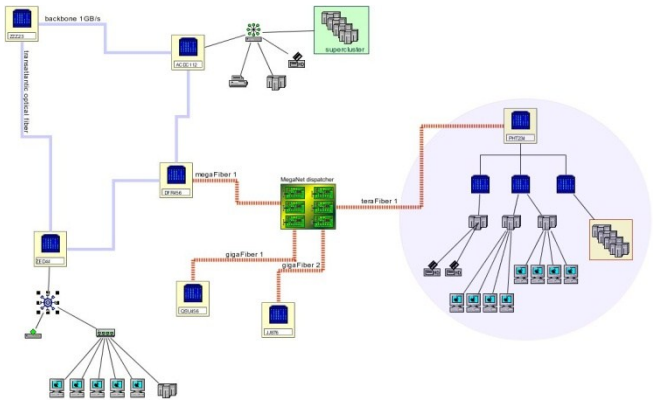
Distributed Systems Chat (DSC)

Protocol & Eisen versie 1, 1 april 2012



Inhoudsopgave

(1) Inleiding

Dit document beschrijft de opdracht voor het practicum gedistribueerde systemen 2011-2012. De opdracht behelst het ontwerpen, bouwen en testen van een server waarmee een backbone van servers voor een gedistribueerd chatsysteem (Distributed Systems Chat, DSC) kan worden gebouwd. Het protocol dat gebruikt dient te worden voor een correcte implementatie van een DSC server vormt hier een belangrijk onderdeel van. De DSC server is een programma dat tijdens het college distributed systems dient te worden ontwikkeld door teams van practicanten.

We realiseren ons dat deze opgaaf op het eerste gezicht veel lijkt op die bij telematica. In de huidige opgaaf wordt echter vooral ook de nadruk gelegd op het bouwen van een robuust

systeem dat de verbindingen weer kan herstellen na uit uitvallen van bepaalde onderdelen. De kennis en ervaring die je bij telematica hebt opgedaan kan je hier natuurlijk wel van pas komen.

Bij deze opgave moet UDP worden gebruikt voor de communicatie tussen de processen.

Voor het programmeren van de server is geen programmeertaal voorgeschreven, deze wordt in overleg met de assistent bepaald. Wel moeten alle ontwikkelde servers en cliënten in het edu netwerk op de FNWI kunnen samenwerken.
Samenwerking. De opdracht moet met groepjes van drie of vier studenten worden gemaakt, waarbij natuurlijk onderling goed en controleerbaar moet worden samengewerkt. In verband met deze controleerbaarheid is aanwezigheid op het practicum verplicht.
Ontwerp. Hoewel hieronder een vrij ver uitgewerkt protocol staat, moet er nog goed worden nagedacht over de afhandeling van de diverse boodschappen zodat een robuust en betrouwbaar systeem ontstaat. Omdat ook wordt geëist dat de gebouwde server kan samenwerken met die die worden gebouwd door de andere groepen, moet het protocol nauwgezet worden geïmplementeerd. Een document met een uitgewerkte analyse van de werking van het protocol en een eerste ontwerp van je server moet aan de assistent worden voorgelegd voordat je begint met programmeren. Je mag pas in de tweede week beginnen met programmeren – ook komt de controlserver (zie onder) pas dan on-line.
Eindproduct. Het eindproduct dat je zal moeten leveren bestaat uit de volgende onderdelen:

- Een demonstratie van de werking van je server op het practicum op dinsdag 15 mei.
- Je hele systeem moet werken op de practicummachines in G0.23-G0.25
- Je mag uitsluitend een van de volgende programmeertalen gebruiken:
  C, C++, Python, Java
- Een verslag, waarin het ontwerpdocument, aangepast op basis van je ervaring bij de implementatie, tevens een korte handleiding en een verslag van de wijze waarop je server is getest. Inleveren op of voor 21 mei via blackboard.
- De goed gedocumenteerde code. Inleveren op of voor 21 mei via blackboard.

