Reeks A

Configuratie van de netwerkinterface (§1.2)

a. Bespreek alle opdrachten (ook hun opties en hun output) en configuratiebestanden (inclusief locatie en inhoud) die onder *Linux* voor de configuratie van de netwerkinterface kunnen gebruikt worden. Vergeet de opstartbestanden niet. Wat is *IP-aliasing* en hoe kan dit ingesteld worden ?
b. Bespreek het equivalent onder *Windows Server*, zowel via de *Command Prompt*, als via de grafische interface.

A) Linux:

Sommige Linux-varianten hebben grafische network configurators (Red Hat: **netcfg, neat** opdracht).

Bij vele UNIX-systemen worden netwerkinterfaces voorgesteld als een speciaal devicebestand (in **/dev**). In Linux-systemen worden ze dynamisch gecreëerd vanuit de software (bij het opstarten van de computer of door een toepassingsprogramma (bv PPP))
Iedere netwerkinterface heeft normaal slechts 1 IP-adres. Wanneer men meerdere IP-adressen toekent aan 1 interface, noemt men dit IP aliasing of virtual hosting.
Voorstelling (UNIX): afkorting van de naam (eth, ppp, fddi, tr ,lo) + nummer.
De loopback interface (lo, 127.0.0.1) is een speciaal soort interface die een computer in staat stelt connecties te maken met zichzelf.

Device drivers = besturingssysteem software die functies (in Linux kernel, houden rekening met methode van aanspreken van de netwerkkaarten) groepeert. Er wordt er één aan elke netwerkinterface gekoppeld. Eén device driver kan diverse netwerkinterfaces ondersteunen.

dmesg (of **/var/log/dmesg**) → namen van interfaces geïnstalleerd op Linux computer.
dmesg | grep eth → toont MAC adres, interrupt en I/O poort bronnen.
Bij het opstarten van de meeste Linux-systemen zorgt het auto-probing mechanisme ervoor dat het netwerkkaarttype gedetecteerd wordt en dat de juiste drivers geïnstalleerd worden. Deze tast verschillende geheugen en I/O-poorten af en vergelijkt de gegevens met de gegevens die er zouden moeten zijn indien er op die locaties een netwerkkaart geïnstalleerd is.

Manueel instellen v/e interface:
/etc/modules.conf of **/etc/conf.modules** bestand wijzigen
Vb: # cat /etc/modules.conf
alias parport_lowlevel parport_pc
alias eth0 ne2k-pci
alias eth1 ne
alias eth2 ne
options ne io=0x330, 0x360 irq= 7,9
alias sound-slot-o es1371
Enkel wijzigen indien:
- niet 100% compatibele klonen
- kaarten die men met verschillende device drivers moet aanspreken
- ISA interface kaarten

Insmod (modprobe) of install
→ aanvullende of recentere drivers éénmalig (insmod) of permanent (install) laden
Vb: # insmod 3c90x.o

install -m 644 3c90x.o /lib/modules/'uname -r'/kernel/drivers/net

lsmod of **/proc/modules** bestand → verifiëren welke device drivers effectief geladen zijn

/lib/modules/'uname -r'/kernel/drivers/net ('uname -r' = release nr kernel)
→ hierin vind je beschikbare drivers terug

ifconfig (of **ip a[ddress]** voor recente distros)
iwconfig voor wireless interfaces

- IP-adressen configureren en status van een interface nagaan
  Vb. # ifconfig eth0 (de parameter is dus de interfacenaam)

  We krijgen info over de instellingen en het verkeer via de interface.
  Belangrijkste: IP-adres, broadcast-adres voor subnet (waarop interface zich bevindt), subnetmasker voor subnet (nog steeds weinig prefixlengte gebruikt), hardwareadres v/d interface (→ verschillende netwerkkaarten v/h zelfde type uit elkaar houden).
  Ook het aantal verzonden/ontvangen pakketten wordt getoond, evenals het aantal collisions, MTU, queulength…

- kijken als interface wordt herkend door de kernel, juiste driver aanwezig, juiste config
  Vb: # ifconfig eth0

- commando opties (# ifconfig argument*)*:
  - **geen**: (= **netstat –i**), info alle geconfigureerde netwerkinterfaces, vlaggen B, L, R, U wijzen op gedefinieerd zijn van broadcast-adres, loopback interface, interface running, interface up. Alles ook terug te vinden in **/proc/net/dev** bestand.

  - **-a**: (= **netstat –ia**), info *alle* (!!!) aanwezige netwerkinterfaces

- andere opties (# ifconfig interfacenaam argument):
  Vb: # ifconfig eth0 193.168.55.23 netmask 255.255.255.0

  - ip adres in dotted decimal notatie eventueel laten voorafgaan door sleutelwoord **inet**
    Indien je verschillende IP-adressen wil toekennen aan dezelfde interfacekaart, dan spreek je die aan via: eth0:0n eth0:1

  - **up, down**: up = onmiddellijk activeren interface (niet expliciet vermelden, config informatie na een down werd bijgehouden en opnieuw geladen), down = deactiveren interface (Vb.: # ifconfig eth0 down), IP-alias verwijderen

  - **netmask + dotted decimal address**: specifieert subnetmasker

  - **broadcast + dotted decimal address**: specifieert broadcast-adres
    opm.: bij niet opgeven van dit en vorige: automatisch uit IP-adres afgeleid

  - **multicast**: interface toelaten multicast berichten te versturen/ontvangen, uitschakelen met **–multicast**.

