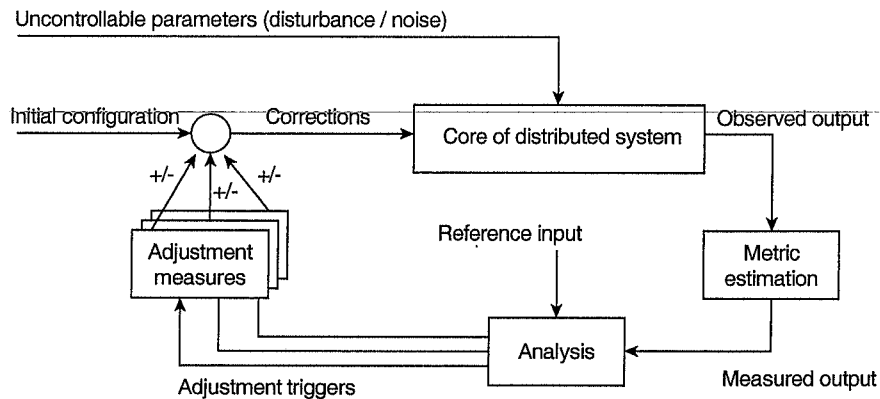
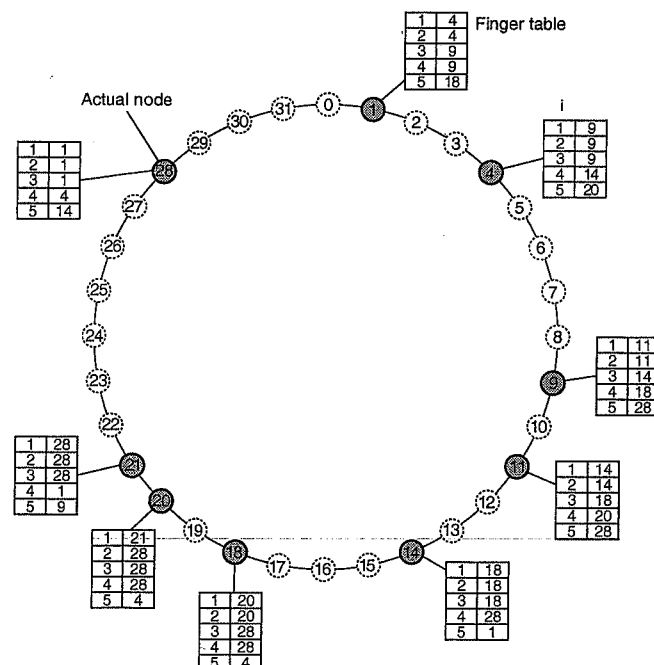


- 1a Geef een voorbeeld hoe een hoge graad van distributie transparantie een negatief effect kan hebben op prestaties. 5pt
- 1b Wat zie je als het grootste schaalbaarheidsprobleem bij een elektronische videoconferentie? 5pt
- 1c Wat zie je als het grootste schaalbaarheidsprobleem bij het realiseren van een systeem dat ingericht wordt volgens een *shared data space* architectuur. 5pt
- 1d Geef een concreet, simpel voorbeeld van een *feedback control loop* in een gedistribueerd systeem. Wees expliciet over elke van de componenten de volgende figuur. 5pt



- 2a Leg de principe werking van een *remote procedure call* (RPC) uit. 5pt
- 2b Een RPC is een voorbijgaande (*transient*), synchrone vorm van communicatie. Wat betekent dit en wat is hiervan het belangrijkste nadeel? 5pt
- 2c Een RPC naar een gerepliceerde server kan transparant gemaakt worden voor zowel de aanroeper als aangeroepene voor wat betreft toegang, replicatie en fout transparantie (*access, replication, failure transparency*). Hoe? Leg je antwoord uit! 5pt
- 3a Leg uit hoe de waarden van de *finger table* voor knoop 9 in het volgende Chord DHT-gebaseerde systeem tot stand zijn gekomen. 5pt



- 3b Stel dat knoop 4 een verzoek krijgt om sleutel 29 op te zoeken. Hoe verloopt dit zoeken. Je **moet** je antwoord uitleggen! 5pt
- 4a Laat zien dat logische klokken niet noodzakelijkerwijs potentiële causale relaties vastleggen. 5pt
- 4b Als men vector klokken gebruikt om causaal geordende multicasting af te dwingen, wordt $VC_i[i]$ opgehoogd alleen als proces P_i een bericht verzendt, and verstuurt dan VC_i als een timestamp $ts(m)$ met bericht m . Hoe moeten we de volgende twee condities interpreteren voordat m afgeleverd kan worden wanneer P_j bericht m ontvangt: 5pt
1. $ts(m)[i] = VC_j[i] + 1$.
 2. $ts(m)[k] \leq VC_j[k]$ for $k \neq j$.
- 4c We spreken altijd zorgvuldig over “potentiële” causale relaties bij middleware. Waarom “potentieel”? 5pt
-
- 5a Wat is het cruciale verschil tussen *data-centric* en *client-centric* consistentie modellen? 5pt
- 5b Leg uit waarom het *writes-follows-reads* client-centric consistentie model er automatisch voor zorgt dat causale relaties in het geval van een netwerk nieuws dienst gerespecteerd blijven. 5pt
- 6a Het *upload/download* model voor bestanden is vooral bekend voor FTP sessies. Het heeft echter ook z’n plek gevonden bij gedistribueerde filesystemen. Waar en hoe wordt het toegepast? 5pt
- 6b In Coda wordt toegestaan dat een client van een lokale kopie mag blijven lezen, zelfs als de server weet dat het betreffende bestand ververs is. Beargumenteer waarom dit consistentie niet in de weg staat. 5pt
- 7a Leg het verschil uit tussen *content-aware* en *content-blind* caching voor Web applicaties. 5pt
- 7b Het repliceren van een database bij een Web applicatie naar *edge servers* kan tot een performance verlies leiden van de applicatie. Hoe kan dat gebeuren? 5pt

Cijferbepaling: Het cijfer wordt bepaald door de punten die per onderdeel behaald zijn, bij elkaar op te tellen (totaal maximaal 90 punten), en vervolgens daar 10 extra punten bij te tellen. Er zijn dus totaal 100 punten te behalen.