

Routing Information Protocol

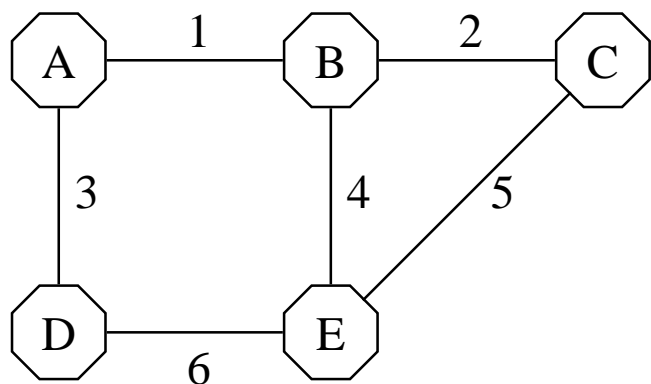
Distance Vector Routing Principles

Routing loops and countermeasures to loops

Bellman-Ford route calculations

RIP

RIP - Routing Information Protocol on sisäisen reitityksen perusprotokolla



Esimerkkiverkko, jossa solmut A, B, C ...

RIP on etäisyysvektoriprotokolla.

Tarkastellaan EV-protokollien toimintaperiaatetta

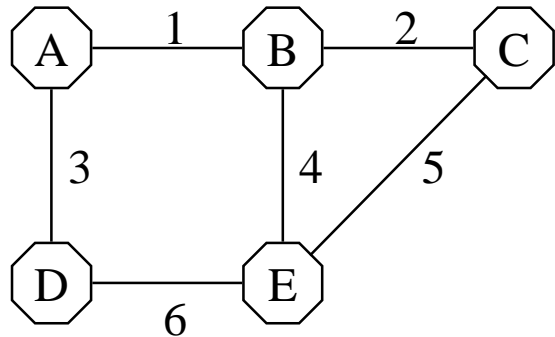
Alkutila: Noodit tuntevat omat osoitteensa ja liitäntänsä, mutta ei muuta

Solmu A muodostaa reititystaulun:

A:sta solmuun	Linkki	Kustannus
A	paikallinen	0

Taulua vastaa etäisyysvektori $A=0$.

Reititystaulujen muodostus käynnistyy, kun kaikki solmut lähettävät toisilleen omat etäisyysvektorinsa kaikista liitännöistä