(2) Netwerkarchitectuur

Tijdens dit practicum dient een gedistribueerd communicatiemedium gebouwd te worden. Dit houdt in dat servers en cliënten die op verschillende platforms worden ontwikkeld moeten communiceren en samenwerken. De cliënten dienen maar met één server tegelijk actief verbinding te maken. Passieve verbindingen zijn altijd mogelijk. De servers moeten aan elkaar gelinkt kunnen worden in een samenhangende boom (dus zonder cykels). Omdat we in ons protocol niet zullen bijhouden langs welke servers een boodschap al is gekomen, zou een lus in de topologie al gauw tot gevolg hebben dat boodschappen langs die lus blijven rondreizen. Met welk adres (welke reeds aanwezige server) een nieuwe server actief verbinding maakt wordt bepaald door de control server, zie ook verderop.

(3) De cliënt

De cliënt voor dit practicum hoeft niet zelf te worden geschreven, maar is reeds door de practicumleiding ontwikkeld. Er is een goede grafische cliënt. Deze werkt onder linux; er zijn geen windows cliënten. De cliënt is te vinden in /opt/stud/opsys/DS2011 op sremote en de practicummachines.

(4) De Server

De server die door de practicanten dient te worden ontwikkeld, moet aan een aantal eisen voldoen:

- De servers moeten elk verbindingen van meerdere cliënten en meerdere servers toestaan.
- De servers moeten berichten kunnen broadcasten naar alle servers en unicasten per verbonden cliënt.
- De servers moeten unicast berichten naar cliënten die niet aan hemzelf verbonden zijn, kunnen relayen/doorsturen naar de juiste andere server
- De interne database van aan het netwerk verbonden cliënten en servers dient consistent te zijn over alle servers, waarbij aangetekend wordt dat niet alle servers dezelfde informatie hebben.
- De servers dienen te voldoen aan het verderop beschreven protocol, uitbreidingen zijn niet verboden, maar worden afgeraden ivm consistentie.
- De servers dienen een configuratiebestand in te kunnen lezen waarin staat:
  - De poort waarop geluisterd moet worden (standaard 2001)
  - De namen van de beheerders van de server, inclusief wachtwoord (cleartext)
  - Het adres met informatie van de controlserver
  - De door deze server te gebruiken identificatie
  - Eventuele uitbreidingen….

  Het formaat van dit configuratiebestand mag zelf worden gekozen en hoeft niet uniform te zijn over alle servers.
- De server mag niet interactief via de console worden gebruikt, alles dient via het configuratiebestand of via commando's uit de cliënten te geschieden. Ook mag er niets naar stdout/stdin geschreven worden, maar dient er een logfile aangemaakt te worden.
- De server mag zelf maar actief naar één andere server linken, maar er mogen meer servers naar dezelfde linken. Het resultaat is een boomstructuur.

(5) Te verzamelen en gebruiken data

Elke server dient tenminste de volgende data bij te houden voor een goede werking van het protocol:

- Voor elke direct aan deze server "verbonden" server tenminste het socketnummer van die server (dus ouder en kinderen)
- Voor alle andere servers in het systeem: een gedeeltelijke route naar die server, feitelijk de eerste "hop".
- Voor elke cliënt tenminste
  - De authorisatiestatus van de cliënt
  - Als die cliënt direct met de huidige server is verbonden: het socketnummer van de cliënt
  - In andere gevallen: de server via welke de cliënt te bereiken is

(6) Berichten

De uitwisseling van informatie en berichten tussen servers en cliënten alsmede tussen servers onderling geschiedt volgens een vast protocol. Hiervan mag niet afgeweken worden, daar dit communicatie tussen verschillende implementaties moeilijk of onmogelijk maakt.

