd anges alltid i ett annat fält i en PDU. Förutom i slutet på de längre fälten, kan det finnas fasta luckor i PDU:er. Dessa ska "paddas", d.v.s. fyllas ut med bytes med värdet 0.

Samtliga nummer-fält, såsom längddefinitioner av andra fält, portnummer, id-nummer, är samtliga unsigned integers av storlek som är förutbestämd av PDU:n.

3.1.2 Data i PDU:er

Data i samtliga PDU:er ska alltid vara lagrat i "network byte order", vilket är samma som "Big Endian order". All text som överförs i kommunikationerna ska vara i UTF-8. Samtliga tal som skickas i protokollet, t ex portnummer, ska vara unsigned, dvs ej teckensatta utan enbart positiva. Strängar ska inte null-termineras, d.v.s. behövs det 4 bytes för att representera en sträng så ska fältet som strängen ryms i vara 4



bytes. Undantaget är PARTICIPANTS som innehåller en lista av identiteter på sessionens deltagares. Där ska varje identitetsnamn null-termineras, inklusive det sista. När det sista nicknamet null-terminerats ska fältet paddas så att det precis slutar i slutet på ett word.

3.1.3 Tidsstämplar i PDU:er

Tidsstämplar anges i Unix-tid, d.v.s. i antal sekunder sedan 00:00:00 UTC första januari 1970. De flesta programmeringsspråk har stöd för att ta fram Unix-tid direkt. Tidsstämplar anges i 32 bitar.

3.1.4 Checksummor i PDU:er

Checksummor beräknas med 8-bitars storlek. Det innebär att alla bytes summeras, och så fort summans värde överstiger 255 subtraheras 255 från summan. Checksumman blir sedan det bitvisa komplementet till summan. Det innebär att om checksumman beräknas igen över hela meddelandet inklusive checksumman ska resultatet bli 255.

3.2 Begränsningar i protokollet

Detta protokoll har vissa begränsningar:

- Ett identitetsnamn kan inte vara längre än 255 tecken (bytes).
- Ett servernamn kan inte vara längre än 255 tecken (bytes).
- En namnserver kan bara hålla koll på 65535 servrar.
- En server kan bara ha 255 anslutna klienter i sessionen.
- Ett meddelande kan inte vara längre än 65535 tecken (bytes).
- All text skall representeras i UTF8.

3.3 Server - namnserver

All kommunikation mellan server och namnserver sker via UDP. När en server ska registrera sig hos en namnserver skickar den först en REG. Som svar får den sedan en ACK med ett unikt ID. Därefter ska servern regelbundet skicka en ALIVE med samma ID. Som svar får den sedan en ACK med sitt eget unika ID. Om namnservern inte får någon ALIVE på 20 sekunder avregistreras servern och ytterligare ALIVE kommer besvaras med NOTREG. Då ska servern registrera om sig med en ny REG. För att undvika att avregistreras ifall ett UDP-datagram försvinner, och då det kan ta lite tid för meddelandena att komma fram skickas en ALIVE var 8:e sekund. Om servern blivit avregistrerad får den skicka ett nytt REG.

3.3.1 REG

Skickas från server till namnserver för registrering av servern (se figur 3).

- **Servernamn-längd:** Längd (0-255) i UTF-8-representation av servernamn (ej paddat, ej null-terminerat).
- **TCP-port:** Port där servern accepterar klienter (0-65535).
- **Servers namn:** UTF-8-representation av servers namn (paddat, ej null-terminerat).

OP: REG	Servernamn-längd	TCP-port
Servers namn		

Figur 3: Beskrivning av en REG-PDU.

3.3.2 ALIVE

Skickas från server till namnserver för att fortsätta vara registrerad (se figur 4).

- **Antal klienter:** Antal klienter (0-255) i sessionen.
- **ID-nummer:** ID-nummer (0-65535) som servern fick som svar på REG-meddelandet.

OP: ALIVE	Antal klienter	ID-nummer
-----------	----------------	-----------

Figur 4: Beskrivning av en ALIVE-PDU.