$A=0$ →

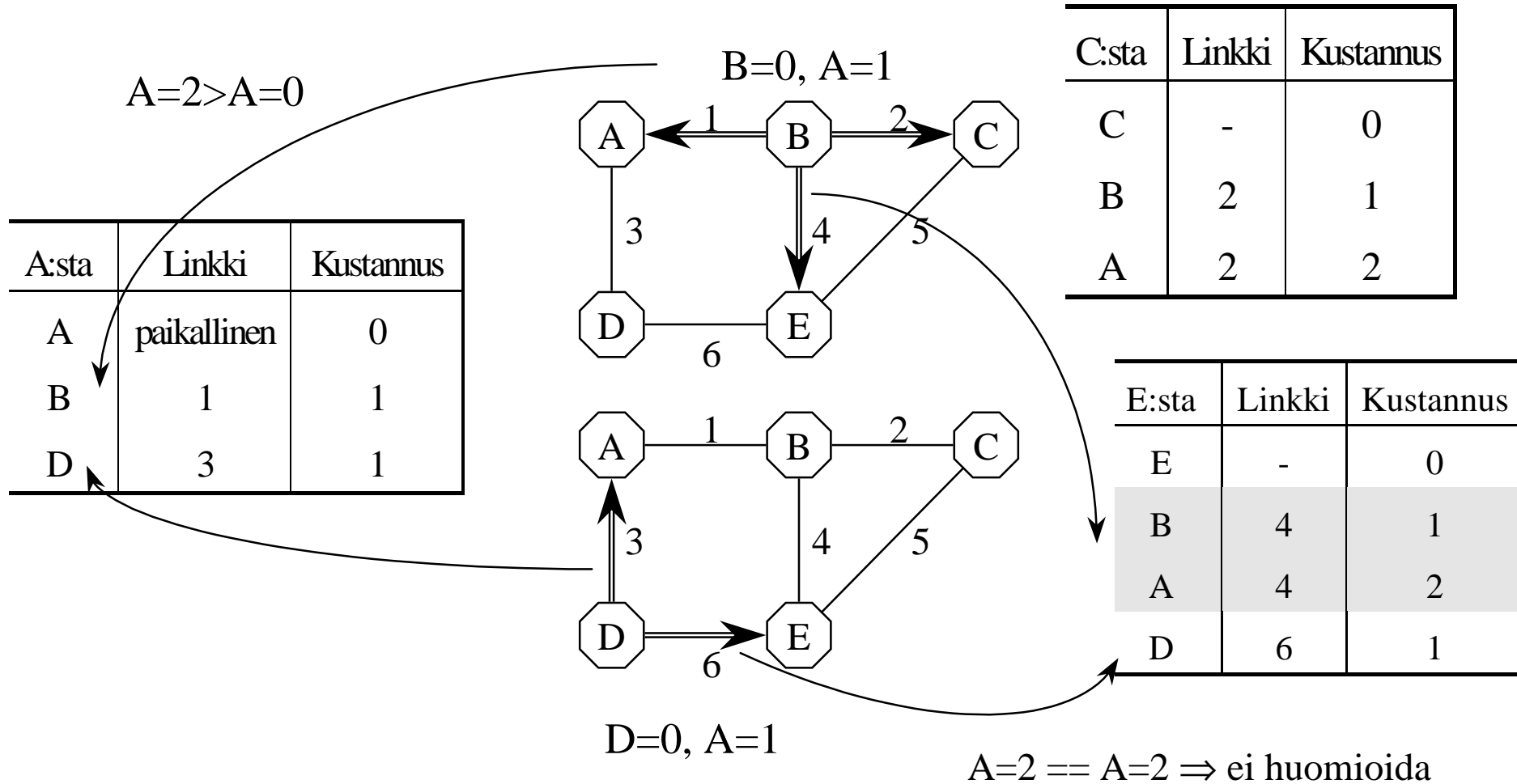
Tarkastellaan vastaottoa solmussa B

B:sta solmuun	Linkki	Kustannus
B	paikallinen	0

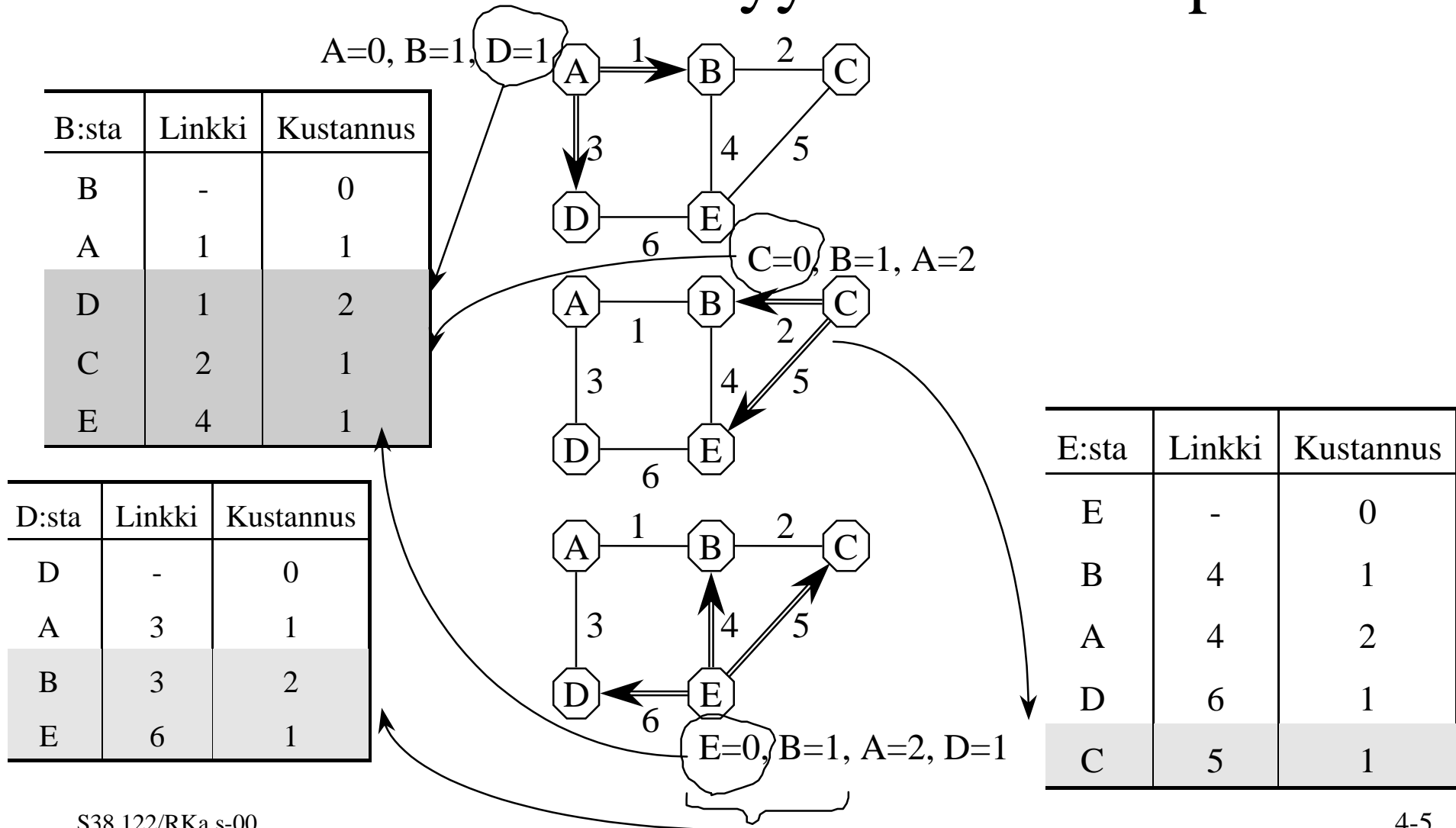
1. B lisää heti etäisyysvektoriin + 1 --> A=1 ja
2. B etsii tulosta omasta taulusta, ei löydy
3. B lisää saamansa tiedon reititystauluunsa, tulos on

B:sta solmuun	Linkki	Kustannus
B	paikallinen	0
A	1	1

B muodostaa oman vektorinsa ja lähettää sen kaikille naapureille



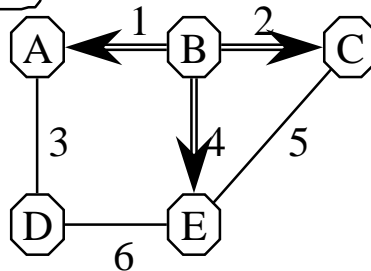
Solmut, joiden reititystaulut muuttuivat lähettävät uudet etäisyysvektorit naapureille



Muutokset lähetetään taas ...

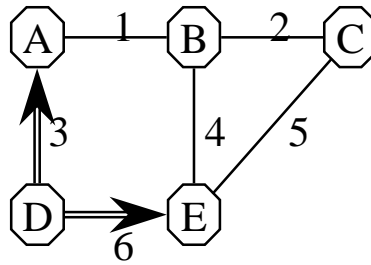
B=0, A=1, D=2, C=1, E=1

A:sta	Linkki	Kustannus
A	-	0
B	1	1
D	3	1
C	1	2
E	1	2

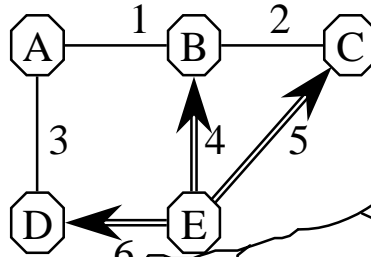


C:sta	Linkki	Kustannus
C	-	0
B	2	1
A	2	2
E	5	1
D	5	2

D=0, A=1, B=2, E=1



D:sta	Linkki	Kustannus
D	-	0
A	3	1
B	3	2
E	6	1
C	6	2



E=0, B=1, A=2, D=1, C=1

A, D, ja C muodostavat vielä vektorit, lähettävät ne mutta ne eivät enää aiheuta muutoksia.

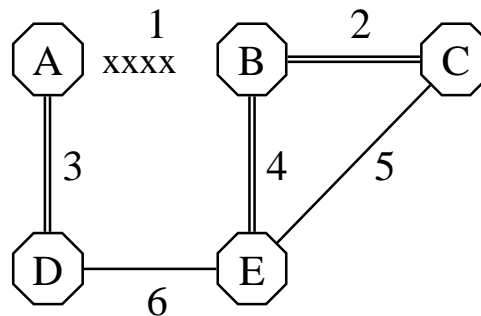
Linkin katkeaminen käynnistää päivityskierroksen

A:sta	Linkki	Kustannus
A	-	0
B	1	inf.
D	3	1
C	1	inf.
E	1	inf.

B:sta	Linkki	Kustannus
B	-	0
A	1	inf
D	1	inf
C	2	1
E	4	1

A=0,B=inf,D=1,C=inf,E=inf

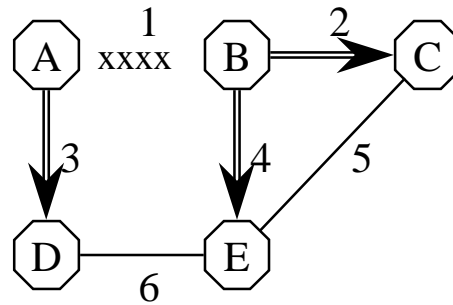
B=0,A=inf,D=inf,C=1,E=1



D, E ja C päivittävät reititystaulunsa

B=0,A=inf,D=inf,C=1,E=1

A=0,B=inf,D=1,C=inf,E=inf



A=1,B=inf,D=2,C=inf,E=inf

D:sta	Linkki	Kustannus
D	-	0
A	3	1
B	3	inf
E	6	1
C	6	2

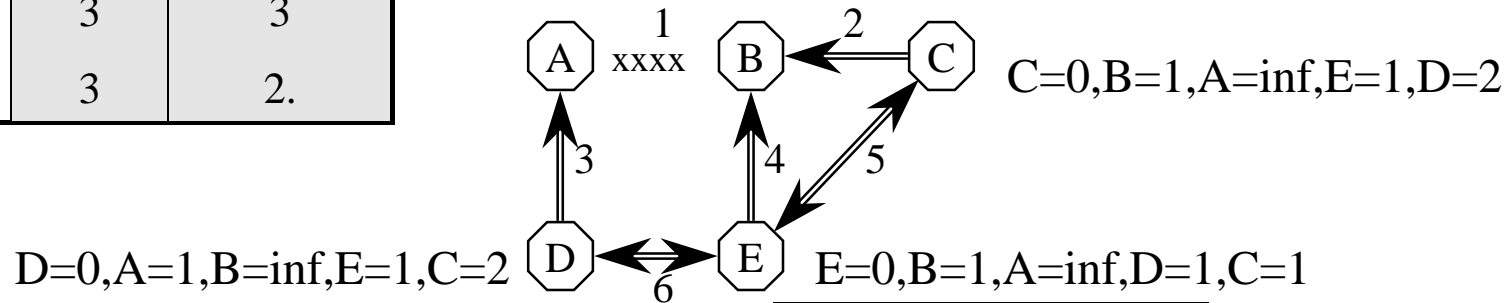
C:sta	Linkki	Kustannus
C	-	0
B	2	1
A	2	inf
E	5	1
D	5	2

E:sta	Linkki	Kustannus
E	-	0
B	4	1
A	4	inf
D	6	1
C	5	1

D, C, E generoivat etäisyysvektorinsa...

