



LUND
UNIVERSITY

EITF45

Internet Routing

JENS ANDERSSON (BILDBIDRAG WILLIAM TÄRNEBERG)



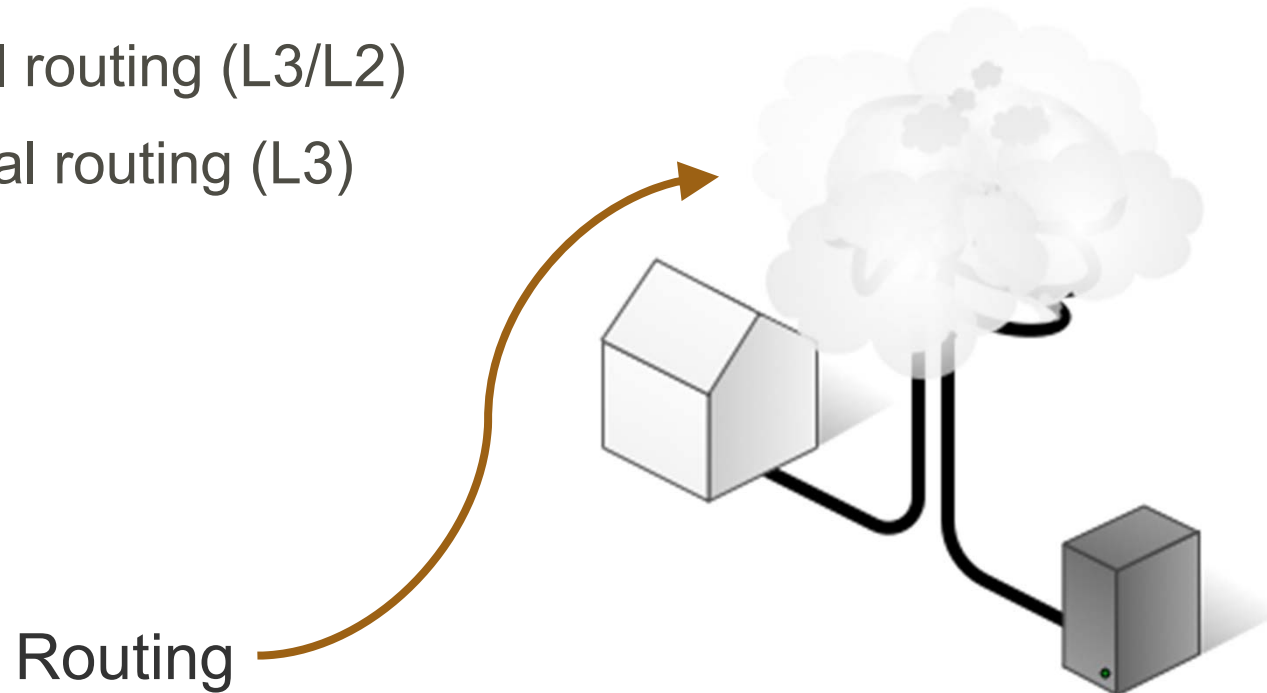
Läsavisning

- Kihl & Andersson:
 - Kap 8, 9.3 – 9.4
- Stallings:
 - Kap 19.1 & 19.2
- Forouzan 5th ed
 - Kap 20.1 – 20.3, 21.1 – 21.2



Agenda

- Internet
- Lokal routing (L3/L2)
- Global routing (L3)



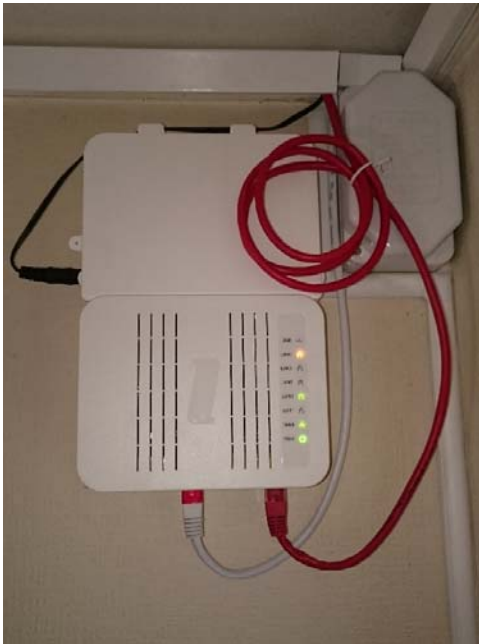
Routern och nät-id

- Routern lär sej hur paket ska skickas bästa väg mot **destinationens nät**
- Routern arbetar med **nät-identiteter/nät-adresser**
- Destinationens **väraddress** är bara intressant för den sista routern på vägen



