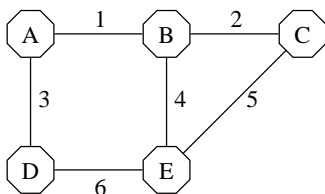


# Routing Information Protocol

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*

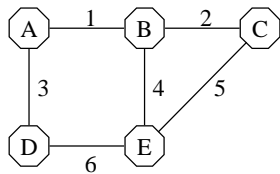
Alkutila: Noodit tuntevat omat osoitteensa ja liitännänsä, mutta ei muuta

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



Tarkastellaan vastaottoa solmussa B

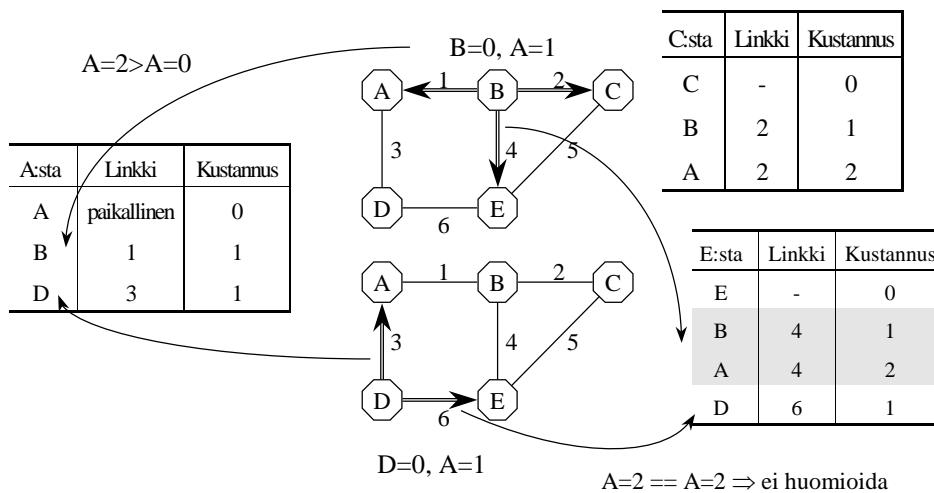
| B:sta solmuun | Linkki      | Kustannus |
|---------------|-------------|-----------|
| B             | paikallinen | 0         |

$A=0$  →

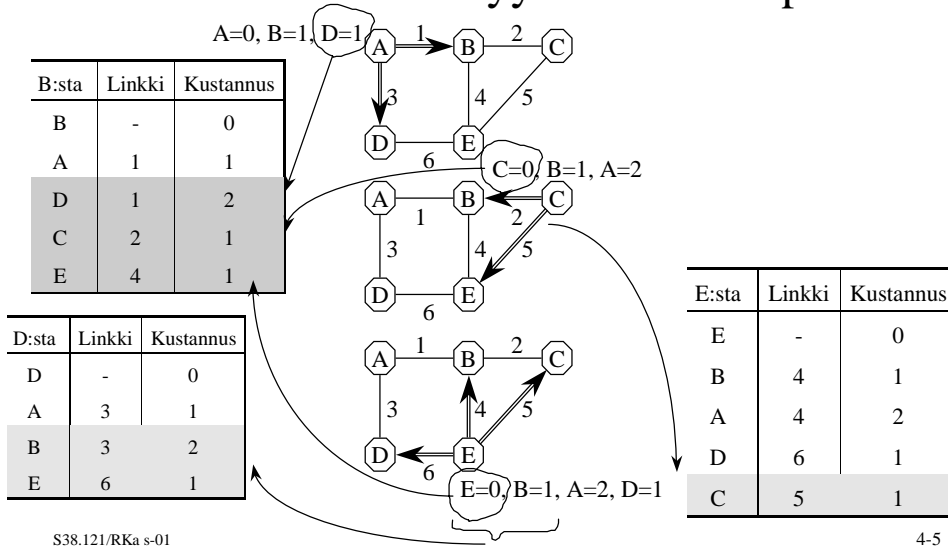
1. B lisää heti etäisyysvektoriin + 1 --> A=1 ja
2. B etsii tulosta omasta taulusta, ei löydy
3. B lisää saamansa tiedon reititystauluunsa, tulos on

