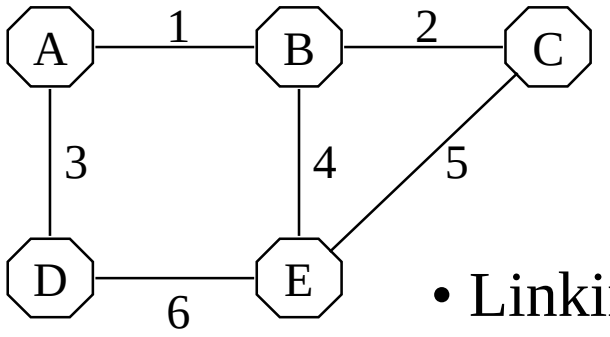
Linkkien tilaan perustuva reititys

Tavoitteena on välttää EV-protokollan luomat silmukat ja skaalautua laajempiin verkkoihin ja monenlaisiin topologioihin.

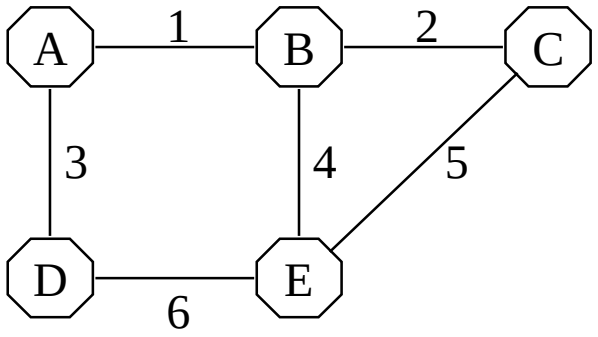
OSPF - Open Shortest Path First on Internetin linkkien tilaan perustuva reititysprotokolla



Esimerkkiverkko

- Linkintilaprotokolla ylläpitää verkon topologiakarttaa.
- Kun topologia muuttuu, kartat päivitetään nopeasti.
- Karttoja käytetään reittilaskennassa.
- OSPF on IETF:n määrittelemä linkintilaprotokolla Internetiä varten - OSPF on suositeltu RIP:n seuraaja.

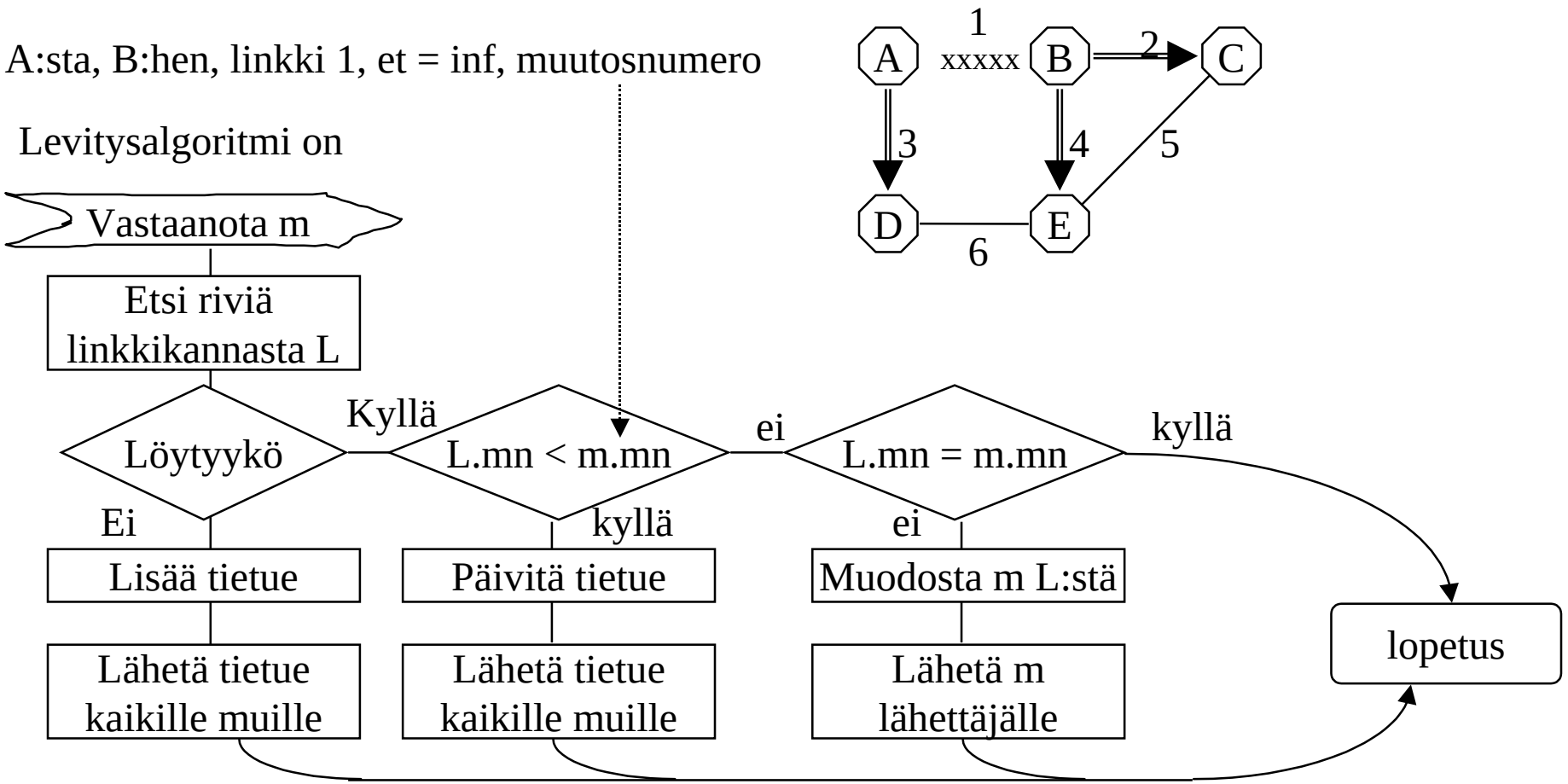
Kartta esitetään täydellisenä linkkien luettelona



- Kustakin rivistä vastaa tietty solmu
- Linkin suunnat esitetään erillisinä

Mistä	Mihin	Linkki	Etäisyys
A	B	1	1
A	D	3	1
B	A	1	1
B	C	2	1
B	E	4	1
C	B	2	1
C	E	5	1
D	A	3	1
D	E	6	1
E	B	4	1
E	C	5	1
E	D	6	1

Levitysprotokolla (flooding) levittää tiedot topologiamuutoksista

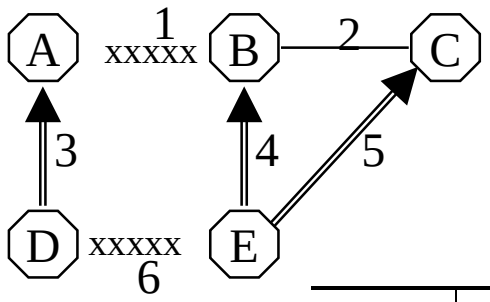


Linkkikanta AB vian levityksen jälkeen

Mistä	Mihin	Linkki	Etäisyys	Mnumero
A	B	1	inf	2
A	D	3	1	1
B	A	1	inf	2
B	C	2	1	1
B	E	4	1	1
C	B	2	1	1
C	E	5	1	1
D	A	3	1	1
D	E	6	1	1
E	B	4	1	1
E	C	5	1	1
E	D	6	1	1

- Muutosnumerojen laskenta alkaa käynnistyksessä 1:stä.
- Modulo aritmetiikka määrittelee mikä on “vähän suurempi kuin”
  --> muutosnumerolaskuri voi pyörähtää ympäri.

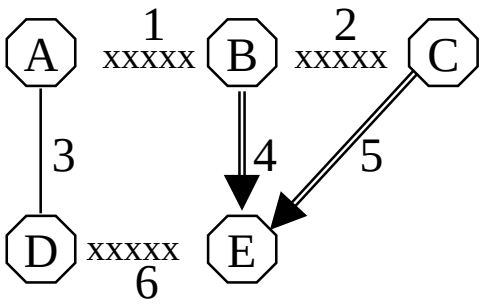
Kun verkko osittuu, puoliskojen kannat eroavat toisistaan



Mistä	Mihin	Linkki	Etäisyys	Mnumero
A	B	1	inf	2
A	D	3	1	1
B	A	1	inf	2
B	C	2	1	1
B	E	4	1	1
C	B	2	1	1
C	E	5	1	1
D	A	3	1	1
D	E	6	inf	2
E	B	4	1	1
E	C	5	1	1
E	D	6	1	1

Mistä	Mihin	Linkki	Etäisyys	Mnumero
A	B	1	inf	2
A	D	3	1	1
B	A	1	inf	2
B	C	2	1	1
B	E	4	1	1
C	B	2	1	1
C	E	5	1	1
D	A	3	1	1
D	E	6	1	1
E	B	4	1	1
E	C	5	1	1
E	D	6	inf	2

Linkki 2 vikaantuu -> kannat eroavat lisää

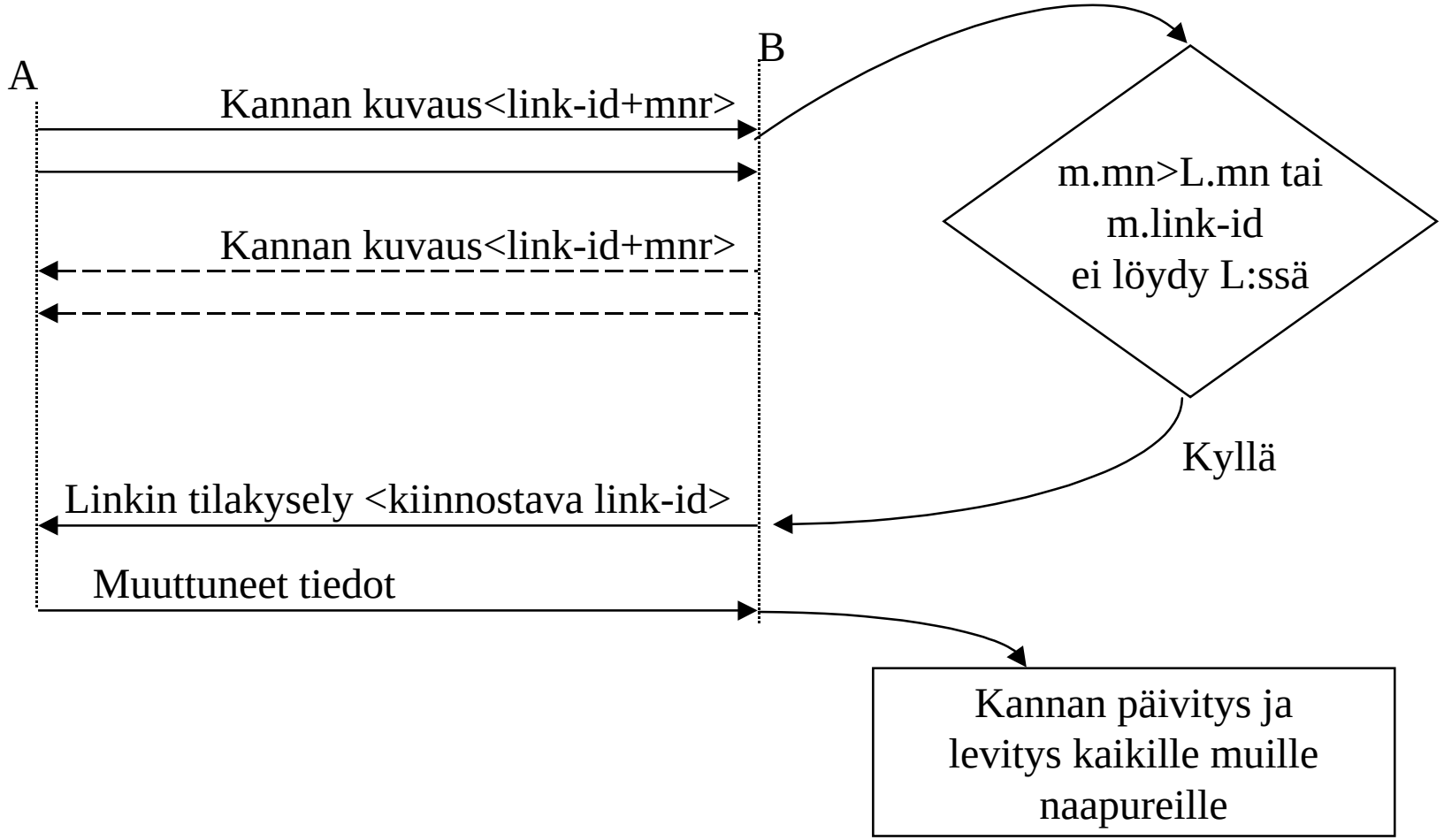


Välittömästi tästä ei aiheudu ongelmaa, mutta jos linkki 1 elpyy ...

B:n, C:n ja E:n kannat:

Mistä	Mihin	Linkki	Etäisyys	Mnumero
A	B	1	inf	2
A	D	3	1	1
B	A	1	inf	2
B	C	2	inf	2
B	E	4	1	1
C	B	2	inf	2
C	E	5	1	1
D	A	3	1	1
D	E	6	1	1
E	B	4	1	1
E	C	5	1	1
E	D	6	inf	2

Osittuneen verkon jälleenyhdistyessä tarvitaan “naapurusten yhteenkasvatusta”



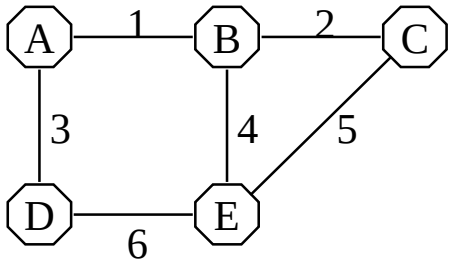
Linkkikantojen yhtenäisyys on varmistettava

- *Levityssanomat* kuitataan linkki kerrallaan
- Kannan *kuvaussanomien* perillemeno varmistetaan
- Kannan kukin tietue suojataan vanhenemisajastimella, jos päivitys ei saavu ajoissa, tietue poistetaan
- Kannan tietueet on suojattu tarkistussummalla
- Sanomat sisältävät autentikointitietoa
- *Mutta: päivityksen ollessa käynnissä toiset solmut ovat paremmin ajan tasalla kuin toiset --> reititysvirheitä*

OSPF perustuu Dijkstran SPF algoritmiin

- SPF - lyhin polku ensin -algoritmi laskee lyhimmän polun lähdesolmun *S* ja kaikkien muiden solmujen välillä
- Aluksi solmut jaetaan evaluoituihin *E* , joista alkavat polut tunnetaan ja muihin *R*.
- Lisäksi tarvitaan polkujen järjestetty lista *O* .