Fysiskt Internet: Hemmanätet

- Brandvägg
- Router
- WLAN
- Fil-server
- NAS
- Tjänsteserver

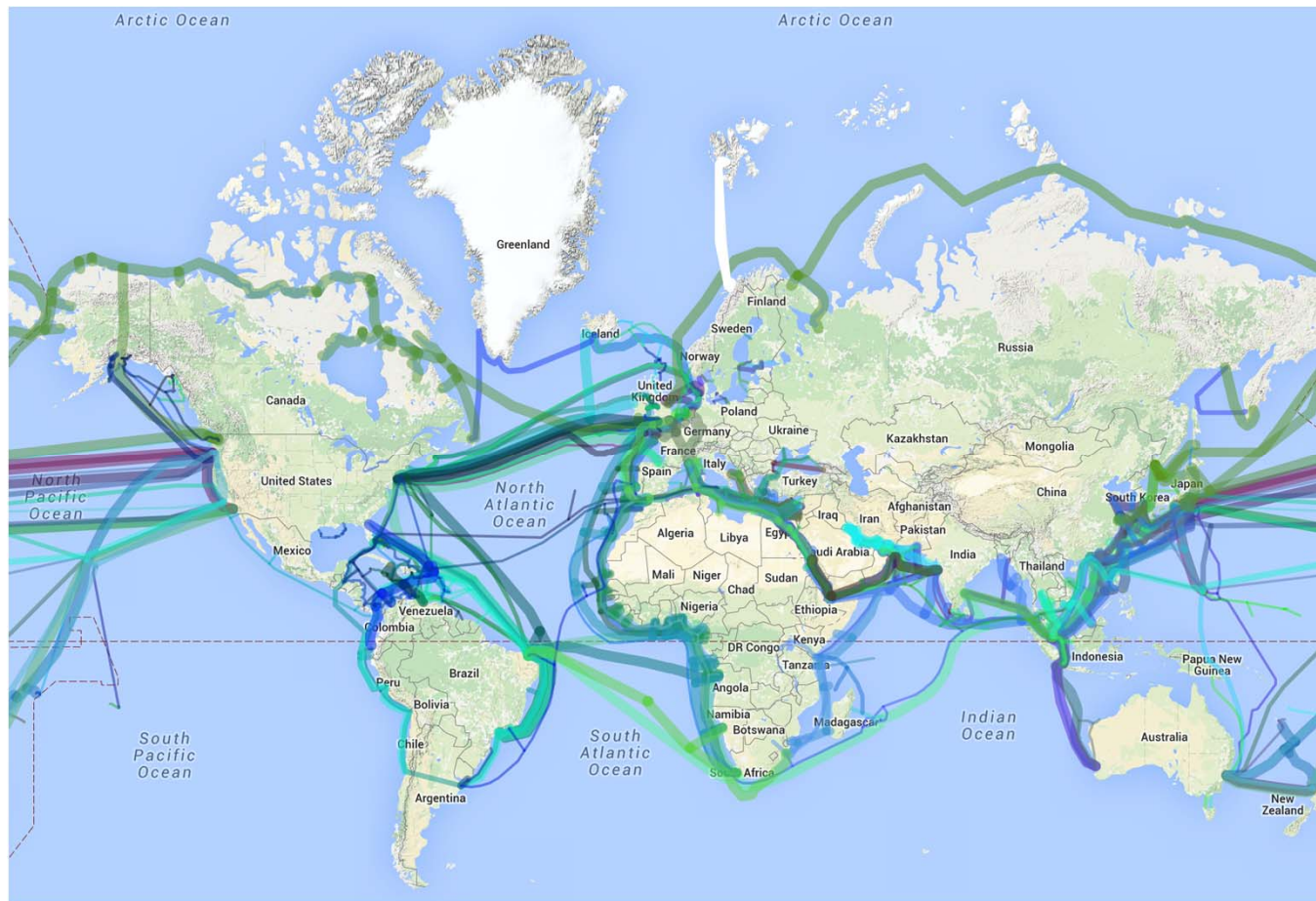


5

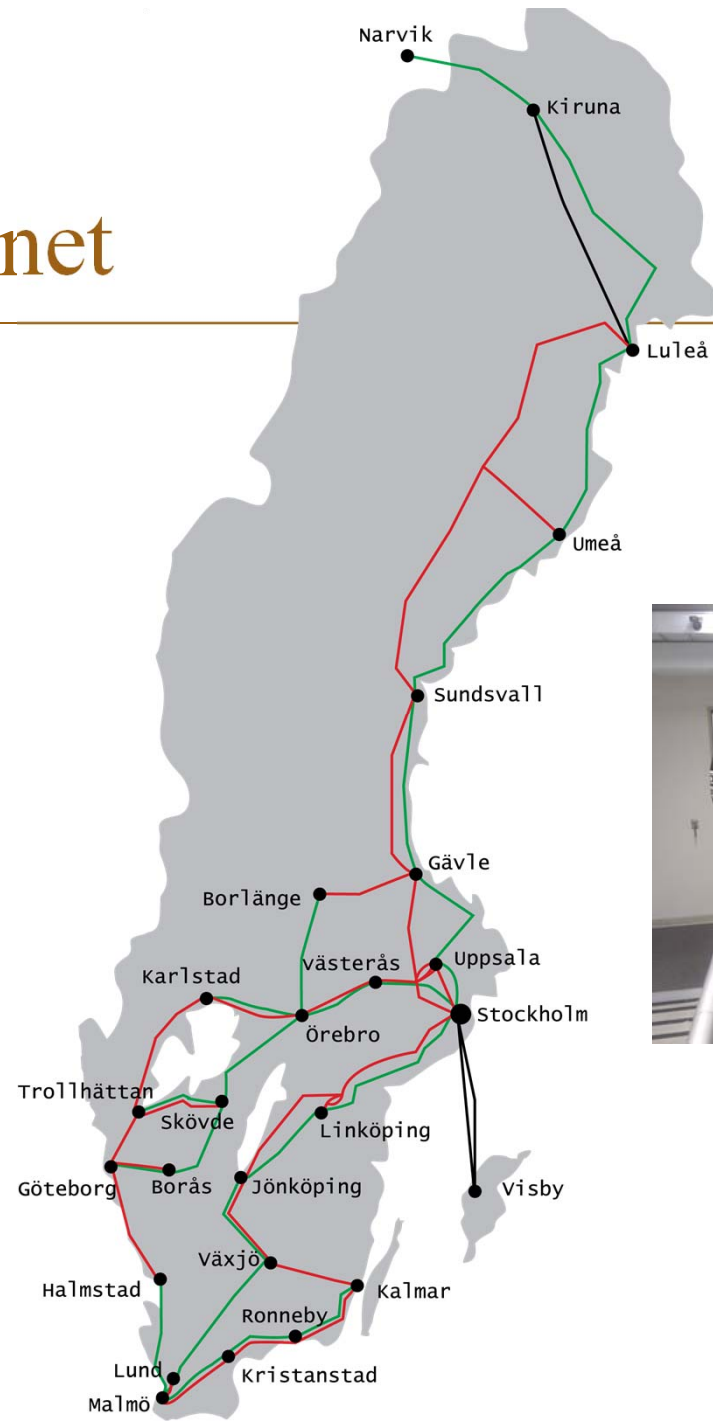


LUND
UNIVERSITY

Fysiskt Internet – Globalt

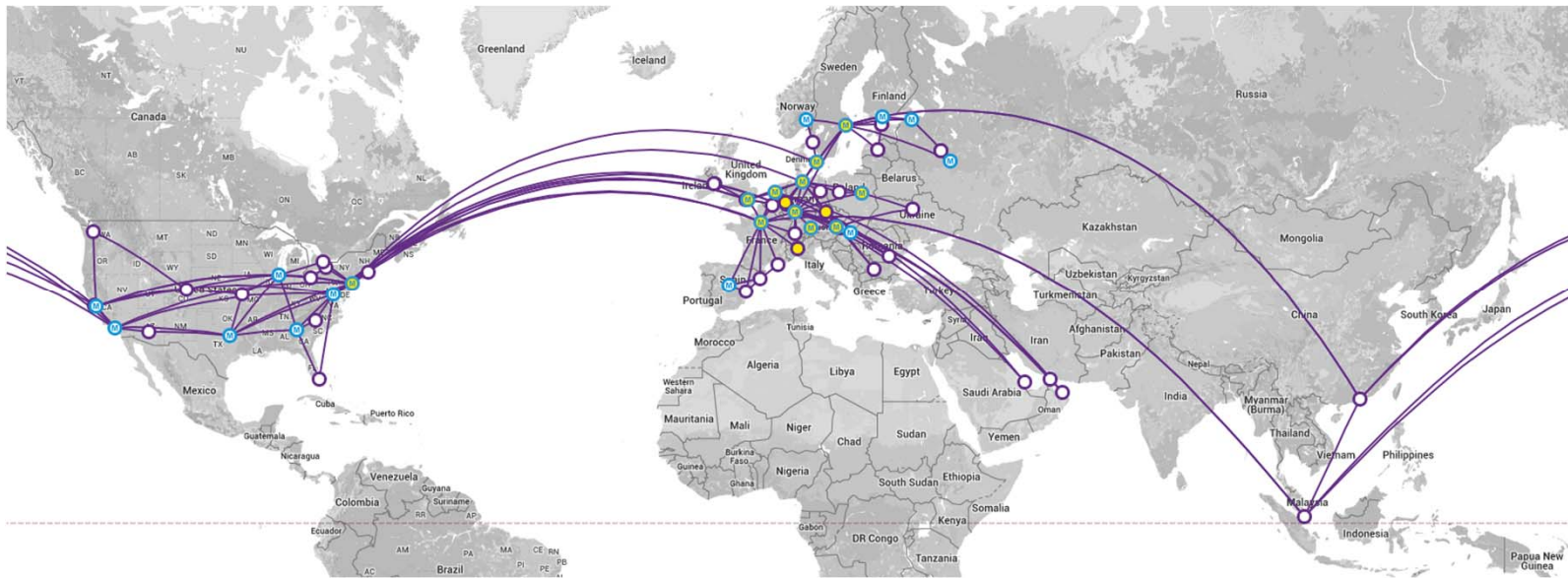


Backbone - Sunet



LUND
UNIVERSITY

TeliaSoneras carrier network

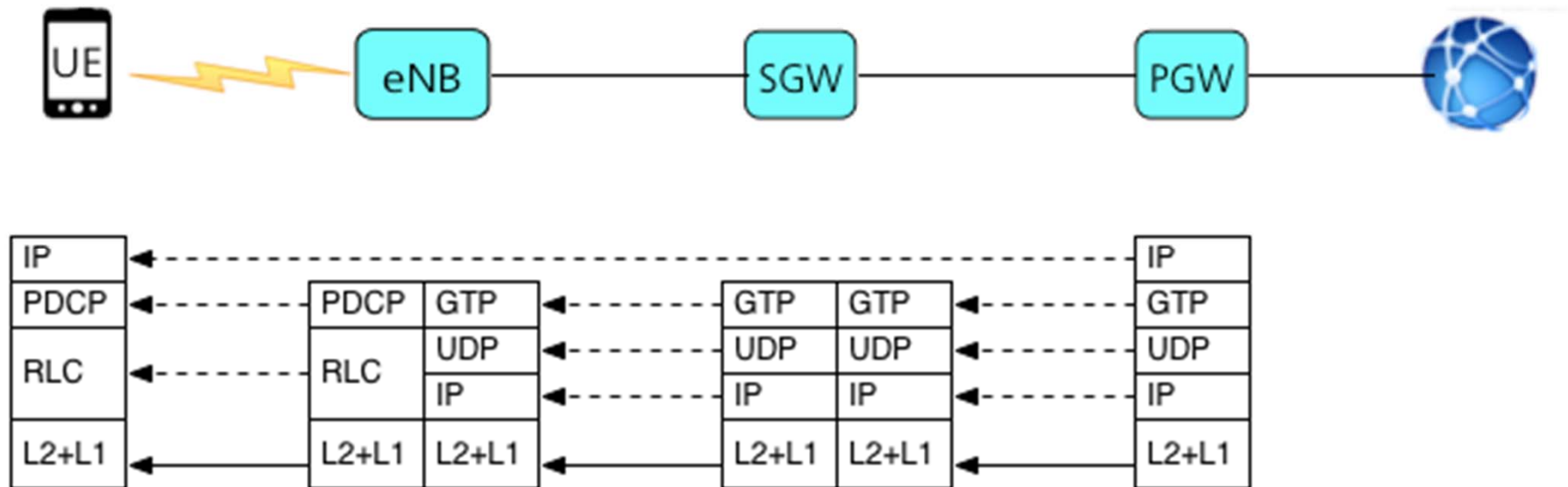


Virtuella Internet



LUND
UNIVERSITY

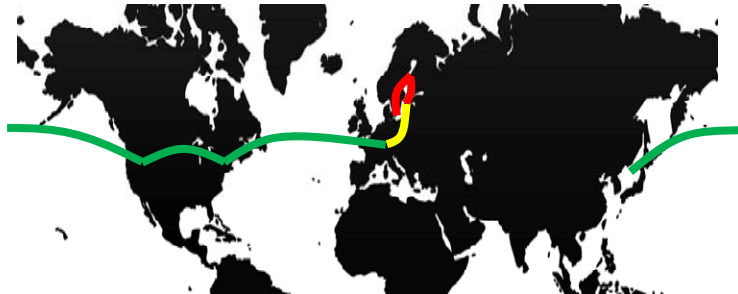
Virtuellt Internet i LTE



Inter Domain Routing = Policy Routing



- Internet är ett nät av nät, inte ett nät
- Avtal, policy viktigare än konnektivitet
 - Vem har vi trafikutbyte med
 - Vem får bära vår trafik



Mer om Inter Domain Routing i
ETSF10 Internetprotokoll



Traceroute

```
traceroute to www.japantimes.co.jp (54.178.172.143), 64 hops max, 52 byte packet
```

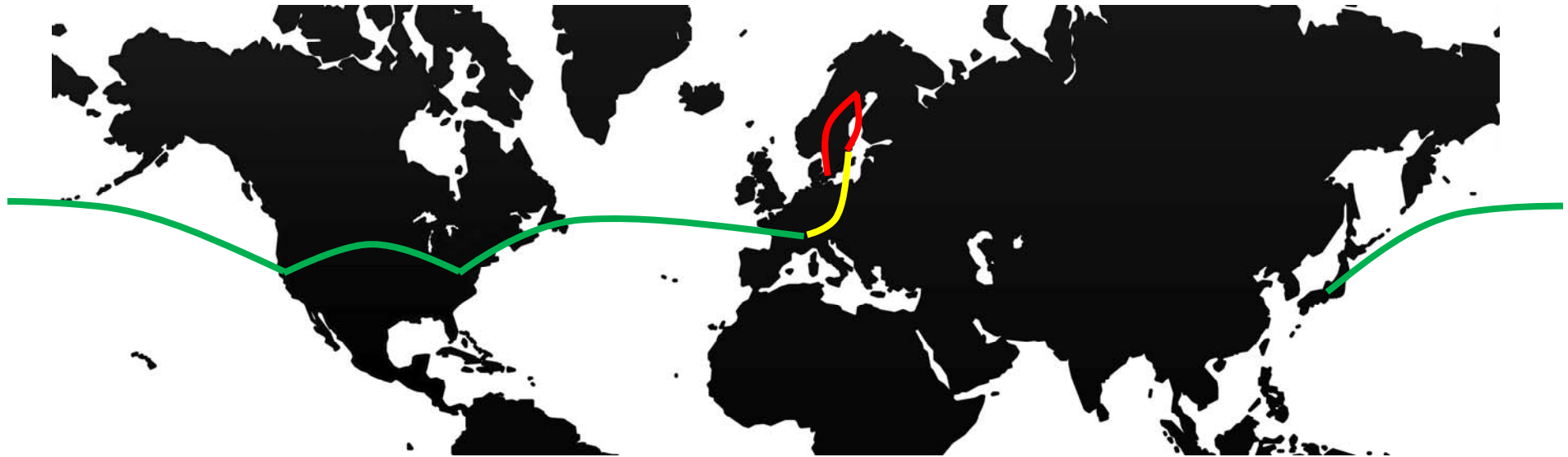
```
1 linksys64996 (10.166.117.223) 1.351 ms 1.228 ms 1.260 ms
2 c213-200-154-20.bredband.comhem.se (213.200.154.20) 2.069 ms 2.012 ms 1.78
3 213.200.165.97 (213.200.165.97) 4.605 ms 5.580 ms
4 213.200.163.33 (213.200.163.33) 3.958 ms
5 kbn-bb4-link.telia.net (80.91.253.244) 4.267 ms 5.590 ms
6 hbg-bb4-link.telia.net (62.115.112.49) 7.616 ms
7 ffm-bb1-link.telia.net (62.115.134.64) 24.637 ms 19.930 ms
8 ffm-b12-link.telia.net (62.115.142.47) 17.866 ms
9 ntt-ic-155239-ffm-b12.c.telia.net (213.248.72.10) 19.154 ms 19.458 ms 18.7
10 ae-5.r21.frnkge03.de.bb.gin.ntt.net (129.250.4.162) 17.665 ms 22.215 ms 1
11 ae-3.r22.londen03.uk.bb.gin.ntt.net (129.250.3.137) 26.374 ms 26.494 ms 2
12 ae-0.r23.londen03.uk.bb.gin.ntt.net (129.250.4.86) 34.629 ms 26.099 ms 26
13 ae-14.r22.osakjp02.jp.bb.gin.ntt.net (129.250.5.221) 286.609 ms 281.092 ms
14 ae-3.r23.osakjp02.jp.bb.gin.ntt.net (129.250.4.121) 298.383 ms 278.948 ms
15 ae-2.r01.osakjp02.jp.bb.gin.ntt.net (129.250.3.199) 278.661 ms 278.195 ms
16 ae-0.amazon.osakjp02.jp.bb.gin.ntt.net (61.200.82.122) 302.717 ms 304.344
17 27.0.0.250 (27.0.0.250) 290.503 ms
18 54.239.52.149 (54.239.52.149) 293.320 ms
19 27.0.0.67 (27.0.0.67) 300.398 ms
20 27.0.0.155 (27.0.0.155) 310.337 ms 306.450 ms 311.632 ms
```

Traceroute till Japan Times (Comhem)



LUND
UNIVERSITY

Traceroute till Japan Times (LTH/Sunet)