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

## B muodostaa oman vektorinsa ja lähettää sen kaikille naapureille



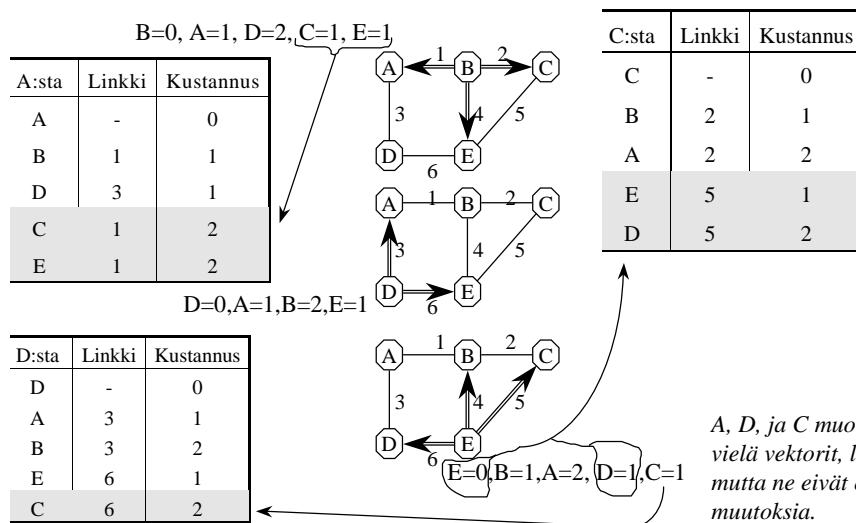
## Solmut, joiden reititystaulut muuttuivat lähettävät uudet etäisyysvektorit naapureille



S38.121/RKa s-01

4-5

## Muutokset lähetetään taas ...



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

S38.121/RKa s-01

4-6

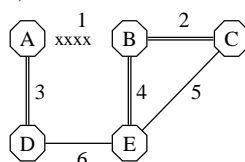
## Linkin katkeaminen käynnistää päivityskierroksen

| 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



S38.121/RKa s-01

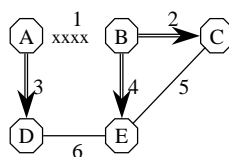
4-7

## D, E ja C päivittävät reititystaulunsa

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

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

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         |

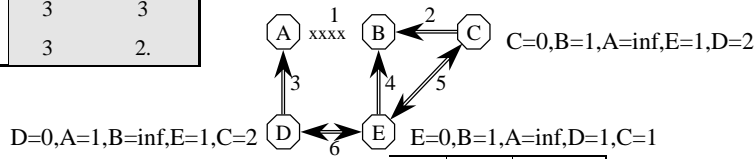
S38.121/RKa s-01

4-8

## 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         |

S38.121/RKa s-01

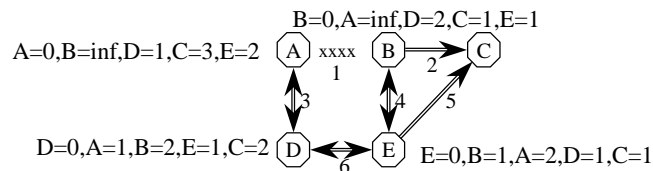
4-9

## 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         |

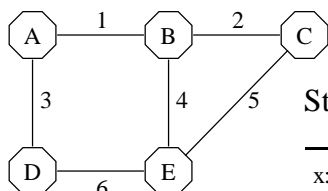


*Tuloksena on, että kaikki voivat taas kommunikoida kaikkien kanssa.*

S38.121/RKa s-01

4-10

## EV-protokolla voi synnyttää transientin reittisilmukan



Oletetaan, että linkin 5 kustannus on 8.