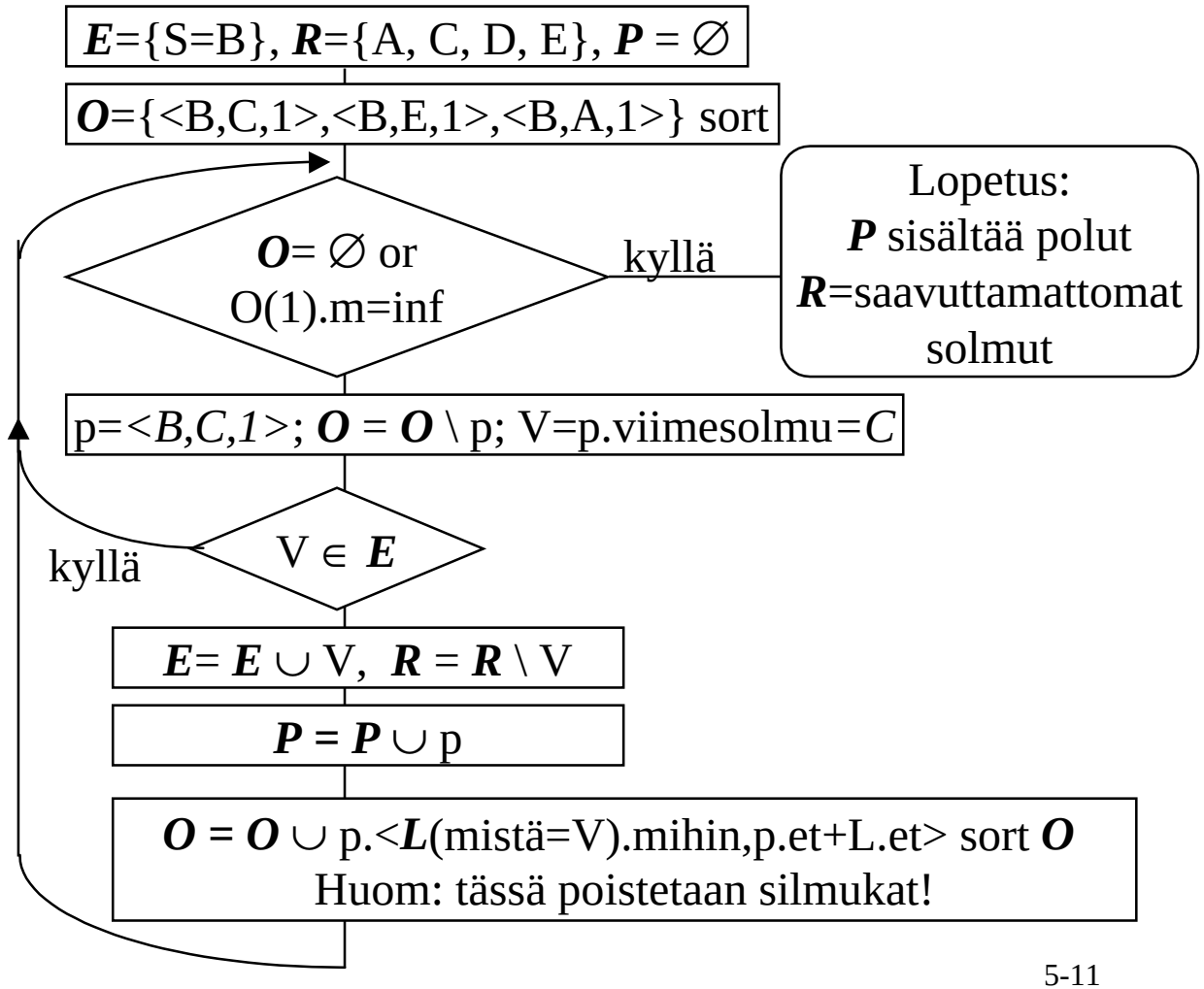
Dijkstran lyhin-polku-ensin algoritmi



L

Mistä	Mihin	Linkki	Etäisyys
A	**B**	**1**	**1**
A	**D**	**3**	**1**
B	**A**	**1**	**1**
B	**C**	**2**	**1**
B	**E**	**4**	**1**
C	**B**	**2**	**1**
C	**E**	**5**	**1**
D	**A**	**3**	**1**
D	**E**	**6**	**1**
E	**B**	**4**	**1**
E	**C**	**5**	**1**
E	**D**	**6**	**1**

Konvergoi nopeammin kuin Bellman-Ford
O(M.logM) < O(N.M)



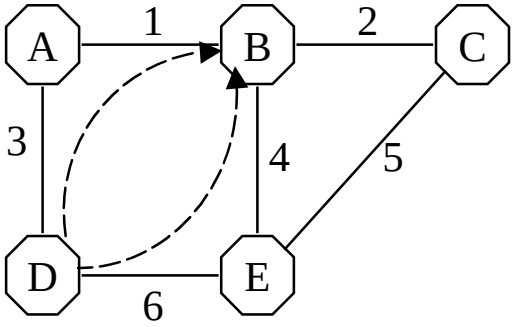
Linkintilaprotokollien etuja ovat

- Linkkikantojen tila konvergoituu nopeasti, ilman silmukoita
- Mittoina voidaan käyttää tarkkoja lukuja. Yksi protokolla tukee monta etäisyysmittaa
  - kapasiteetti, viive, hinta, luotettavuus.
- Ylläpitää useita polkuja yhteen kohteeseen.
- Ulkoisten polkujen erillinen esitys.

Useiden etäisyysmittojen käyttö edellyttää

- Mittojen tallennusta jokaiselle linkille (***L.et1, L.et2 ...***)
- Erillisten reititystaulujen laskemista mitoille (***P***(et1), ***P***(et2) ...)
- Linkkiprotokolla kuljettaa kaikki mitat
- Käyttäjäsanomiin merkataan vaadittu mitta.
- Reittisilmukka on mahdollinen, jos solmut käyttävät eri mittaa samalle käyttäjän paketille.

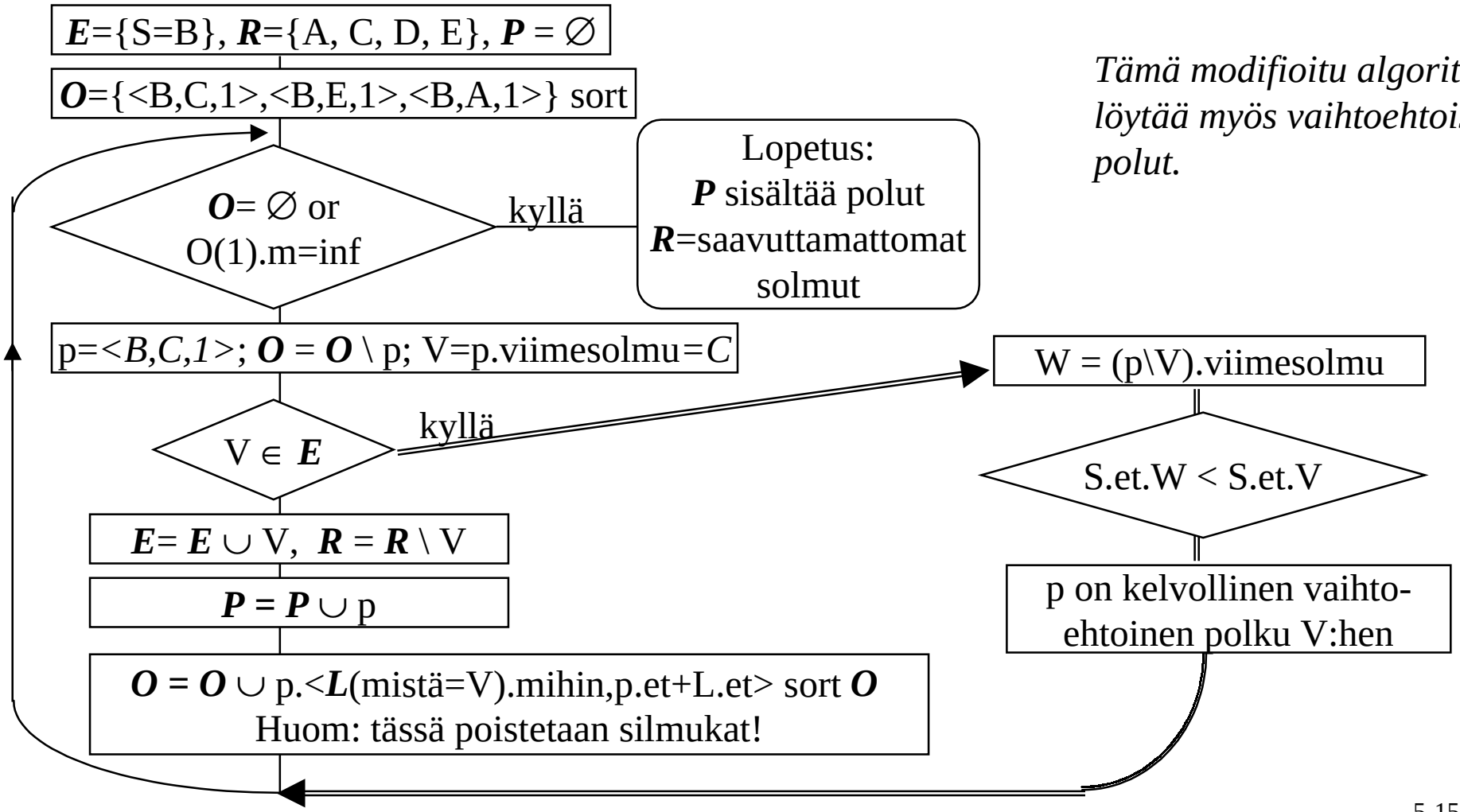
Liikenteen jakaminen vaihtoehtoisille samanmittaisille poluille tehostaa verkon käyttöä



- Solmujen kokonaisjonopituus laskee
- Keskimääräinen viive laskee
- Päästä päähän viiveen vaihtelut pienenee
- Vikatilanteissa liikenteen heilahtelut vähenee

- Vaarana on pakettien järjestyksen muuttuminen, koska polkuviiveet (solmujen jonojen pituudet) vaihtelevat
- Hankaluus: menossa oleva liikenne ei pysy nykypolulla (no route pindown) --> epästabiilisuus mahdollista (2000)
- Milloin polut ovat riittävän samanmittaisia?

Sääntö A->Y...->X, jos Y.et.X < A.et.X rajaa vaihtoehtoiset reitit (monotonisiin)



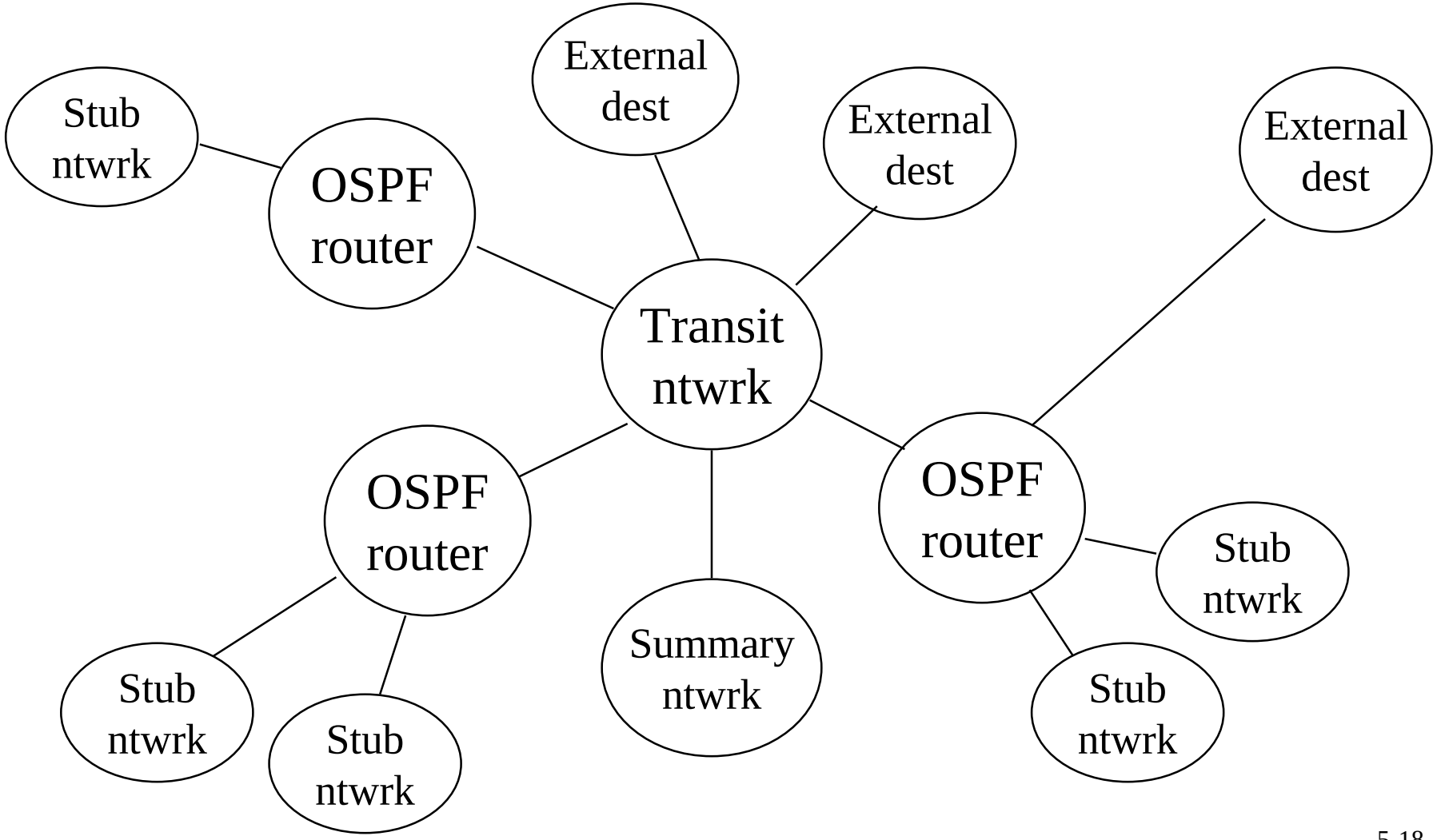
Tämä modifioitu algoritmi löytää myös vaihtoehtoiset polut.

Linkintilaprotokolla kuvaa useita ulkoisia reittejä tarkoilla (halutuilla) mitoilla

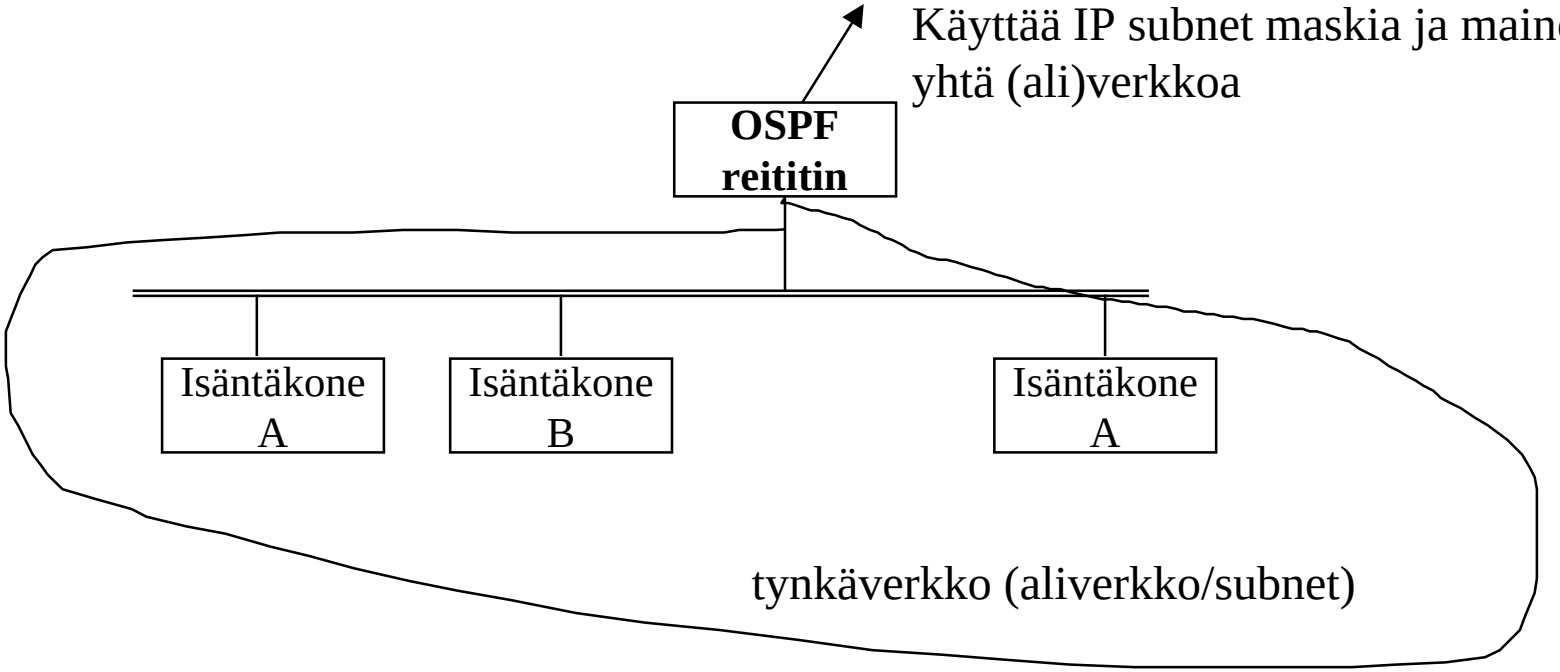
- EV-protokollan mahdollisuudet kuvata reittejä ovat rajalliset äärettömyyteen laskennan ja Bellman-Fordin (O(N**2) monimutkaisuuden takia
- Linkintilaprotokolla on vapaa yo rajoitteista. SPF reittilaskennan monimutkaisuus on O(N.logN) - N=ulkoisten reittien määrä
- Esim. noin 30 000 ulkoista reittiä =>10exp9 vs. 450 000

Tästä alkaa OSPF protokollan periaatteiden kuvaus

OSPF sees the network as a graf



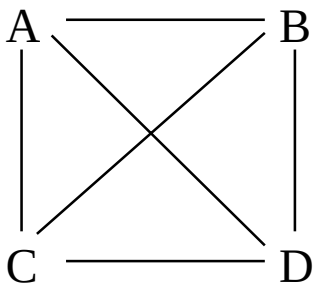
OSPF erottelee reitittimet ja isäntäkoneet