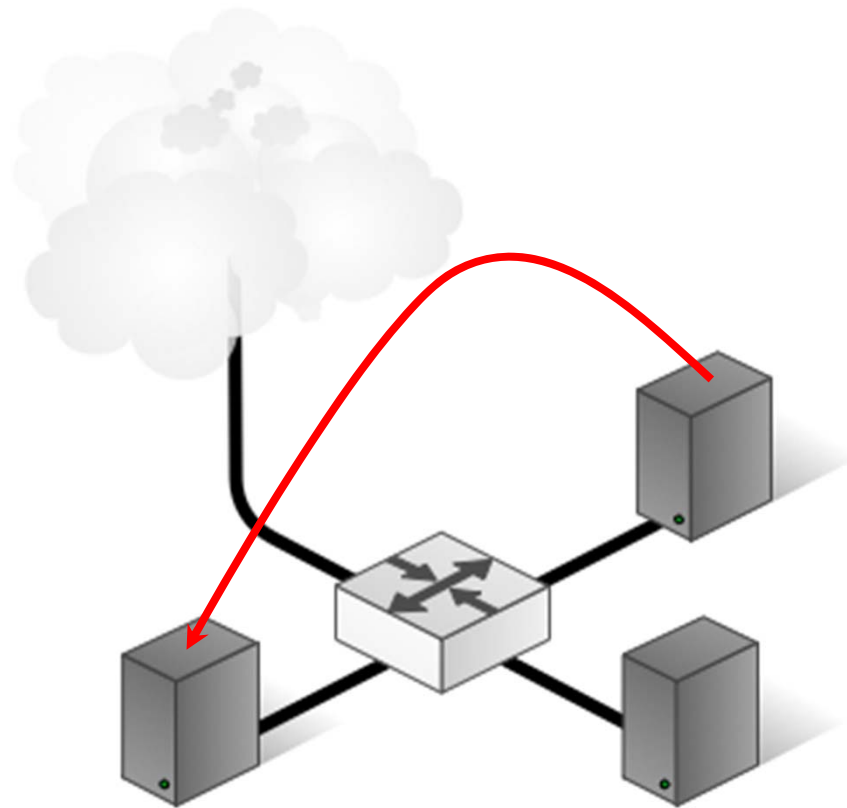


LUND
UNIVERSITY

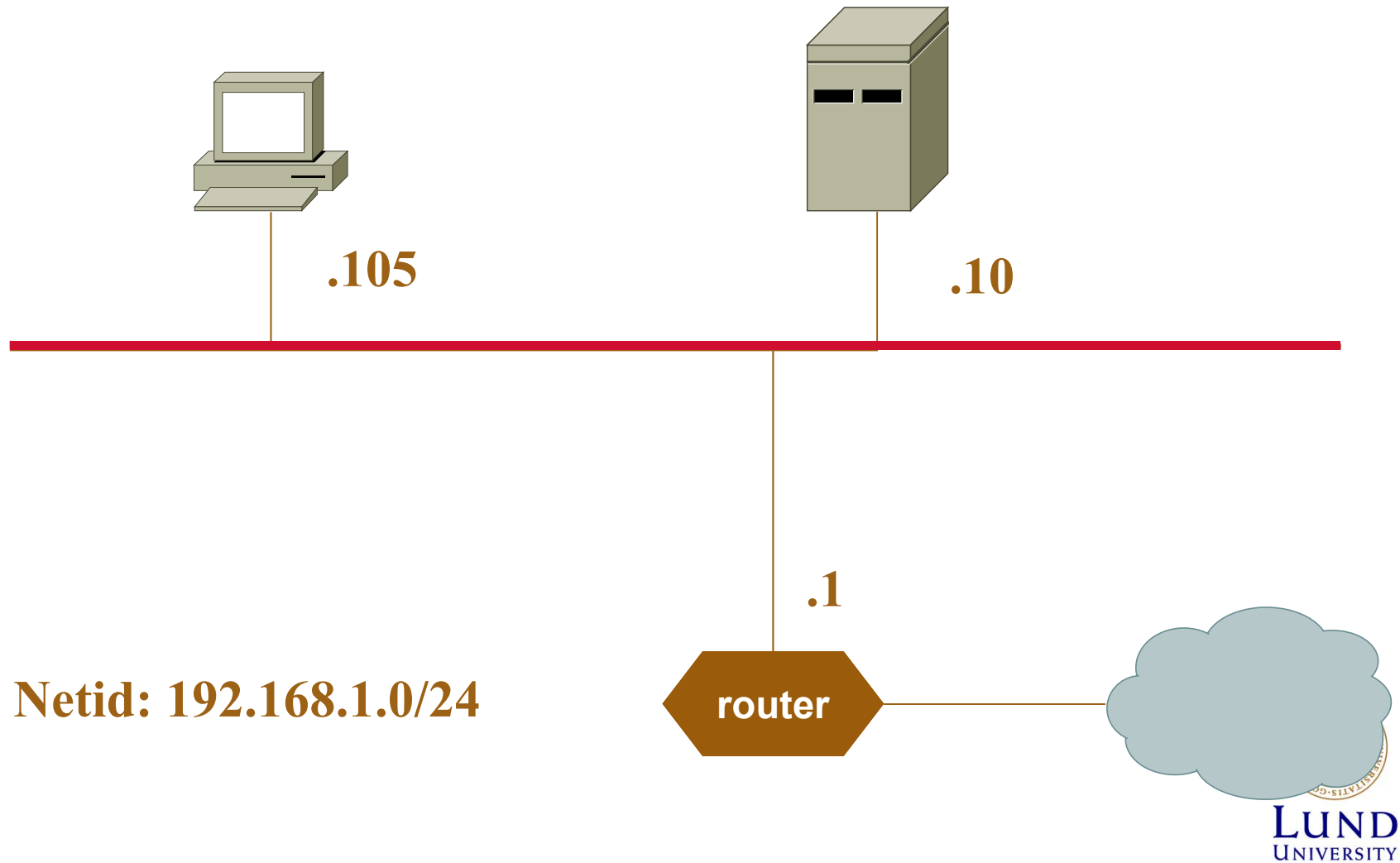
Lokal Routing



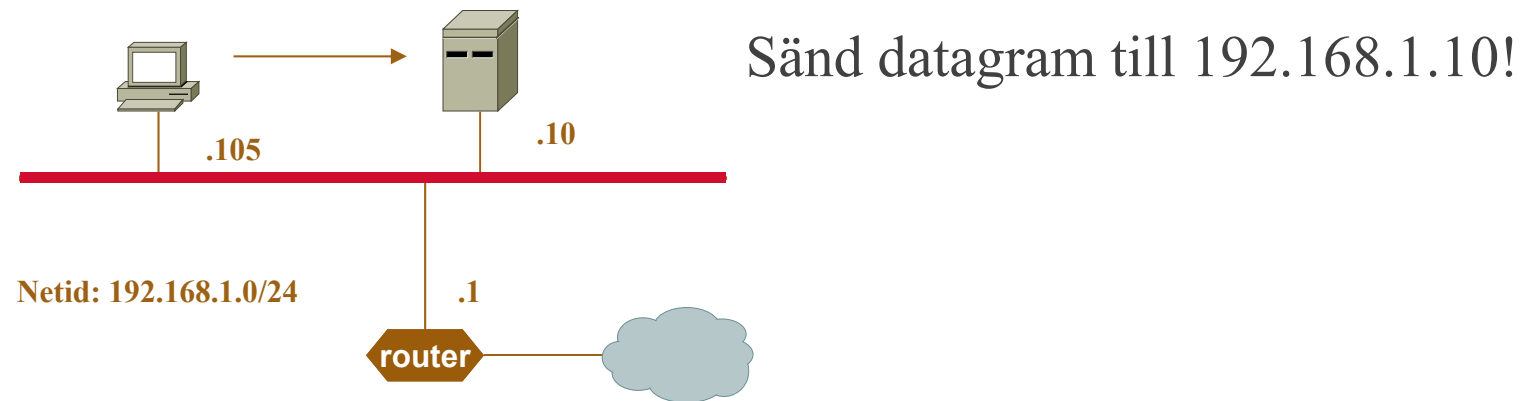
Lokal routing



Lokal Routing & ARP (1)



Lokal Routing & ARP (2)



Är destinationen på samma nät?

Sändaren jämför egen nät-id med destinationens nät-id.

i detta fall JA

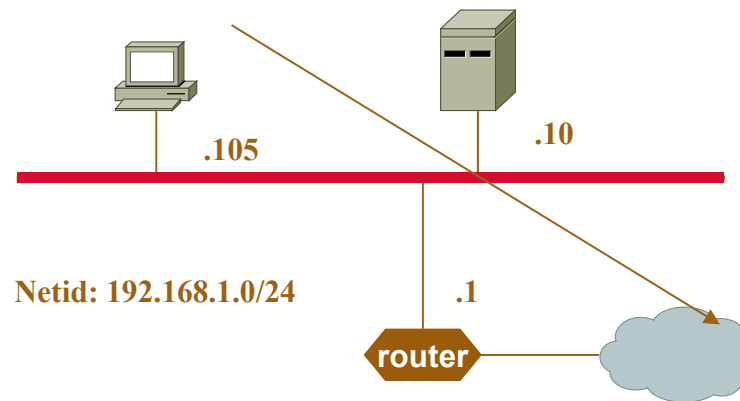
Är destinationens MAC-adressen i ARP-cache?

om JA använd den

om nej använd ARP för destinationen



Lokal Routing & ARP (3)



Sänd datagram till 10.0.100.35!

Är destinationen på samma nät?

Sändaren jämför egen nät-id med destinationens nät-id.

i detta fall NEJ

Är def. gateway MAC-adressen känd och i ARP-cache?

om JA använd den

om NEJ använd ARP för def. gateway



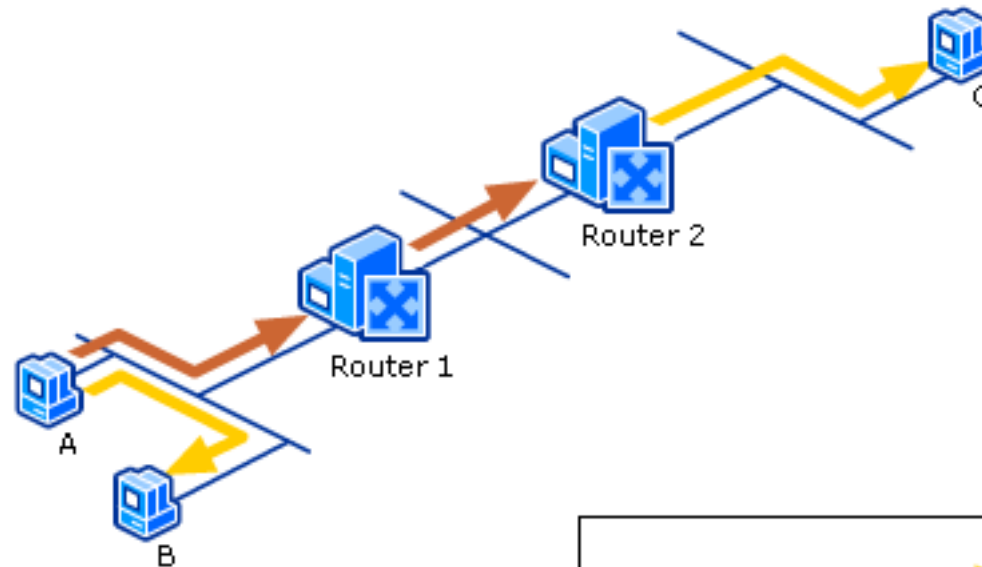
LUND
UNIVERSITY

Lokal routing och IPv6

- Fungerar analogt med IPv4 och ARP
- ARP ersätts med Neighbour Discovery Protocol i ICMPv6



ARP på alla länkar

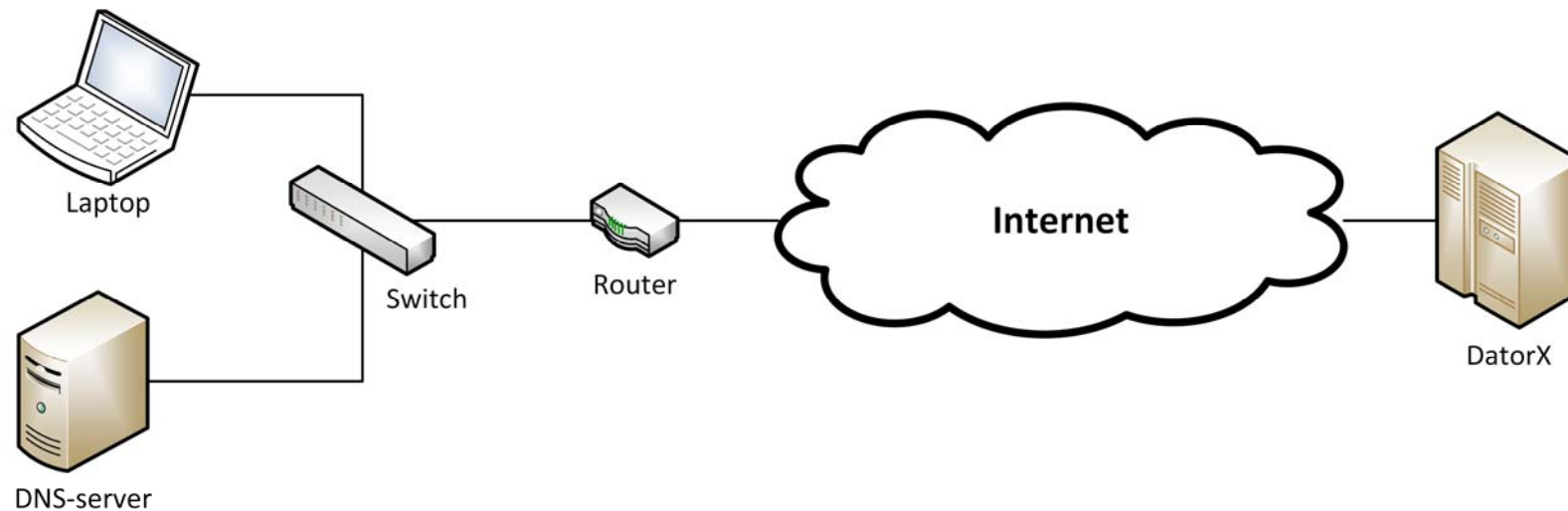


Från dator A till dator C

1. Dator A: ARP request (broadcast) "Router1"
2. Router 1: ARP reply med MAC
3. Router 1: ARP request (broadcast) "Router2"
4. Router 2: ARP reply med MAC
5. Router 2: ARP request (broadcast) "Dator C"
6. Dator C: ARP reply med MAC



Tentafråga



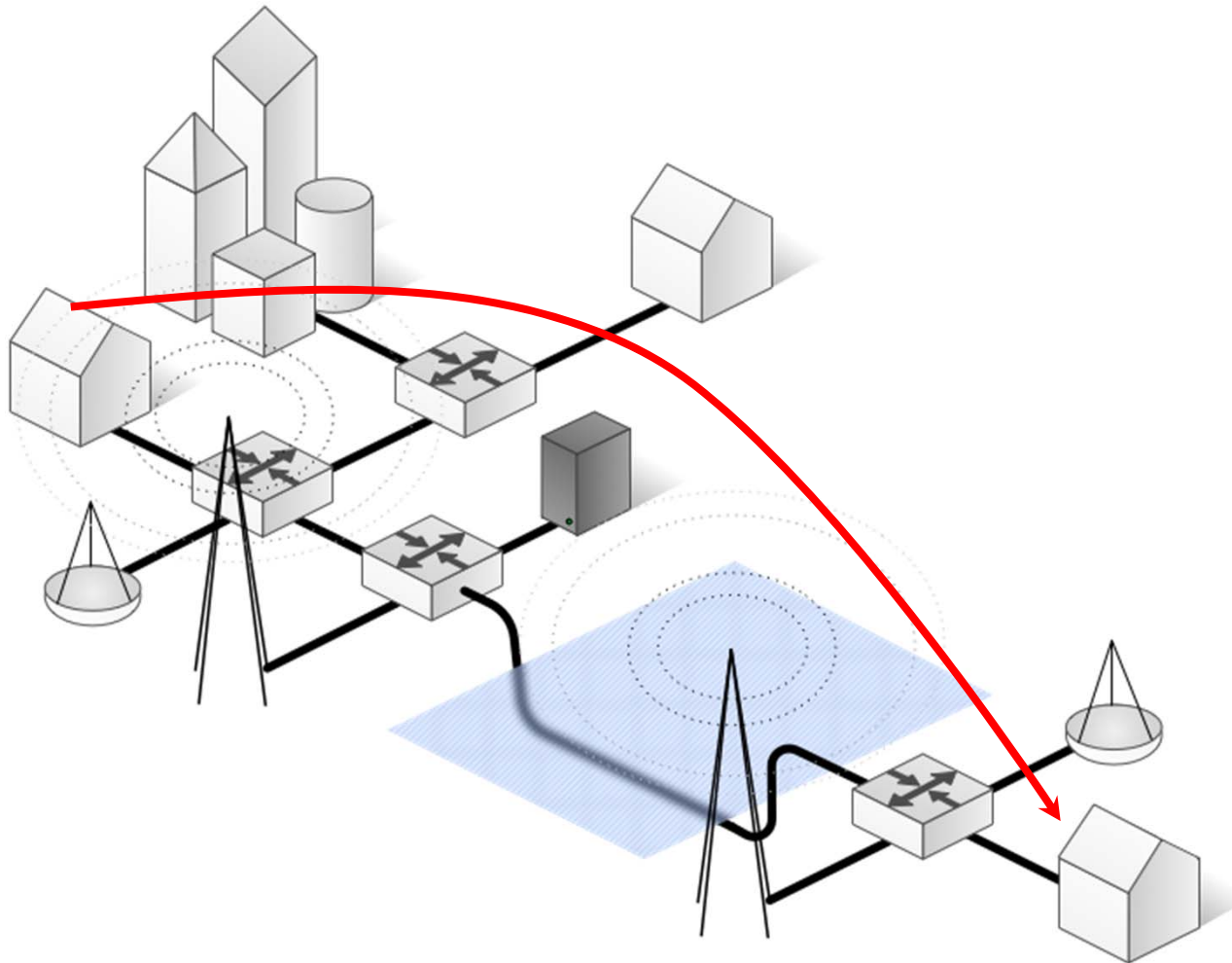
Laptop ska skicka ett IP-paket till DatorX. Vilken IP-address frågas efter i laptoppens ARP request?