(6.1) Berichtformaat

De berichten hebben het volgende formaat

- De eerste 16 bits zijn een integer die de lengte van het record inclusief deze 2 bytes aangeeft.
- De volgende 2 bytes zijn een integer die het type bericht aangeeft.
- De volgende 2 bytes zijn een unsigned integer die het volgnummer van de boodschap op het kanaal tussen zender en ontvanger weergeeft.
- De 3 voornoemde integers dienen in network byte order te worden verzonden
- Daarna volgt de rest van het bericht, een bericht is maximaal 200 bytes lang, inclusief de eerste 6 bytes. Deze bytes moeten allemaal "human-readable" ascii karakters bevatten. Dus: letters, cijfers, spaties, en de tekens `` `~!@#$%^&*()-_=+[{]};:'"\|,<.>/? ``
- Een bericht wordt niet afgesloten met een bepaald karakter. Een '\0' of '\n' aan het einde van een bericht is expliciet niet toegestaan.

Fouten: Bij ontvangst van foute data mag onmiddellijk de socketverbinding worden verbroken.

Woord: Een woord zoals verderop gebruikt, is een reeks tekens gescheiden door één of meerdere spaties en/of tabs.

Naam: Een naam voor een cliënt mag alleen bestaan uit letters, cijfers en underscores. De maximale lengte van een naam is 12 tekens. Hoofdletter gevoelig!

Serverident: Een server gebruikt steeds maar één enkel UDP poortnummer, voor zowel zenden als ontvangen. Voorbeeld: 146.50.18.121:2000. Andere partijen kunnen deze ophalen uit de IP header informatie.[1] Omdat servers herstart kunnen worden, genereert daarnaast een unieke tag van 32 bytes. De rol van de tag is ervoor te zorgen dat een herstarte server met hetzelfde ipadres en poortnummer toch als een andere server kan worden herkend. De server moet deze tag dus zelf genereren en wel zo dat hij uniek is. Gebruik als ingrediënten b.v. ipadres, poortnummer en de tijd tot op microseconden.
Je kan b.v. ook gebruik maken van /dev/random of /dev/urandom
Je moet met een herkenbaar stukje tekst beginnen: T*nn*S*procid* voor team nummer *nn* en server met proces id *procid* – dan is jouw server herkenbaarder in de log van de control server.

(6.2) Berichttypen

Hieronder volgt een lijst met de mogelijke berichttypen, tussen welke partijen deze berichten verzonden worden en de data die ze verder bevatten.

Berichttype: Registreer cliënt
Typenummer: 100
Tussen: Cliënt – server
Parameters: naam [wachtwoord]

De rest van het bericht dient 1 of 2 woorden te bevatten: het eerste is de naam van de cliënt, het tweede woord een eventueel wachtwoord. De naam moet uniek zijn voor het hele systeem en de server zal dan ook pas een "registratie gelukt" bericht sturen als dit zo is.

[1] In C b.v. te vinden in de struct sockaddr *`address` in recvfrom()

De ontvangende server moet ip adres en poortnummer van de afzender uit de ip header halen.

Berichttype: Cliënt toegevoegd
Typenummer: 110
Tussen: Server - server en server - cliënt
Parameters: volgnr, lijstlengte, naam
Format v.d. parameters: `%6d%6d%s`

Aan de toegevoegde cliënt of server wordt in een serie berichten van dit type alle namen van verbonden cliënten doorgegeven. Omdat deze boodschappen ook worden gebruikt om een reeks namen door te geven, wordt een volgnummer en een totaal aantal te versturen elementen meegegeven. Volgnummers lopen van 1 – lijstlengte. Gebruik voor de getallen een veldbreedte van 6 tekens.
Bij de aanmelding van een server moeten ook dergelijke boodschappen worden uitgewisseld (zie verderop). Omdat er niet altijd clienten zullen zijn, mogen dan volgnummer en lijstlengte beide nul te zijn.

Berichttype: Cliënt verwijderen
Typenummer: 120
Tussen: Cliënt-server
Parameters: woord*

Dit bericht wordt verzonden als de cliënt-server verbinding verbroken wordt. De rest van het bericht bevat een eventuele tekst die naar iedereen als groet verzonden wordt.