  - **promisc**: promiscuous mode aanzetten, uitzetten met **–promisc**

- **pointtopoint + adres andere kant v/d verbinding**: voor point-to-point verbindingen
  (Vb.: # ifconfig ppp0 193.190.88.17 pointtopoint 193.168.55.14 netmask 255.255.255.252)

ifrename: manipuleren van de default naamgeving van de netwerkinterfaces

→ commando niet na elke reboot opnieuw uitvoeren:
ifcfg-eth0 bestand aanmaken in **/etc/sysconfig/network-scripts**

```
# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
IPADDR=198.258.46.1
NETMASK=255.255.255.192
ONBOOT=yes
```

Wordt door opstartscripts, meer bepaald door **/etc/rc.d/init.d/network** script, ingelezen
Interfaces waarvoor deze files aanwezig zijn → afsluiten/activeren door **ifdown xxx** en **ifup xxx** (xxx = interfacenaam in naam van bestand, vb eth0).

B) Windows:

De *Plug and play* functionaliteit van windows spoort tijdens het starten van de computer alle netwerkkaarten automatisch op en er wordt een interface (verbinding genoemd) gecreëerd in de map 'Network Connections' (Start > control panel > Network Connections). Krijgt standaard naam Local Area Connection (+ nummer).
Het netwerkkaarttype wordt bij het opstarten automatisch gededecteerd en de juiste drivers worden geladen/geïnstalleerd.
Rechtsklik op interface roept *menu* op:

- *rename* : geef interface andere naam
- *properties* :
  - controleer/wijzig hardware/software instellingen (bij lan: 2 tabbladen, nl. general en advanced, configure knop op general tabblad → aanpassingen instellen, tabbladen general, advanced, driver, resources)
  - aanvinken show icon in taskbar when connected → status inschakelen voor elke actieve connectie
  - selecteer Internet Protocol (TCP/IP) + klik op properties → stip vakje use following IP-adres → invullen IP-adres en subnetmasker. IP-aliasing toepassen → op advanced klikken en op IP settings tabpagina voor ieder bijkomend IP-adres op Add klikken en gegevens invullen. Dit kan ook via **ipconfig /all** (zie verder)
- *status* : verkeer monitoren (duur van verbinding, verzonden/ontvangen pakketten, eventueel compressie en statistische gegevens over fouten), ook mogelijk via **netstat** (optie **–e**: ethernet statistieken, optie **–s**: statistieken voor TCP, UDP, ICMP en IP protocollen, optie **–p protocol**: keuze van TCP, UDP, ICMP, IP: **netstat –esp IP**) , verbinding verbreken → op disable klikken
- *disable* : verbinding verbreken

ipconfig /all
→ Equivalent van **winipcfg** (oudere versies van windows), toont info TCP/IP netwerkconfiguratie v/ alle netwerkinterfaces.
→ Als computer is geconfigureerd met IP-adres dat kopie is van bestaand IP-adres → subnetmasker wordt als 0.0.0.0 weergegeven.

netsh

→ Het instellen van de TCP/IP-netwerkconfiguratie vanuit een Command Prompt is enkel mogelijk als de Routing and Remote Access Service actief is
→ De interactie met de Routing and Remote Access Service vanuit een Command Prompt gebeurt met **netsh**-opdracht.

Vb: # netsh interface ip set address "interfacenaam" static 193.190.172.3 255.255.255.192

→ Overzicht bekomen van alle interfaces met hun IP-adressen
Vb: # netsh interface ip show ipaddress

Statische routing (§1.3 zonder subsecties)

a. Bespreek het *doel* van routing, de *werking*, en de belangrijkste *componenten* ervan. Behandel de *terminologie* en *problematiek* die het routing proces kenmerkt.
b. Vergelijk *statische* en *dynamische* routing, zonder in detail in te gaan op routingprotocollen.
c. Geef de verschillende alternatieven om de *routingtabel van niet-routers* te configureren. Indien er hiertoe op Linux of Windows bijzondere componenten moeten geïnstalleerd of geconfigureerd worden, bespreek hoe dit moet gebeuren.

A)

Doel:
Aan eindgebruikers de illusie creëren dat ze allen verbonden zijn via één en hetzelfde netwerk, ook al zijn ze fysiek aangesloten op verschillende netwerken met uiteenlopende technologieën.
Componenten:
Gateway = computers die tegelijkertijd deel uitmaken van meerdere LAN en/of WAN netwerken en die bereid zijn om berichten van het ene netwerk naar het andere door te geven. Meestal zorgt dit voor zware belasting → er wordt gekozen voor toestel met specifieke hardware/software voor deze taken (=(hardware) router), als een gewone pc dit doet spreekt men van gateway of software router.
Functie:
Berichten doorspelen van het ene naar het andere netwerk (rechtstreeks als hij verbonden is op het netwerk van de bestemmeling (*direct delivery*) of onrechtstreeks via andere router (*indirect delivery*)).
Het is de gezamenlijke taak van alle routers van het internetwerk om er voor te zorgen dat het bericht doorgestuurd wordt naar een router die zich op het netwerk van de eindbestemming bevindt.

Terminologie:
- *routing*: doorsturen van berichten van ene router naar andere
- *hops*: de keten van routers die het bericht achtereenvolgend doorloopt om bij zijn bestemming te raken noemen we opeenvolgende hops
- *hopafstand* tussen 2 toestellen: minimum aantal routers nodig om een bericht tussen de twee toestellen te zenden (direct delivery → hopafstand = 0)
- *diameter*: maximale hopafstand in het internetwerk
- *routingtabel*: lijst die voor elk netwerk in het internetwerk (waarvan de router zelf geen deel uitmaakt) aangeeft naar welke volgende router (forwarding address) hij het bericht moet doorsturen, en van welke interface hij hierbij gebruik moet maken. Bevat netwerkadres, forwarding address, interface, metriek, lifetime veld.
- *netwerkadres*: unieke identificatie voor elk netwerk in internetwerk
- *metriek*: getal dat voor elk pad naar een eindbestemming via een bepaalde router de kost aanduidt om die eindbestemming te bereiken (verschillende mogelijkheden om waarde te kiezen: hopafstand, bandbreedte, … → TOS routing)
- *lifetime veld*: houdt bij hoe lang de route als geldig wordt beschouwd
- *TOS routing*: op basis van het Type Of Service veld in een IP-datagram wordt de juiste berekeningswijze voor de metriek gekozen