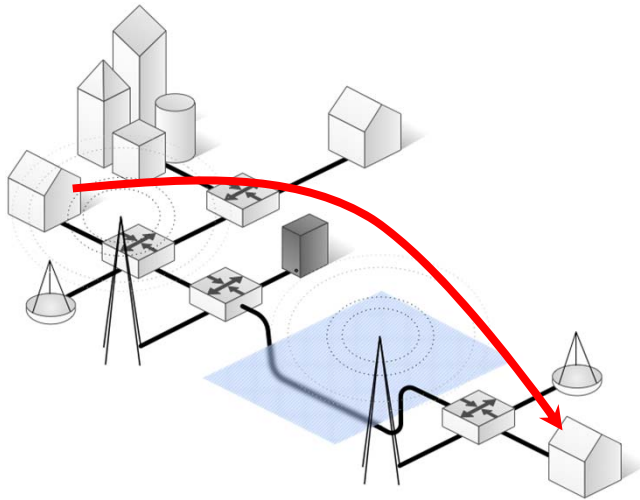
Global routing



Global routing



Global routing

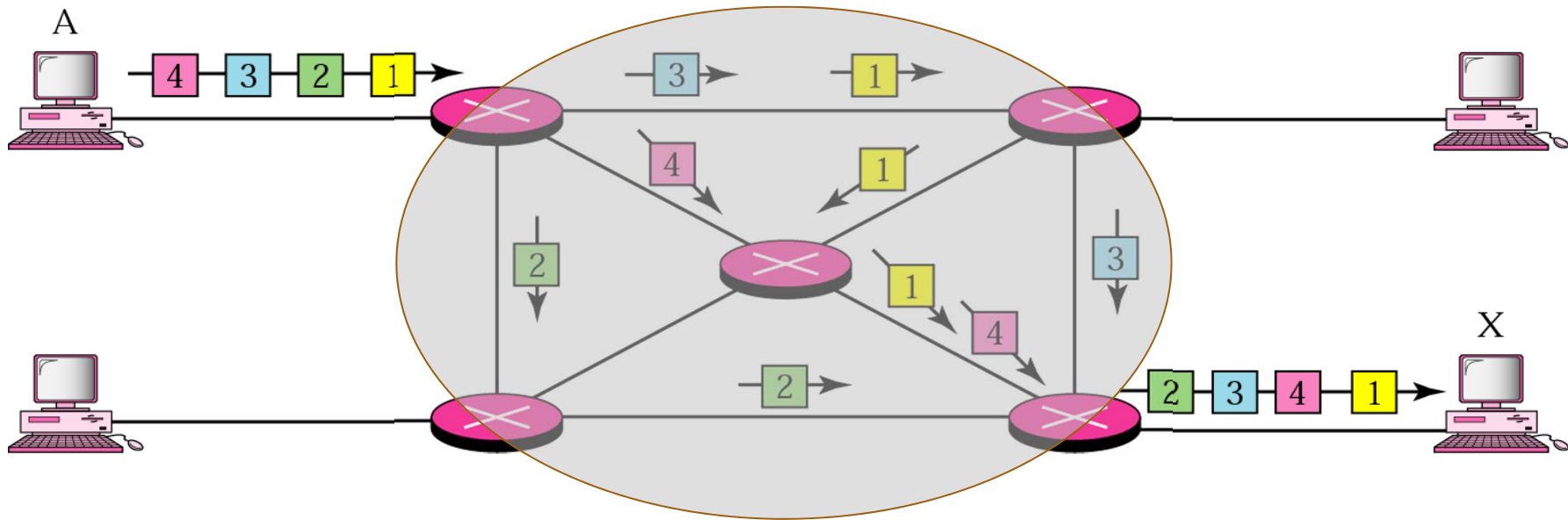


Problem att lösa:

- Enskilda noder skickar vidare till nästa nod
- Varje nod måste ha **kunskap om hela(?) nätet** (destinationer)
- Bestämma **bästa väg** från nod till destinationen
- Bestämma ***Next Hop*** på bästa väg



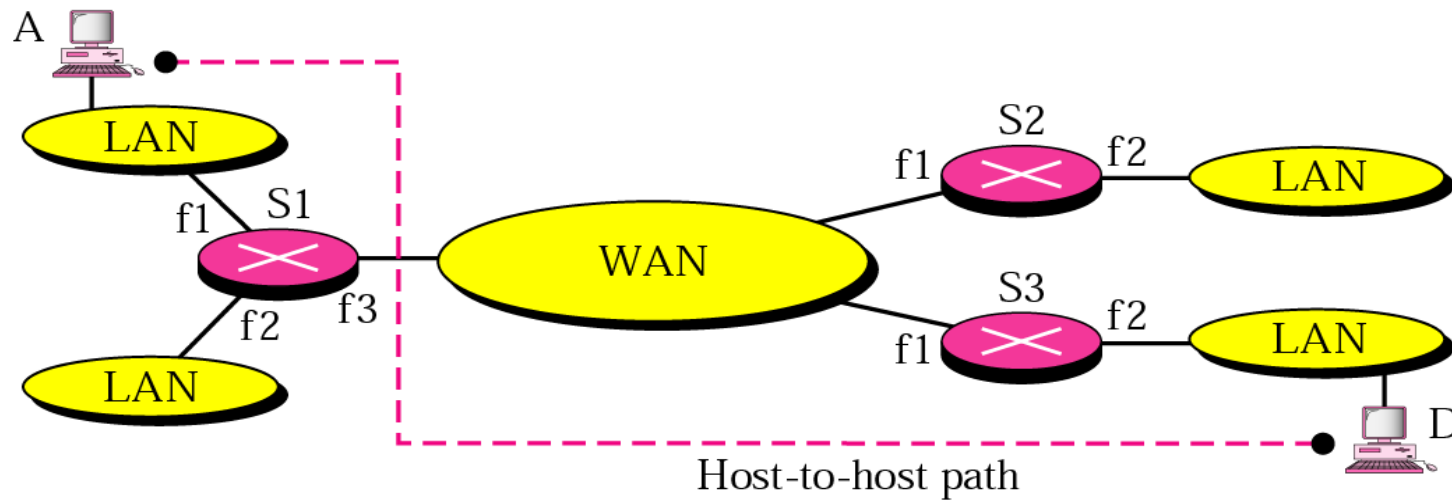
Datagram och nät



- Paketen tar olika vägar
 - kan komma fram i oordning
- Dator A behöver inte veta vilken väg paketet kommer att ta eller om det kommer fram (gäller IP-lagret)



Nätverkslagret /Lager 3

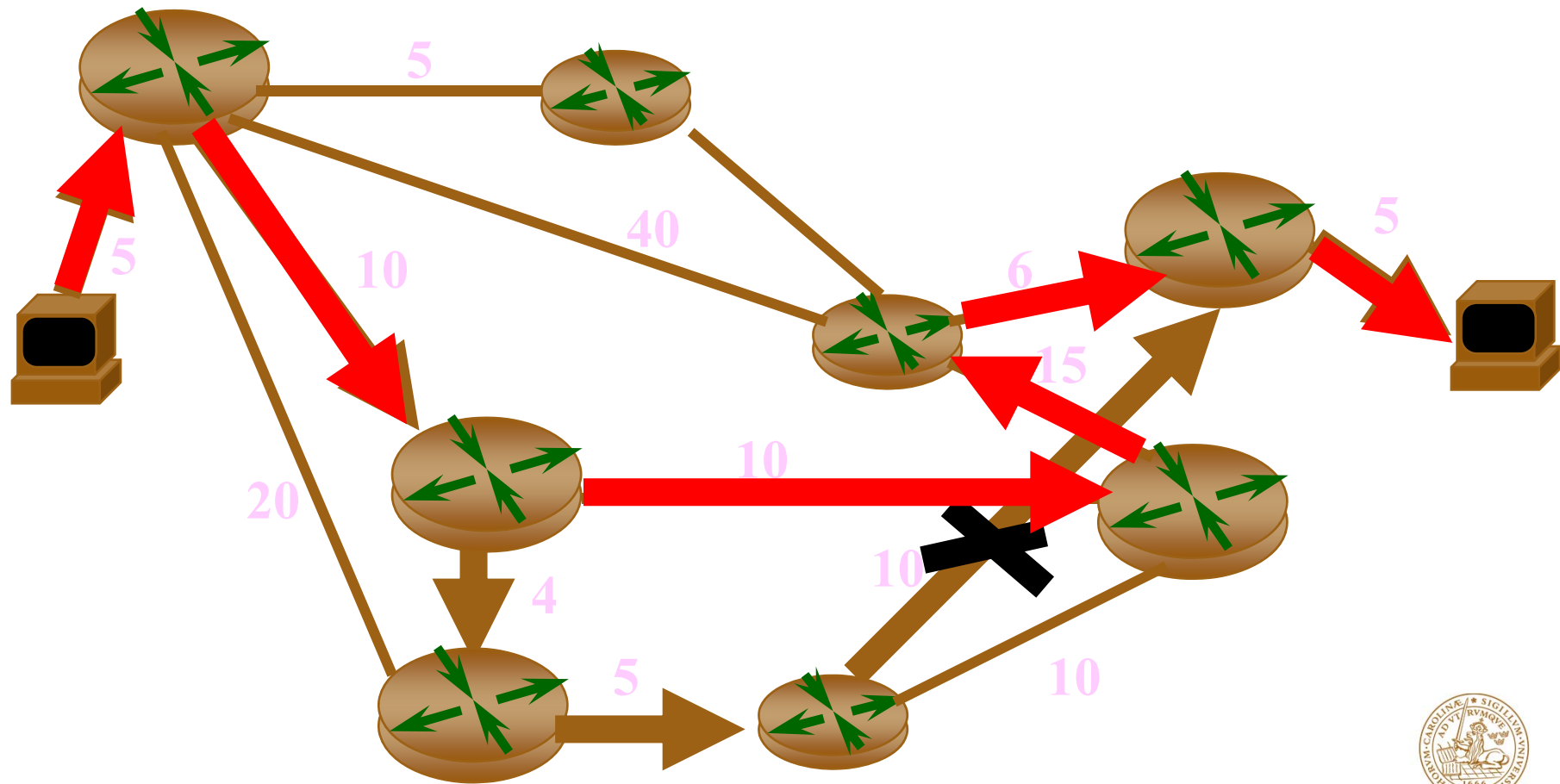


Routern

- En router förmedlar paket mellan nätverk baserat på nätverkslagrets adresser
- **Routing-beslut fattas utifrån nät-identitet** (net id), inte värd-identitet (*host id*)
- En router gör "intelligenta" beslut om bästa väg för paketets vidare leverans mot slutdestinationen



Uppgift: Välj bästa väg!