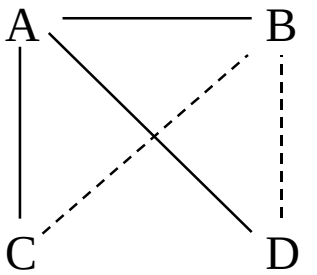
OSPF tukee yleislähetysverkkoja

Yleislähetysverkossa

- *kaikki voivat lähettää kaikille*
- *yksi voi lähettää kaikille tai joukolle*
- *jos siinä on N reititintä, niillä on N*(N-1)/2 naapuruussuhdetta ja*
- *jokainen reititin mainostaa N-1 reittiä toisiin reitittimiin + yhtä tynkäverkkoa*



N*(N-1)/2 naapuruussuhdetta

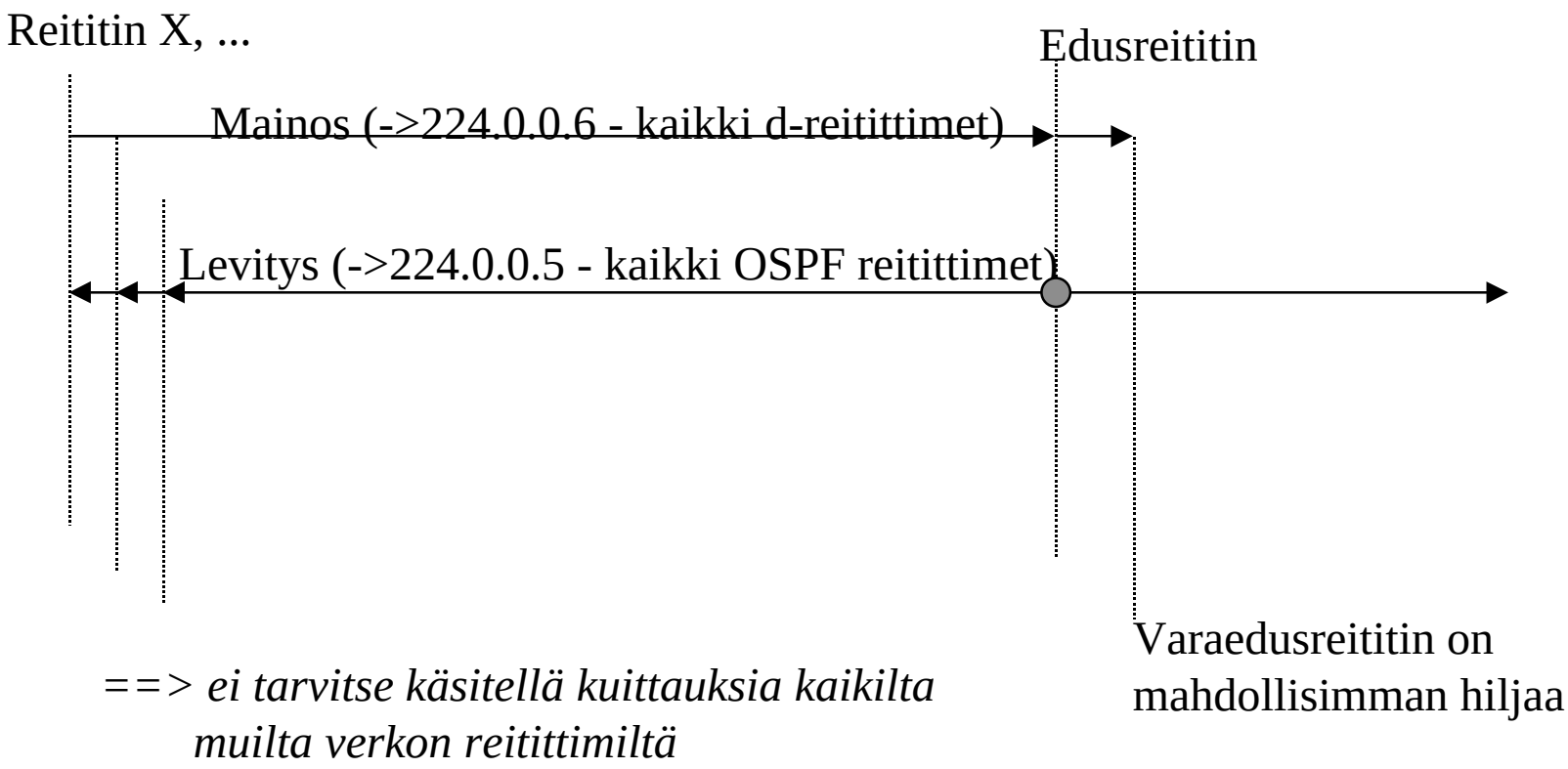


(vara)

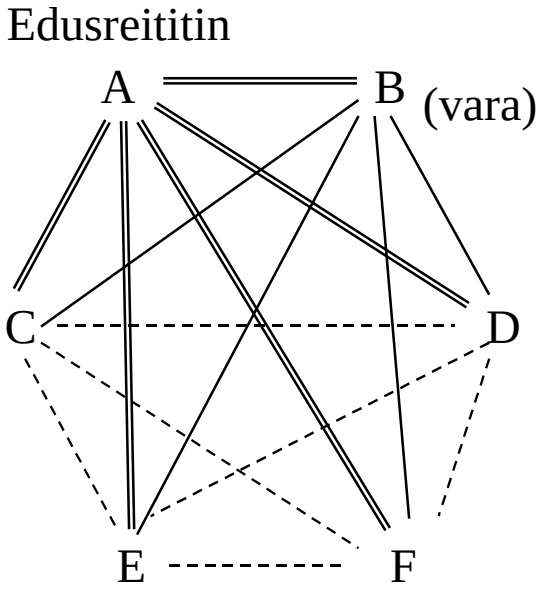
Naapuruus rajattu vain edus-
(designated) reitittimeen (A) ==>

- A täytyy valita Hello-protokollan avulla
- Linkkikantojen synkronointi yksinkertaistuu
- Varaedus (backup designated) reititin valitaan samalla kertaa

OSPF levitysprotokolla yleislähetysverkossa



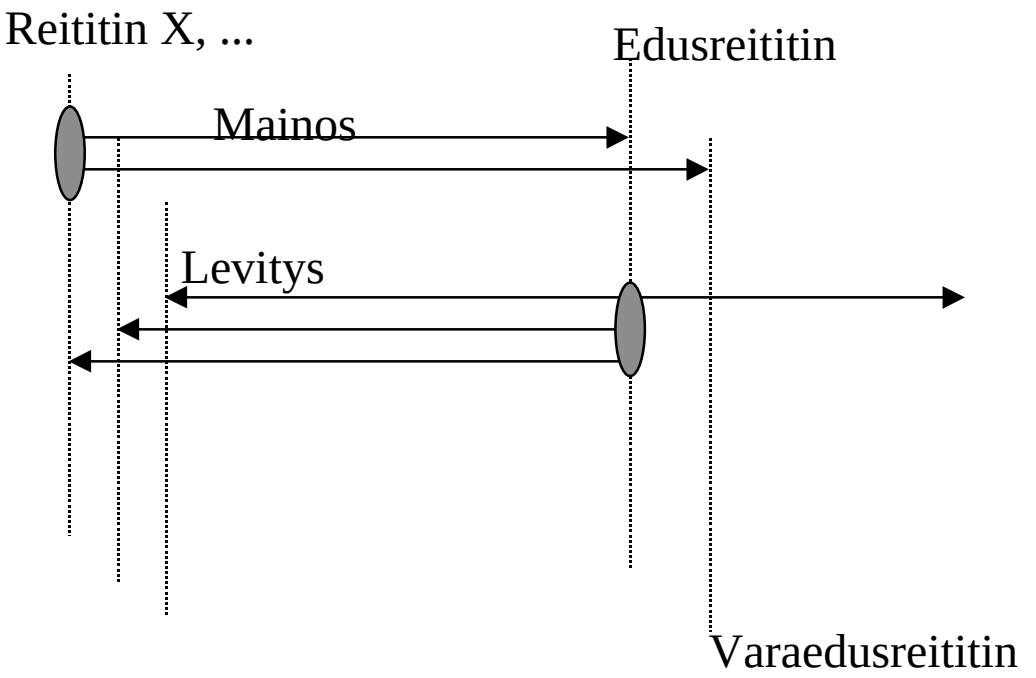
Ei-BC verkoissa OSPF toimii samoin kuin BC-verkoissa paitsi, että BC-lähetykset korvataan yksipistelähetyksillä



Kiinteä yhteys edus-reitittimeen

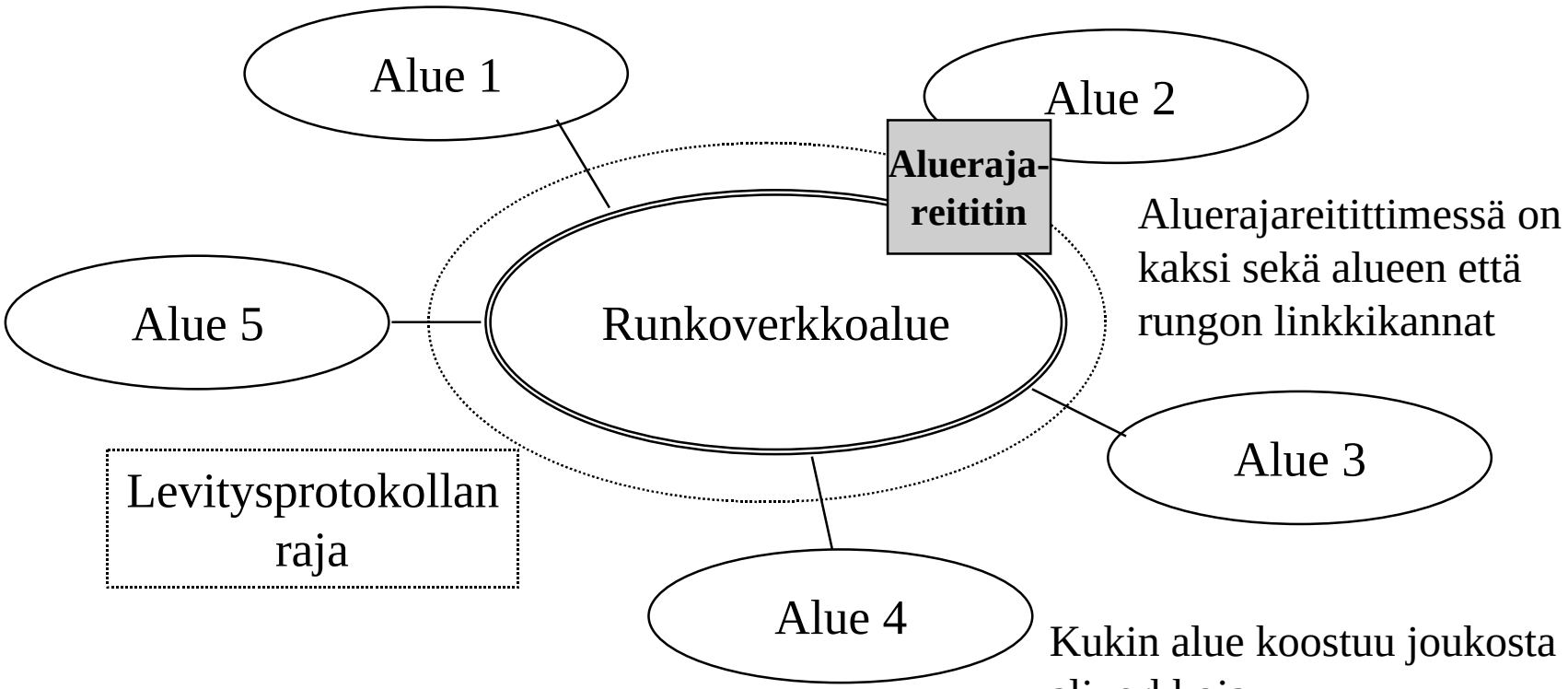
Kiinteä yhteys vara-reitittimeen

Valinnainen yhteys muiden reitittimien välillä



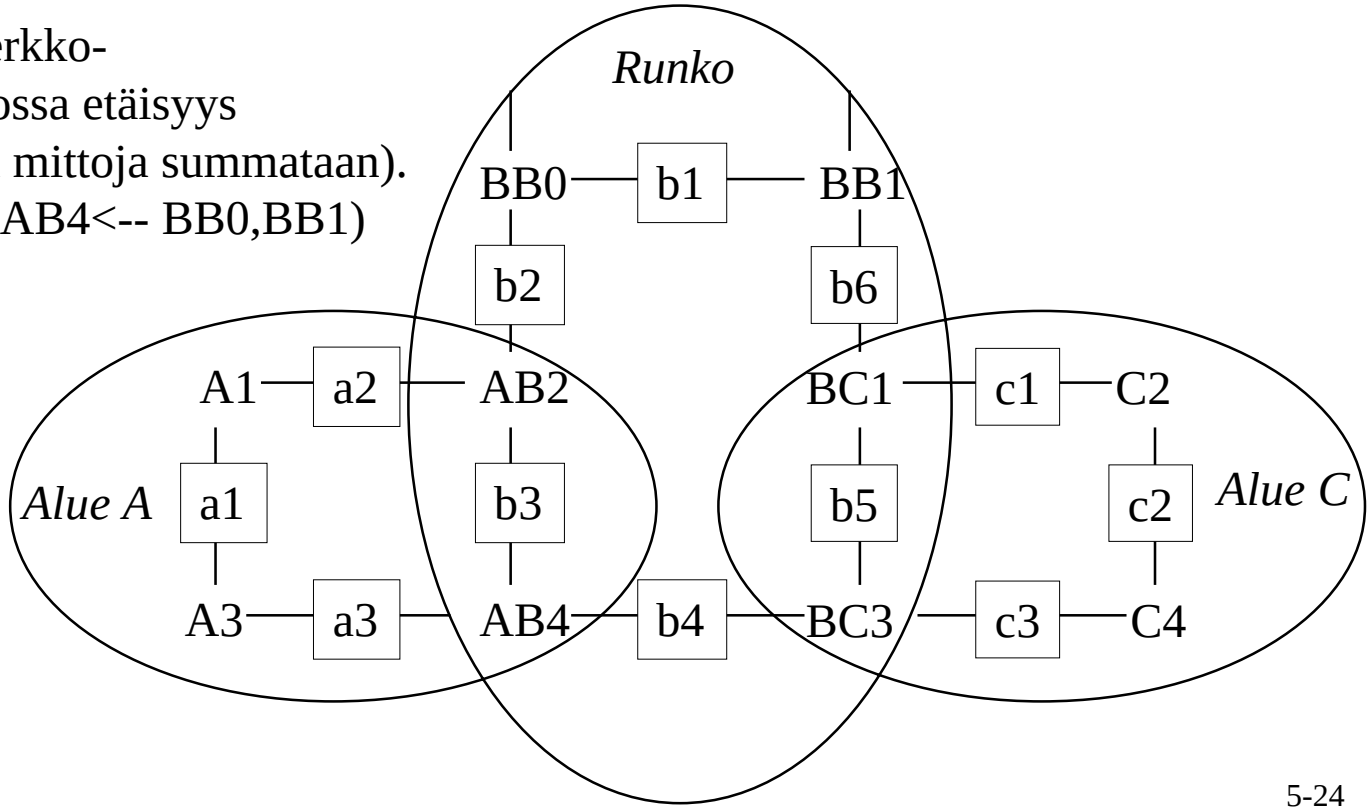
Huom: kiinteitä yhteyksiä kannattaa minimoida, koska ne ovat kalliita

Jakamalla laaja verkko alueisiin OSPF helpottaa levitystä ja pienentää linkkikantoja