Problematiek:
Meest voorkomende fouten:
- *Routing loops*: wanneer routingtabellen voor een bepaalde eindbestemming een pad construeren dat terugverwijst naar één van de intermediaire routers v/h pad (gaat door tot levensduur bericht verstreken is).
- *Black holes*: wanneer een router op het pad naar de eindbestemming niet meer functioneert en dit falen nog niet is opgenomen in de routingtabel van de vorige router op het pad.

→ Routingtabellen van alle routers moeten meestal aangepast worden na elke topologiewijziging van het internetwerk.

Methode:
1. op zoek naar alle regels in routingtabel waar IP-adres van bestemming deel uitmaakt van netwerkadres. Dit doet men door de bits van het IP-adres en het netwerkadres te vergelijken over de lengte van de netwerkprefix. Of door het IP-adres te AND-en met het subnetmasker.
2. kies de meeste specifieke regel die hieraan voldoet (meest aantal bits overeenstemmen)
3. kies deze met kleinste metriek
4. indien geen gevonden → *ICMP-Destination Unreachable* bericht gestuurd naar oorspronkelijke afzender

→ **I**nternet **C**ontrol **M**essage **P**rotocol: doel is niet om IP-datagrammen betrouwbaarder te maken maar om afzender op de hoogte te stellen van netwerkproblemen. ICMP-bericht kan zelf geen ICMP-bericht tot gevolg hebben → stortvloed vermijden.

B)
Statisch:
Netwerkbeheerder maakt de routingtabellen voor elke router manueel aan.
Voordeel:
- Gemakkelijk te configureren
- Geen routing protocol overhead

Nadeel:
- enorm werk
- zeer foutgevoelig: bij elke topologiewijziging moet dit werk opnieuw gebeuren!

Dynamisch:
Routers communiceren met elkaar d.m.v. routingprotocollen zoals RIP en OSPF. Alle routers melden (=adverteren) aan alle andere routers met welke netwerken ze

rechtstreeks verbonden zijn, en/of welke ze onrechtstreeks kunnen bereiken. Hiermee construeren alle routers hun eigen routingtabel.
Voordeel:

- configuratie past zichzelf aan bij elke topologiewijziging (route flapping= bij uitvallen/terugkeren van router het wijzigen van routingtabellen, er wordt naar oorspronkelijke of nieuwe stabiele toestand teruggekeerd).

Nadeel:

- configureren van routingprotocollen
- overhead bij uitwisseling van berichten

C)
Statisch:
routingtabel handmatig invullen (meestal éénmalig bij installatie)
Dynamisch:

1. *Router-discovery met ICMP*: Opsporen gegevens default gateway op netwerksegment wanneer er geen default gateway is ingesteld. Computers verzenden *ICMP-Router Sollicitation* berichten naar het multicast adres voor alle routers: 224.0.0.2. De hiervoor geconfigureerde routers zenden *ICMP-Router Advertisements* terug (ook periodiek om beschikbaarheid mee te delen want beperkte lifetime voor standaardgateway in routingtabel). Router discovery met ICMP is standaard ingeschakeld op Windows computers. Op Windows Server moet Routing and Remote Access Service ingeschakeld zijn om ICMP-Router Advertisements te genereren.
2. *ICMP-Redirect*: als router Y bericht krijgt om door te zenden naar router X maar afzender kon router X zelf bereiken → router Y zendt ICMP-redirect naar afzender → optimaliseren van zijn routingtabel. Wordt niet veel gebruikt → veiligheidsrisico. Enkel host-routes (route naar individueel toestel) worden toegevoegd met zeer beperkte lifetime.
3. *Eavesdropping (wiretapping)*: Uitwisseling tussen routers vaak via broadcast → werkposten kunnen passief meeluisteren en hun routingtabellen aanpassen (met vb. Silent RIP, op Linux: activeren met **routed –q**, op Windows: RIP Listener installeren via Configuratiescherm > Add Remove Windows Components > Network Services > RIP Listener). Er wordt enkel naar RIPv1 berichten geluisterd.

Vraag3. Dynamische routing (§1.5)

a. Bespreek het *doel* en de *voordelen* van dynamische routing. Behandel de *terminologie* en de *problematiek* die dynamische routing kenmerkt.
b. Beschrijf het routing proces op *Internet schaal* (inclusief de relatie tot ISPs).
c. Maak een *classificatie van routingprotocollen*, volgens twee criteria. Geef van elke klasse de meest courante *vertegenwoordigers*. Het is niet de bedoeling in te gaan op een gedetailleerde vergelijking tussen de verschillende klassen en hun vertegenwoordigers.

A)

Doel:
Statisch geconfigureerde routers zijn weinig foutbestendig (vb. tijdelijk uitvallen router = wijzigen topologie) → *dynamische* (automatische) invulling routingtabellen is goed alternatief .

Voordelen:

- Tijdelijk falen van router wordt vlugger vastgesteld en doorgegeven aan andere routers (routes hebben beperkte lifetime, ctu zenden van berichten tss routers → routingprotocol).