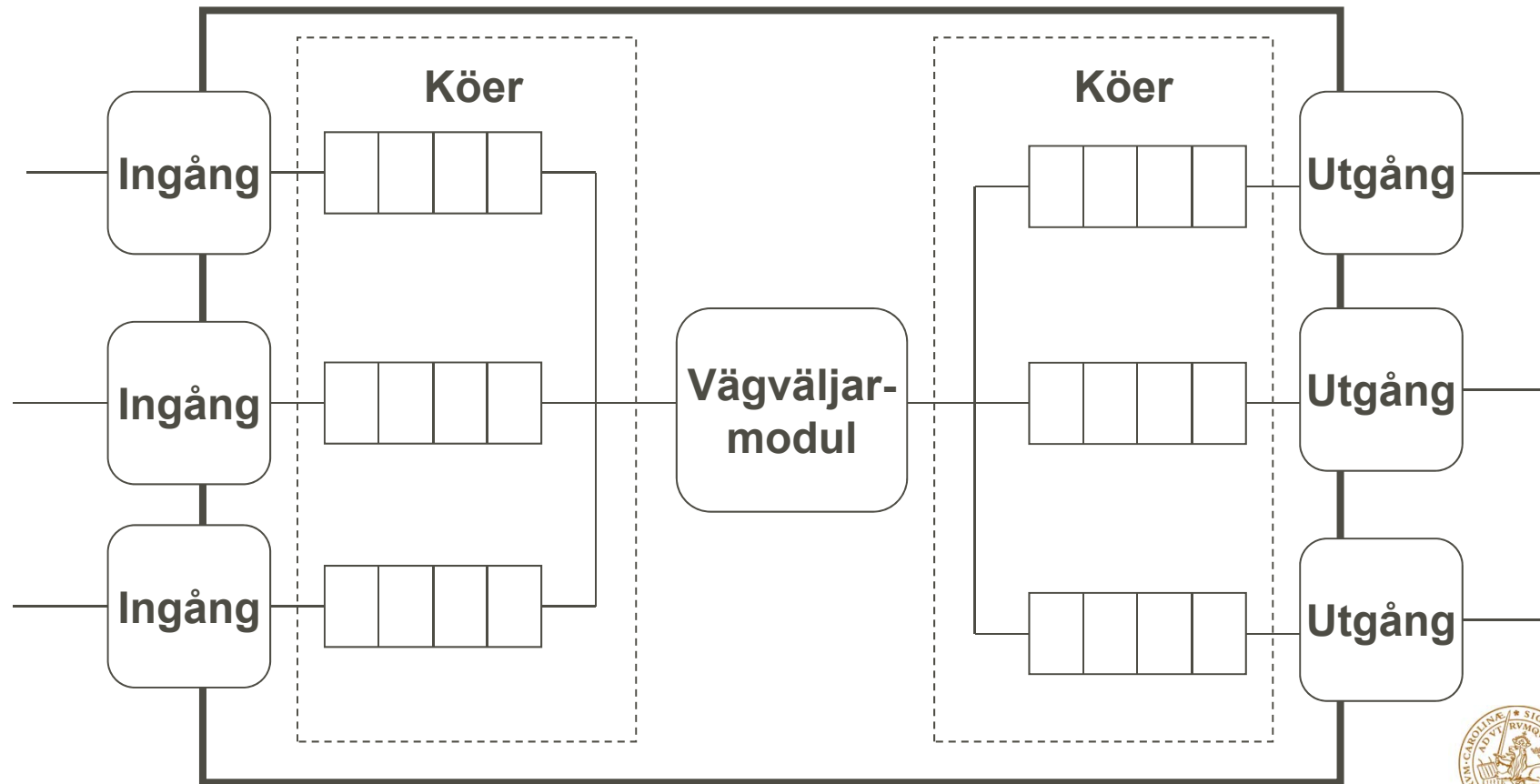


I alla lägen!



LUND
UNIVERSITY

Routern, schematiskt



Routingprinciper

- Ingen “intelligens”
- Centraliserad
- **Distribuerad**

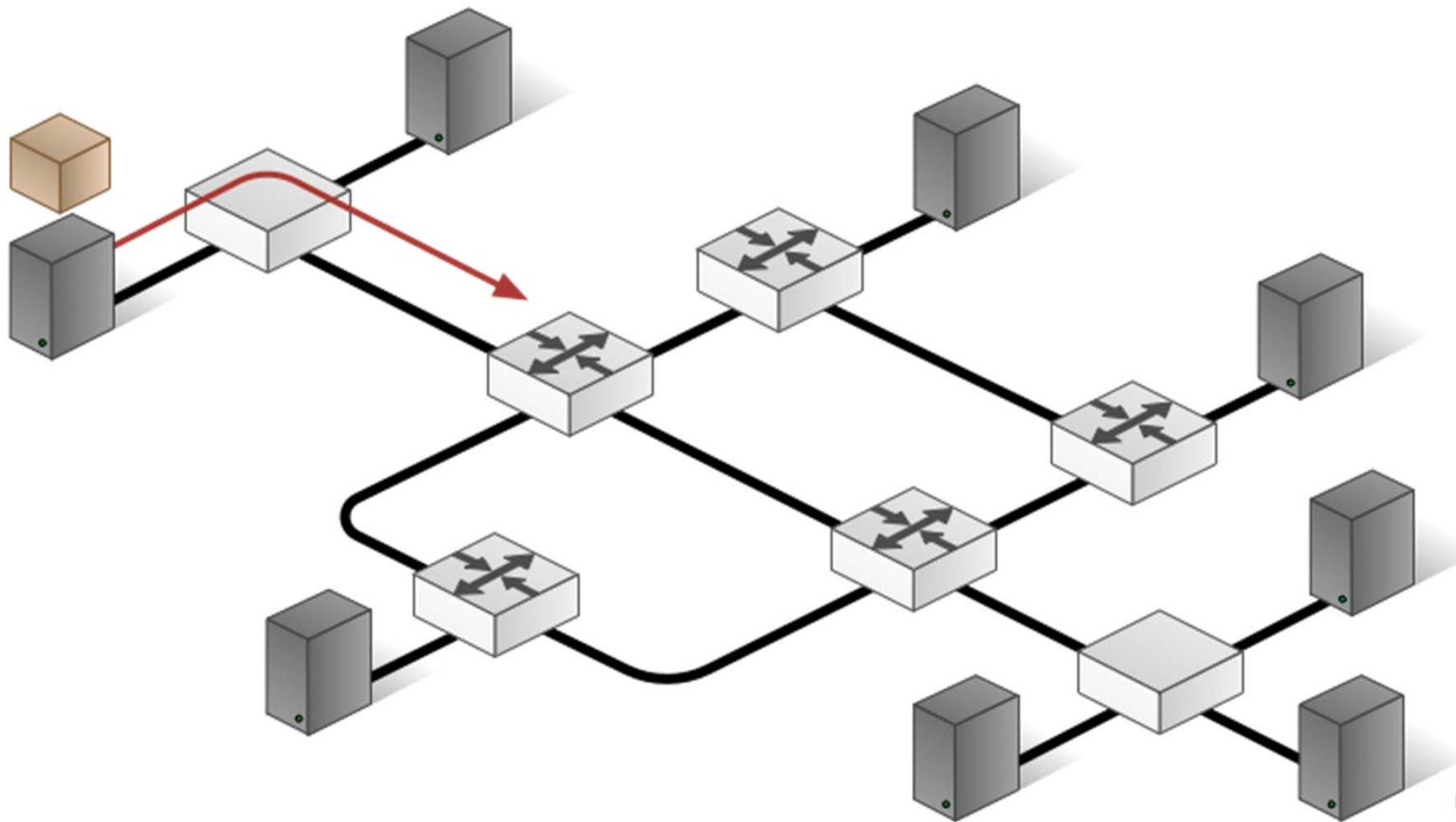


Ingen “intelligens”: Flooding

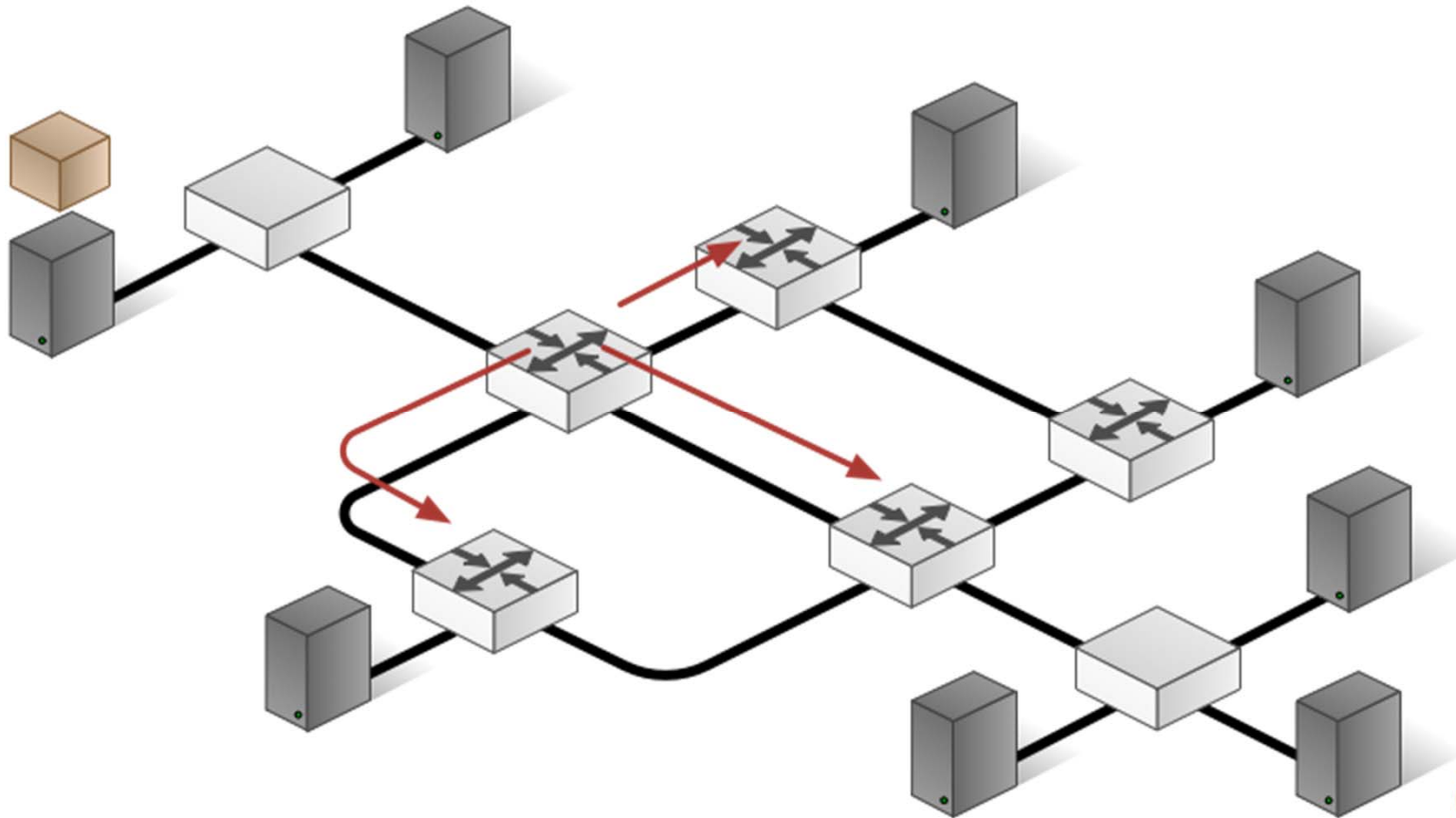
- Skicka ut all paket/datagram
 - På alla portar/interface/linkar
 - Utom ingress-porten/interfacet/linken