3.3.3 ACK

Skickas från namnserver till server som bekräftelse på REG eller ALIVE (se figur 5).

- **ID-nummer:** ID-nummer (0-65535) som servern ska använda i sina ALIVE- meddelanden.

OP: ACK	Pad	ID-nummer
---------	-----	-----------

Figur 5: Beskrivning av en ACK-PDU.

3.3.4 NOTREG

Skickas som svar på ALIVE om ingen server med det aktuella ID:t är registrerad (se figur 6).

- **ID-nummer:** ID-nummer (0-65535) som servern skickade i sitt ALIVE- meddelanden.

OP: NOTREG	Pad	ID-nummer
---------------	-----	-----------

Figur 6: Beskrivning av en NOTREG-PDU.

3.4 Klient - namnserver

För att begära en serverlista skickar klienter en GETLIST till en namnserver via TCP. Som svar kommer en SLIST.

3.4.1 GETLIST

Skickas från klient till namnserver för att begära en lista med aktiva servrar (se figur 7).

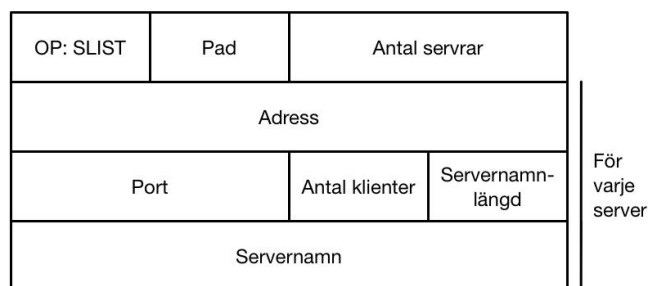
OP: GETLIST	Pad
----------------	-----

Figur 7: Beskrivning av en GETLIST-PDU.

3.4.2 SLIST

Skickas som svar på GETLIST (se figur 8).

- **Antal servrar:** Antal servrar (0-65535) i PDU:n.
- För varje server:
 - o **Adress:** IPv4-adress (4-fält å 0-255) till servern.
 - o **Port:** Portnummer (0-65535) till servern.
 - o **Antal klienter:** Antal klienter (0-255) anslutna till den specifika servern.
 - o **Servernamnlängd:** Längd (0-255) i UTF-8-representation av servernamn (ej paddat, ej nullterminerat).
 - o **Servernamn:** UTF-8-representation av servers namn (paddat, ej null-terminerat).



Figur 8: Beskrivning av en SLIST-PDU.

3.5 Klient - server

Kommunikation mellan klient och server sker via en TCP-anslutning. När anslutningen öppnats skickar klienten en JOIN och får som svar en PARTICIPANTS. Därefter kan klienten skicka MESS för att skicka meddelanden. Dessutom kan anslutningen avslutas genom att skicka en QUIT. Alla andra PDU:er från en klient är ogiltiga och ska besvaras med en QUIT följt av att servern stänger anslutningen.

När en klient har anslutit till sessionen (efter att PARTICIPANTS skickats) kan servern skicka:

- MESS om någon annan klient skickat ett meddelande
- PJOIN till alla som var med sedan tidigare i session (d.v.s. ej till den nya klienten)
- PLEAVE om en annan klient lämnat sessionen
- QUIT om servern avslutas

När en klient skickar MESS till servern ska meddelandet inte ha någon tidsstämpel eller någon avsändare. Detta läggs till i servern så att klienter inte ska kunna ange ett annat namn eller en annan tid än den korrekta. När servern tar emot meddelanden är det viktigt att de skickas till alla klienter i sessionen i samma ordning.

3.5.1 JOIN

Skickas från klient till server för att ansluta till sessionen (se figur 9).

- **Identitetslängd:** Längd (0-255) i UTF-8-representation på namnet på klienten (ej paddat, ej null-terminerat).
- **Identitet:** UTF-8-representation av identiteten på klienten (paddat, ej null-terminerat).