- Nauwelijks manueel onderhoud (enkel initiële configuratie van de routing protocol software) → geschikt voor grote internetwerken

Terminologie:
- *Geconvergeerd* internetwerk: alle routingtabellen zijn correct ingevuld → internetwerk is geconvergeerd naar stabiele toestand → optimaal pad voor elke verbinding.
- *Route flapping*: uitwisseling van gegevens tussen routers (meestal bij topologie-wijziging). Tijdens deze periode bevindt het internetwerk zicht in onstabiele toestand.
- *Convergentieperiode*: tijd nodig om tot stabiele toestand terug te keren nadat router/verbinding uitvalt, hangt af van diverse factoren (topologie van het netwerk, het gebruikte routingprotocol, de opgetreden fout,…)

Problematiek:
- Bijkomend netwerkverkeer (vooral in WAN merkbaar).
- Tijdens de onstabiele toestand van het internetwerk (convergentieperiode) kunnen *routing loops* of *black holes* voorkomen.

B)

Het Internet bevat honderdduizenden subnetwerken en routers. Indien om het even welk paar routers met elkaar hun routingtabellen zouden uitwisselen, dan zouden de convergentieperiode en de impact op het netwerkverkeer enorm groot zijn. Daarom wordt het Internet opgedeeld in administratieve domeinen, meestal autonomous systems (AS) genoemd. Alle subnetwerken, routers en verbindingen van eenzelfde AS vallen meestal onder het beheer van 1 organisatie, dikwijls een ISP (Internet Service Provider).

Alle Routers van een AS wisselen onderling volledige routinginformatie uit. In elke AS configureert men 1 of meerdere routers als Border Router. Deze wisselen informatie uit met Border Routers van andere Ass via verbindingen die tot geen AS behoren (niemandsland, gedemilitariseerde zone) → elke router behoort tot één AS → makkelijk voor beheerders.

Verschillende ASs wisselen onderling zo weinig mogelijk gedetailleerde info uit → adresbereiken zoveel mogelijk *aggregeren* → de gedetailleerde van een AS blijft verborgen voor ander AS, waardoor de grootte routingtabellen beperkt blijft. Let wel: enkel aggregeren van subnetten als hoogste orde bits van subnetten in AS overeenstemmen en niet voorkomen in subnet ander AS.

Op het Internet ontmoeten ISP van verschillende niveau's zich in regional peering points of regional exchanges. Alle Tier-1 ISPs zijn op bijna alle regional peering points aanwezig. Omdat ze zoveel verkeer moeten uitwisselen ontmoeten Tier-1 ISPs elkaar ook nog op *private peering points*.
ISPs van lagere niveaus kunnen enkel via regional peering points verkeer uitwisselen met Tier-1 ISPs uitwisselen. Tier-2 ISPs beheren internetwerken, meestal beperk tot 1 land en wisselen rechtstreeks data uit met Tier-1 ISPs. Tier-2 ISPs wisselen verkeer uit met andere Tier-2 ISPs op regional peering points of maken hiervoor tegen betaling gebruik van diensten van een tier-1 ISP. Tier-2 ISPs verkoopt diensten aan Tier-3 ISPs, enz…
Internetwerk beheerd door dezelfde ISP vormt één enkel AS.

C)

Op basis van Routingprotocollen:

- **IGP: Interior Gateway Protocollen**: intradomain routingprotocollen

    - opvullen routingtabellen binnen zelfde AS
    - berekenen beste route naar om het even welke eindbestemming en deze info verspreiden over alle routers van het AS
    - *vertegenwoordigers*:
        - **RIP: R**outing **I**nformation **P**rotocol
        - **EIGRP: E**nhanced **I**nterior **G**ateway **R**outing **P**rotocol
        - **OSPF: O**pen **S**hortest **P**ath **F**irst **P**rotocol
        - **IS-IS: I**ntermediate **S**ystem to **S**ystem **I**ntermediate protocol
    - *Routingdomein*: verzameling netwerken (in één AS) waarin routers onderling de gedetailleerde routinginformatie uitwisselen op basis van hetzelfde IGP. (een AS kan opgesplitst worden in verschillende routing domeinen)

- **EGP: Exterior Gateway Protocollen**: Inter-domain routingprotocollen
    - mogelijk om routing info verschillende ASs over grenzen heen te zenden
    - enkel ASBRs (AS Border Routers) moeten EGP ondersteunen.
    - moeten rekening houden met routing policy

  ➔ *vertegenwoordigers*:
    - **EGP: E**xterior **G**ateway **P**rotocol
    - **GGP: G**ateway to **G**ateway **P**rotocol,
    - **BGPv4: B**order **G**ateway **P**rotocol → elk AS moet een uniek id (autonomous system number, ASN) hebben. → meeste gebruikt op Internet.

<u>Op basis van technologie</u>:

- Distance vector protocollen
  ➔ BGPv4, RIP, EIGRP
- Link state protocollen
  ➔ OSPF, IS-IS

➔ Verschillen in welke routing informatie uitgewisseld wordt en de manier waarop.

Vraag 4. RIP (§1.6 behalve §1.6.3) en behalve 1.6.4?

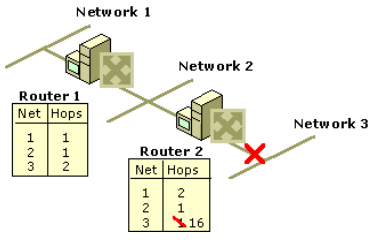
a. Geef een gedetailleerde beschrijving van de *werking* van RIP. Bespreek de *mogelijkheden*, *beperkingen*, en *problemen*. Bespreek in het bijzonder de gehanteerde *metriek*, en hoe RIP berichten verpakt worden (cfr. het OSI 7-lagen model).
b. Wat wordt bedoeld met *reductie van de convergentieperiode* (inclusief oorzaken) ? Bespreek de verschillende technieken om dit te verwezenlijken.
c. Bespreek de verschillende verbeteringen van *RIPv2* ten opzichte van *RIPv1*.