Berichttype: Cliënt verwijderen
Typenummer: 130
Tussen: Server-server, server-cliënt en beheerder-server
Parameters: woord*

Dit bericht wordt verzonden als de cliënt-server verbinding verbroken is. De rest van het bericht bevat de naam van de verwijderde cliënt en een tekst die naar iedereen als groet verzonden wordt. Mocht de cliënt deze niet zelf invullen of wordt de verbinding onverwachts verbroken dan dient de server dit zelf in te vullen

Berichttype: Echo request (ping)
Typenummer: 140
Tussen: Server-server, cliënt-server, server-cliënt, controlserver-server en server-controlserver
Parameters: identificatie

De rest van het bericht dient 1 woord te bevatten: een identificatienummer / tekenreeks. Het antwoord op dit bericht dient binnen 2 seconden verstuurd te zijn, anders mag de verbinding verbroken worden.
Omdat met UDP wordt gewerkt, is het niet altijd zeker of berichten aankomen. Daarom mag indien geen pong wordt ontvangen, het ping-verzoek een paar keer worden herhaald met hetzelfde identificatienummer.

Berichttype: Echo response (pong)
Typenummer: 150
Tussen: Server-server, server-cliënt en cliënt-server, controlserver-server en server-controlserver
Parameters: identificatie

De rest van het bericht dient 1 woord te bevatten: het identificatienummer / tekenreeks van het voorafgaande Echo Request. Bericht dient binnen 2 seconden na het ontvangen van een ping verstuurd te zijn, anders mag de verbinding verbroken worden door de andere partij.

Berichttype: Wijzig naam
T*ypenummer*: 160
Tussen: Cliënt-server
Parameters: newname

Dit bericht wordt verzonden als een cliënt zijn naam wijzigt. Het eerste en enige woord van het bericht is de nieuwe naam. De cliënt mag pas aannemen dat het gelukt is als een correcte statusmelding is ontvangen.

Berichttype: Wijzig naam
Typenummer: 170
Tussen: Server-server, server-cliënt
Parameters: oldname newname

Dit bericht wordt aan alle aangesloten cliënten verzonden als een cliënt zijn naam heeft gewijzigd. Het eerste woord in het datadeel is de oude naam, het tweede de nieuwe naam.

Berichttype: Tekstbericht
Typenummer: 200
Tussen: Cliënt-server
Parameters: bestemming woord*

De rest van het bericht bestaat uit 2 delen: het eerste woord is de naam van degene aan wie het bericht gericht is of de tekst #all als het bericht naar iedereen gaat. De rest is de door de persoon ingevoerde tekst.

Berichttype: Actiebericht
Typenummer: 210
Tussen: Cliënt-server
Parameters: bestemming woord*

De rest van het bericht bestaat uit 2 delen: het eerste woord is de naam van degene aan wie het bericht gericht is of de tekst #all als het bericht naar iedereen gaat. De rest is de door de persoon ingevoerde tekst. Voor de servers is er geen verschil in behandeling van tekstberichten en actieberichten.

Berichttype: Tekstbericht
Typenummer: 300
Tussen: Server-server, server-cliënt
Parameters: bron bestemming woord*

De rest van het bericht bestaat uit 3 delen: het eerste woord is de naam van de afzender, het tweede woord is de naam van degene aan wie het bericht gericht is of de tekst #all als het bericht naar iedereen gaat. De rest is de door de persoon ingevoerde tekst.

Berichttype: Actiebericht
Typenummer: 310
Tussen: Server-server, server-cliënt

Parameters: bron bestemming woord*

De rest van het bericht bestaat uit 3 delen: het eerste woord is de naam van de afzender, het tweede woord is de naam van degene aan wie het bericht gericht is of de tekst #all als het bericht naar iedereen gaat. De rest is de door de persoon ingevoerde tekst.

Berichttype: Registratie gelukt
Typenummer: 500
Tussen: Server-cliënt
Parameters: (geen)

Wordt verzonden als de registratie gelukt is, de rest van het bericht kan leeg zijn.