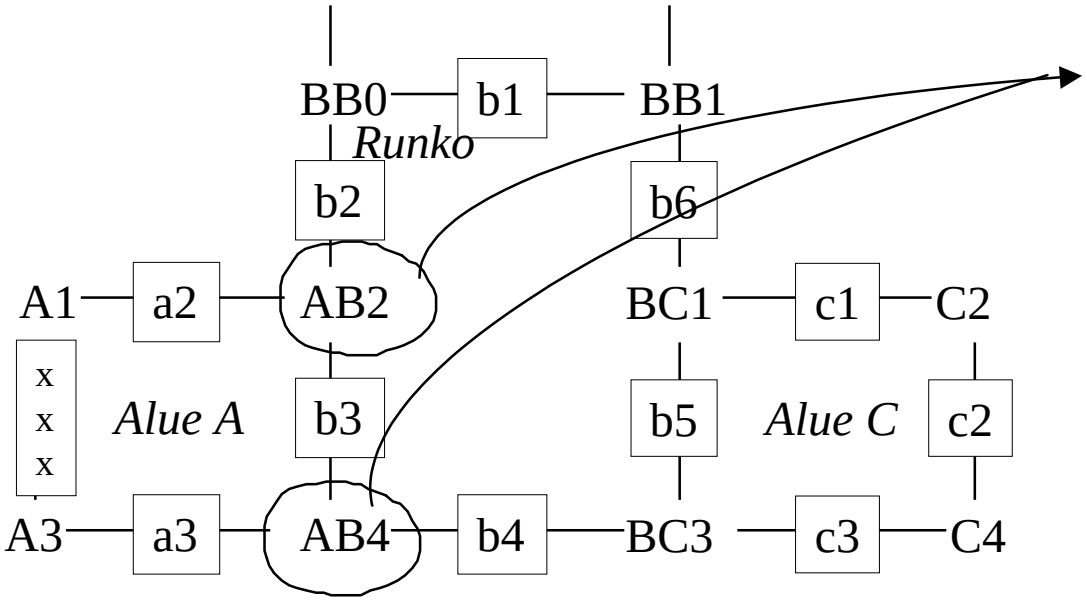


Muiden alueiden (ali)verkot kuvataan yhteenvetotietueissa, joiden mitta lasketaan RIP:n tapaan

Alueen A linkkikanta:
- a1
- a2
- a3
- rungon ja Alueen C aliverkko-
  tietueet (<- AB2,AB4), jossa etäisyys
  ABx--bz tai ABx--cy (eli mittoja summataan).
- ulkoiset tietueet (<-AB2,AB4<-- BB0,BB1)

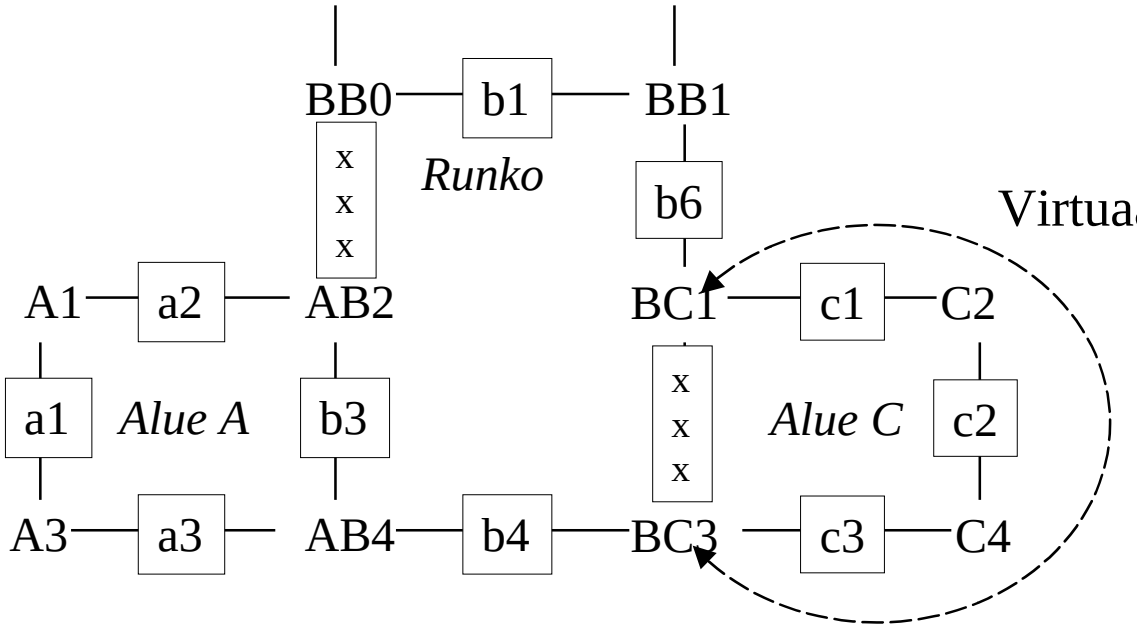


OSPF elpyy helposti alueiden vikatilanteista



AB2 ja AB4 mainostavat vain niitä verkkoja, jotka ne todella voivat saavuttaa: AB2: a2 ja AB4: a3.
Runko ei tunne Alueen A tarkkaa rakennetta, vaikkakin se tuntee A:n kaikki saavutettavat aliverkot.

Virtuaalilinkki voi pelastaa rungon jakaantuessa lohkoihin vikojen takia



Virtuaalilinkki alueen C kautta:
etäisyys=c1+c2+c3

Tynkäalueella (stub area) kaikki ulkoiset reitit summataan oletusreittiin

- Jos OSPF alueella on vain yksi reunareititin, kaikki liikenne ulkoiseen Internetiin ja ulkoisesta Internetistä kulkee tämän reunareitittimen kautta. Ei maksa vaivaa mainostaa kaikkia Internetin reittejä tällaisella alueella.
- Reunareitittimiä voi olla myös useita, mutta niistä ei voi valita sopivinta kohteen perusteella
- NSSA - “Not So Stubby Area” on alue, jolla kaikki ulkoiset reitit on summattu oletusreittiin, paitsi jotkut.

LSA types in OSPF are

LS Type = 1 Router LSA -- describes set of active If and neighbors

LS Type = 2 Network LSA -- describes a network segment (BC or NBMA) along with the IDs of currently attached routers

LS Type = 3 Summary LSA --

LS Type = 4 AS Border Router summary LSA

LS Type = 5 AS- external LSA -- descr ext routes

Hierarchical Routing

LS Type = 6 Group Membership LSA (MOSPF - Multicast)

LS Type = 7 NSSA LSA -- to import limited external info

LS Type = 8 (proposed) external attributes LSA (in lieu of Internal BGP)

Alueen OSPF reitittimillä on samansisältöinen linkkien tilatietokanta

Kannassa on 5 “LS” tietuetyyppiä
1.reititin (router LSA)
2. Verkko (network LSA)
3. IP-verkon yhteenveto
4. reunareitittimen yhteenveto
5. ulkoinen
.. yhteenvedoilla on sama formaatti
6. Monilähetys LSA
7. NSSA tietue
jne ...

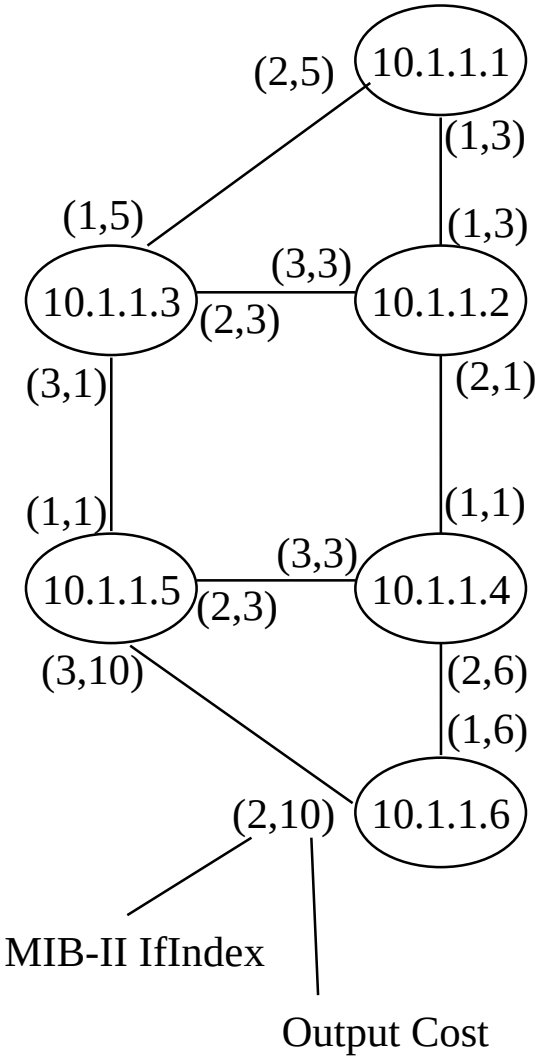
Yhteinen linkintilamainoksen otsikko

LS ikä	**optiot**	**LS tyyppi**
Linkin tila ID		
Mainostava reititin		
LS järjestysnumero		
LS tarkistussumma	**pituus**	

Avain

LS ikä - sekunteina mainoksesta
optiot: E - ulkoiset linkit
T - palvelun tyyppi -- eri mittoihin perustuva reititys

Example of the Router LSA



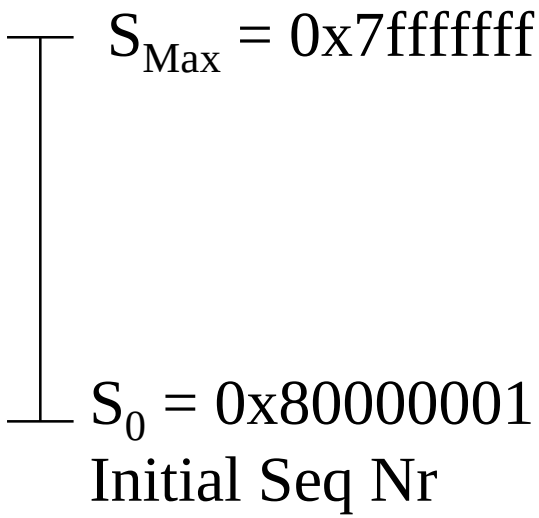
0 seconds	LS Age		Options	LS Type
	Link State ID			
	Advertising Router			
	LS Sequence Number			
0x9b47	LS Checksum		Length	
0 (ordinary)	Router Type	0	Nrof links	
	Link ID			
	Link Data			
1 (pt-t-pt), 0	Lnk Type	#Tos met	Metric	
	Link ID			
	Link Data			
1 (pt-t-pt), 0	Lnk Type	#Tos met	Metric	
	Link ID			
	Link Data			
3 (stub netw), 0 Tos metrics	Lnk Type	#Tos met	Metric	

E-bit, LS Type 1, (Router LSA)
10.1.1.1
10.1.1.1
0x80000006
60 bytes
3
10.1.1.2 (Neighb)
IfIndex 1 (Unnum)
3
10.1.1.3 (Neighb)
IfIndex 2 (Unnum)
5
10.1.1.1
255.255.255.255
0

Router 10.1.1.1's router-LSA

Length = 24 + 3 * 12 = 60 bytes
Router with 100 interfaces:
length = 24 + 100 * 12 = 1224 bytes

LSA Sequence Numbers

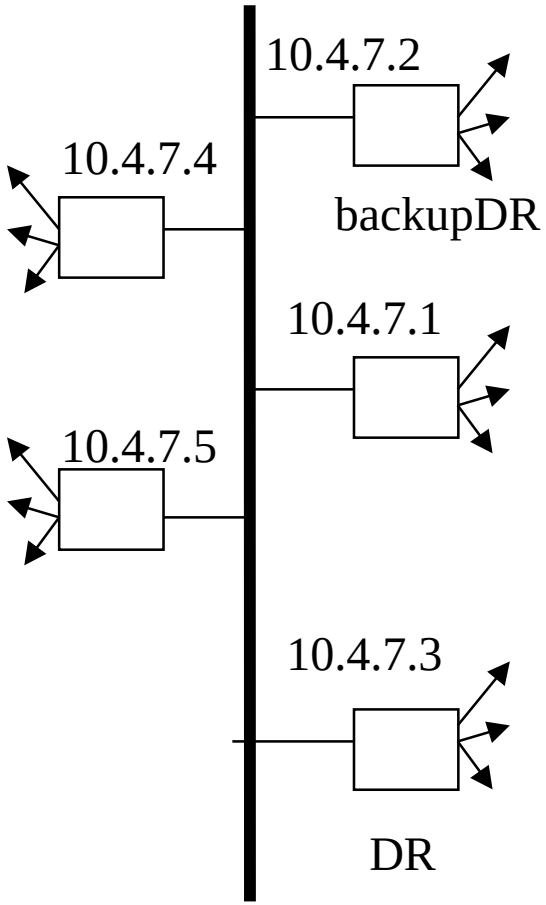


- To roll the space over, first delete record with S_{Max}
- A router may update a self originated record only once in 5 sec.
- In absense of errors rolling the space over takes at least 600 years.
- LS Age is updated during flooding at each step. Records with max Age are discarded. This breaks inf loops.

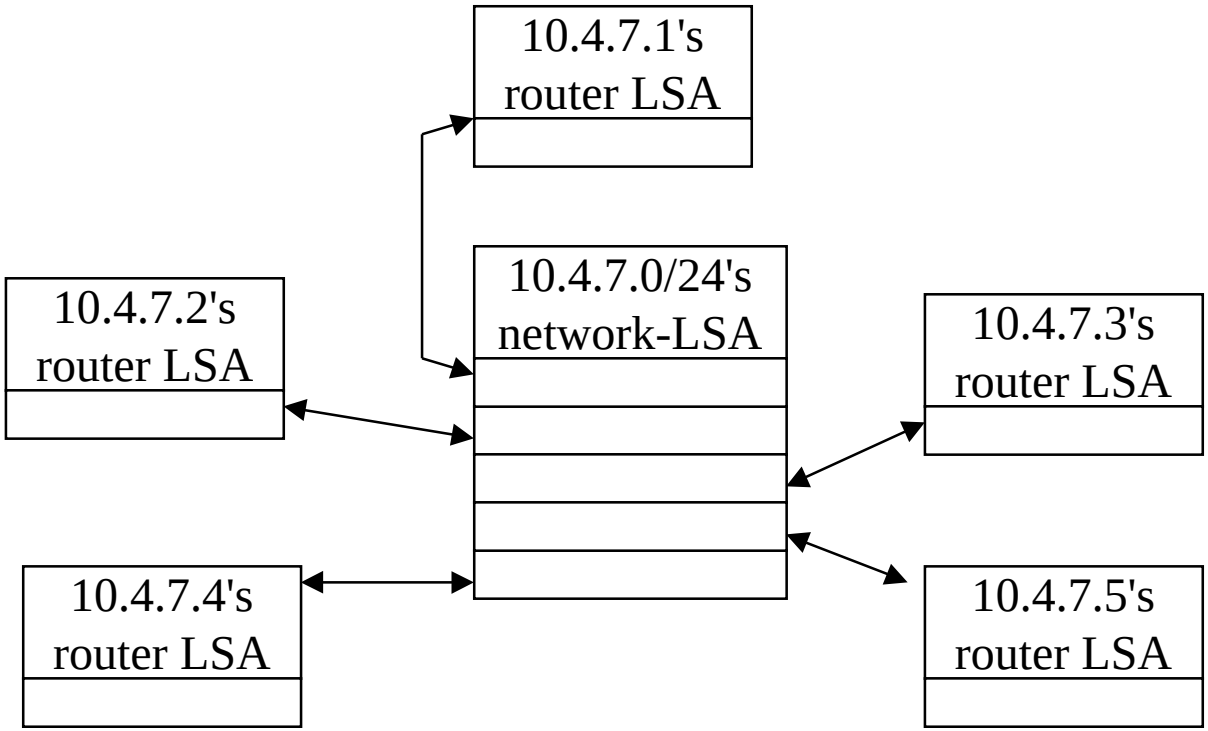
OSPF timeouts - LS Age field

Constant	Value	Action of OSPF router
MinLSArrival	1 second	Max rate at which a router will accept updates of any LSA via flooding
MinLSInterval	5 seconds	Max rate at which a router can update an LSA
CheckAge	5 min	Rate to verify an LSA Checksum in DB
MaxAgeDiff	15 min	When Ages differ more than 15 min, they are considered separate. Smaller LS age - newer!
LSRefreshTime	30 min	A Router must refresh any self-originated LSA whose age has reached 30 min.
MaxAge	1 hour	LSA is removed from DB.