Stabiili lähtötila reiteillä C:hen olisi:

| x:sta<br>C:hen | Linkki<br>x:stä | Kustannus |
|----------------|-----------------|-----------|
| A->C           | 1               | 2         |
| B->C           | 2               | 1         |
| C->C           | -               | 0         |
| D->C           | 3               | 3         |
| E->C           | 4               | 2         |

*Keskitytään vain kunkin reitin ensimmäiseen linkkiin*

## Linkki 2 vioittuu...

EV A:sta B:hen saapuu ensin

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

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

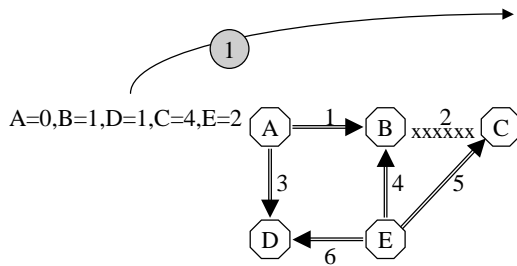
| x:sta<br>C:hen | Linkki<br>x:stä | Kustannus |
|----------------|-----------------|-----------|
| A->C           | 1               | 2         |
| B->C           | 2               | inf       |
| C->C           | -               | 0         |
| D->C           | 3               | 3         |
| E->C           | 4               | 2         |

välitila

| x:sta<br>C:hen | Linkki<br>x:stä | Kustannus |
|----------------|-----------------|-----------|
| A->C           | 1               | 4         |
| B->C           | 1               | 3         |
| C->C           | -               | 0         |
| D->C           | 3               | 3         |
| E->C           | 4               | 4         |

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

## A ja E lähettävät etäisyysvektorinsa



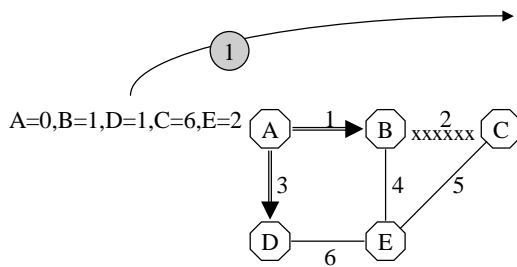
| x:sta<br>C:hen | Linkki<br>x:stä | Kustannus |
|----------------|-----------------|-----------|
| A->C           | 1               | 4         |
| B->C           | 1               | 5         |
| C->C           | -               | 0         |
| D->C           | 3               | 5         |
| E->C           | 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



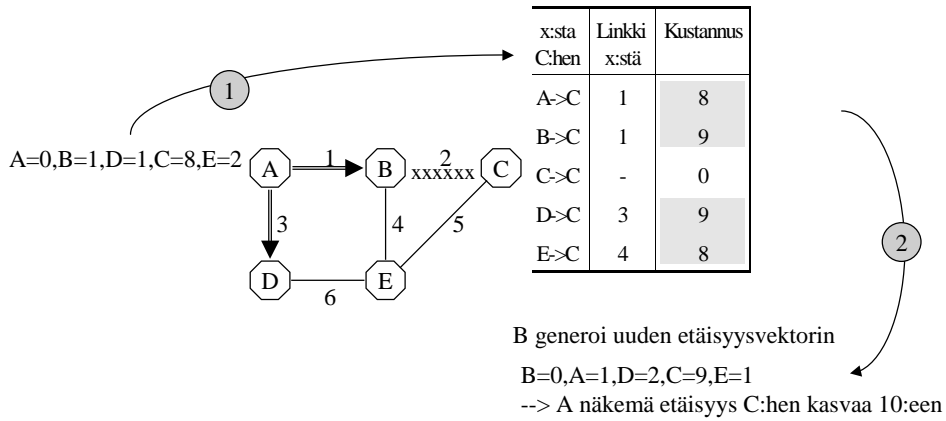
| x:sta<br>C:hen | Linkki<br>x:stä | Kustannus |
|----------------|-----------------|-----------|
| A->C           | 1               | 6         |
| B->C           | 1               | 7         |
| C->C           | -               | 0         |
| D->C           | 3               | 7         |
| E->C           | 4               | 6         |