Berichttype: Registratie mislukt
Typenummer: 510
Tussen: Server-cliënt
Parameters: reden (=woord woord*)

Wordt verzonden als de registratie mislukt is. De rest van het bericht bevat de reden van het mislukken van de inlogpoging.

Berichttype: Naamwijziging gelukt
Typenummer: 520
Tussen: Server-cliënt
Parameters: (geen)

Wordt verzonden als de naamswijziging gelukt is, de rest van het bericht kan leeg zijn.

Berichttype: Naamwijziging mislukt
Typenummer: 530
Tussen: Server-cliënt
Parameters: reden (=woord woord*)

Wordt verzonden als de naamswijziging mislukt is. De rest van het bericht bevat de reden van het mislukken van de poging.

Berichttype: Registreer server
Typenummer: 600
Tussen: Server-server
Parameters: de eigen tag.

Dit verstuurt een server om zich aan te melden bij een andere server. Het is dus de server-server versie van berichtype 100.
De ontvangende server moet ip adres en poortnummer van de afzender uit de ip header halen.

Berichttype: Adres server
Typenummer: 601
Tussen: Server-control server
Parameters: de eigen tag.

Dit verstuurt een server bij aanmelding naar de control server.
De ontvangende server moet ip adres en poortnummer van de afzender uit de ip header halen.

Berichttype: Adres server

Typenummer: 602
Tussen: Control server- server
Parameters: ipadres, poortnummer en tag van de toegewezen parent(in een woord) of “none”

Dit verstuurt een control server bij aanmelding naar de nieuwe server. Parameter is of een serverident en tag, gescheiden door een dubbele punt, of de tekenreeks “none” (zonder “”). Voorbeeld:
146.50.19.123:2001:T*nn*S*procid*qwertyqwertyqwerty
Deze tag geeft dus een unieke identificatie van de “parent” server.

Berichttype: Peer verbinding verbroken
Typenummer: 603
Tussen: Server-control server
Parameters: Tag van de uitgevallen verbinding (in een woord)

Dit verstuurt een server naar de control server als een van zijn partner servers niet reageert. Dat kan dus de parent of een van de kinderen zijn.
De huidige implementatie van de control server neemt een 603 altijd voor waar aan en probeert niet of de betreffende server misschien toch reageert.

Berichttype: Hergroepeer
Typenummer: 604
Tussen: Control server- server
Parameters: tag van de uitgevallen server (in een woord) en eventueel een tweede woord equivalent aan de parameter van een 602 bericht.

Deze wordt door de controlserver verstuurd als een peer niet meer reageert. Wanneer dit bericht ontvangen wordt, dienen alle cliënten die via de dode server verbonden zijn te worden afgemeld. Als de uitgevallen node de huidige parent is zal er een tweede woord aanwezig zijn met de identificatie van de nieuwe parent is. Met deze zal verbinding moeten worden gemaakt om het netwerk te herstellen.

Berichttype: Adres server
Typenummer: 610
Tussen: Cliënt-control server
Parameters: geen

Dit verstuurt een cliënt bij aanmelding naar de control server om een serveradres op te vragen.

Berichttype: Adres server
Typenummer: 611
Tussen: Control server- cliënt
Parameters: ipadres en poortnummer toegewezen server (in een woord) of “none”

Antwoord op een 610. Dit verstuurt de control server bij aanmelding naar een cliënt. Parameter is een serverident zonder tag. Als er geen servers bekend zijn, wordt er noodgedwongen geen adres opgestuurd, maar “none”.

Berichttype: Stop server
Typenummer: 700
Tussen: Beheerder-server
Parameters: geen

De berichttypen 700-799 zijn gereserveerd voor uitbreidingen die server-specifiek zijn. Alleen het commando 700 dient gebruikt te worden om een server te kunnen stoppen. Dit commando mag uiteraard alleen door beheerders gegeven worden. Beheerders identificeren zichzelf tijdens het aanmelden of een naamswijziging. Daarom is hier geen extra authenticatie nodig.