A)

Werking:
Interior, distance vector protocol.
Periodiek adverteert router alle route vectoren van zijn routingtabel naar alle routers die hij op subnetwerk-niveau rechtstreeks kan bereiken. Op basis van deze advertisements vullen routers hun routingtabellen systematisch bij. Uiteindelijk beschikken alle routers over de topologie informatie van het volledige netwerk.
De uitwisseling van berichten gebeurt niet-synchroon: elke router adverteert zijn gegevens op onafhankelijke tijdstippen van de andere routers. De ontvanger bevestigt RIP-berichten niet (ze worden slechts ingekapseld in UDP-segmenten).
RIP gebruikt standaard de hop afstand als metriek: bij het ontvangen van een RIP-Adverstisement verhoogt de router de metriek van de ontvangen routes met 1 vooraleer zijn routingtabel aan te passen. (berekening van metriek kan ook anders geconfigureerd worden)
RIP bericht gebroadcast (default 30s), ook bij stabiele configuratie.

Mogelijkheden/beperkingen/problemen:

- Maximale diameter internetwerk = 15 (als alle verbindingen een metriek 1 krijgen). RIP beschouwt routes met metriek 16 als onbereikbaar.
- Veel bandbreedte verbruikt door RIP berichten.
- Wanneer een eindbestemming langs verschillende paden bereikt kan worden, moet volgens de RIP-standaard elk van deze paden in de routetabellen opgenomen worden, met hun corresponderende metriek. De meeste implementaties wijken hier echter van af, om de grootte van de routingtabellen te beperken: enkel de paden met de kleinste metriek worden bewaard.
- *Trage convergentie* bij topologiewijzigingen, 3 oorzaken:
  - Enkel routes met kleinste metriek bewaren (metriek enkel verhoogd als bericht van eerste intermediaire hop wordt ontvangen).
  - Lifetime parameter (default 3min) = 6x pauze tussen advertisements. Na 4 advertisements met metriek 16 → route verwijderen.
  - Adverteren = periodiek en onafhankelijk → zekere willekeur in uitwisselingsproces waarbij resultaat beïnvloed wordt door volgorde van verzending door verschillende routers.
- *Count-to-Infinity probleem*:



  - Subnetwerk 3 is onbeschikbaar. Router 2 stelt zijn metriekwaarde voor subnetwerk 3 in op 16. Wanneer router 1 nu een advertisement stuurt voordat router 2 een advertisement stuurt (50% kans) dan zal router 2 merken dat router 1 subnetwerk 3 wel nog kan bereiken en zijn metriekwaarde aanpassen. Hierna zal router 2 een advertisement sturen. Router 1 zal daardoor zijn metriekwaarde voor subnetwerk 3 met 1 verhogen, enz. Na een tijdje (meer dan 3 minuten) zullen beide routers een metriekwaarde 16 hebben voor subnetwerk 3. Pas dan zijn beide routers geconvergeerd naar een stabiele toestand.

    - Reden voor 16 = ∞ → bij keuze groter dan 16 zou convergentieperiode alleen maar stijgen
- *Routing loop problemen*:
    - Tijdens convergentieperiode
    - Gepingpong van berichten tss 2 routers → einde als ttl=0
    - Kans groter dat netwerken overbelast raken → verlies pakketten

Metriek:
Hop afstand gebruikt voor metriek: ontvangen advertisement → metriek +1 (configureerbaar voor specifieke verbindingen) → routingtabel aanpassen.
Routes met metriek >= 16 → als onbereikbaar aanzien.

Verpakking:
RIP berichten worden door de ontvanger niet bevestigd → slechts ingekapseld in UDP segmenten (poortnummers 520)

B)

Reductie van convergentieperiode: tijd verkleinen om bij wijziging in het internetwerk opnieuw tot een stabiele toestand te komen.
Oorzaken: zie hierboven
Technieken: (grootste kans op slagen als ze globaal toegepast worden).

- *Split horizon*
    - Routers adverteren routes niet langer op het subnetwerk waarlangs ze deze routes vernomen hebben
    - Vermijdt count-to-infinity en routing loop in situatues waar er slechts 1 pad bestaat voor elke bestemming.
    - Geen bescherming tegen alle vormen van lussen: indien er meerdere paden mogelijk zijn, wordt de kans op fouten gereduceerd, maar niet uitgesloten.
    - Gunstig effect op overlast van RIP protocol op netwerk
- *Poison reverse*
    - Agressieve variant split horizon, blijft adverteren op alle netwerken, maar vermeldt routes op subnetwerk langs waar ze vernomen zijn met metriek 16
    - Meer reductie van kans op count-to-infinity en routing loop dan bij split horizon
    - Geen gunstig effect op overlast van RIP protocol op netwerk
- *Triggered updates*
    - Routers die triggered updates ondersteunen, adverteren niet allen periodiek, maar ook telkens de metriek van een route verandert. Een triggered update hoeft enkel de wijziging te adverteren.
    - Ook als lifetime veld van een route 0 is, wordt deze route bijna onmiddellijk met metriek 16 gebroadcast (er is nog kleine pauze → broadcastlawine vermijden)
    - Aanzienlijke reductie convergentieperiode
    - Wel nog steeds mogelijkheid tot lussen (door bv verlies van pakketten)
    - Verhoogd aantal broadcastberichten
    - In principe moet elke router die men wegneemt vooraf een triggered update versturen waarin alle routes met metriek 16 aangekondigd worden, anders langere convergentieperiode
- *General RIP request*
    - Tijdens opstartfase gebroadcast

- Routers die dit ondersteunen beantwoorden met hun volledige routingtabel
- Advertisement wordt wel gericht gestuurd, niet gebroadcast → vermijdt dat nieuwe router 30s moet wachten