Network-LSA reduces LinkDB for BC networks



DR - designated router

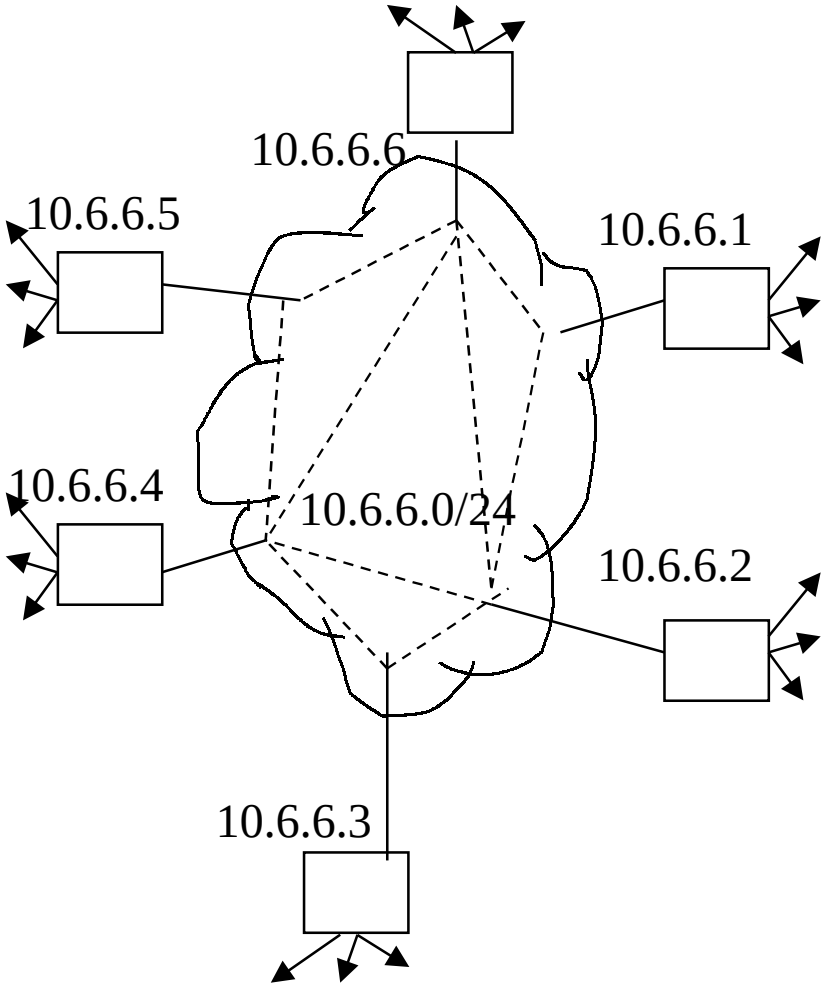


Network-LSA is generated by DR
Network-LSA reflects DB sync status with DRboth ways
This reduces nrof link records from O(n*(n-1)) to n*2.
Particularly important if Network is ATM or FR with a lot of routers attached!

Nonbroadcast Multiaccess(NBMA) subnets support many routers communicating directly but do not have BC capability

- Examples are ATM, Frame Relay, X25
- IP routing requires more manual configuration
- Designated router and backup DR concept reduce the nrof adjacencies
- The model is prone to failures that may be hard to track

Point-to-multipoint subnet is more robust but less efficient



- There is no DR nor backup DR
- Every OSPF router maintains adjacencies with all neighbors with whom it has direct connectivity
- Alternative is a set of NBMA networks
- Next hop routing protocol improves scalability

OSPF packets - the protocol itself

- OSPF works directly on top of IP. OSPF protocol number is 89.
- For most packets TTL = 1, except for hierarchical routing
- Dest IP address = Neighbors IP address or AllSPFRouters (224.0.0.5) or AllDRouters (224.0.0.6)
- Packet types are
  - Type 1: Hello
  - Type 2: Database Description packet
  - Type 3: Link State Request packet
  - Type 4: Link State Update packet
  - Type 5: Link State Acknowledgement packet

OSPF protokolla toimii suoraan IP:n päällä

OSPF:ssä on 3 osaprotokollaa:

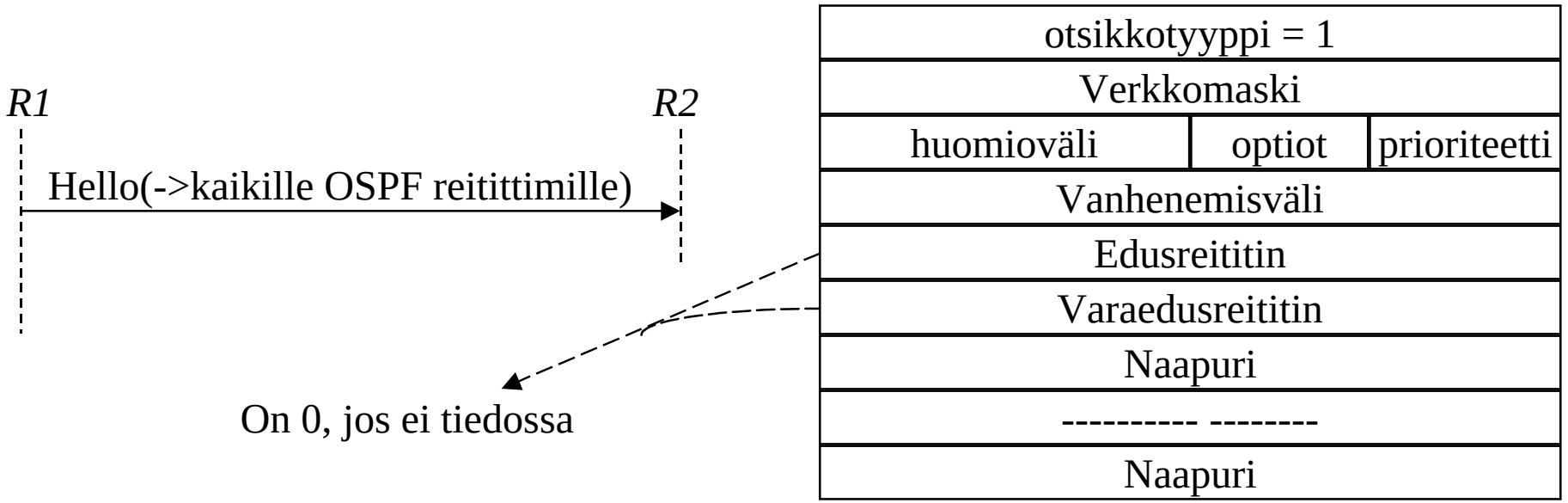
- Huomio (Hello) protokolla
- Tiedonvaihto (exchange) protokolla
- Levitys (flooding) protokolla

OSPF sanomien yhteinen otsikko on:

Versio	Tyyppi	paketin pituus
Reitittimen ID		
Alueen tunnus		
Tarkistussumma		Autentikointityyppi
Autentikointi		
Autentikointi		

OSPF nykyversio on 2.
Tyyppi erottelee OSPF sanomat toisistaan
Autentikointi perustuu salasanoihin, jolloin sen merkitys on rajallinen tai uudempiin kryptografisiin menetelmiin (1995).

Huomioprotokolla varmistaa, että linkit toimivat ja valitsee edus- ja varaedusreitittimen



- Naapuri - lista naapureista, joilta on tullut viestejä viimeisen vanhenemisvälin aikana
- Huomioväli kertoo, kuinka usein paketteja lähetetään.
- Optiot - E -ulkoisia linkkejä, T - TOS reititys onnistuu.
- Prioriteetti kertoo kelpoisuudesta edusreitittimeksi.
- Huomioviestien pitää kulkea linkillä molempiin suuntiin, jotta linkki kelpaisi reitiksi

Huomioprotokollan avulla valitaan edus- ja varaedusreititin

1. Vaalikelpoisuus saavutetaan yhden vanhenemisvälin jälkeen
   mikäli kaksisuuntainen saavutettavuus on OK.

2. Varaedusreitittimeksi valitaan korkeimman prioriteetin omaava niistä,
   jotka ilmoittautuivat. Tasatilanteessa valitaan suurimman ID:n omaava.

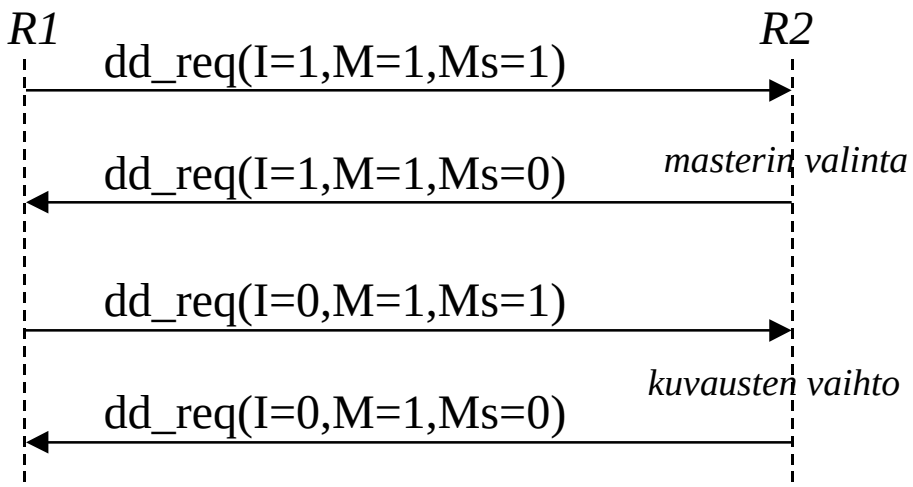
3. Jos kukaan ei ilmoittautunut varaedusreitittimeksi, valitaan korkeimman
   prioriteetin omaava naapuri. Tasatilanteessa suurimman ID:n omaava.

4. Edusreitittimeksi valitaan ilmoittautuneista yo. säännöllä.

5. Jos yksikään ei ilmoittaudu edusreitittimeksi, varaedusreititin ylennetään.
   Kohdat 2 ja 3 suoritetaan uudestaan varaedusreitittimen valitsemiseksi.

6. Muutoksia minimoidaan siten, että korjattu entinen edusreititin
   ei kiirehdi ilmoittautumaan uudelleen. Kohtia 2....5 suoritetaan jatkuvasti.

Vaihtoprotokolla alkusynkronoi linkkikannan edusreitittimen kannan kanssa

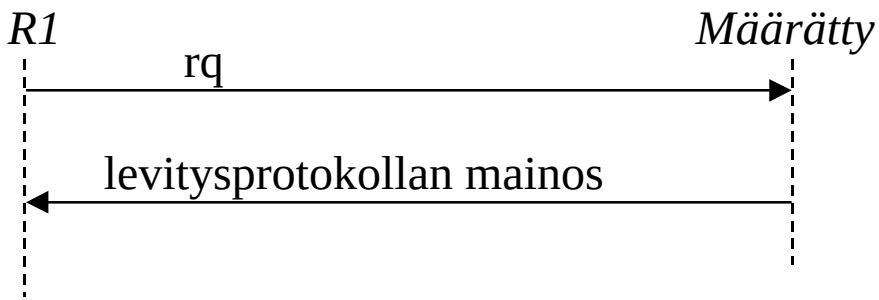


otsikkotyyppi = 2 (dd)			
0	0	optiot	0 IMMs
dd järjestysnumero			
Linkin tilatyyppi			
Linkin tila ID			
Mainostava reititin			
Linkin tilan järjestysnumero			
Linkin tilan tark.summa		Linkin tilan ikä	
----- -----			

- Master lähettää oman kantansa kuvauksen järjestysnumeroiduissa paketeissa
- Slave kuittaa paketit lähettämällä vastaavat omat kuvauksensa.
- Vaihto jatkuu kunnes molemmat ovat lähettäneet omat kuvauksensa.
- Erot kirjataan kyseltävien tietueiden listaan.

I = initialize
M= more
Ms = Master/slave
(aloituspaketti on harmaa)

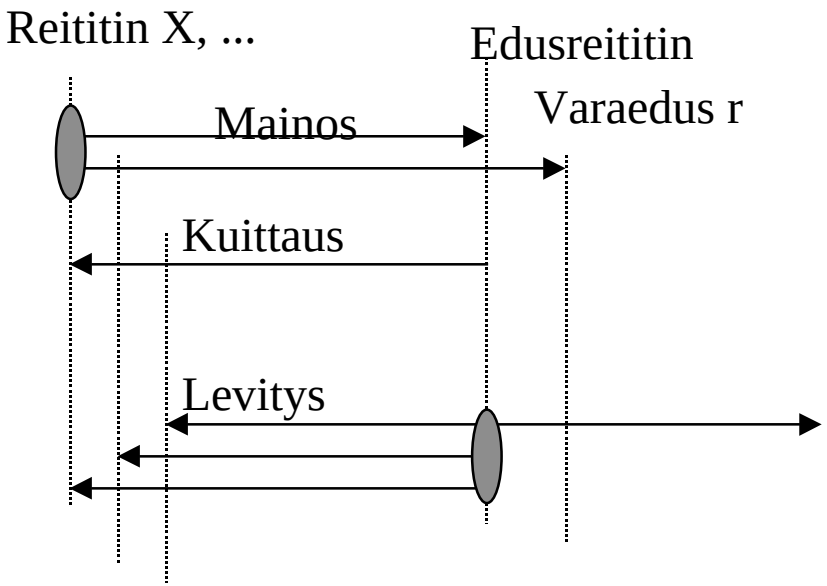
Tietueiden sisältö kysellään pyyntöpaketeilla, jotka kuitataan levitysprotokollan paketein



otsikkotyyppi = 3 (rq)
Linkin tilatyyppi
Linkin tila ID
Mainostava reititin
----- -----

- Vastausta odotetaan uudelleen lähetyksen väli, jos ei vastausta, pyyntö toistetaan.
- Jos kyseltävät tietueet eivät mahdu yhteen pakettiin, tietueet jaetaan useaan kyselyyn.
- Jonkin mennessä pieleen, palataan herkästi roolien uudelleen neuvotteluun.
- Tietuesisältöjen kysely voi alkaa heti, kun yksikin eroava tietue on havaittu, jolloin dd-pakettien ja rq pakettien vaihto tapahtuu rinnan.

Levitysprotokolla ylläpitää alueen linkkikantojen yhtenäisyyttä



otsikkotyyppi = 4 (upd.)
Mainosten lukumäärä
Linkin tilamainokset(ks.tietueformaatti)
----- -----

otsikkotyyppi = 5 (ack.)
Linkin tilamainosten otsikot
----- -----

- Alkuperäisen mainoksen lähettää aina linkistä vastaava reititin.
- Mainosta levitetään levityssääntöjen mukaan koko alueelle (age = age + 1).
- Uuden tietueen kuittaus voidaan BC verkossa korvata levityssanomalla
- Yksi ack voi kuitata monta mainosta.

Linkkitietueilla on ikä, vanhat poistetaan kannasta

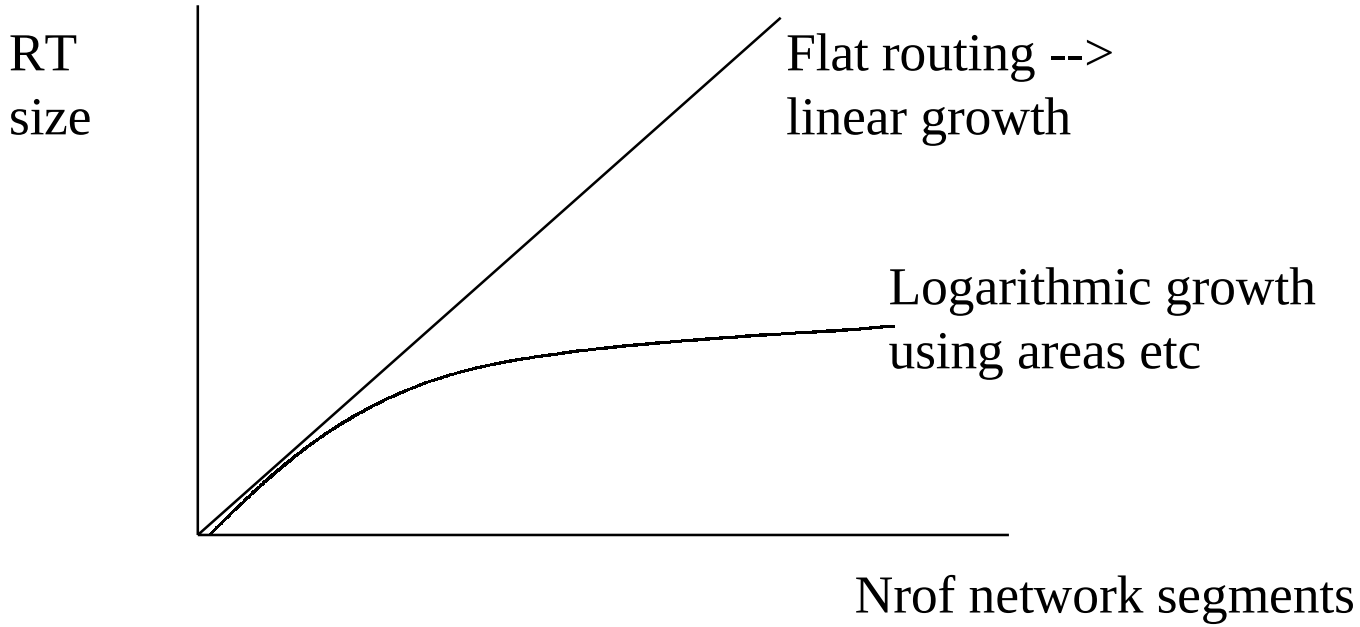
1. Ikä = mainosten kulkemien linkkien lkm + sekunnit vastaanotosta
2. Maksimi-ikä = 1 tunti
3. Jokaista tietuetta pitää mainostaa vähintään 30 min välein.
   Uusi mainos nollaa iän ja inkrementoi tietueen järjestysnumeron.
4. Kun ikä tulee täyteen (MaxAge=1h), lähetetään mainos.
5. MaxAge mainos hyväksytään ja levitetään - tämä poistaa vanhan tiedon.
6. Jos mainoksen ikäero kantaan on pieni mainosta ei levitetä, jotta saman tiedon kopiot eivät kuormittaisi verkkoa
7. Jos MaxAge tietuetta ei löydy, mainos ei aiheuta toimenpiteitä, (koska reititin on jo ehtinyt poistaa vanhan tiedon.)