OP: JOIN	Identitets- längd	Pad
Identitet		

Figur 9: Beskrivning av en JOIN-PDU.

3.5.2 PARTICIPANTS

Skickas som svar på JOIN om klienten tillåts ansluta till sessionen (se figur 10).

- **Antal identiteter:** Antal klientidentiteter (0-255) i PDU:n.
- **Längd:** Längd av klientidentiteter (0-65535) i UTF-8-representation inklusive null-terminering av varje klientidentitet (inklusive sista), exklusive padding.
- **Identiteter:** UTF-8-representation av alla klientidentiteter som är med i sessionen (inklusive klienten som precis anslöt) med en 0-byte efter varje identitet (inklusive efter sista identiteten) (paddat, null-terminerat).

OP: PARTICIPANTS	Antal identiteter	Längd
Identiteter ...		

Participants exempel

OP: PARTICIPANTS	2	9	
P	O	\0	K
A	L	L	E
\0	\0	\0	\0

Figur 10: Beskrivning av en PARTICIPANTS-PDU.

3.5.3 QUIT

Skickas från server eller klient innan anslutningen till den andra parten stängs (se figur 11). Eftersom en TCP-anslutning används skickas inget svar på en QUIT.

OP: QUIT	Pad
----------	-----

Figur 11: Beskrivning av en QUIT-PDU.

3.5.4 MESS

Innehåller ett meddelande, skickas från klient till server eller från server till klienter (se figur 12).

- **Identitetlängd:** Längd (0-255) på avsändarens identitet. Ska vara 0 när PDU:n skickas från klienten.
- **Checksumma:** Checksumma (0-255) enligt beskrivning ovan. Beräknas om av servern när tidsstämpel och klientidentitet lagts till.
- **Meddelandelängd:** Längd (0-65535) i UTF-8-representation (ej paddat, ej null-terminerat).
- **Tidsstämpel:** Unix-tid (0-4294967295, antal sekunder från 1970 till att meddelandet skickades). Ska vara 0 då PDU:n skickas från klienten och sätts av servern.
- **Meddelande:** UTF-8-representation av meddelandet (paddat, ej null-terminerat).
- **Klientidentitet:** UTF-8-representation av klientidentiteten för avsändaren (paddat, ej null-terminerat). Ska inte skickas med från klienten utan läggs till av servern.

OP: MESS	Pad	Identitet- längd	Check- summa
Meddelandelängd		Pad	
Tidsstämpel			
Meddelande			
Klientidentitet			

Figur 12: Beskrivning av en MESS-PDU.

3.5.5 PJOIN

Skickas från server till klient när en annan klient ansluter till sessionen (se figur 13).

- **Identitetlängd:** Längd (0-255) i UTF-8-representation för klientens identitet (ej paddat, ej null-terminerat).
- **Tidsstämpel:** Unix-tid (0-4294967295, antal sekunder från 1970 till att klienten anslöt till sessionen).
- **Klientidentitet:** UTF-8-representation av klientens identitet (paddat, ej null-terminerat).

OP: PJOIN	Identitet- längd	Pad
Tidsstämpel		
Klientidentitet		

Figur 13: Beskrivning av en PJOIN-PDU.

3.5.6 PLEAVE

Skickas från server till klient när en annan klient lämnar sessionen (se figur 14).

- **Identitetlängd:** Längd (0-255) i UTF-8-representation för klientens identitet (ej paddat, ej null-terminerat).
- **Tidsstämpel:** Unix-tid (0-4294967295, antal sekunder från 1970 till att klienten lämnade sessionen).
- **Klientidentitet:** UTF-8-representation av klientens identitet (paddat, ej null-terminerat).

OP: PLEAVE	Identitet- längd	Pad
Tidsstämpel		
Klientidentitet		

Figur 14: Beskrivning av en PLEAVE-PDU.