C)

- RIPv1 berichten gebroadcast, voordeel is mogelijkheid tot Silent RIP, nadeel is hoeveelheid broadcastverkeer: alle aangesloten computers moeten broadcast-berichten tot op UDP niveau verwerken.
  RIPv2 berichten kunnen naar multicast adres 224.0.0.9 gestuurd worden → stoort de niet-routers niet
- RIPv1 maakte uitsluitend gebruik van zelfidentificerende adressen (A, B, C). Prefixlengte of subnetmasker werden niet in RIPv1 berichten opgenomen. Supernetten door RIPv1 routers foutief geadverteerd, ook routes naar subnetten verkeerd in routingtabel.
  RIPv2 berichten nemen expliciet het subnetmasker in de routes op → **C**lassless **I**nter**d**omain **R**outing op Internet niveau mogelijk.
- Men gebruikt het Next Hop veld van een RIPv2 bericht om adres van eerste intermediaire hop van een route aan te duiden. Bij RIPv1 is dit altijd 0: de afzender van het bericht wordt beschouws als eerste intermediaire hop. Bij RIPv2 wordt dit wel ingevuld om te vermijden dat niet optimale routes worden geadverteerd (dubbele hop).
  In zekere zin vergelijkbaar met ICMP-redirect techniek.
- RIPv1: geen beveiligingsmechanisme om authenticiteit van berichten te garanderen. RIPv2 ondersteunt eenvoudige authentificatie (wachtwoorden) alsook Message Digest 5.
  Bij authentificatie: eerste 20 bytes na header authentificatiegegevens

Vraag 5. OSPF (§1.8 inclusief §1.8.1)

a. Geef een gedetailleerde beschrijving van de *werking van OSPF*, inclusief de diverse mechanismen van berichtenuitwisseling en de OSPF routers met een bijzondere functie, maar zonder in te gaan op de uitwerking van het *algoritme van Dijkstra* en het concept van *OSPF area's*.
b. Beschrijf wat er precies gebeurt indien er een nieuwe router in een door OSPF gestuurd internetwerk wordt opgenomen.
c. Hoe worden OSPF berichten verpakt (cfr. het OSI 7-lagen model) ?
d. Wat is *TOS routing*, en in hoeverre ondersteunt OSPF dit ?

A)

Interior, link state routing protocol → elke OSPF router meldt enkel op subnetwerken hij rechtstreekse toegang heeft. Deze meldingen, **L**ink **S**tate **A**dvertisement**s** (LSAs) worden niet enkel naar zijn buren, maar naar elke router op het internetwerk verstuurd. Deze berichten worden in principe enkel na heropstarten van de router vertsuurd, of nadat de router een verandering in topologie van het netwerk opgemerkt heeft. Niet gebroadcast → geen eavesdropping.
De meldingen zijn *gesynchroniseerd* en de ontvanger *bevestigt* elk bericht expliciet (anders worden OSPF pakketten periodiek terug opgestuurd).
Autonomous **S**ystem **B**order **R**outer**s** (routers in het routing domein die ook aangesloten zijn op subnetwerken die niet tot routing domein behoren)vermelden in hun LSA ook routes buiten het routing domein die via hen bereikbaar zijn (rechtstreeks en onrechtstreeks). Dit noemt men *external route LSAs* → dikwijls geaggregeerd tot één enkele *default route LSA*.

Elke router verzamelt (compileert) de van LSAs die hij van andere routers van het routing domein ontvangt in een **L**ink **S**tate **D**ata**b**ase. Elke LSA bevat een tijddstempel (lolly nummer) waardoor de ontvangende router het verschil ziet tussen oude en nieuwe informatie. De LSDB bevat dus een volledige inventaris van alle routers en van alle netwerken waarop deze een aansluiting hebben. (Nadat alle LSAs uitgewisseld zijn, bevatten alle routers dezelfde LSB)
Na compilatie van de LSB, berekent elke router op basis hiervan een routingtabel. Dit gebeurt meestal adhv het algoritme van Dijkstra.

Synchronisatie van de links state databank (2 fasen)

1. Gebruik van *flooding* om LSAs te verspreiden: elke router stuurt LSA met eigen configuratie en ontvangen LSAs van andere routers door. Na korte tijd heeft elke router alle LSAs van de andere routers binnen het routing domein.
   Er worden groepen van naburige routers (adjacencies) gevormd om te vermijden dat elke router met elke andere router van het routing domein zou moeten controleren of hun LSDBs gesynchroniseerd zijn. Adjacency is gevormd van zodra routers van elkaar bewust zijn en zelfde gecompileerde LSDB hebben.

   Adjecencies worden dynamisch gevormd: tijdens zijn initialisatie stuurt de router een Hello-pakket om zijn bestaan aan te kondigen. Inhoud pakket = eigen router ID + router IDs van routers waarvan hij Hello pakket heeft ontvangen. OSPF routers veronderstellen transitiviteit: ze beschouwen routers die vermeld staan in Hello-pakketten van andere routers ook als rechstrreekse buren (ook al heeft hij van hen nog geen Hello-pakket ontvangen) → detectieproces versnellen.
   Dit is de 1ste functie Hello-pakket: weten met welke routers adjacency vormen.

2. *Database exchange proces*: elk koppel routers van een te construeren adjacency vormt een master/slave verhouding en wisselt *database description* pakketten uit. Deze pakketten beschrijven welke LSAs zicht momenteel in de LSDB van de router bevindt. Routers vergelijken dit met hun eigen LSDB. Indien er LSAs ontbreken of recenter zijn, vraagt hij die op mbv *Link State Request* pakketten, die de andere beantwoordt met *Link State Update* pakketten. (De ontvangst van *Link State Update* pakketten worden expliciet bevestigd) Wanneer alle koppels *Link State Request* en *Link State Update* pakketten uitgewisseld en bevestigd hebben, dan is de adjacency gevormd. Tenzij bij topologiewijziging is er geen uitwisseling LSDB gegevens meer.