B generoi uuden etäisyysvektorin

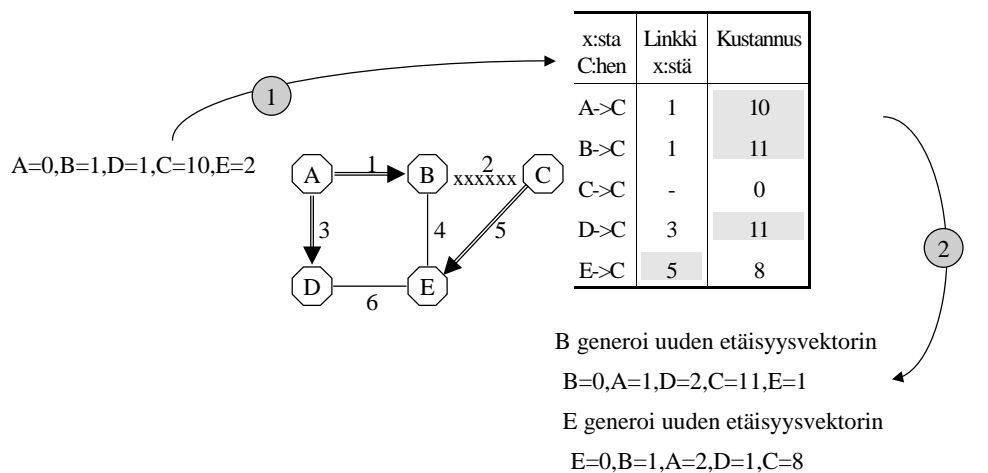
B=0, A=1, D=2, C=7, E=1

--> A näkemä etäisyys C:hen kasvaa 8:een

## A lähettää uuden etäisyysvektorin



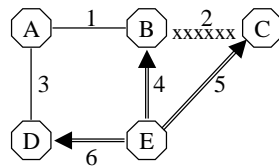
## A lähettää uuden etäisyysvektorin





## E lähettää uuden etäisyysvektorin

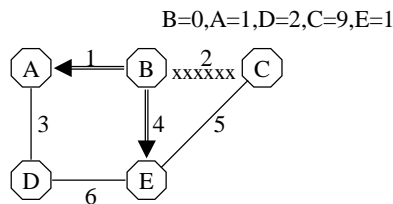
| x:sta<br>Chen | Linkki<br>x:stä | Kustannus |
|---------------|-----------------|-----------|
| A->C          | 1               | 10        |
| B->C          | 4               | 9         |
| C->C          | -               | 0         |
| D->C          | 6               | 9         |
| E->C          | 5               | 8         |



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

## B lähettää EV:n, mutta taulut on jo OK

| x:sta<br>Chen | Linkki<br>x:stä | Kustannus |
|---------------|-----------------|-----------|
| A->C          | 1               | 10        |
| B->C          | 4               | 9         |
| C->C          | -               | 0         |
| D->C          | 6               | 9         |
| E->C          | 5               | 8         |



B=0, A=1, D=2, C=9, E=1

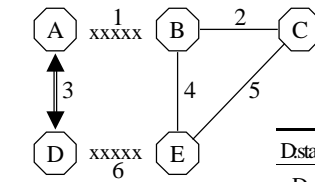
- Jokainen päivityskierros korjasi kustannuksia 2:lla
- Prosessi etenee satunnaisessa järjestyksessä, koska siinä on aitoa rinnakkaisuutta.
- Prosessin aikana verkon tila on huono, EV-protokollasanomia voi hukkua pallottelevien käyttäjäviestien aiheuttamassa ruuhkassa.

## 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

| Dsta | Linkki | Kustannus |
|------|--------|-----------|
| D    | -      | 0         |
| A    | 3      | 1         |
| B    | 6      | inf       |
| E    | 6      | inf       |
| C    | 6      | inf       |



1  
D ei ehdi lähettää EV:tään

| Dsta | Linkki | Kustannus |
|------|--------|-----------|
| 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=inf. ("split horizon with poisonous reverse")

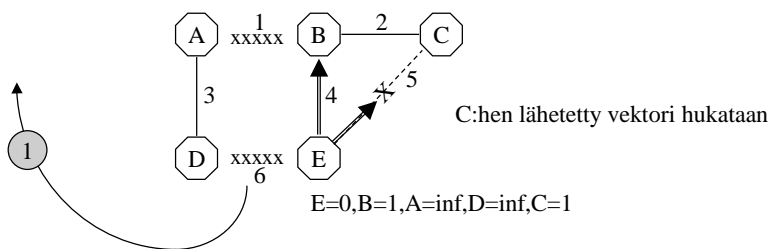
⇒ kahden solmun silmukat kuolee heti.