Berichttype: Quit, server stopt
Typenummer: 701
Tussen: Server-cliënt, server-server, server-control server
Parameters: Reden

Een server kan dit bericht versturen naar zijn buren als hij gaat stoppen, b.v. na de ontvangst van een 700 bericht.

(7) Cliënt-Server communicatie

Een cliënt maakt verbinding met de server door een UDP/IP boodschap (loginbericht) te sturen naar het ip-adres van de server. De poort waarop de server luistert staat niet vast, maar veelal zal poort 2001 worden gebruikt. De cliënt kan eventueel aan deze gegevens komen door een boodschap te sturen naar de control server (boodschappen 610 en 611).

De server verstuurt daarop een statusmelding (OK, naam onjuist, wachtwoord onjuist) en indien OK zal hij de lijst met namen sturen. Ook zal de server de naam van de nieuwe cliënt verspreiden over het netwerk. Er komt geen verbinding tot stand in de TCP zin, maar de server registreert de cliënt en zal verdere boodschappen van hem acepteren en aan hem afleveren.

Merk op dat het huidige protocol het mogelijk maakt dat na het overlijden van een cliënt een nieuwe client op dezelfde node de associatie met de server overneemt, ook als de oude cliënt een beheerder was (met wachtwoord).

Vervolgens zal er een reeks berichten uitgewisseld worden van de types ‘bericht’, ‘actie’ en ‘naamswijziging’. De server zal deze naar alle aangesloten cliënten versturen, inclusief de zendende cliënt.

Bij wijziging van namen zal de server dat aan alle aangesloten cliënten/servers melden. Deze dienen daarop hun namenlijst aan te passen. Als een naamswijziging niet correct is (er is al iemand met die naam bijvoorbeeld) zal de server een errorcode terugsturen. Ook is het mogelijk om administratieve handelingen te doen zoals het afsluiten van de server via een bepaald commando. Verdere uitbreidingen staan vrij.

De sessie wordt normaliter beëindigd met een quit commando, welke de server naar alle cliënten doorstuurt. Als een cliënt of server meer dan 30 seconden stil blijft dient er een ping verzonden te worden. Wordt deze niet binnen 2 seconden beantwoord, dan is de ander kennelijk niet meer beschikbaar en mag deze bij alle deelnemers worden afgemeld (berichttype 130 voor clienten en 603 voor servers). Dat mag ook bij foutief verzonden data. Als de server zelf stopt moet er wel naar alle cliënten en servers een ‘quit’ bericht worden verzonden. Daarin staat de reden van het verbreken.

(8) Server-Server communicatie

De servers vinden elkaar uiteraard niet vanzelf. Hiervoor is de control server. Deze draait op een nader op te geven machine op poort 2001. Een server dient hiermee een paar 601 en 602 berichten uit te wisselen. De control server zal dan een adres van een reeds actieve server geven. Dit adres zal een ip4-adres zijn in ascii tekens, gevolgd door een poortnummer en een tag (zie 6.1), bijvoorbeeld "146.50.9.13:2000:T37S1583qwerty123456...". Als de server de string "none" teruggeeft is er nog geen server actief en blijf je standalone. Elke 10 seconden zal er gepingd worden volgens het eerder genoemde protocol. Verder mogen alleen boodschappen van het type 603 en eventueel 701 worden gestuurd.

Een server meldt zich niet aan met een code 100, maar met een code 600. We noemen de server waarbij hij zich aanmeldt de "parent" van een server. Als een server zich aanmeldt zal hij van zijn parent een lijst met namen ontvangen. Hij moet dan de namen doorsturen naar alle reeds verbonden cliënten en servers (alleen van belang bij heraanmelding). Mocht hij al cliënten hebben met dezelfde naam, dan dient hij deze te verwijderen en de verbindingen te verbreken.