   Leden van adjacency zenden wel *periodiek* Hello-pakketten → tonen dat ze actief zijn (2de functie Hello-pakket). Bij uitblijven Hello-pakket (default 40s) wordt router aanzien als uitgevallen → opnieuw uitwisselen Link State Update pakketten (+bevestigen). Idem als router uitgevallen verbinding detecteert. Hierbij opnieuw gebruik van *flooding*.

   Opm: enkel wijzigingen worden dus doorgestuurd <> distance vector protocollen.

Deze synchronisatie van LSDBs → zeer efficiënt → beperking netwerkbelasting
Na synchronisatie → elke router berekent routetabel (pad met kleinste totale kost) volgens *algoritme van Dijkstra*. Er wordt een **S**hortest **P**ath **F**irst tree aangemaakt met zichzelf als root → tree bevat pad met minimale kost voor elk netwerk en elke router als eindbestemming.

Op elk subnetwerk wordt een **D**esignated **R**outer (DR) verkozen en elke andere router vormt enkel adjaceny met deze → anders teveel verkeer: n routers → n(n-1)/2 adjacencies.
Wanneer een router een wijziging verneemt, dan geeft die dit door aan de DR via multicast 224.0.0.6. De DR verwittigt de andere routers anderen via multicast 224.0.0.5.

Hierdoor neemt de netwerkbelasting lineair toe met het aantal routers op het subnetwerk, en niet kwadratisch.
De DR-functie is eigenschap van interface van de router en niet van router zelf. Een router kan in een aantal subnetwerken een DR zijn, en in andere niet.
Bij het uitvallen van een DR moet een nieuwe verkozen worden en moeten alle adjacencies opnieuw opgebouwd worden. Hierdoor wordt het routing proces in internetwerk ernstig verstoord. Om de convergentieperiode te versnellen wordt ook een **B**ackup **D**esignated **R**outer (BDR) aangesteld. Ook BDR vormt een adjacency met alle andere routers. BDR luistert samen met DR naar multicast 224.0.0.6 maar houdt zich anders op de vlakte.
Bij het uitvallen van de DR wordt de BDR de nieuwe DR en wordt een nieuwe BDR gekozen.
De DR en BDR worden verkozen tijdens de uitwisseling van Hello-pakketten. (3de functie van Hello-pakketten). Indien er in het subnetwerk nog geen DR of BDR is, wordt deze gekozen aan de hand van de hoogste router prioriteit of (bij gelijke prioriteit) de router met de hoogste ID's. Prioriteit van interface = 0 → nooit DR of BDR zijn → dikwijls bij fysieke ster: hub is dan DR.

B)

Figuur 1.59 p47!!!
Van zodra opname van router in internetwerk → uitwisseling Hello-pakketten → vorming adjacency + uitwisselen database description pakketten. Link state request en link state updates tussen routers → gesynchroniseerde LSDB → nieuwe SPF tree en routingtabel berekenen.
Link state update verzonden naar andere routers (van adjacency) met enkel LSA van nieuwe router. Via flooding wordt gesynchroniseerde toestand snel bereikt.

C)

OSPF pakketten worden rechtstreeks ingekapseld in IP-datagrammen.

D)

TOS routing = **T**ype **O**f **S**ervice routing. Door het instellen van de TOS velden in een IP-datagram kan een router een andere metriek berekenen afhankelijk van de voorkeuren van de afzender (vb.: **C**ost, **S**ecurity, **B**andwith).
OSPF ondersteunt dit. LSAs kunnen verschillende metrieken bevatten voor verschillende TOS-waarden (zeker metriek voor TOS nul → wordt gebruik bij niet vermelden specifieke metriek) en moeten vermelden als router TOS ondersteunt of niet.
Als een router TOS-routing ondersteunt, moet deze een SPF tree berekenen voor elke legale TOS-waarde. Bij het berekenen van de SPF tree voor een niet-nul TOS mogen geen routers opgenomen worden die geen TOS-routing ondersteunen. Dat kan tot gevolg hebben dat bepaalde bestemmingen niet bereikbaar zijn voor een bepaalde TOS-waarde. Deze paketten worden dan doorgestuurd via TOS nul routes.

Vraag 6. RIP , OSPF en EIGRP (§1.7 & §1.8.4)

a. Van welke *categorieën* routingprotocollen zijn RIP en OSPF typische vertegenwoordigers ?
b. Maak een *gedetailleerde vergelijking* tussen de mogelijkheden en beperkingen van deze twee categorieën.
c. Tot welke categorie behoort *EIGRP* ? Bespreek de meerwaarde van dit protocol in vergelijking tot de andere vertegenwoordiger in deze klasse, en (in detail) hoe dit gerealiseerd wordt.

A)

RIP: Interior Gateway, Distance Vector Protocol
OSPF: Interior Gateway, Link State Protocol

B)

Distance Vector Protocol (RIP):

+ *Eenvoudiger te begrijpen*: één type bericht, één tabel, eenvoudig mechanisme (adverteren + updaten)
+ *Eenvoudig te configureren*: daemon (unix) of service (win) opstarten op routers
+ *Geen complexe verwerking* van de advertisements: de ontvangen routes kunnen eenvoudig in de locale routetabel opgenomen worden.
+ *Goede resultaten*, voorwaarde: netwerktopologie relatief eenvoudig en verbindingen vallen zelden uit
– *Grote routingtabellen* door verschillende paden → minder geschikt voor grote routing domeinen. Diameter maximaal 15.
– *Veel netwerk verkeer* door periodieke advertisement (volledige routingtabel !) → veel redundante informatie → niet echt geschikt voor WAN
– *Lange convergentieperioden* want verkeer niet gesynchroniseerd noch bevestigd. Kans op routingproblemen (routing loops, count-to-infinity, black holes) tijdens deze periode.