3.6 Felhantering

Om en klient eller server tar emot en ogiltig eller felaktigt formaterad PDU ska anslutningen som PDU:n skickades över stängas. Servern skickar då ut meddelanden som berättar detta till samtliga anslutna klienter. En konvention som gäller är att meddelanden utan identitetsinformation och med fältet IDLEN satt till 0 är systemmeddelanden från servern. Det är då lämpligt att klienten presenterar detta meddelande på ett speciellt sätt, t.ex. med speciella tecken före/efter texten, för att skilja servermeddelande från vanliga meddelanden.

4 Servicegränssnitt

I detta avsnitt presenteras vilka tjänster som protokollet ska erbjuda till användare av protokollet. En användare av en implementation av protokollet kan bara använda tjänsten via klienten. Server och namnserver används bara internt för att lösa protokollets tjänster.

Först ges en övergripande beskrivning av tjänsterna som protokollet kan tillhandahålla. Därefter beskrivs flödena som sker inom protokollet när de olika tjänsterna används av användaren.

4.1 Tillhandahållna tjänster

De tjänster som tillhandahålls är följande:

- Klienten kan ansluta till en session som administreras av en server. Användaren anger adress och portnummer till den namnserver som ska användas. När klienten kommunicerat med namnservern presenteras listan av servrar (och därmed vilka sessioner det finns) för användaren. Användaren väljer en server (session) att ansluta till.
- Klienten kan ansluta direkt till en session som administreras av en server. Användaren anger då adress och portnummer till den server som ska användas.
- Användaren kan skicka meddelanden till andra deltagare i sessionen.
- Användaren kan läsa meddelanden från andra deltagare i sessionen.
- Användare kan få meddelanden om någon deltagare anslutit eller lämnat sessionen.
- Användaren eller servern kan avsluta kopplingen till sessionen.

4.2 Start av klient

I det följande ges en stegvis beskrivning av vad som händer inom protokollet när en klient startar.

1. Användare → Klient: Starta
 - 1.1. Om användare anger adress och portnummer till namnserver
 - 1.1.1. Klient: Öppna TCP-socket till namnserver
 - 1.1.2. Klient → Namnserver: GETLIST
 - 1.1.3. Namnserver → Klient: SLIST
 - 1.1.4. Klient: Stäng TCP-socket
 - 1.1.5. Klient → Användare: Presentera serverlista
 - 1.1.6. Användare → Klient: Vald server
2. Klient: Öppna TCP-socket till server
3. Klient → Server: JOIN
4. Server → Klient: PARTICIPANTS
5. Klient → Användare: Lista med deltagare

4.3 Klient använder chatttjänsten



I det följande ges en stegvis beskrivning av vad som händer inom protokollet när chattjänsten används efter att klienten anslutit till en session.

1. Användare → Klient: Meddelande
 - 1.1. Användare → Klient: Meddelande
 - 1.2. Klient → Server: MESS
2. Server → Klient: MESS
 - 2.1. Server → Klient: MESS
 - 2.2. Klient → Användare: Meddelande
3. Server → Klient: PJOIN
 - 3.1. Server → Klient: PJOIN
 - 3.2. Klient → Användare: Identitet på deltagare som anslutit till sessionen
4. Server → Klient: PLEAVE
 - 4.1. Server → Klient: PLEAVE
 - 4.2. Klient → Användare: Identitet på deltagare som lämnat sessionen

4.4 Avslutning av klients användning av chattjänsten

I det följande ges en stegvis beskrivning av vad som händer inom protokollet när användandet av chattjänsten avslutas för klienten.

1. Användare → Klient: Meddelande: Avsluta
 - 1.1. Användare → Klient: Meddelande: Avsluta
 - 1.2. Klient → Server: QUIT
 - 1.1. Server: Stäng TCP-socket till denna klient
 - 1.2. Server → Övriga klienter: PLEAVE
 - 1.3. Klient: Stäng TCP-socket
2. Server → Klient: QUIT
 - 2.1. Server → Klient: QUIT
 - 2.2. Server: Stäng TCP-socket
 - 2.3. Klient → Användare: Server har avslutat
 - 2.4. Klient: Stäng TCP-socket