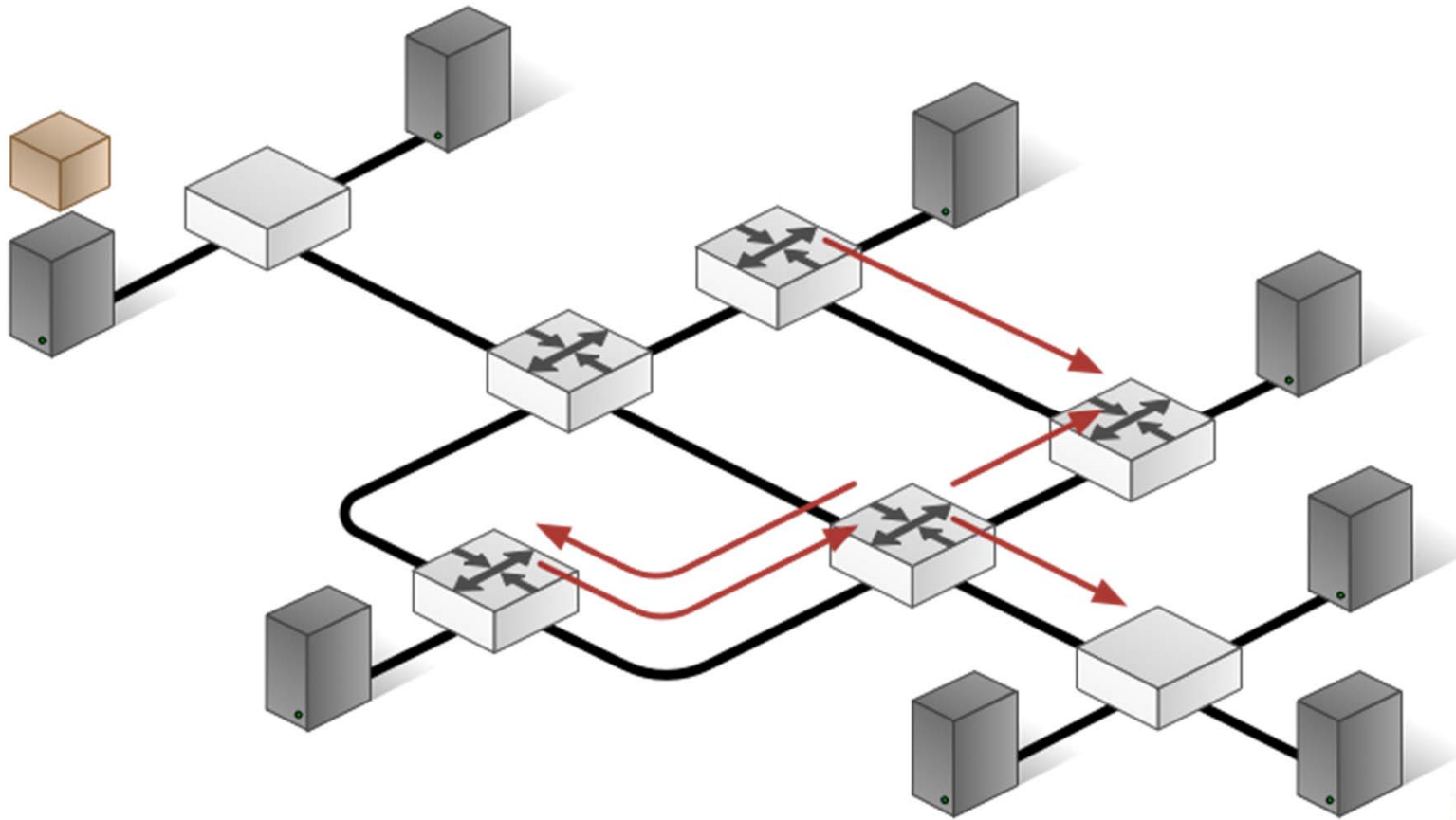
Flooding



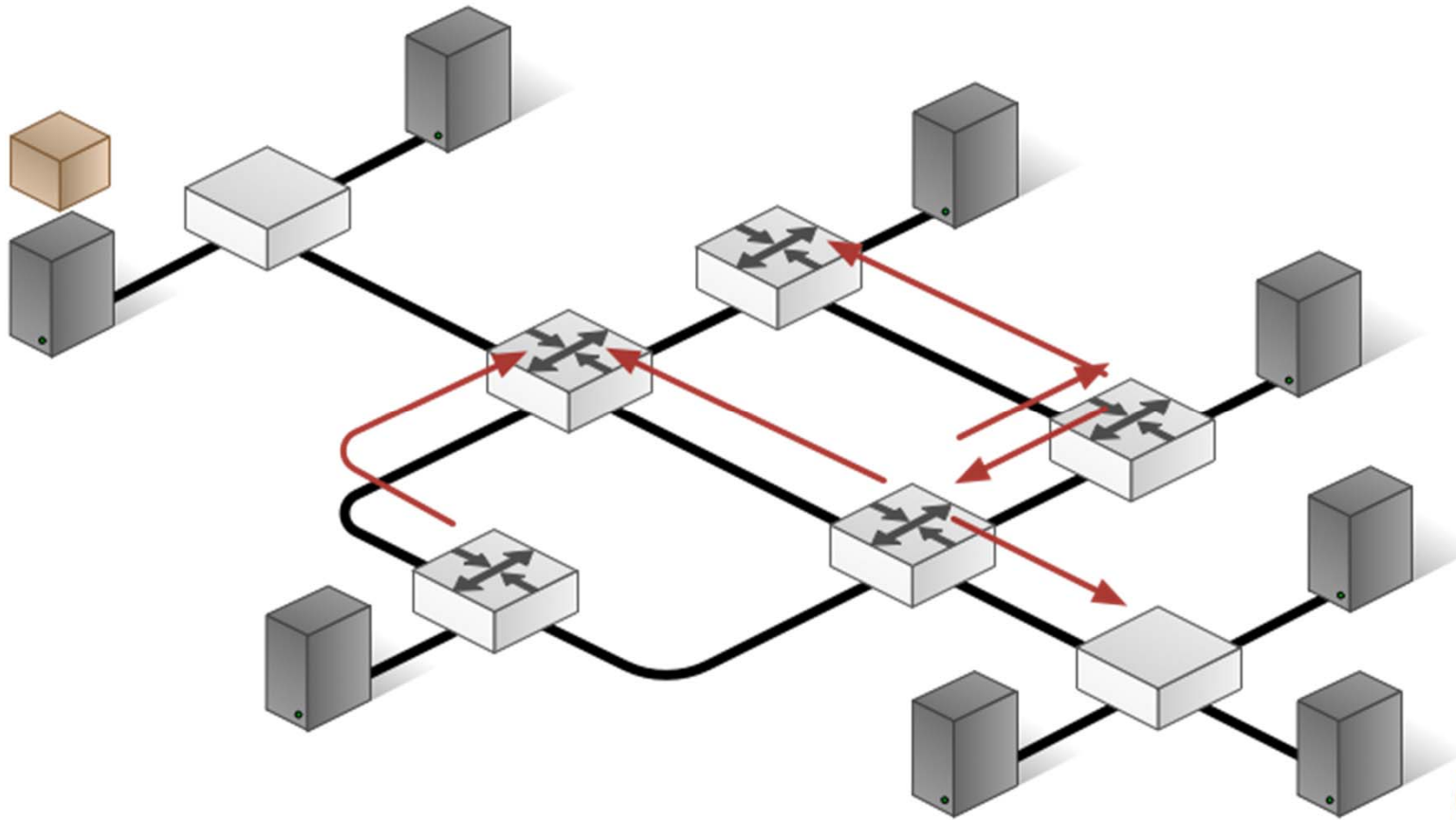
Flooding



Flooding



Flooding



Ingen “intelligens”: Flooding

- Skicka ut all paket/datagram
 - På alla portar/interface/linkar
 - Utom ingress-porten/interfacet/linken
- Problem?
 - Paket som loopar
 - Onödig trafik
 - Två lösningar
 - » TTL-räknare
 - » Kom ihåg vilka paket som redan hanterats

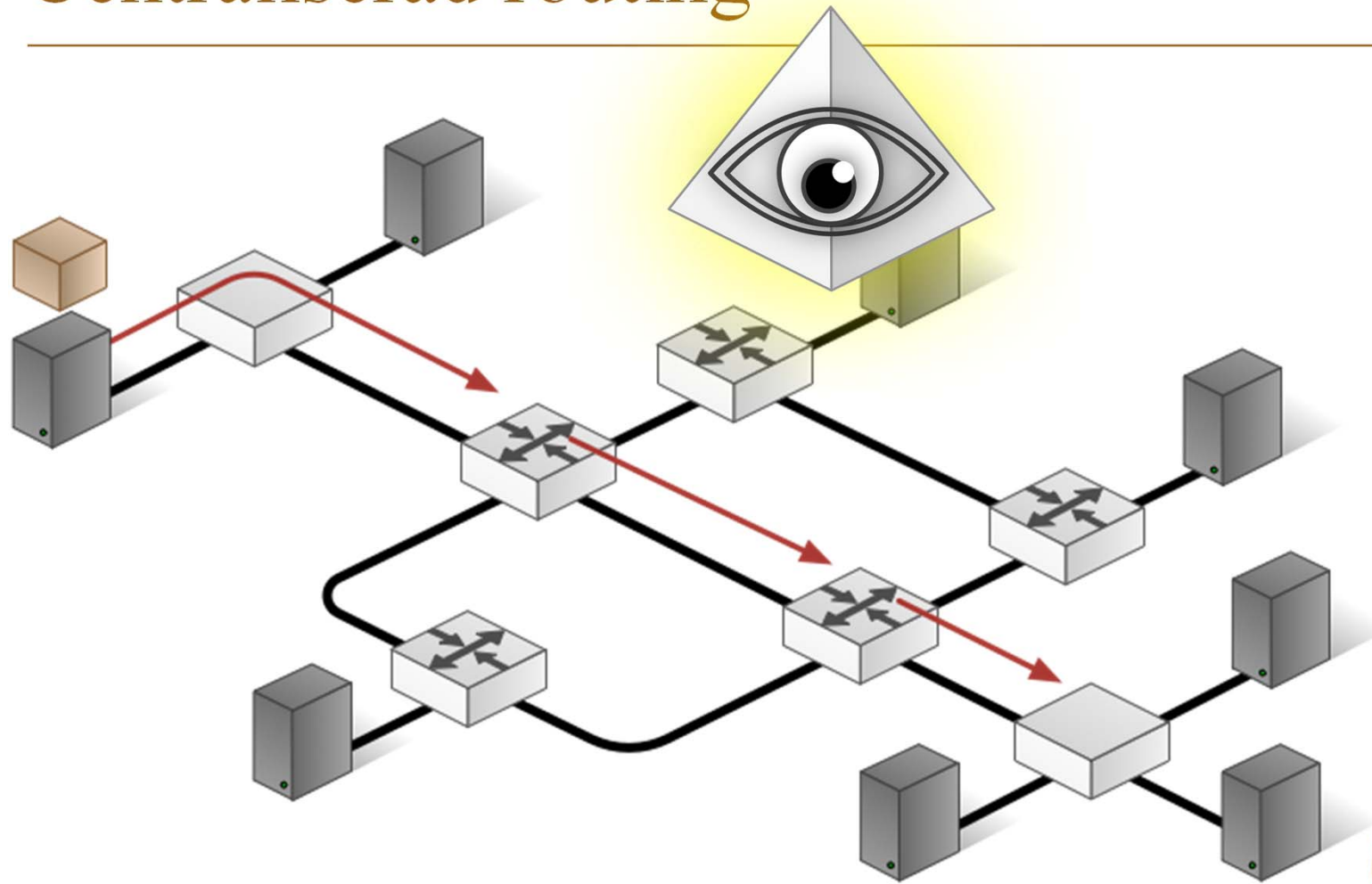


Centraliserad routing

- Separera *control plane* och *data plane*
- Databas och algoritm centralt
 - Noderna i nätet uppdaterar den centrala funktionen
- Paketförmedlingen distribuerad
 - självklart! eller?
- *Software Defined Networks*



Centraliserad routing

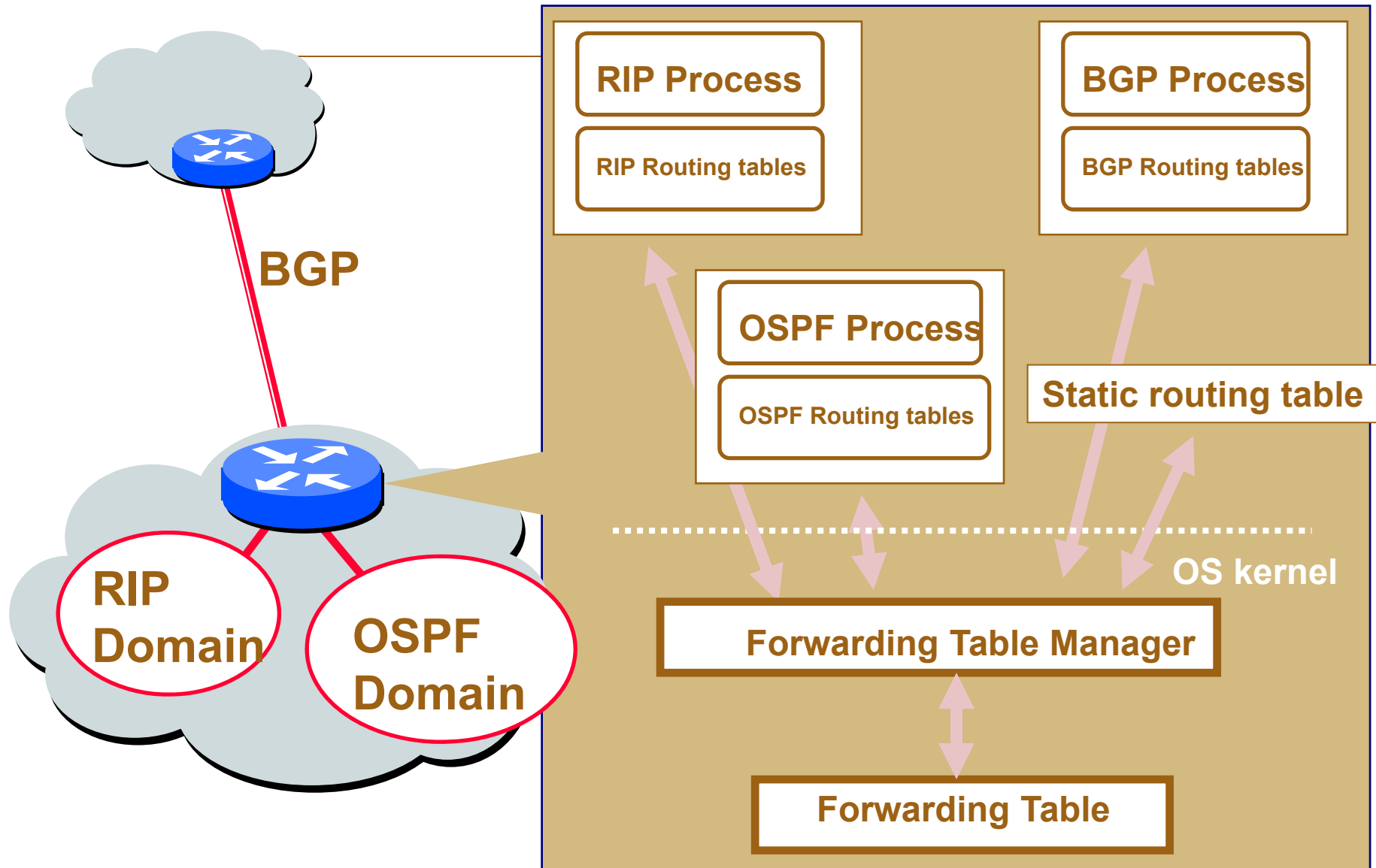


Distribuerad routing

- Routingprocessen (*control plane* och *data plane*) distribuerad till alla routrar
- Två metoder
 - **Distance Vector**
 - » Varje nods information om bästa vägar distribueras till nodens grannar
 - » Bästa väg e-2-e fås fram genom jämförelse med alla möjliga *next hop*
 - » Enkelt, låga krav på processor och minne
 - **Link State**
 - » Information om lokal om topologi flödas (*flooding*) till alla noder
 - » Bästa väg e-2-e till alla noder beräknas lokalt i varje nod (trädbyggnad)
 - » Komplicerat med krav på processorkraft och minne