Link State Protocol (OSPF):

+ *Kleinere routingtabellen* want slechts één enkele router per eindbestemming
+ *Minder netwerkbelasting tijdens convergentiefase*: enkel gewijzigde LSAs doorsturen. Toch is belasting niet te verwaarlozen, vooral bij route flapping (verbinding uitvalt → tweemaal flooding van een LSA naar alle routers + herberekening van alle routingtabellen)
+ *Minder netwerkbelasting na convergentie*: enkel periodiek Echo-pakketten (bevatten router ID en geen LSA info → beperkte lengte) → geschikt voor LAN en WAN
+ *Voor grote tot zeer grote routingdomeinen*: al dan niet nog opsplitsen in verschillende areas
+ *Kortere convergentieperioden*: geen routing loops of count-to-infinity want elke router zelfde LSDB
+ *Alle routers* hebben *exact beeld* van de *topologie* van het internetwerk

- *Complexer*: 5 verschillende berichten en 3 soorten procedures (Hello, database exchange, flooding van LSA)
- *Meer werk voor netwerkbeheerder*: nood aan planning, configuratie en opvolging tenzij slechts één area (backbone). Netwerkbeheerder moet grenzen definiëren (= toekennen Area ID), aggregeren van external route LSAs, …
- *Berekening* van routingtabel uit LSDB is *geheugen- en processorintensief*.

C)
EIGRP: **E**nhanced **I**nterior **G**ateway **R**outing **P**rotocol is een Interior Gateway, Distance Vector Protocol
Net als RIPv2 gebaseerd op periodiek multicasten van routingtabel naar al zijn buren.
Werkt met lagere frequentie voor periodieke updates: 90s t.o.v. 30s voor RIP.
Eveneens toepassing split horizon en triggered updates → kans ontstaan lussen daalt.

Twee belangrijkste verbeteringen t.o.v. RIP:

1. Advertisements bevatten voor elke route vier metrische waarden:
   Statisch: afgeleid uit type subnetwerk of ingesteld door beheerder v/ router
   - *d (delay):* uitgedrukt in eenheden van 10 µs, gecodeerd in 3 bytes: som transmissie oponthoud in de verbindingen tussen router en bestemming
   - *b (bandwith):* 3 bytes: bandbreedte van de zwakste verbinding in pad naar bestemming

   Dynamisch: verkregen door verkeer in verbindingen statistisch te analyseren
   - *r (reliability):* betrouwbaardheid: 1 byte: percentage pakketten dat bij bestemming aankomt, r=255 → 100% aangekomen
   - *l (load):* belasting: 1 byte: bezettingsgraad van drukste verbinding

   Berekening van ***samengestelde metriek*** Δ volgens formule (zie cursus p. 42) met k1-k5 parameters per router ingesteld.
   Elke EIGRP *router i* berekent metriek Δij naar om het even welke *eindbestemming j* door alternatief pad in overweging te nemen met elk van zijn naburen x als eerste intermediaire hop. Metriek van alternatieve paden beschouwd als samengestelde van Δxj (buur naar eindbestemming) + δix (verbinding met buur). Pad met kleinste metriek bepaalt route die router i in routingtabel opneemt.
   $\Delta ij = \min (\delta ix + \Delta xj)$

2. Implementatie ***Diffusing Update (DUAL) algoritme*** → vermijden van routing loops.
   Stel: router i krijgt van nabuur y nieuwe waarde (δiy of Δiy). Als $\delta iy' + \Delta iy' < \Delta ij$, dan wordt router y eerste intermediaire hop voor router i naar eindbestemming j. Router i werkt routingtabel bij en zendt triggered update naar alle buren. Geen kans op routing loops want *kleinere metriek gekozen*.
   Is de *nieuw berekende metriek groter*: niets doen tenzij buur die de update gegenereerd heeft eerste intermediaire hop naar eindbestemming is (y=x).
   Nabuur z acceptabel: als $\Delta zj < \Delta ij$ (oude waarde).
   Twee mogelijkheden:
   I. *Ten minste één acceptabele buurman*: $\Delta iy' = \min (\delta iz + \Delta zj)$ → z eerste intermediaire hop (bijwerken routingtabel) → router i zendt triggered update naar alle naburen.
   II. *Geen acceptabele buurman*: router i doet *diffusieberekening* (zolang berekening niet klaar → route naar bestemming j bevroren = *activeren* van route → geen routing loops, wel black hole en tijdelijke onbereikbaarheid van bestemmeling).
   Stap 1 van proces: router i zendt query (bijzondere vorm van update) naar alle naburen z (behalve deze waarvan update ontvangen) met $z \neq x$. In query: bestemming j + aangepaste metriek: : $\Delta ij' = (\delta ix' + \Delta xj')$. Als reactie wordt nieuwe metriek vanuit standpunt van buurman verwacht $\Delta zj'$.

Stap 2 van proces:

- buren houden route in passieve toestand (vb. omdat router i niet eerste intermediaire hop of andere buurman kan geselecteerd worden) → onmiddellijk antwoord met aangepaste metriek in routingtabel
- OF: ze moeten route activeren → sturen zelf queries naar alle buren (= DUAL de diffusie van de update), als router van alle buren antwoord ontvangen → stuurt antwoord naar router waarvan query ontvangen zodar diffusie wegebt. Uiteindelijk antwoord bij eerste router → stabiele toestand.