Summary of OSPF subprotocols

	Hello msg (1)	DD (2)	LS rq (3)	LS upd (4)	LS ack (5)
Hello protocol	X				
Database exchange		X	X	X	X
Flooding protocol				X	X

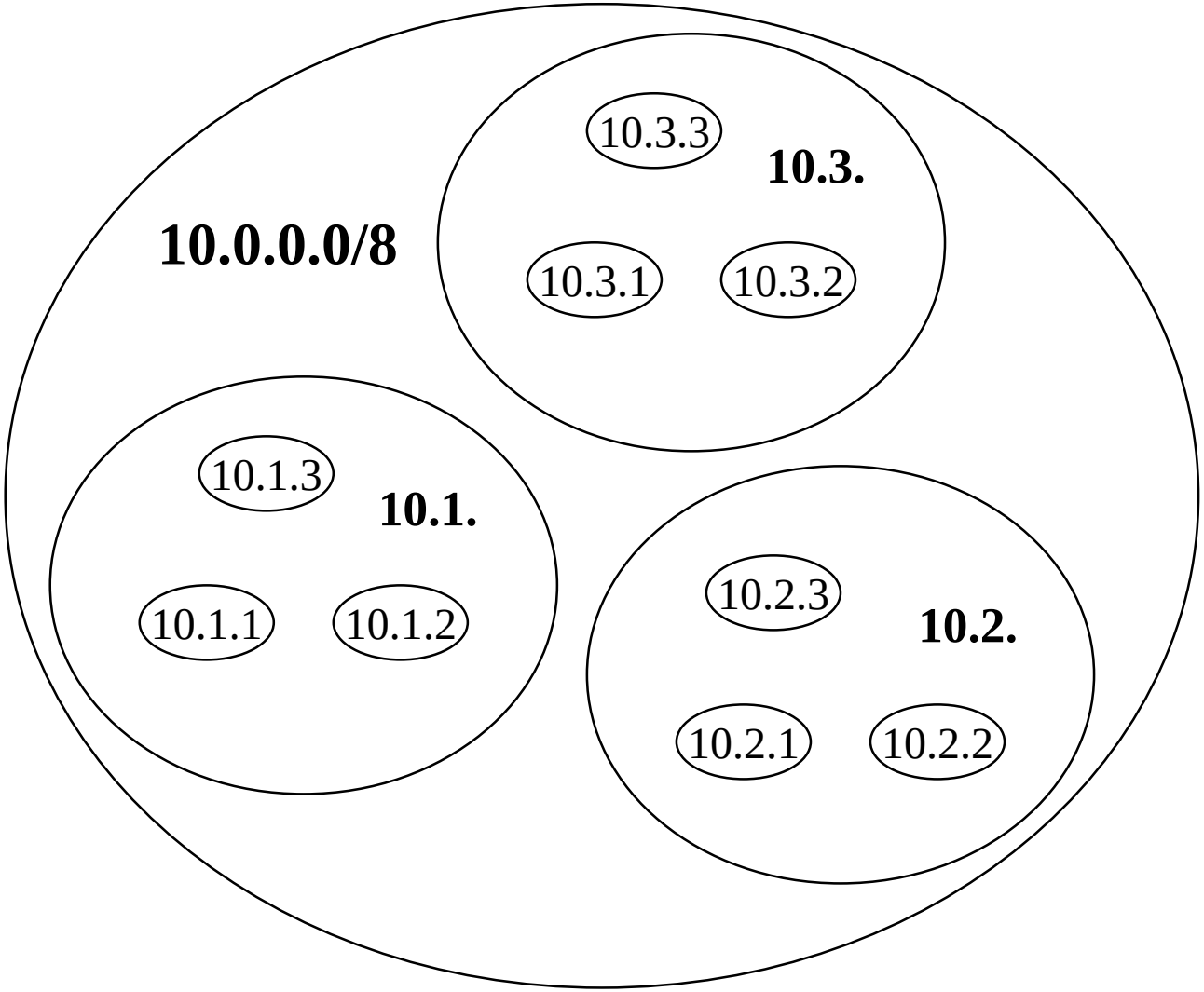
OSPF without Dijksta's algoritm and with more generic data objects is SCSP (Server Cache Synchronization Protocol) which is proposed as the basis for *Telephony Routing Information Protocol* - studied in our Lab.

The purpose of hierarchical routing in OSPF is to reduce routing table growth



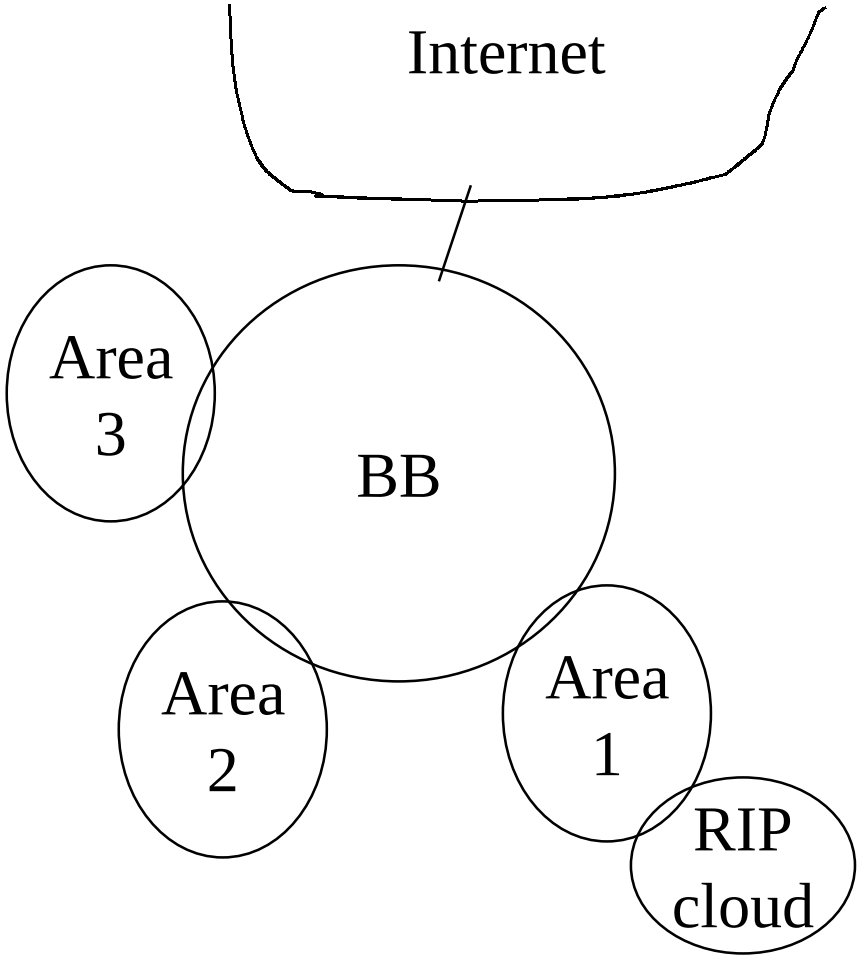
The cost is: sometimes suboptimal routes.

Example of use of routing hierarchy



Example:
- 16 segments in each lowest level network
- flat routing:
  RTsize= 16*9=144
- areas 10.1.1:
  16 local routes +
  10.1.2/24
  10.1.3/24
  10.2/16
  10.3/16
  == 20 RT entries!

OSPF supports 4 level routing hierarchy



Level	Description
1	Intra-area routing
2	Inter-area routing
3	External Type 1 metrics
4	External Type 2 metrics

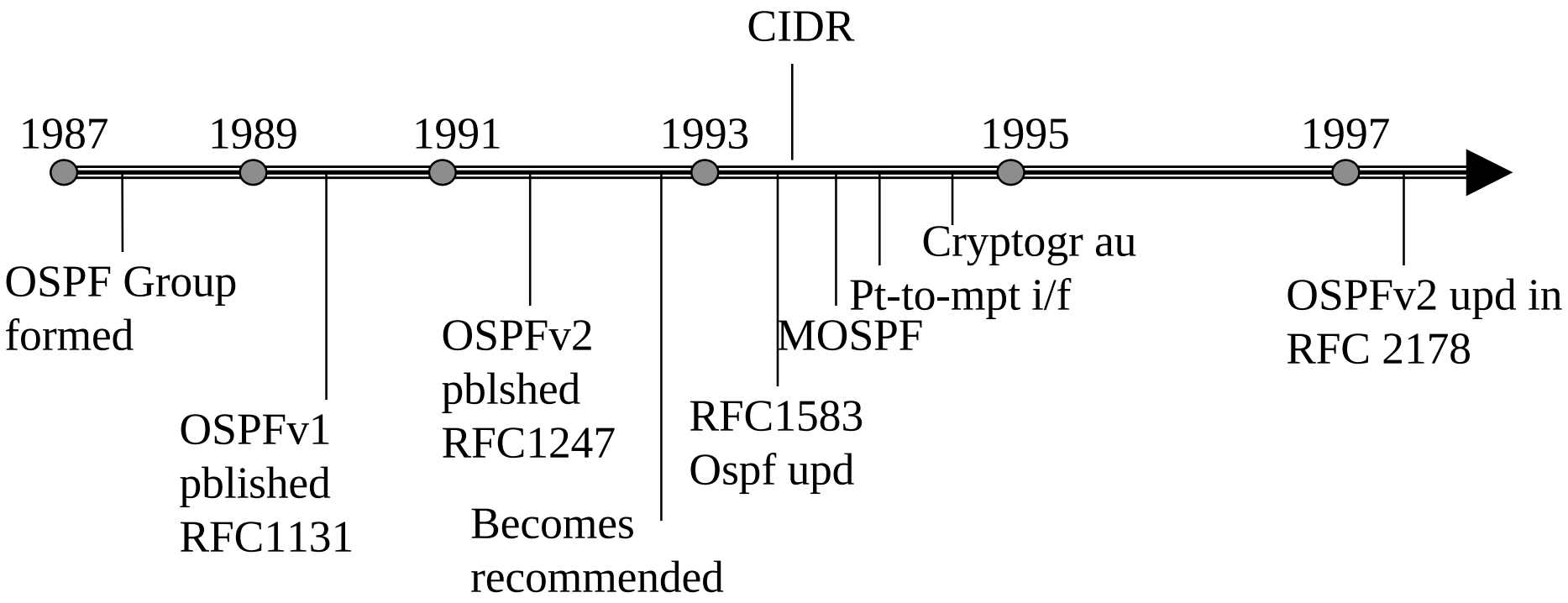
- Type 1 metrics are of the same order as OSPF metrics, e.g. hop count (for RIP and OSPF)
- Type 2 metrics are always more significant then OSPF internal metrics

Why is it difficult to route packets around network congestion?

- BBN ARPANET link state metric varied with the length of the output queue of the link --> lead to route trashing.
- The problem is there is no route pin-down for existing traffic.
- By limiting the range of the metric changes, an equilibrium could be reached. Nevertheless routing instability is the problem.

When QoS or Class of Service a'la DiffServ is introduced this problem again becomes important.

OSPF development history



CIDR - Classless Inter Domain Routing

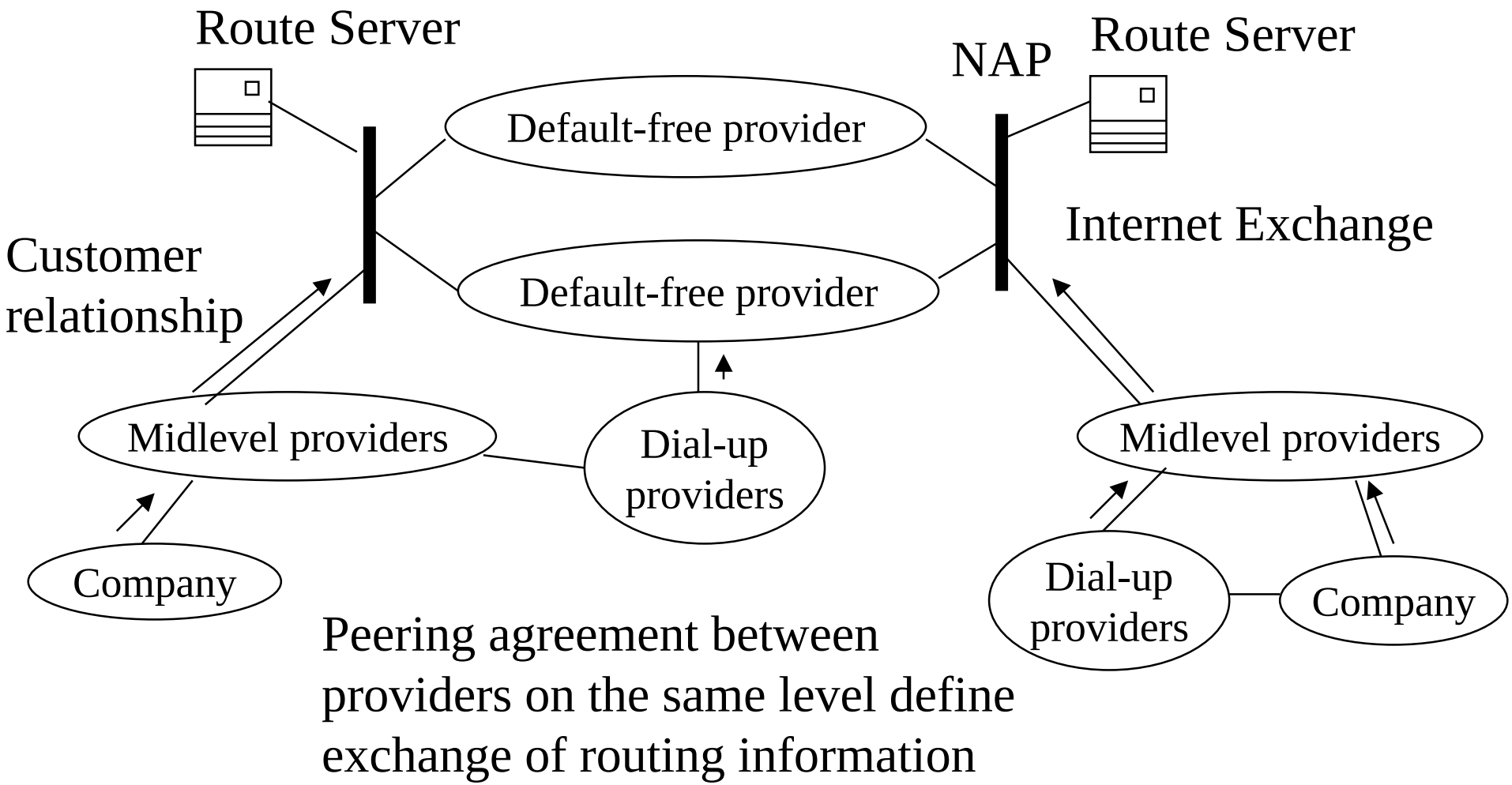
Internetin kasvu on pakottanut ottamaan käyttöön CIDR osoitearitmetiikan, jolla IP-osoitteavaruus saadaan tehokkaampaan käyttöön.

Forwarding process
Router implementation

CIDR vaikuttaa useisiin reititysprotokolliin

- AS - Autonomous System on Internetin hallinnollinen alue, eli osa verkosta, jolla on yksi omistaja.
- AS:lla käytössä on yleensä yksi (sisäinen) reititysprotokolla, esim OSPF.
- AS välillä käytetään ulkoista reititystä.