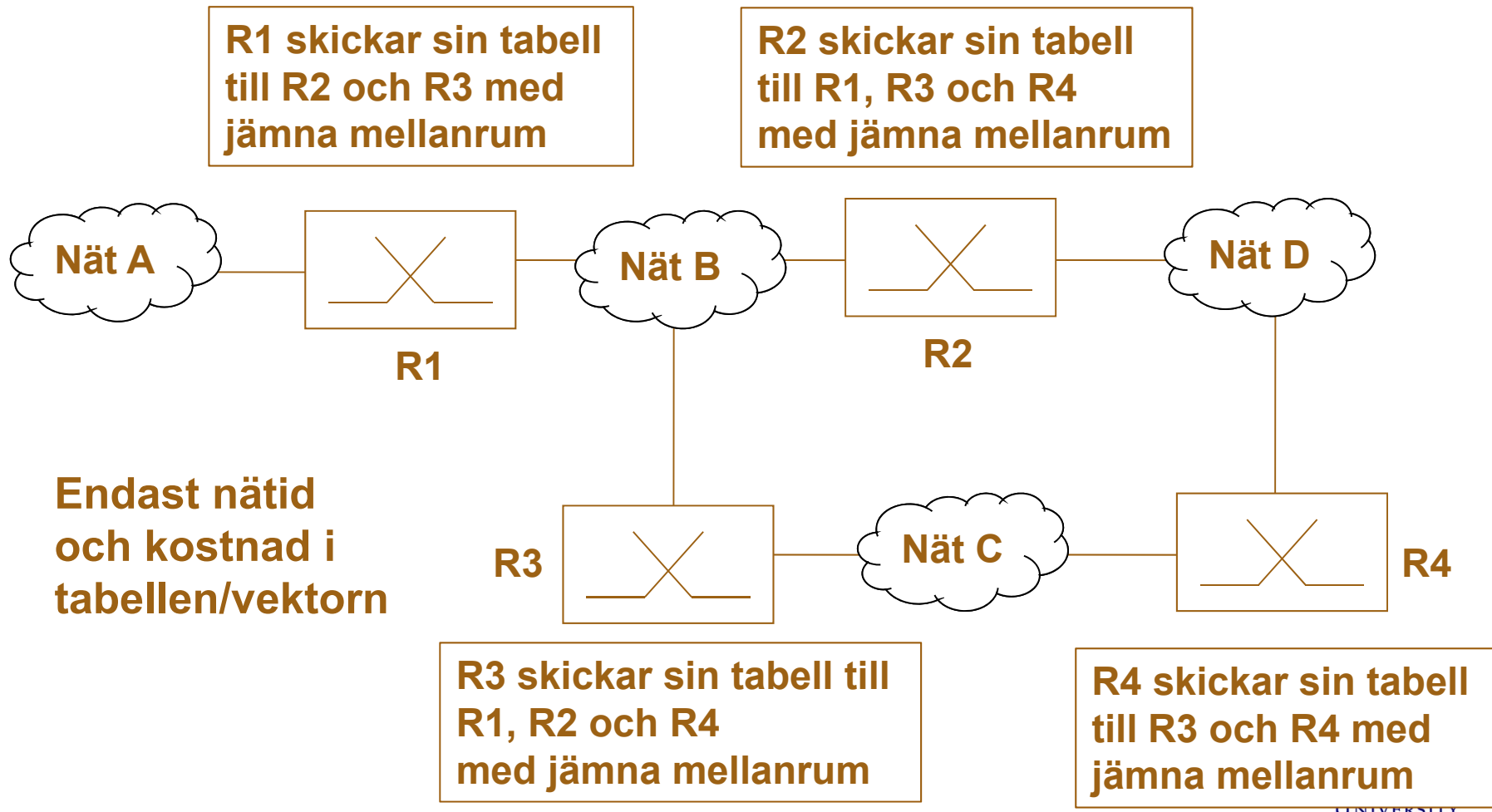
Routing Tables and Forwarding Table



Distance Vector



Distance Vector: Princip



Distance vector: princip

- Alla kända bästa vägar **skickas till grannar**
 - Periodiskt
 - Vid varje förändring
- Routingtabeller **uppdateras** vid
 - Info om nya noder
 - Ändrad kostnad eller vägar/*paths*
- ”Global kunskap sprids lokalt”



En distance vector

Allmänna fallet

Nod	Kostnad
-	-
-	-
-	-
-	-
-	-

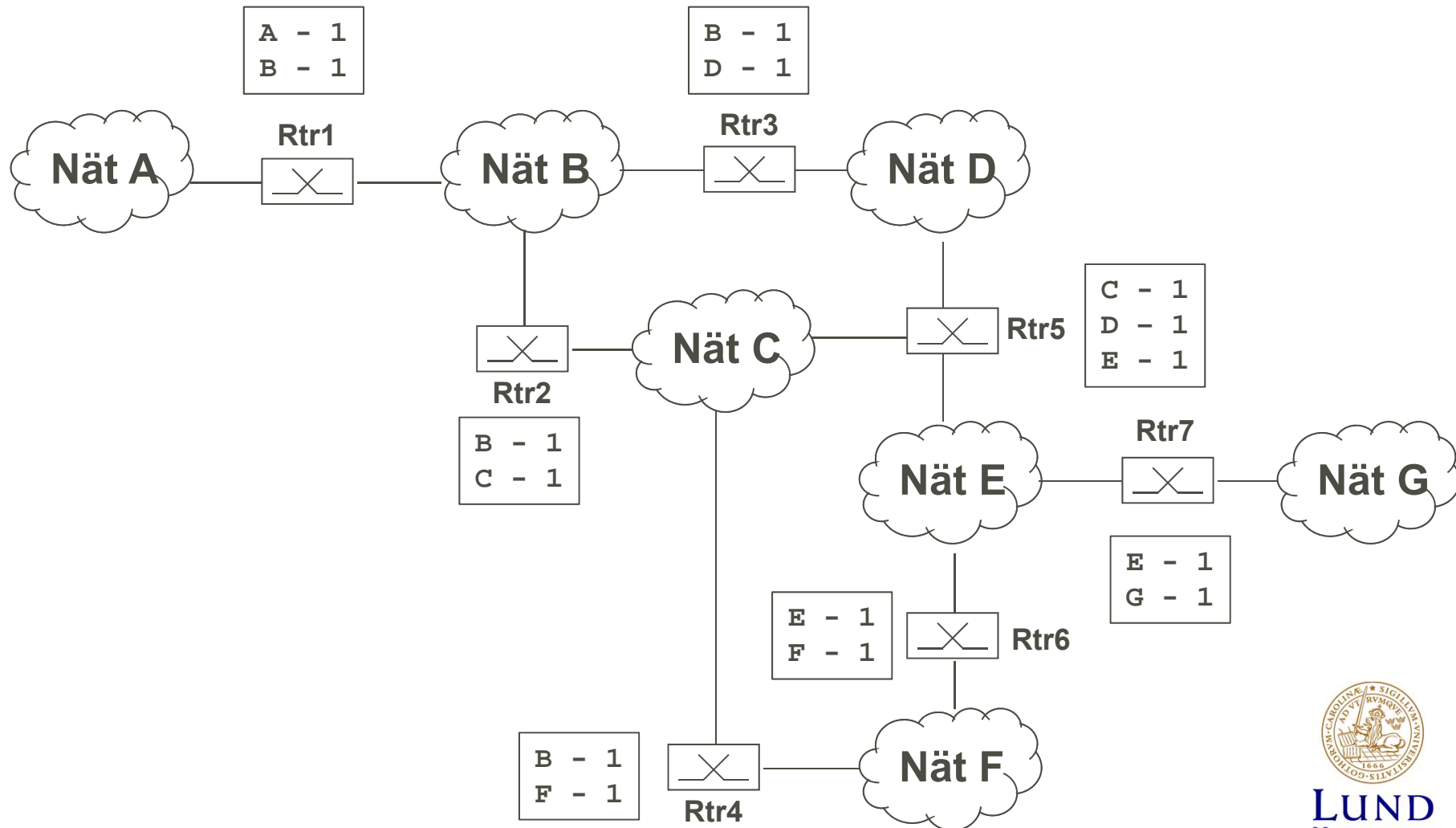
För routing

Destination	Kostnad
-	-
-	-
-	-
-	-
-	-

Next Hop = den som skickar vektorn



Exempel - Hop count



Uppdatering av routingtabell

Rtr3

Ursprunglig

B - 1
D - 1

Från Rtr5 +1

C 2
D 2
E 2

Uppdaterad

B - 1
C rtr5 2
D - 1
E rtr5 2

Rtr1

Ursprunglig

A - 1
B - 1

Från Rtr2 +1

B 2
C 2

Uppdaterad

A - 1
B - 1
C rtr2 2

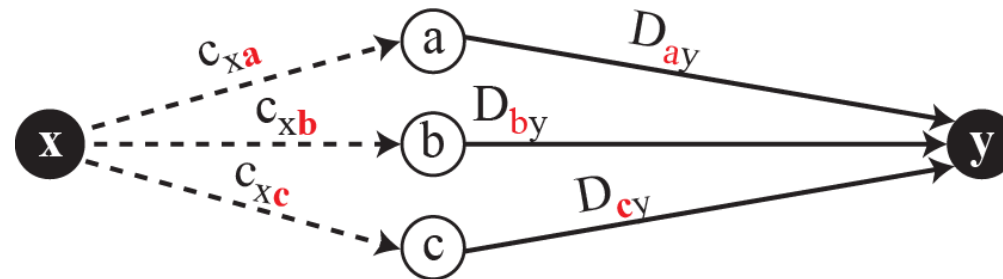


Bellman-Fords algoritm

```
(1)      if (advertised destination not in table) then
           update table
(2)      else
(2.a)    if (advertised next-hop = next-hop in table) then
           replace entry
(2.b)    else
(2.b.i)  if (advertised hop count < hop count in table) then
           replace entry
(2.b.ii) else
           do nothing
```

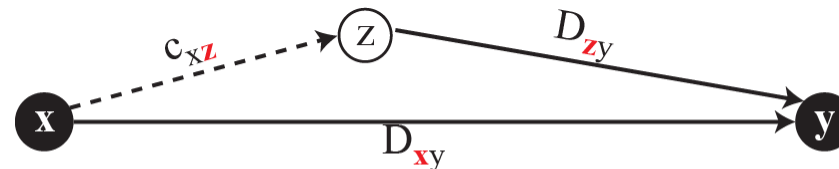


Bellman-Fords ekvation



a. General case with three intermediate nodes

$$D_{xy} = \min\{(c_{xa} + D_{ay}), (c_{xb} + D_{by}), (c_{xc} + D_{cy}) \dots\}$$



b. Updating a path with a new route

$$D_{xy} = \min\{D_{xy}, (c_{xz} + D_{zy})\}$$

Not! D_{xy} kan ändras utan att nod z tillkommit!



Tentafråga

NetID	Cost	Next Hop
A	3	rtrX
B	1	---
C	4	rtrY

Uppdatering från rtrY

NetID	Cost
A	1
C	2
D	4

Hur ser den nya routingtabellen ut efter uppdatering från rtrY?



Distance Vector, funderingar

- Periodiska uppdateringar!?
 - Hur hitta grannar?
 - Hur upptäcka att en granne försvinner?
- Problem med länkar och noder (bortom grannar) som försvinner.
 - Finns inget naturligt sätt att säga ”avbrott”