A:sta	Linkki	Kustannus
A	-	0
B	1	inf.
D	3	1
C	3	3
E	3	2.

B:sta	Linkki	Kustannus
B	-	0
A	1	inf
D	4	2
C	2	1
E	4	1



D:sta	Linkki	Kustannus
D	-	0
A	3	1
B	6	2
E	6	1
C	6	2

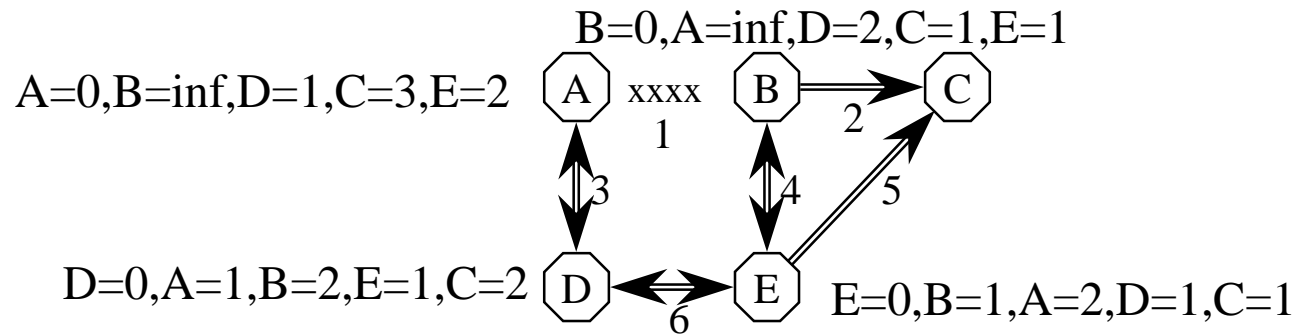
E:sta	Linkki	Kustannus
E	-	0
B	4	1
A	6	2
D	6	1
C	5	1

A, B, D, E generoivat etäisyysvektorinsa

A:sta	Linkki	Kustannus
A	-	0
B	3	3
D	3	1
C	3	3
E	3	2

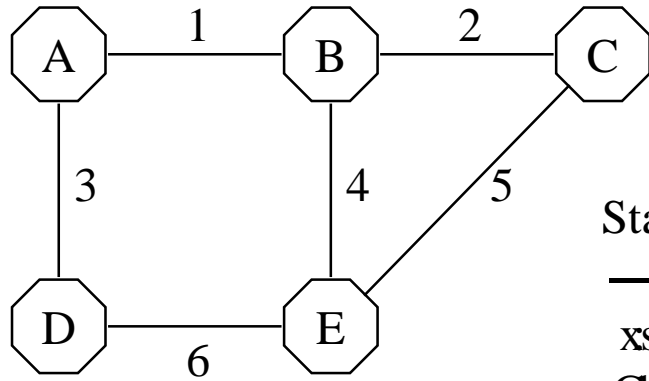
B:sta	Linkki	Kustannus
B	-	0
A	4	3
D	4	2
C	2	1
E	4	1

C:sta	Linkki	Kustannus
C	-	0
B	2	1
A	5	3
E	5	1
D	5	2



Tuloksena on, että kaikki voivat taas kommunikoida kaikkien kanssa.

EV-protokolla voi synnyttää transientin reittisilmukan



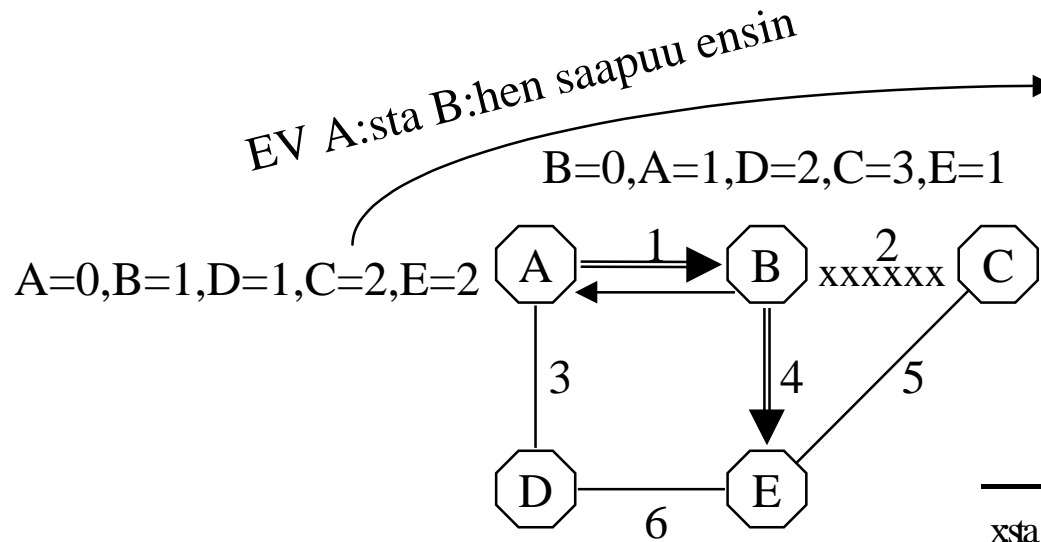
Oletetaan, että linkin 5 kustannus on 8.

Stabiili lähtötila reiteillä C:hen olisi:

Chn xsta	Linkki xstä	Kustannus
A-C	1	2
B-C	2	1
G-C	-	0
D-C	3	3
E-C	4	2

Keskitytään vain kunkin reitin ensimmäiseen linkkiin

Linkki 2 vioittuu...



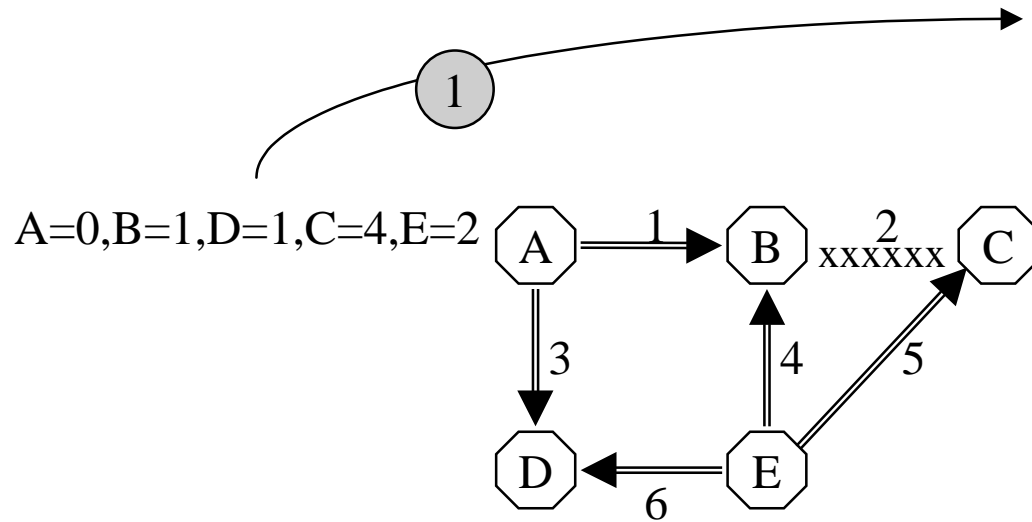
xsta	Linkki	Kustaus
Chen	xstä	
A>C	1	2
B>C	2	inf
G>C	-	0
D>C	3	3
E>C	4	2

välitila

Kaikki viestit C:hen ohjataan B:lle, joka lähettää ne A:lle, joka lähettää ne B:lle... kunnes TTL=0. (Bouncing effect - pallottelu)