Organisation of the Internet as Autonomous Systems

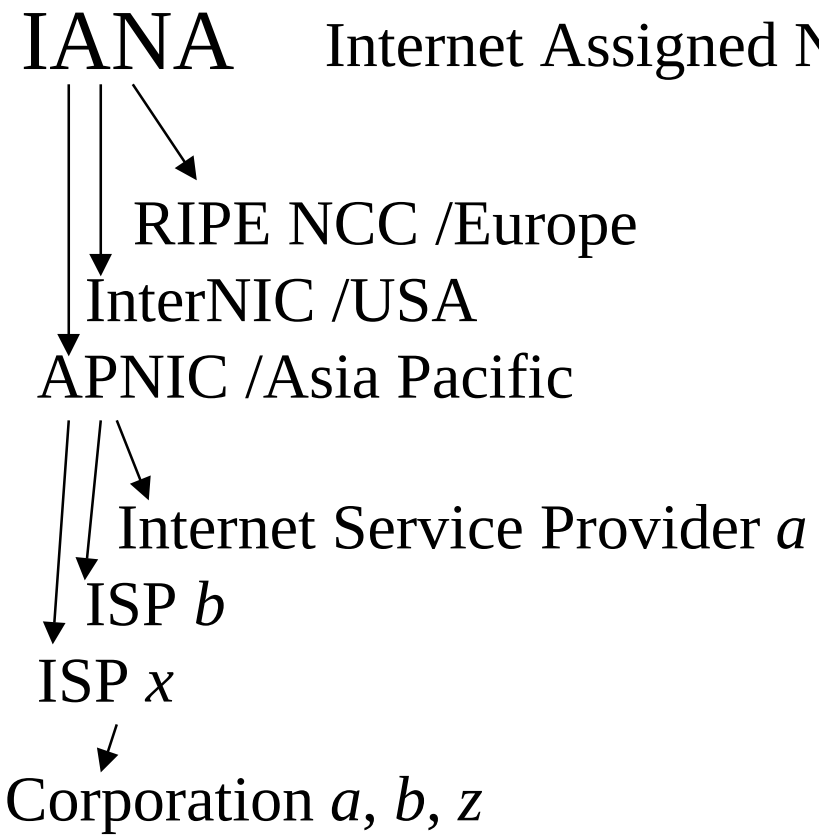


Peering agreement between providers on the same level define exchange of routing information

History of the Internet Core

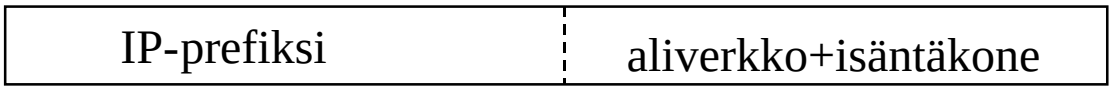
…..1985 Arpanet
…..1987 NSFNET 56k lines
…..1992 NSFNET T1 lines (1.5M)
…. 1995 NSFNET T3 lines (24M)
1995 NSFNET decommissioned
1995… Commercial (UUNET,MCI, Sprint...

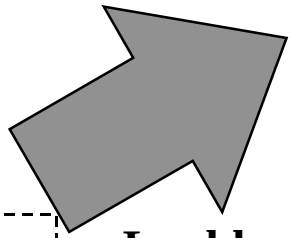
Internet Addresses are assigned by a hierarchy of registrars



- This model leads to provider addressing.
- Due to Provider addressing an ISP needs to advertise shorter prefixes leading to savings in routing table size in the Backbone

CIDR aritmetiikka perustuu 32-bitin IP-osoitteen jakamiseen vapaasti verkko- ja häntäosaan





MSB(t)	Verkkotaso	Host	Luokka
0	7 bittiä	24 bittiä	A
10	14 bittiä	16 bittiä	B
110	21 bittiä	8 bittiä	C

Joukko peräkkäisiä C-luokan osoite-lohkoja esitetään näin:

194.51.120.0 - 194.51.127.255 =

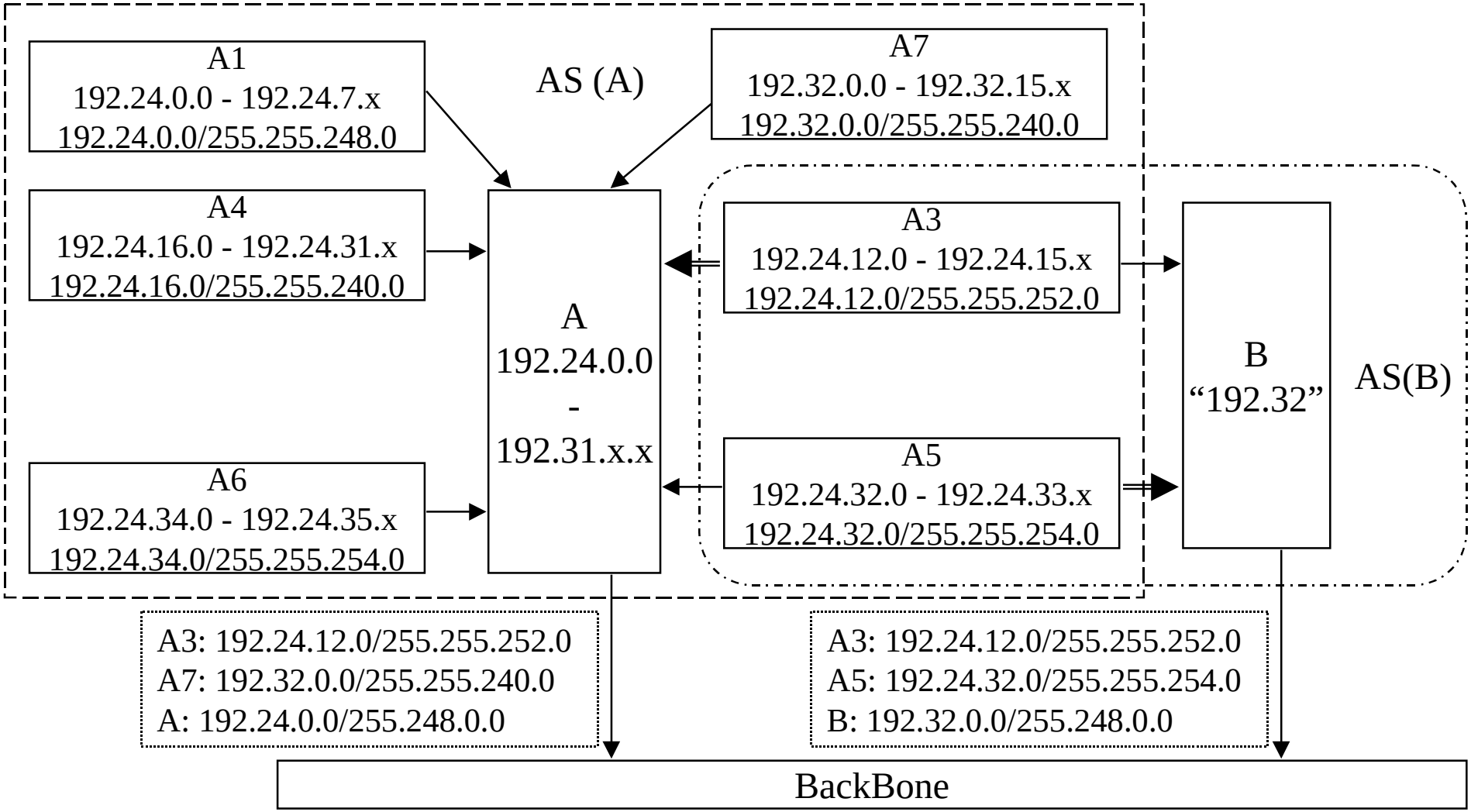
alku = 194.51.120.0
maski = 255.255.248.0

CIDR muuttaa reittien mainostusperiaatetta

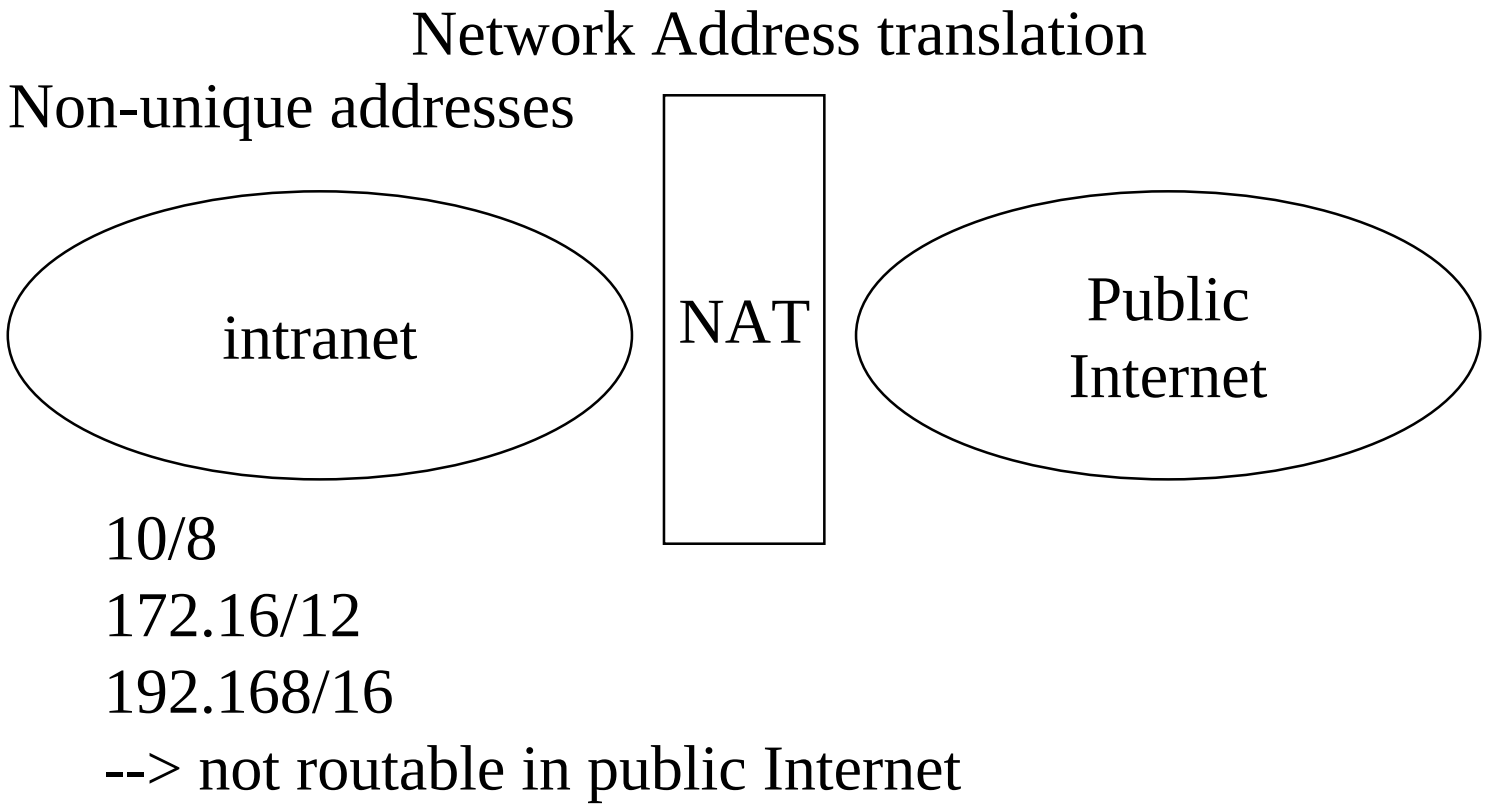
Sääntö1: Reititys kohteeseen tapahtuu aina pisimmän matchaavan osoitteen perusteella. ---> useaan AS:ään liittyvien (moni-kotisten) verkkojen osoitteita ei voi aggregoida tavalliseen tapaan.

Sääntö 2: Verkko, joka aggregoi useita reittejä, tuhoaa paketit, jotka matchaavat summattuun osoitteeseen, mutta eivät yhteenkään summan tekijään. (tällä estetään silmukoiden muodostumista).

Esimerkki


A1
192.24.0.0 - 192.24.7.x
192.24.0.0/255.255.248.0
AS (A)
A7
192.32.0.0 - 192.32.15.x
192.32.0.0/255.255.240.0
A4
192.24.16.0 - 192.24.31.x
192.24.16.0/255.255.240.0
A
192.24.0.0
-
192.31.x.x
A3
192.24.12.0 - 192.24.15.x
192.24.12.0/255.255.252.0
B
“192.32”
AS(B)
A6
192.24.34.0 - 192.24.35.x
192.24.34.0/255.255.254.0
A5
192.24.32.0 - 192.24.33.x
192.24.32.0/255.255.254.0
A3: 192.24.12.0/255.255.252.0
A7: 192.32.0.0/255.255.240.0
A: 192.24.0.0/255.248.0.0
A3: 192.24.12.0/255.255.252.0
A5: 192.24.32.0/255.255.254.0
B: 192.32.0.0/255.248.0.0
BackBone

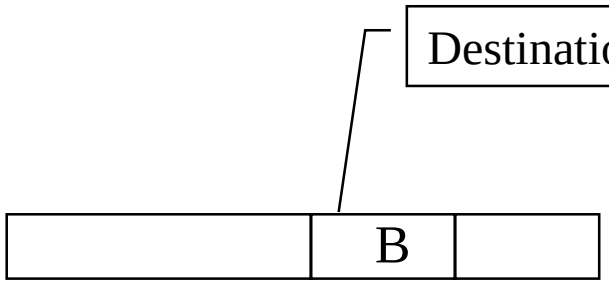
Network Address Translation (NAT) preserves address space and improves security



Packet Forwarding and Router Architectures

Packet forwarding in a router

RFC 2097 - view routing table
by SNMP MIB



Incoming packet

Forwarding Information Base

Network address	Mask	Port
a.b.c.d	n	

Nrof lines approaches 100 000

B and Mask(I) =
Network Address

Mask presentation
- nrof bits in prefix
  or
- prefix by 1's, host by 0's

⇒ List of Candidate destination ports

⇒ Take the longest match
(= longest mask)

Modifications of forwarding process

- Multipath routing
  - e.g. hash(source IP address, destination IP address) produces one of the possible next hops.
- TOS routing
  - never widely used
  - has been removed from recommendations (RFC 2178 in 7/97)
- Source Routing (strict or loose)
  - max 9 hops can be specified in header options
  - has performance penalty
  - is considered a security hole (all packet may be dropped

More modifications of the forwarding process

- When there are too many packets to forward, some need to be dropped. To maintain a fair service drop algorithms are used
  - e.g Random Early Detection (RED)
- Scheduling algorithms manage the share of connections in the available bandwidth
  - e.g give 15 kbit/s to an audio conference or half of the link bandwidth to interactive services
  - Weighted Fair Queuing (WFQ) and Class Based Queuing (CBQ) are examples of scheduling algorithms

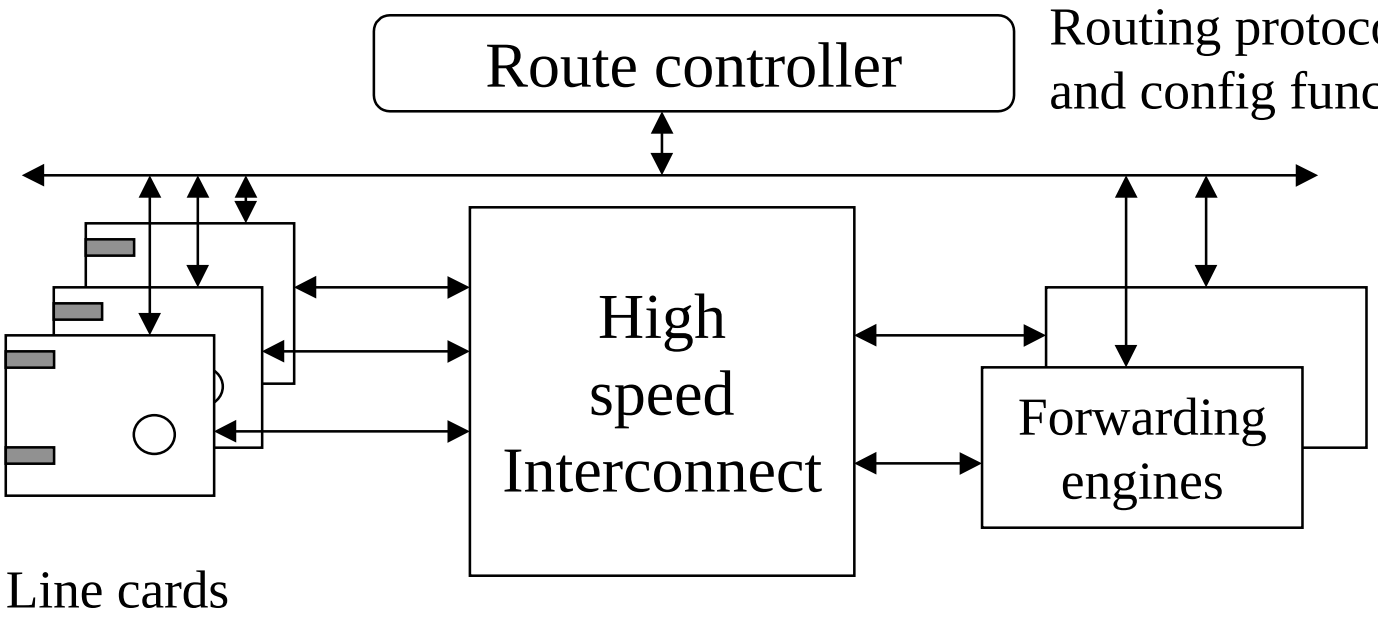
Routers support Security and problem resolution