Pas daarna stuurt hij zelf een lijst met de namen met aan of via hem verbonden cliënten aan de parent. Ook deze stuurt de namen door naar alle verbonden cliënten en servers. De communicatie tussen twee servers lijkt erg op die tussen server en cliënt, met het verschil dat er veel berichten een ander typenummer en andere parameters hebben.

De berichten die voor cliënten bestemd zijn die niet direkt met de huidige server verbonden zijn, moeten allemaal naar de server verzonden worden via welke de cliënt te bereiken is.

Als de parent van een server niet meer reageert of is afgemeld (door een fout of door het afmelden van de parent), dan meldt de server zich als volg opnieuw aan bij het systeem:

- Hij markeert de informatie over alle via de parent te bereiken cliënten en servers als ongeldig.
- Hij maakt zonodig (b.v. omdat een 604 niet is ontvangen) een nieuwe verbinding met de control server om een nieuwe parent toegewezen te krijgen (zoals boven beschreven).
- Hij meldt zich eventueel aan bij de nieuw toegewezen parent volgens eerder genoemd protocol, waarbij cliënten weer als geldig gemarkeerd kunnen worden.
- Hij stuurt naar alle van hem afhankelijke cliënten en servers een 130 bericht voor de cliënten die daarna nog als ongeldig gemarkeerd overblijven.
- Hij stuurt informatie over nog onbekende deelnemers door naar alle van hem afhankelijke deelnemers.

Als de verbinding met een child server wegvalt, dan stuurt een server een 130 bericht voor alle langs die weg te bereiken (nu onbereikbare) cliënten aan de nog wel verbonden partijen.

(9)Beheerders/moderatoren

Bij elke server hoort in principe ook een beheerder/moderator die zich als een cliënt met wachtwoord bij die server aan kan melden. Deze heeft de volgende eigenschappen:

- De beheerder krijgt van alle tekst en beheersberichten een afschrift. Beheersberichten moeten als berichttype 310 worden ingepakt om te voorkomen dat de cliënt mogelijk foutief reageert.

- De beheerder is alleen bekend op de eigen server – de naam wordt dus niet via type 110 berichten verspreid.
- De beheerder kan wel naar iedereen berichten versturen. De afzender heet dan "mod@*ip_adres*" waar *ip_adres* het IP adres van de betreffende server is (inclusief poortnummer).
- De beheerder mag een cliënt verwijderen middels een "130" bericht.

(10) Eindnotities

Zoals alles in het leven staat niets vast: het protocol zal aan veranderingen onderhevig zijn, ook tijdens de looptijd van het practicum. De te ontwikkelen programma's dienen te voldoen aan de laatst bekende versie van het protocol. Mocht iemand onvolkomenheden / fouten in het protocol vinden, dan zal dit zeker aangepast worden.. Voor alle vragen mbt het practicum kun je je wenden tot de assistent. In het cursusjaar 2011-2012 is dat Merlijn Wajer. Hij is te bereiken via mail(m.b.w.wajer@uva.nl).

De practicumsessies zijn voor iedereen verplicht, niet aanwezig betekent dat je geen cijfer krijgt voor het practicum. Wel kan in overleg met de assistent voor teams een andere tijd gekozen worden. De beoordeling van de resultaten ligt in handen van de assistent. Deze zal kijken naar o.a. inzet, voortgang, correctheid en volledigheid. Het eindresultaat zal een cijfer zijn op de schaal van 0 t/m 10. Het cijfer moet voldoende zijn, en telt dan voor 50% mee in het eindcijfer.

Zie ook de planning en andere informatie op Blackboard.

Belangrijke mededeling:
De berichten die servers naar cliënten versturen moeten **ook** verstuurd worden naar de cliënt die ze veroorzaakt. Een cliënt mag er ook niet van uitgaan dat een bericht goed ontvangen is voordat hij hem terugkrijgt.

Maart 2011
Dick van Albada (G.D.vanAlbada@uva.nl)
Merlijn Wajer (*M.B.W.Wajer@uva.nl*)