xsta	Linkki	Kustaus
Chen	xstä	
A>C	1	4
B>C	1	3
G>C	-	0
D>C	3	3
E>C	4	4

A ja E lähettävät etäisyysvektorinsa



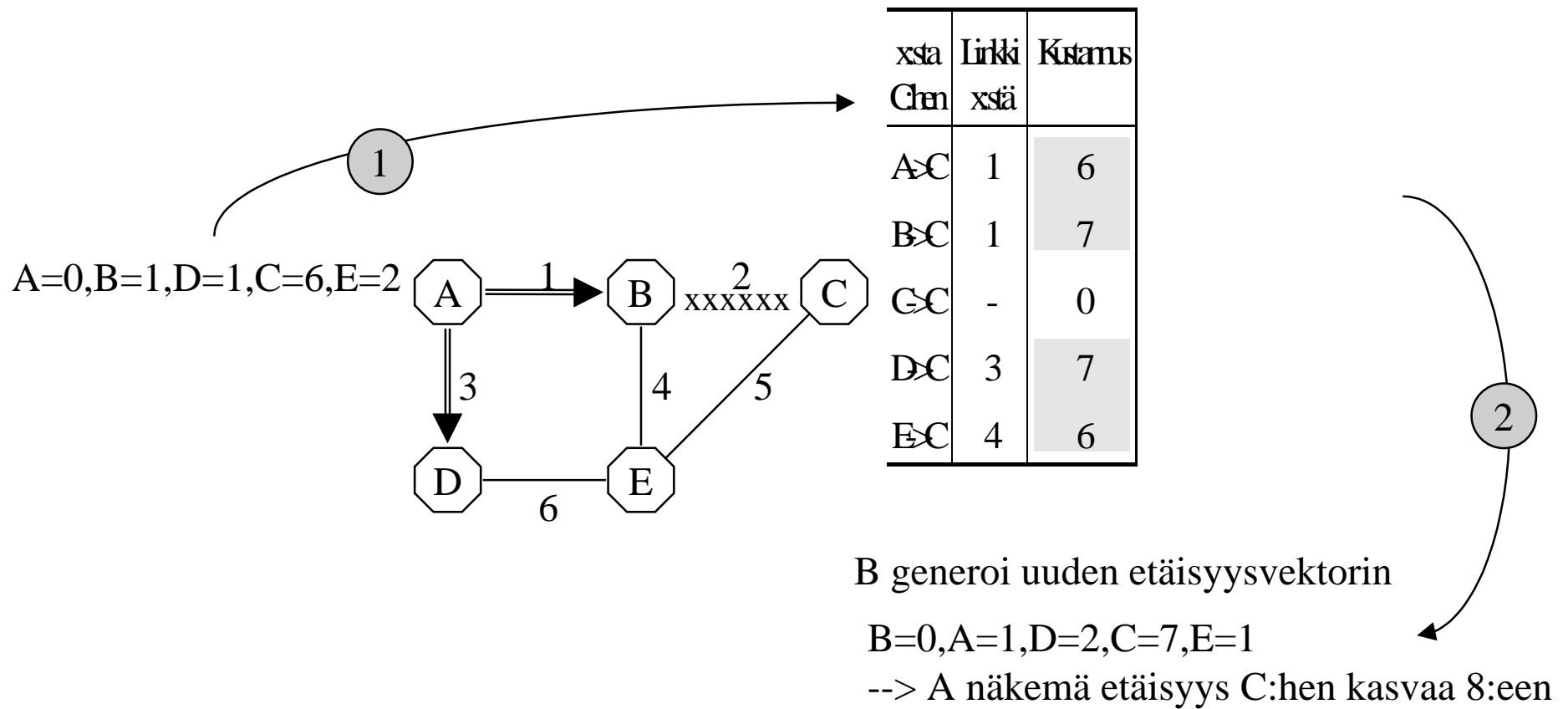
xstä	Linkki	Kustannus
C:n	xstä	
A	C	1 4
B	C	1 5
G	C	- 0
D	C	3 5
E	C	4 4