## Kolmen solmun silmukat ovat silti mahdollisia

| x:sta | Linkki | Kustannus |
|-------|--------|-----------|
| Chen  | x:stä  |           |
| B->D  | 4      | 2         |
| C->D  | 5      | 2         |
| E->D  | 6      | inf       |

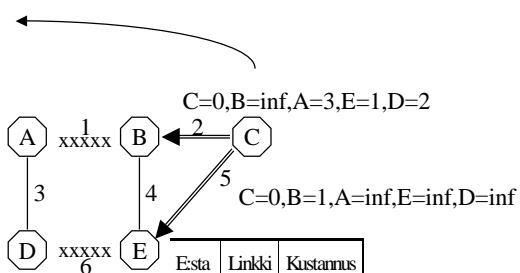


| x:sta | Linkki | Kustannus |
|-------|--------|-----------|
| Chen  | x:stä  |           |
| B->D  | 4      | inf       |
| C->D  | 5      | 2         |
| E->D  | 6      | inf       |



## On C:n aika mainostaa myrkytetyin vektorein

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

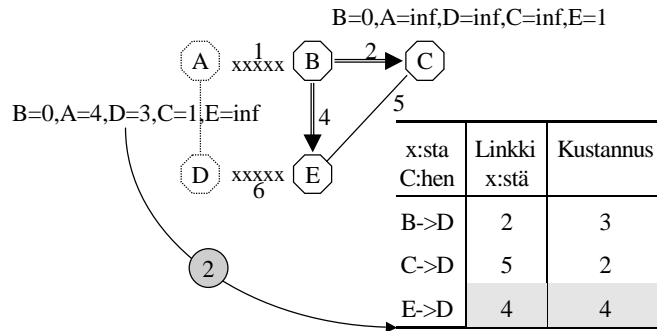


| E:sta | Linkki | Kustannus |
|-------|--------|-----------|
| E     | -      | 0         |
| B     | 4      | 1         |
| A     | 6      | inf       |
| D     | 6      | inf       |
| C     | 5      | 1         |

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

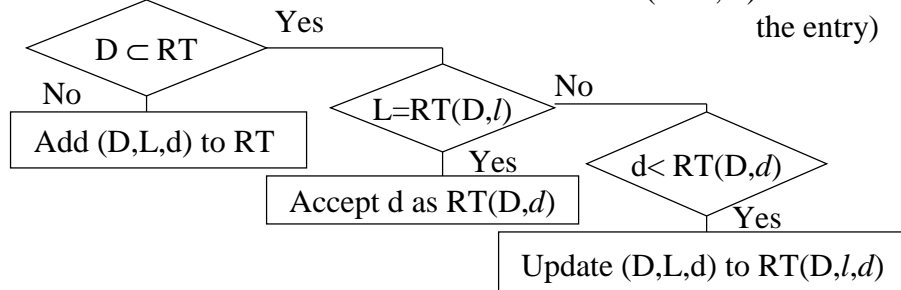
D = Destination, d = distance + 1  
L = link of reception

Legend:

RT - Routing Table

RT(dest) - RT-entry

RT(Dest, x) - x-field of the entry



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

## Tapahtuman 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    |    |     |
| $N$ |   |    | j:hin    |    |     |

Linkkitaulussa on molemmat suunnat erikseen!  
Sarake vastaa solmun etäisyysvektoria!

## Alustetut etäisyys- ja linkkimatriisit ovat

|     |          |          |          |          |
|-----|----------|----------|----------|----------|
| $D$ | 1        | ..       | ..       | $N$      |
| 1   | <b>0</b> | $\infty$ | $\infty$ | $\infty$ |
| :   | $\infty$ | <b>0</b> | $\infty$ | $\infty$ |
| $j$ | $\infty$ | $\infty$ | <b>0</b> | $\infty$ |
| :   | $\infty$ | $\infty$ | $\infty$ | <b>0</b> |
| $N$ | $\infty$ | $\infty$ | $\infty$ | <b>0</b> |

|     |    |           |           |           |
|-----|----|-----------|-----------|-----------|
| $H$ | 1  | ..        | ..        | $N$       |
| 1   | -1 | -1        | -1        | -1        |
| :   | -1 | <b>-1</b> | -1        | -1        |
| $j$ | -1 | -1        | <b>-1</b> | -1        |
| :   | -1 | -1        | -1        | <b>-1</b> |
| $N$ | -1 | -1        | -1        | <b>-1</b> |