- Security includes e.g. preventing unauthorised access to a company intranet
  - we talk about Firewalls
  - forwarding needs to check filtering rules on IP addresses and TCP port numbers
  - ISP routers may check all source IP addresses to trace security attacks
- A router may support RMON MIB
  - router allows traffic tracing for routing problem analysis

Routers can collect Statistics

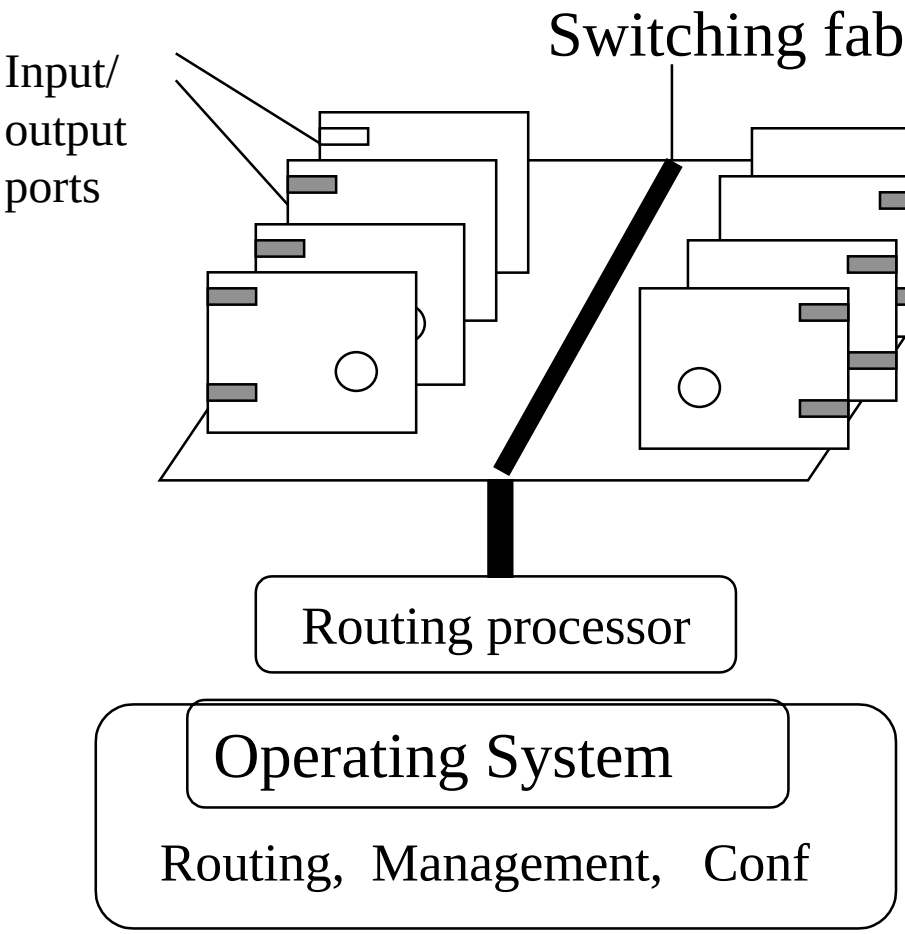
- Statistics are needed for Network Planning,
- Inter ISP accounting and
- Usage based charging

A non-trivial router architecture is



Forwarding capacity is increased by sharing the load between several forwarding engines

A faster router architecture is



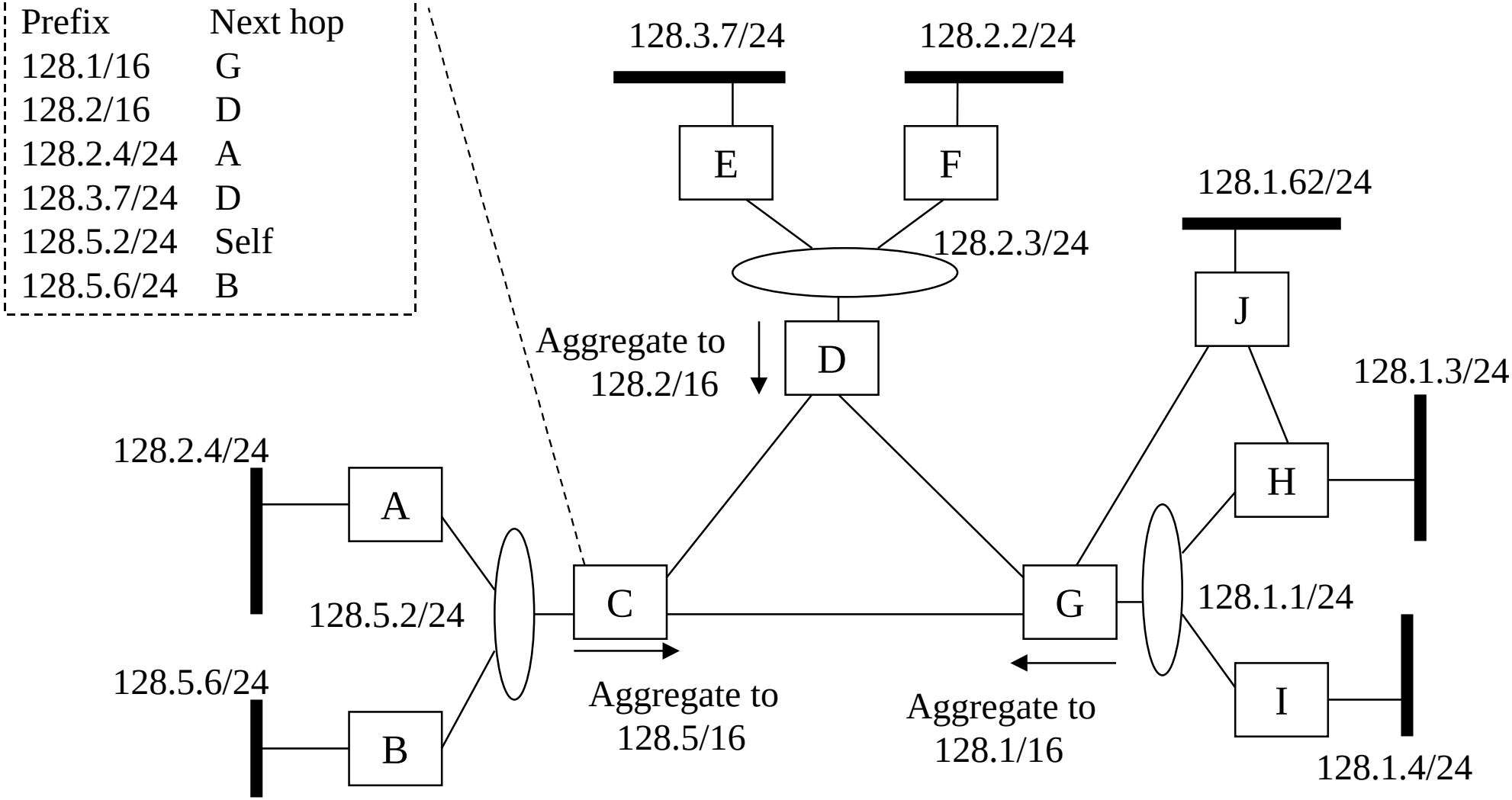
- Ports are located on line cards
- FIB is produced by Routing processor and downloaded to the line cards
- Switching fabric is a bus or shared memory

FIB Lookup speed is determined by nrof reads

- E.g. access time 30ns x 8 reads = 240 ns ==> >4 M lookups/s
- May need to backtrack --> poor worst case performance.
- Ways to improve performance:
  - Hardware oriented techniques
  - Table compaction techniques
    -- e.g. long trie branches with few leaves are packed into a node
  - Hashing techniques
    -- problem is unknown mask length -->
    -- (e.g. binary search on prefix length)

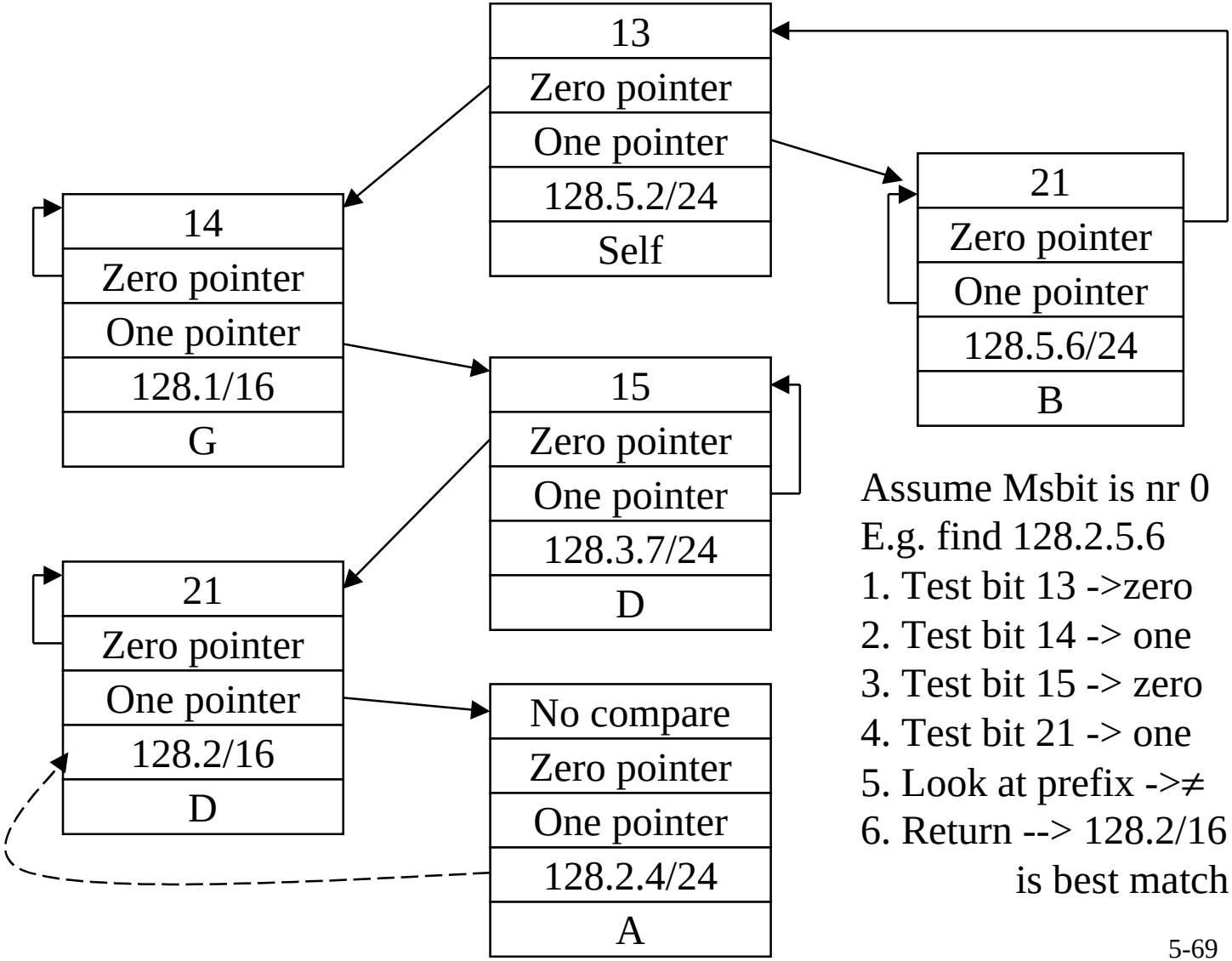
Route aggregation example

Prefix	Next hop
128.1/16	G
128.2/16	D
128.2.4/24	A
128.3.7/24	D
128.5.2/24	Self
128.5.6/24	B



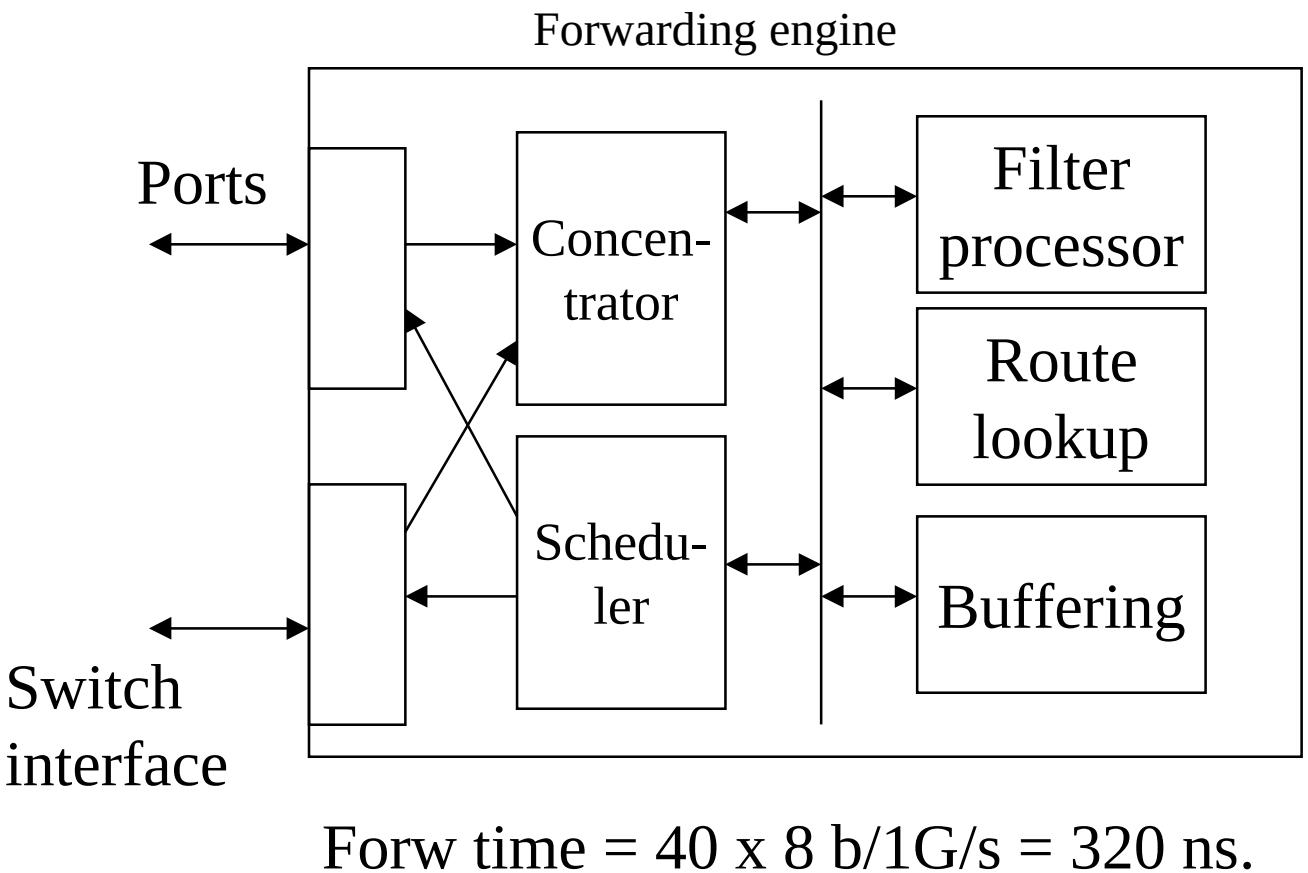
Route lookup may be based on Patricia tree

Prefix	Next hop
128.1/16	G
128.2/16	D
128.2.4/24	A
128.3.7/24	D
128.5.2/24	Self
128.5.6/24	B



Assume Msbit is nr 0
E.g. find 128.2.5.6
1. Test bit 13 ->zero
2. Test bit 14 -> one
3. Test bit 15 -> zero
4. Test bit 21 -> one
5. Look at prefix ->≠
6. Return --> 128.2/16
is best match

Forwarding speed can be increased by parallel processing



Forw time = 40 x 8 b/1G/s = 320 ns.

Example based on Bell Labs prototype.

Boxes are 33MHz… 66MHz FPGAs.

Can process all headers prior to buffering at 1Gbit/s line speeds!

--> can provide QoS.