B generoi uuden etäisyysvektorin

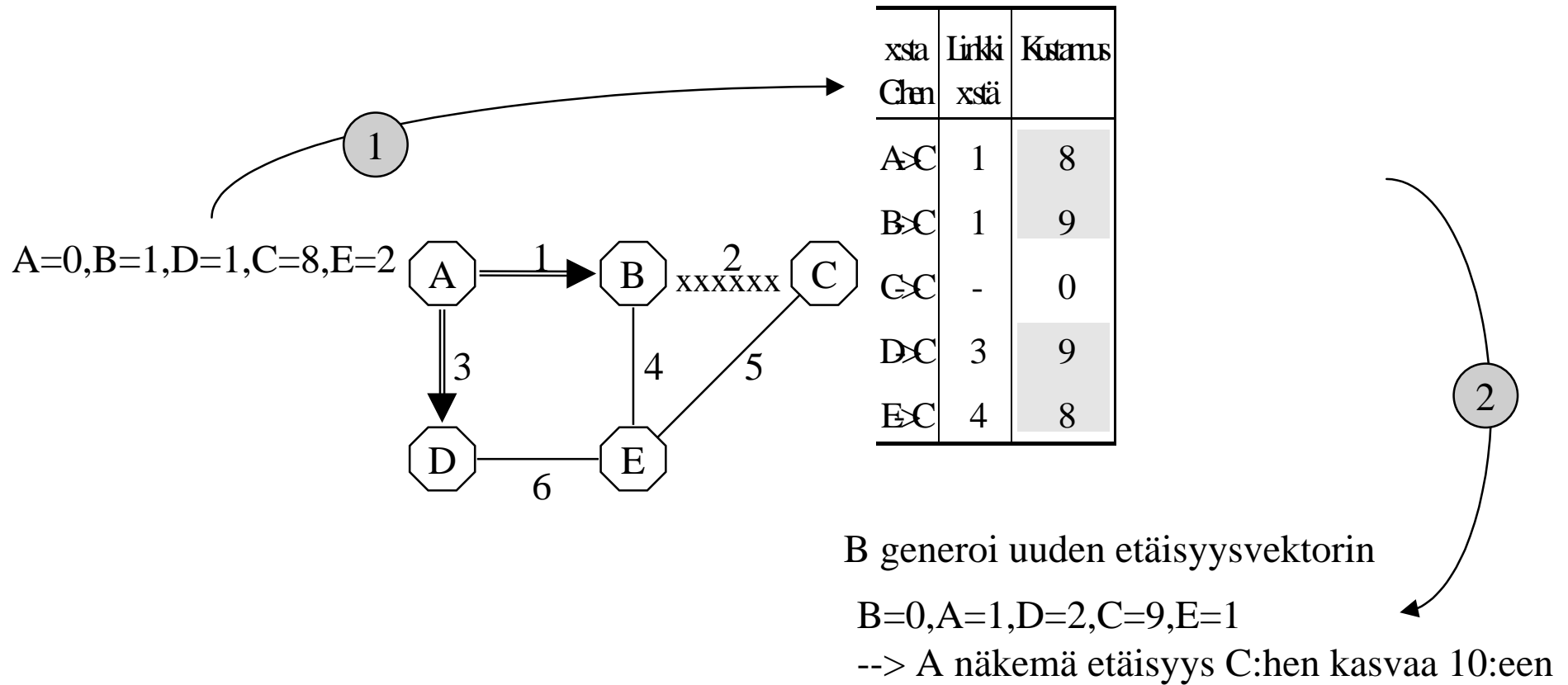
B=0, A=1, D=2, C=5, E=1

--> A näkemä etäisyys C:hen kasvaa 6:een

A lähettää uuden etäisyysvektorin

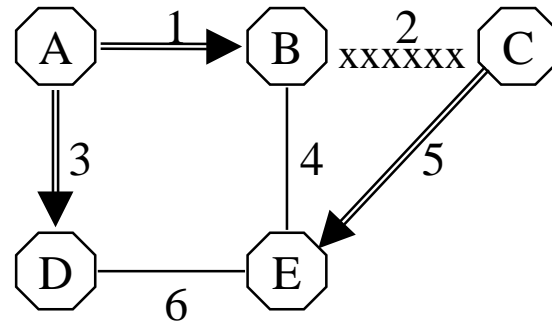


A lähettää uuden etäisyysvektorin



A lähettää uuden etäisyysvektorin

A=0,B=1,D=1,C=10,E=2



xstä	Lirki	Kustannus
Chen	xstä	
A→C	1	10
B→C	1	11
G→C	-	0
D→C	3	11
E→C	5	8

B generoi uuden etäisyysvektorin

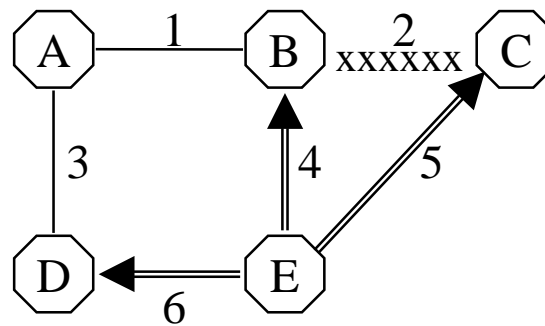
B=0,A=1,D=2,C=11,E=1

E generoi uuden etäisyysvektorin

E=0,B=1,A=2,D=1,C=8

E lähettää uuden etäisyysvektorin

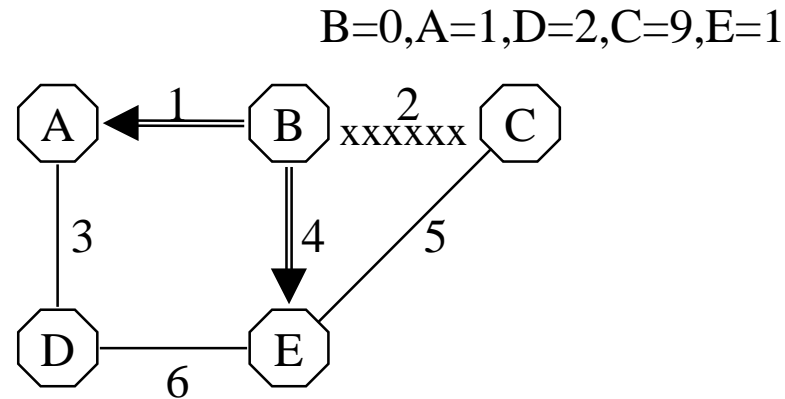
xsta	Linkki	Kustannus
Chen	xstä	
A C	1	10
B C	4	9
G C	-	0
D C	6	9
E C	5	8



E=0, B=1, A=2, D=1, C=8

B lähettää EV:n, mutta taulut on jo OK

xstä	Linkki	Kustaus
Chn	xstä	
A-C	1	10
B-C	4	9
G-C	-	0
D-C	6	9
E-C	5	8



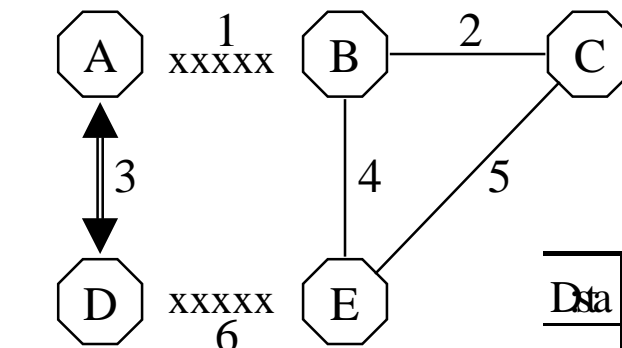
- Jokainen päivityskierros korjasi kustannuksia 2:lla
- Prosessi etenee satunnaisessa järjestyksessä, koska siinä on aitoa rinnakkaisuutta.
- Prosessin aikana verkon tila on huono, EV-protokollasanomia voi hukkaa pallottelevien käyttäjäviestien aiheuttamassa ruuhkassa.

Irralliset saarekkeet aiheuttavat laskemisen äärettömään

Kun linkin 1 vikaantumisesta on toivuttu, myös linkki 6 vikaantuu. Kaikki linkkikustannukset=1.

2 A=0,B=3,D=1,C=3,E=2

Dst	Linkki	Kustaus
D	-	0
A	3	1
B	6	inf
E	6	inf
C	6	inf



1 D ei ehdi lähettää EV:tään

Dst	Linkki	Kustaus
D	-	0
A	3	1
B	3	4
E	3	3
C	3	4

Tuloksena on silmukka, kustannukset kasvavat 2:lla joka kierroksella. On sovittava, että max etäisyyttä suurempi kustannus = inf.

Silmukoita voidaan vähentää karsimalla etäisyysvektoreista tietoa ja generoimalla EV:t heti taulun muututtua

Karsintasääntö = Jos solmu A lähettää solmulle X solmun B kautta, B:n ei kannata yrittää tavoitella X:ää A:n kautta

⇒

A:n ei kannata mainostaa B:lle lyhyttä etäisyyttään X:ään.

Toteutusvariaatiot: 1. A ei mainosta etäisyyttään X:ään B:lle lainkaan

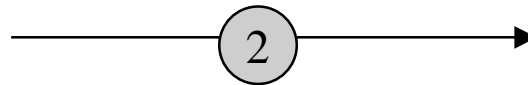
⇒ edellisen esimerkin silmukkaa ei synny

2. A mainostaa B:lle: $X = \text{inf}$. (“split horizon with poisonous reverse”)

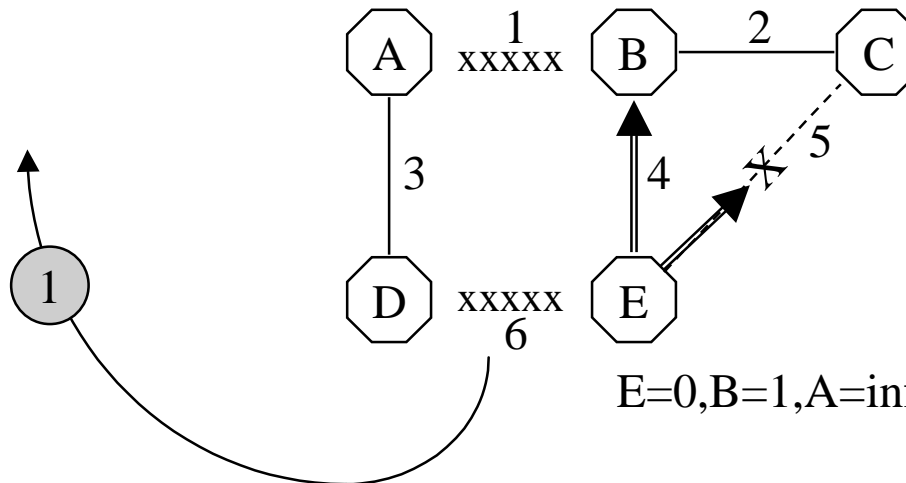
⇒ kahden solmun silmukat kuolee heti.