*Huom: Linkkivektori sisältää molemmat suunnat erikseen.*

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

## 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.

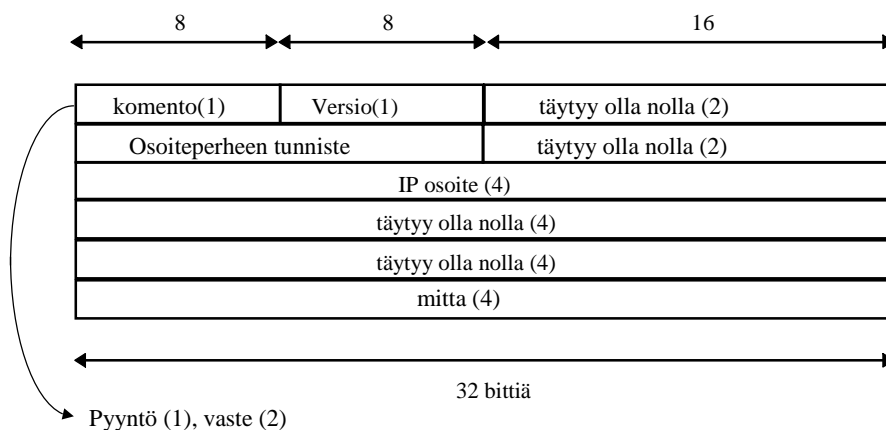
## RIP protokollan peruspiirteitä ovat

- 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



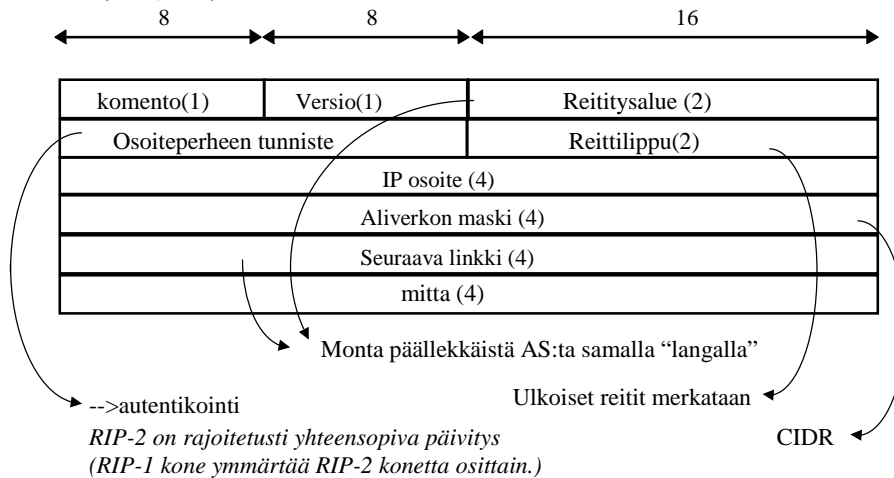


## RIP reititystaulun rivi sisältää

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

## RIP versio 2 - sanomat

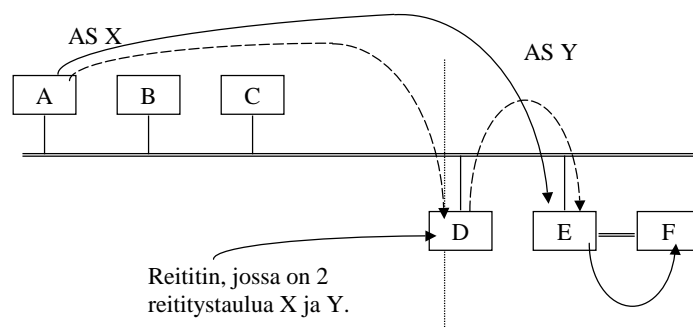
RFC-1388 (1387,1389)



## 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 reitittimeen aliverkosta riippumatta
- RIP-2 korjaa tilannetta kertomalla ulos aliverkon ja aliverkkomaskin

## Reititusalue ja seuraava linkki



Seuraava linkki ==> 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 pakkaus aika+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.
- Korjaus ehdotus perustuu lähetysten kuittausmoodiin, jossa periodisia lähetyksiä ei tapahdu  
--> RIP sanomien puuttuessa oletetaan, että naapuri on edelleen tavoitettavissa  
--> Tieto kaikista vaihtoehtoisista reiteistä talletetaan.