Kolmen solmun silmukat ovat silti mahdollisia

xsta	Linkki	Kustannus
Chen	xstä	
B-D	4	2
G-D	5	2
E-D	6	inf



xsta	Linkki	Kustannus
Chen	xstä	
B-D	4	inf
G-D	5	2
E-D	6	inf

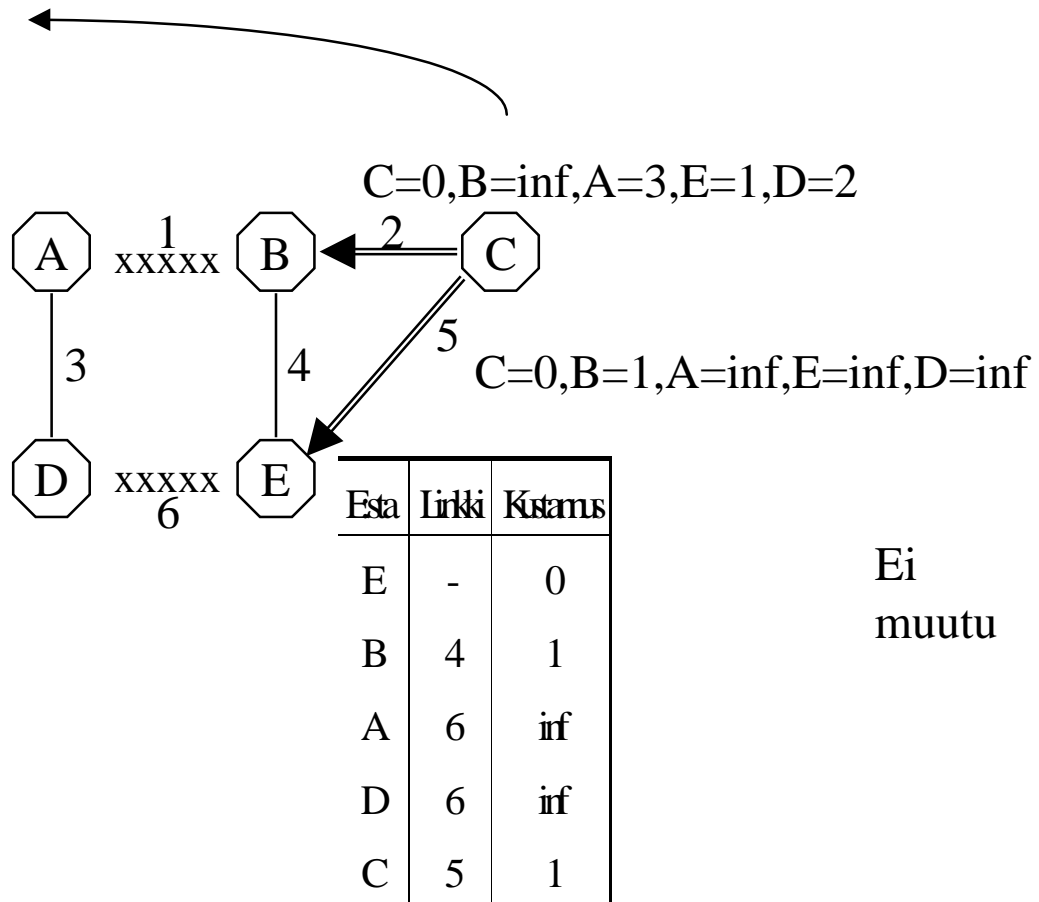


C:hen lähetetty vektori hukataan

$E=0, B=1, A=\text{inf}, D=\text{inf}, C=1$

On C:n aika mainostaa myrkytetyin vektorein

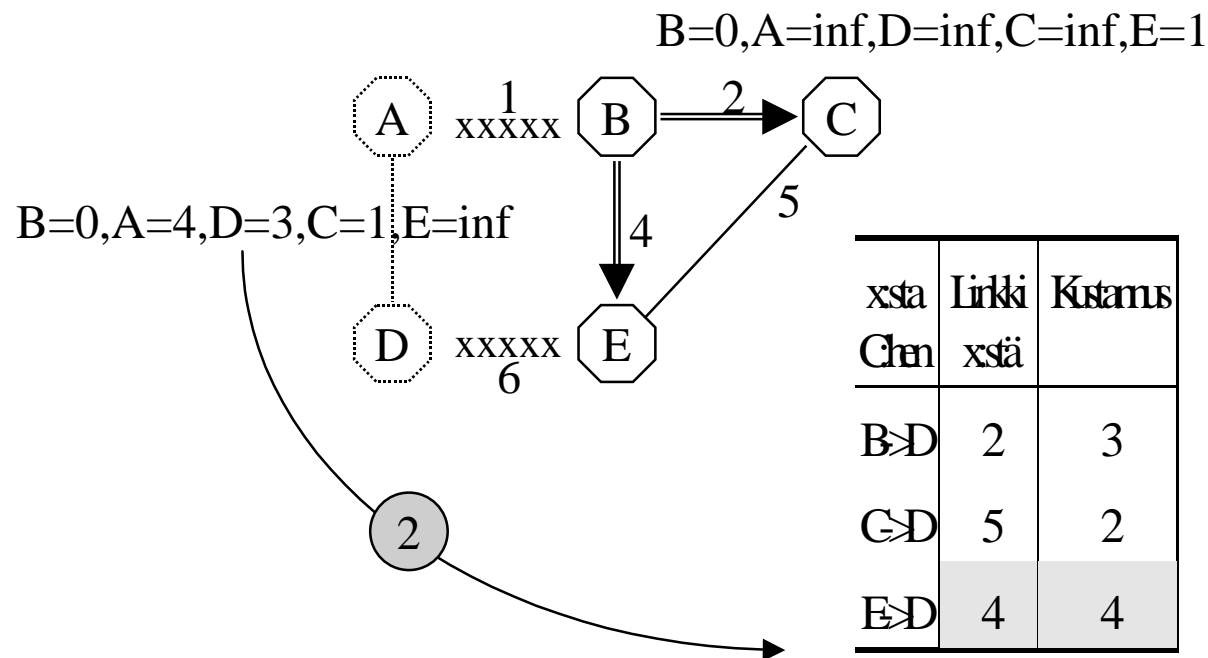
Bsta	Linkki	Kustaus
B	-	0
A	2	4
D	2	3
C	2	1
E	4	1



Ei
muutu

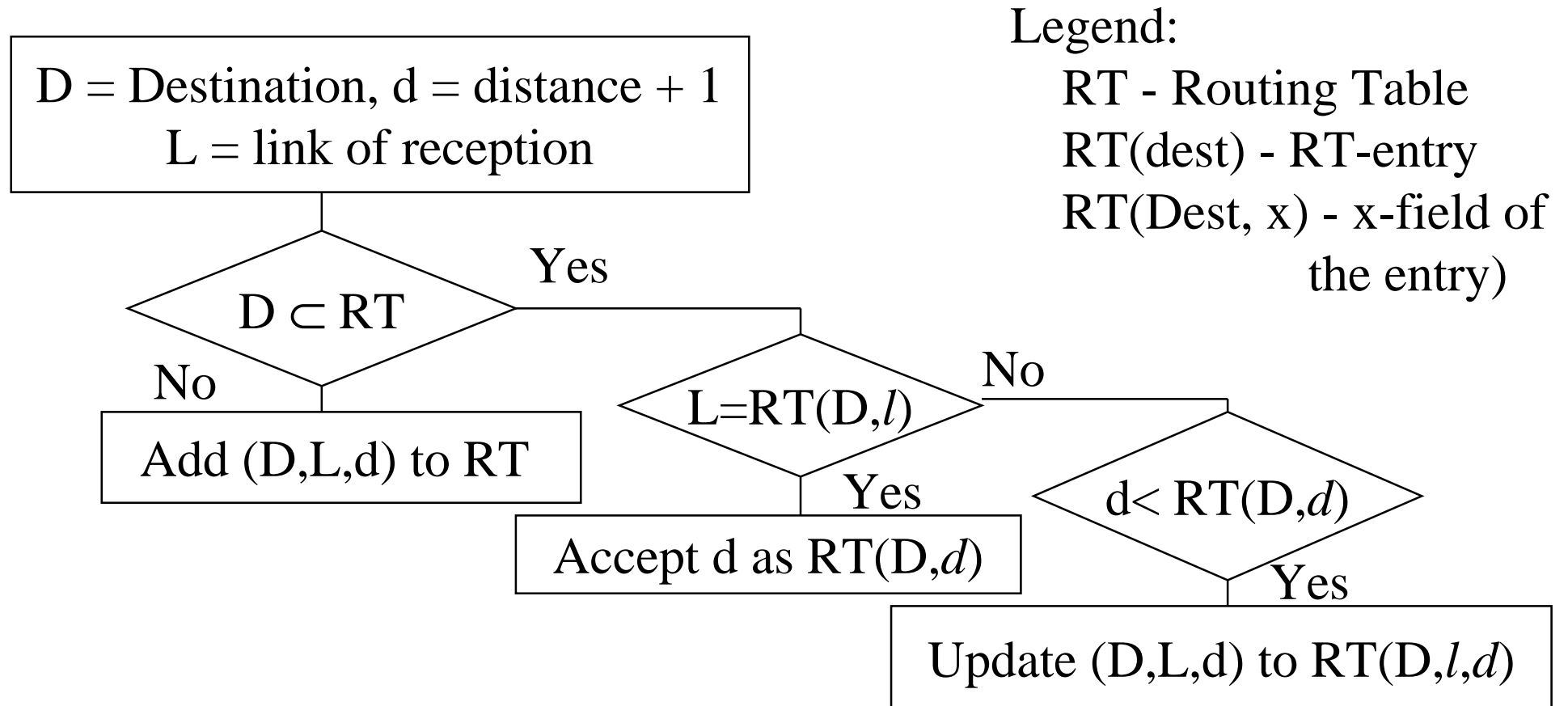
Kolmen solmun silmukka on valmis...

- 1 B muodostaa myrkytetyt vektorit



- 3 Reitit D:hen eivät enää muutu, lasketaan äärettömään, mikä viimein purkaa silmukan: linkillä 5 kerrotaan kustannus 4, C:n käsitys etäisyydestä D:hen alkaa kasvaa ...

Received Distance Vectors are processed so



Note: this is simplified, shows only the principle!

Milloin EV-protokollan kannattaa lähettää

- lähetys hetki on kompromissi
 - + tiedon välitön päivittäminen
 - + pakettien katoamisesta toipuminen
 - + naapureiden monitorointitarve
 - kaikkien muutosten lähettäminen yhtä aikaa
 - protokollan aiheuttama liikennekuorma
- + nopeasti
- hidastele

Tapahduman laukaisemat päivitykset parantavat RIP:n toimintaa

- Reititystaulukon riveillä on virkistys- ja vanhenemisajastin
- RIP lähettää aina virkistysajastimen lauetessa ja heti kun muutos havaitaan
- Laukaistu päivitys nopeuttaa laskua äärettömään ja vähentää silmukoiden syntyä

EV-protokollat perustuvat Bellman-Ford algoritmiin

- Keskitetty versio:
1. Olkoon N solmujen lukumäärä ja M linkkien lukumäärä.
 2. L on M -rivinen linkkitaulukko, $L[l].m$ - linkin mitta,
 $L[l].s$ - linkin alkupää
 $L[l].d$ - linkin kohde
 3. D on $N \times N$ matriisi, jossa $D[i,j]$ on etäisyys i :stä j :hin
 4. H on $N \times N$ matriisi, jossa $H[i,j]$ on linkki, jolla i lähettää j :lle

D	1	..	i	..	N
1					
:					
j	etäisyys i:stä j:hin				
:					
N					

Linkkitaulussa on molemmat suunnat erikseen!
Sarake vastaa solmun etäisyysvektoria!

Bellman-Ford algoritmi on

1. Alustetaan: Jos $i=j$ silloin $D[i,j] = 0$, muuten $D[i,j] = \text{inf}$.
Alustetaan $\forall H[i,j] = -1$.
2. $\forall l$ ja \forall kohteille k aseta $i = L[l].s$, $j = L[l].d$ ja laske $d = L[l].m + D[j,k]$
3. Jos $d < D[i,k]$, aseta $D[i,k] = d$; $H[i,k] = l$.
4. Jos edes yksi $D[i,k]$ muuttui, toista kohta 2, muutoin algoritmi päättyy.

Alustetut etäisyys- ja linkkimatriisit ovat

$$\begin{array}{c}
 D \quad 1 \quad .. \quad .. \quad N \\
 \left| \begin{array}{ccccc}
 1 & \mathbf{0} & \infty & \infty & \infty & \infty \\
 : & \infty & \mathbf{0} & \infty & \infty & \infty \\
 & \infty & \infty & \mathbf{0} & \infty & \infty \\
 : & \infty & \infty & \infty & \mathbf{0} & \infty \\
 N & \infty & \infty & \infty & \infty & \mathbf{0}
 \end{array} \right|
 \end{array}
 \qquad
 \begin{array}{c}
 H \quad 1 \quad .. \quad .. \quad N \\
 \left| \begin{array}{ccccc}
 1 & -1 & -1 & -1 & -1 & -1 \\
 : & -1 & \mathbf{-1} & -1 & -1 & -1 \\
 & -1 & -1 & \mathbf{-1} & -1 & -1 \\
 : & -1 & -1 & -1 & \mathbf{-1} & -1 \\
 N & -1 & -1 & -1 & -1 & \mathbf{-1}
 \end{array} \right|
 \end{array}$$

Huom: Linkkivektori sisältää molemmat suunnat erikseen.

Aluksi D-matriisissa täyttyvät yhden linkin päässä olevien solmujen väliset etäisyydet, seuraavaksi kahden linkin päässä olevat, jne.

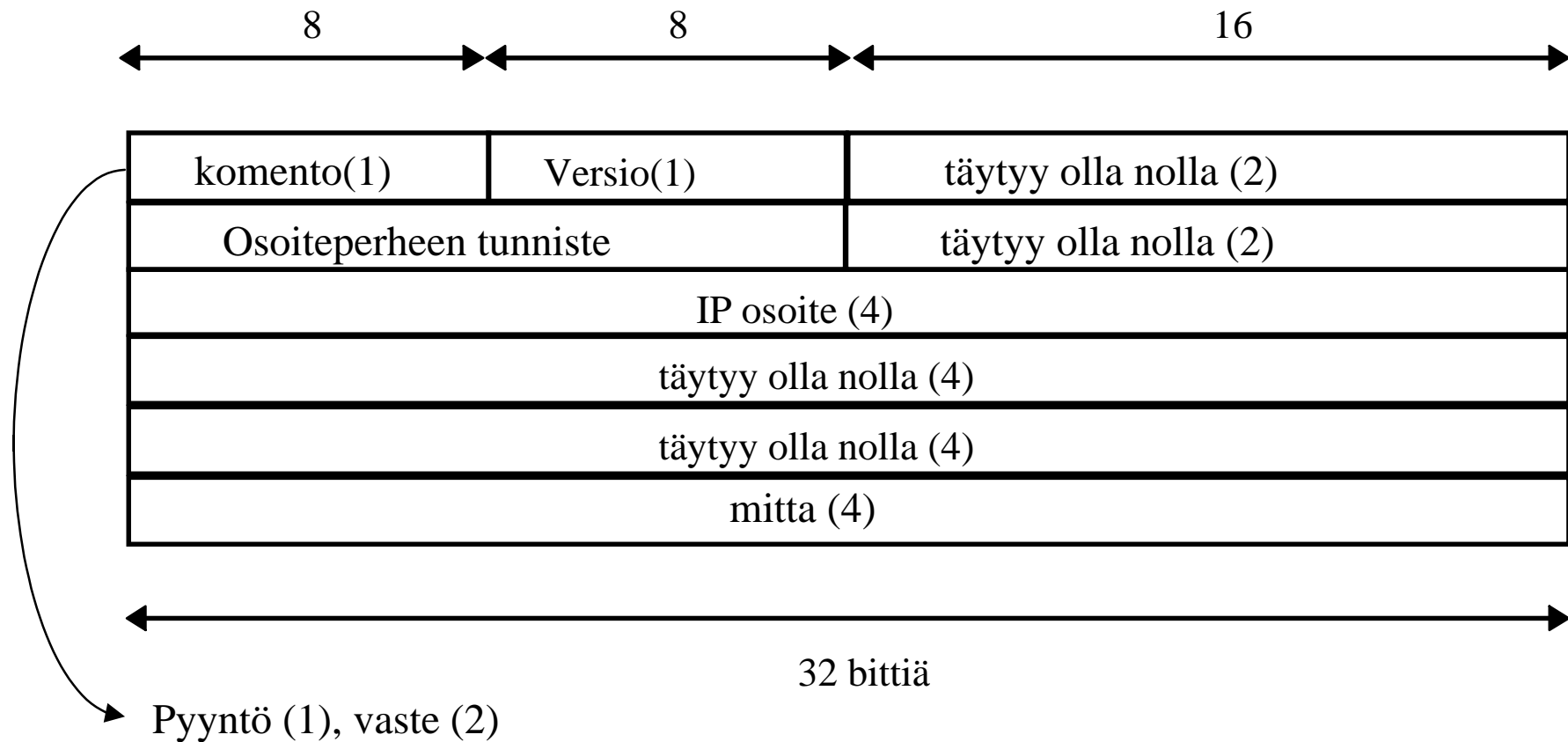
RIP protokollan peruspiirteitä ovat

- RIP versio 1 - RFC-1058
- RIP:iä käytetään autonomisen järjestelmän sisällä
- Reititystaulun rivi esittää isäntäkonetta, verkkoa tai aliverkkoa (subnet)
 - <netid,subnetid,host> esittää isäntäkonetta
 - <netid,subnetid,0> esittää aliverkkoa
 - <netid,0,0> esittää verkkoa
 - <0.0.0.0> esittää reittiä ulos autonomisesta järjestelmästä
- Etäisyys = hop count = polun peräkkäisten linkkien lukumäärä, 16=inf.

RIP:n piirteitä ovat myös

- RIP toimii sekä jaetun median (Ethernet) että yksipisteverkoissa (pt-to-pt)
- RIP toimii UDP:n ja IP:n päällä.
- RIP lähettää 30 s välein, yli 180s vanha reititysriivi --> etäisyys asetetaan inf:iin
- Ajastimen käynnistämiä lähetyksiä täytyy satunnaistaa, jotta RIP liikenne tasoittuisi.
- RIP käyttää myrkytettyjä vektoreita
- Lähetykset naapuri-aliverkkoon aggregoidaan

RIP sanomaformaatti on



RIP reititystaulun rivi sisältää

- Kohteen osoite
- Etäisyys kohteeseen
- Seuraavan reitittimen osoite
- “Äsken” päivitetty -lippu
- Useita ajastimia (virkistys- , vanhenemis- ...)

RIP versio 2 - sanomat

RFC-1388 (1387,1389)



komento(1)	Versio(1)	Reititysalue (2)
Osoiteperheen tunniste		Reittilippu(2)
IP osoite (4)		
Aliverkon maski (4)		
Seuraava linkki (4)		
mitta (4)		

Monta päällekkäistä AS:ta samalla "langalla"

--> autentikointi

Ulkoiset reitit merkataan

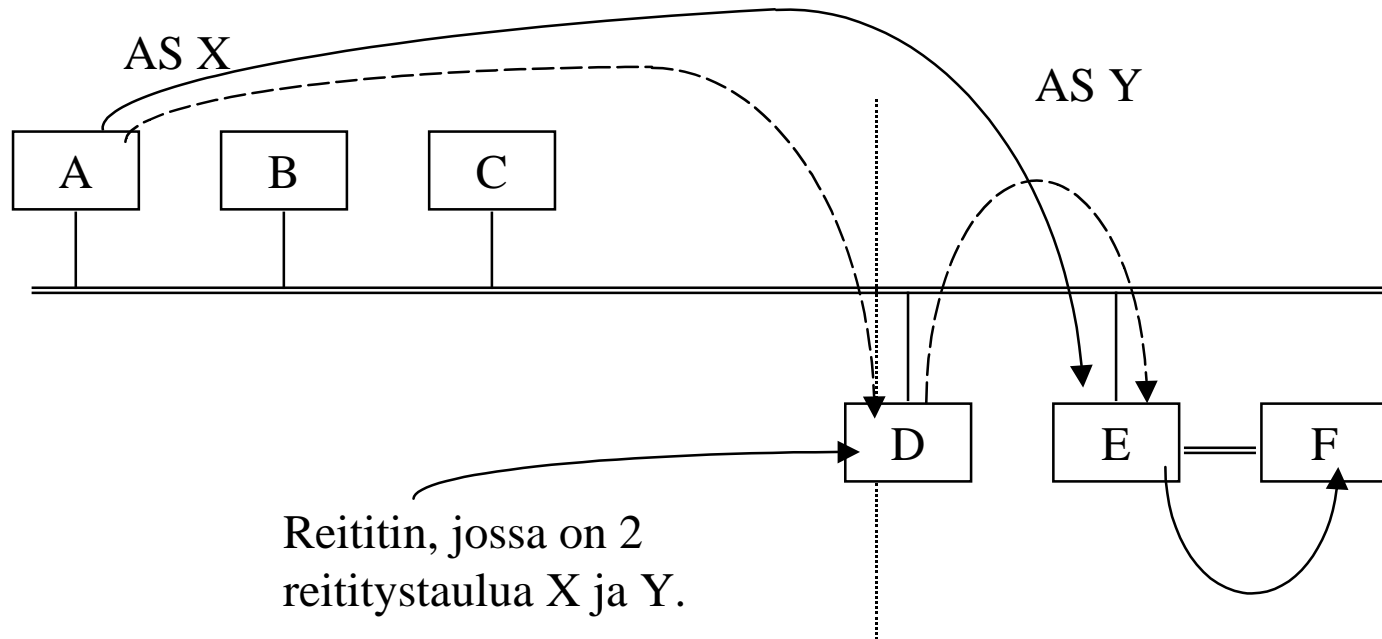
CIDR

*RIP-2 on rajoitetusti yhteensopiva päivitys
(RIP-1 kone ymmärtää RIP-2 konetta osittain.)*

Reititys aliverkosta toiseen

- RIP-1:n aliverkkomaski ei ole tunnettu aliverkon ulkopuolella, vaan ulos kerrotaan ainostaan netid -->
- Isäntäkonetta ja aliverkkoa ei voi erottaa toisistaan -->
- Kaikki aliverkot pitää yhdistää kaikkiin ja ulkoa reititettävä verkon lähimpään reitittemeen aliverkosta riippumatta
- RIP-2 korjaa tilannetta kertomalla ulos aliverkon ja aliverkkomaskin

Reititysalue ja seuraava linkki



Seuraava linkki \implies D mainostaa X:ssä, että etäisyys F:ään on f ja seuraava linkki on E!

Huomioita RIP:stä

- Reitittimillä on spontaani taipumus synkronoida lähetyshetkensä. Tämä lisää virheiden todennäköisyyttä verkossa. Siksi lähetyshetket satunnaistetaan 15s ... 45s välille.
- Syy:
lähetysväli = vakio+sanoman pakkausaika+yhtä aikaa tulleiden sanomien käsittelyaika.
- Kun RIP:ä käytetään ISDN linkin yli -->uusi puhelu/30s --> kallista.
- Hidas alusverkko --> jonojen pituudelle rajoituksia. RIP lähettää sanomansa (25 riviä/sanoma) putkeen --> RIP sanomia voi kadota.
- Korjausehdotus perustuu lähetyksen kuittausmoodiin, jossa periodisia lähetyksiä ei tapahdu
--> RIP sanomien puuttuessa oletetaan, että naapuri on edelleen tavoitettavissa
--> Tieto kaikista vaihtoehtoisista reiteistä talletetaan.