Mer i ETSF10 Internetprotokoll



Link State

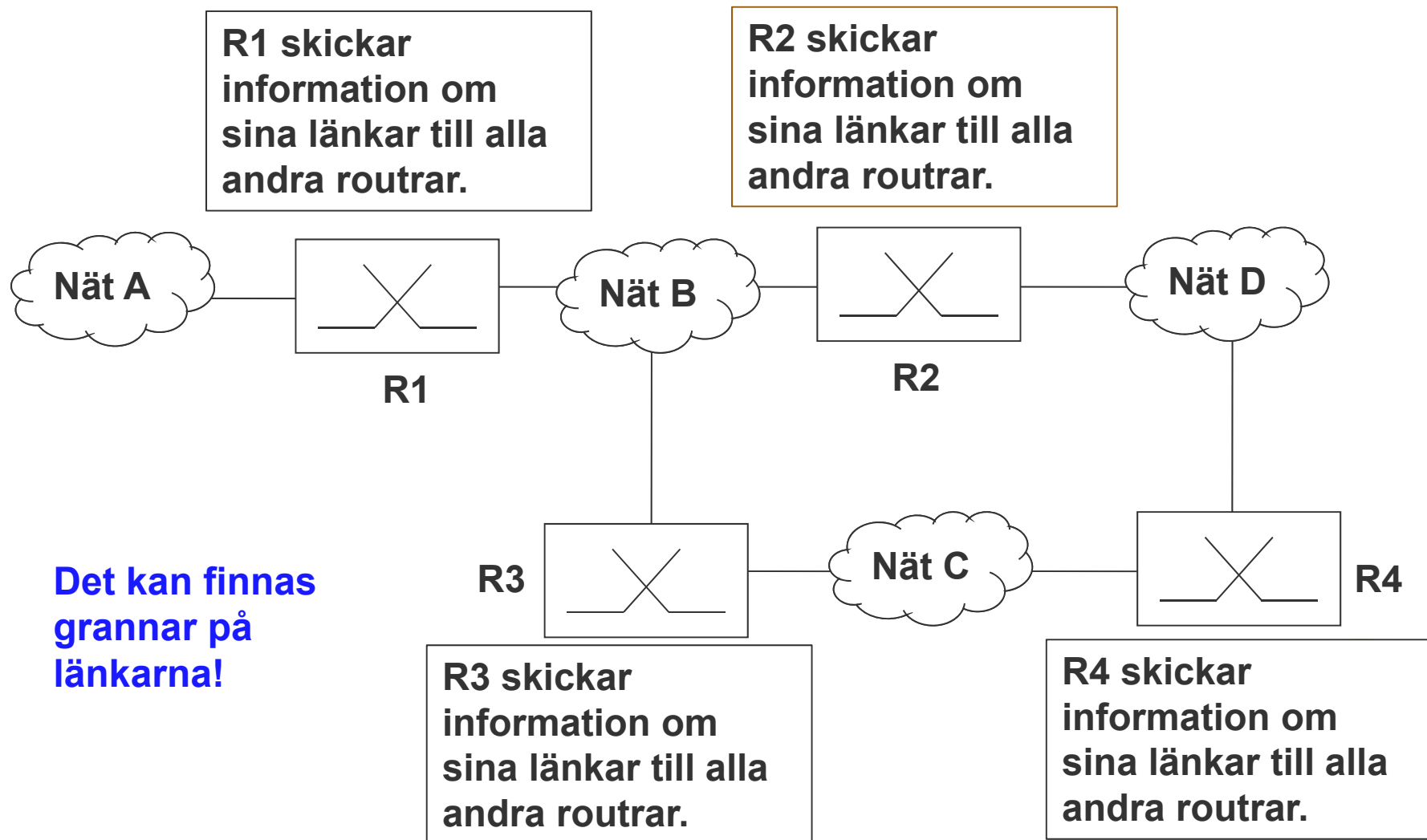


Link state: princip

- **Lokal topologi** info **flödas** globalt (LSA)
 - Vid lokal förändring
 - Periodiskt (i praktiken mycket sällan, typ varje halvtimme)
- Skapa databas i varje node med alla kända link states
- Uppdatera routing-tabell när ny information läggs in i databasen (Shortest Path First)
- "Lokal kunskap sprids globalt"



Link State: princip



LSA (*Link State Advertisement*)

Advertiser	Network ID	Cost	Neighbour



Link State Databas, exempel

Advertiser	Network ID	Cost	Neighbour
Rtr 1	Net A	8	---
Rtr 1	Net B	4	Rtr 2
Rtr 1	Net B	4	Rtr 3
Rtr 2	Net B	4	Rtr 1
Rtr 2	Net C	2	Rtr 4
Rtr 2	Net C	2	Rtr 5
Rtr 3	Net B	4	Rtr 2
Rtr 3	Net B	4	Rtr 2
Rtr 3	Net D	10	Rtr 5
.....

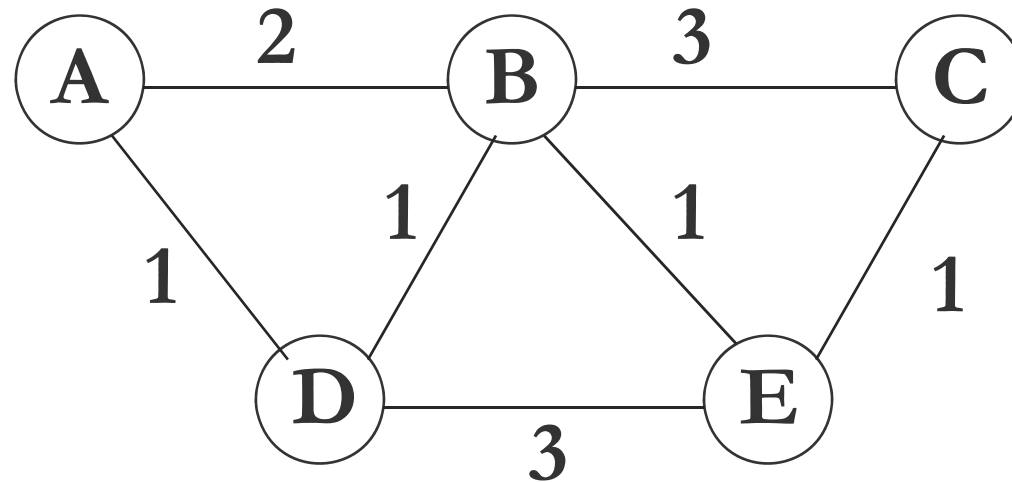


Dijkstras algoritim: Shortest Path First

1. Identify the root (the node itself)
2. Attach all neighbor nodes temporarily
3. Make link and node with least cumulative cost permanent
4. Choose this node
5. Repeat 2 and 3 until all nodes are permanent



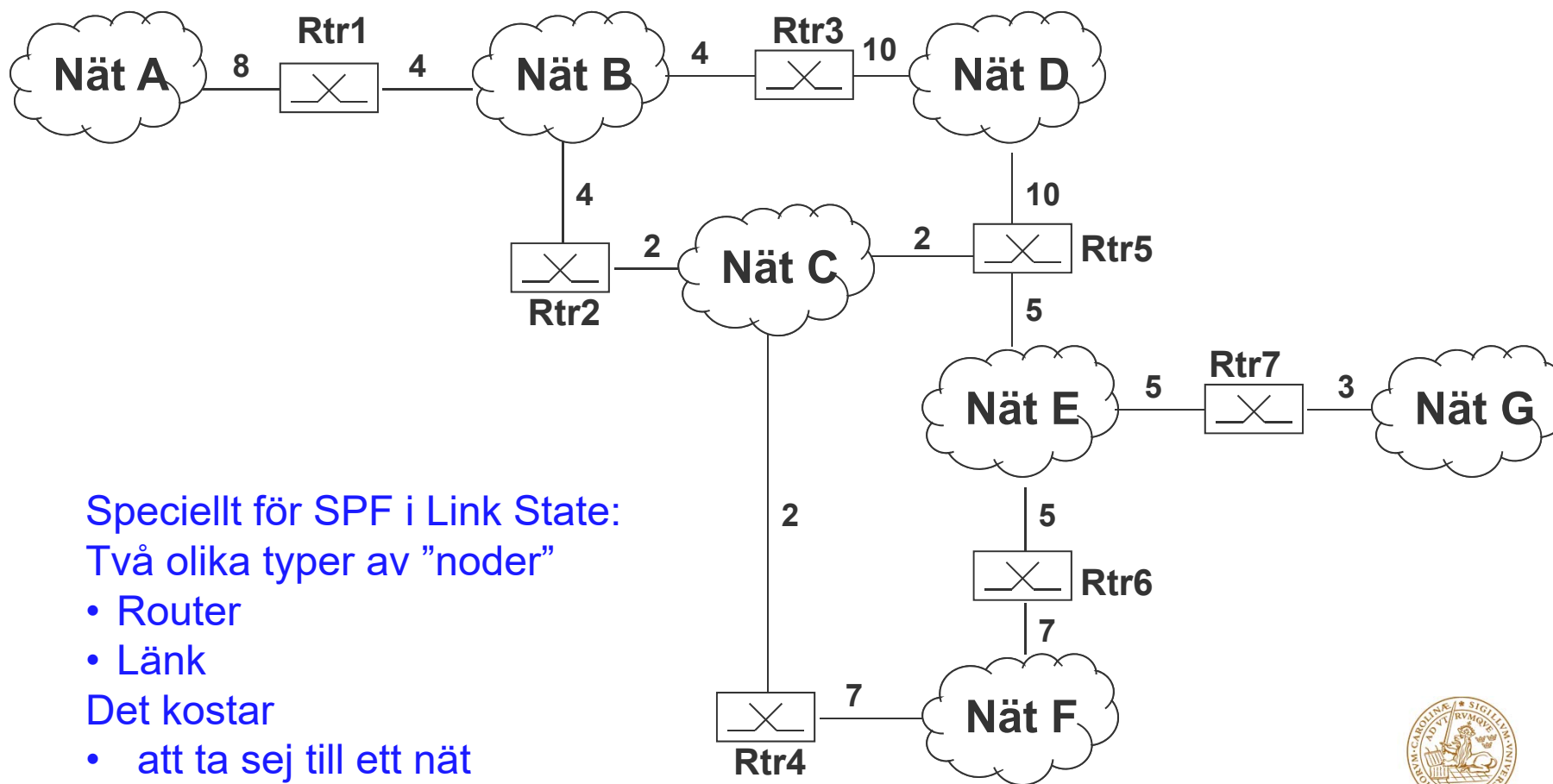
Tentafråga



Rita ett Shortest Path Fast-träd med Dijkstars algoritm utgående från nod A.



Link State: ett exempel



Speciellt för SPF i Link State:

Två olika typer av "noder"

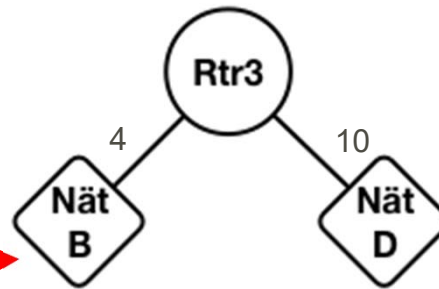
- Router
- Länk

Det kostar

- att ta sej till ett nät
- inget att ta sej från ett nät



SFP Rtr 3: steg 1

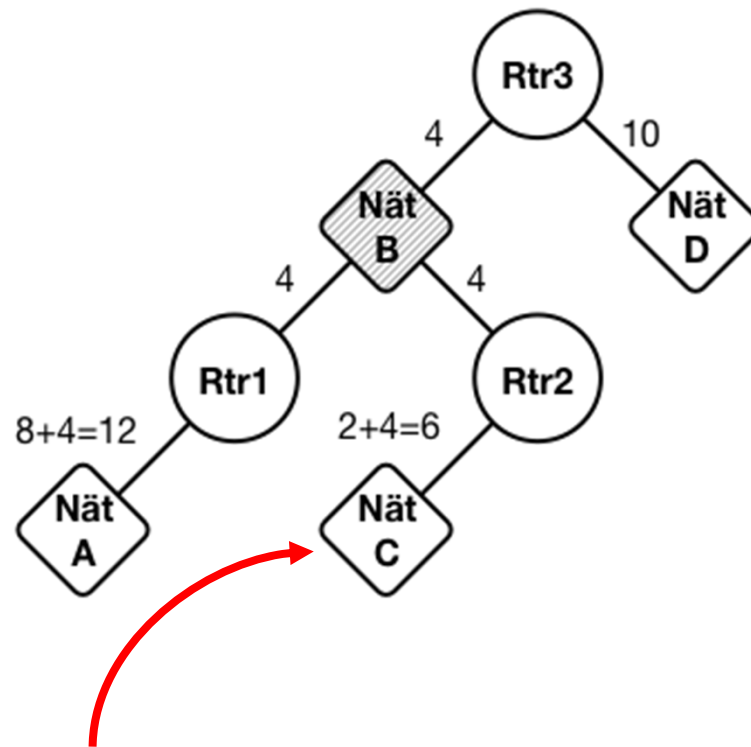


“Billigast” -> Permanent



LUND
UNIVERSITY

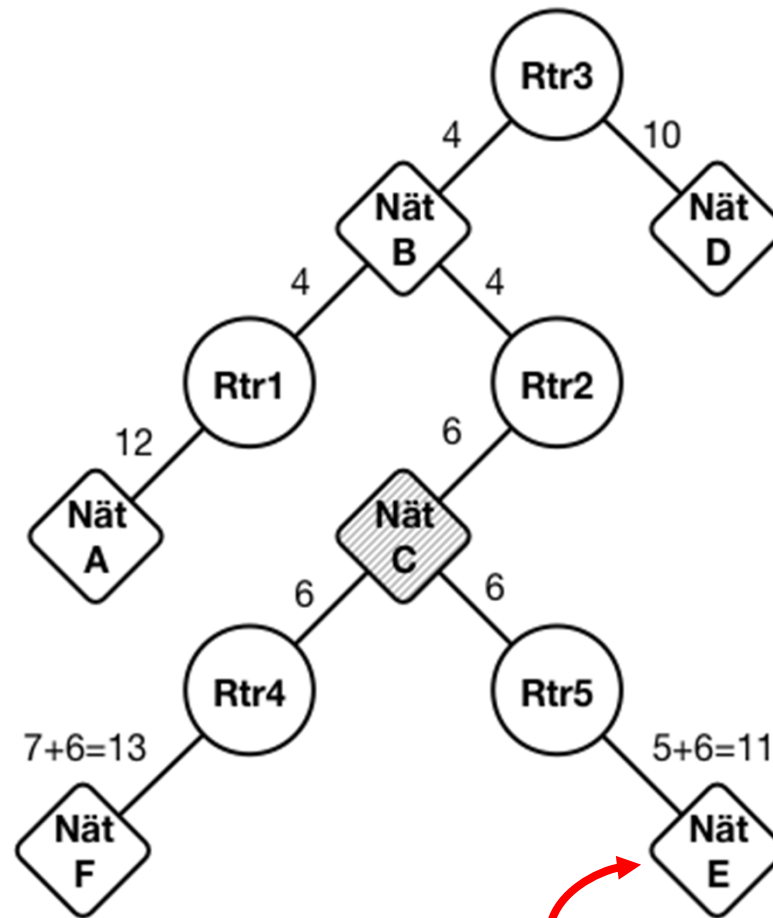
SFP Rtr 3: steg 2



“Billigast” -> Permanent



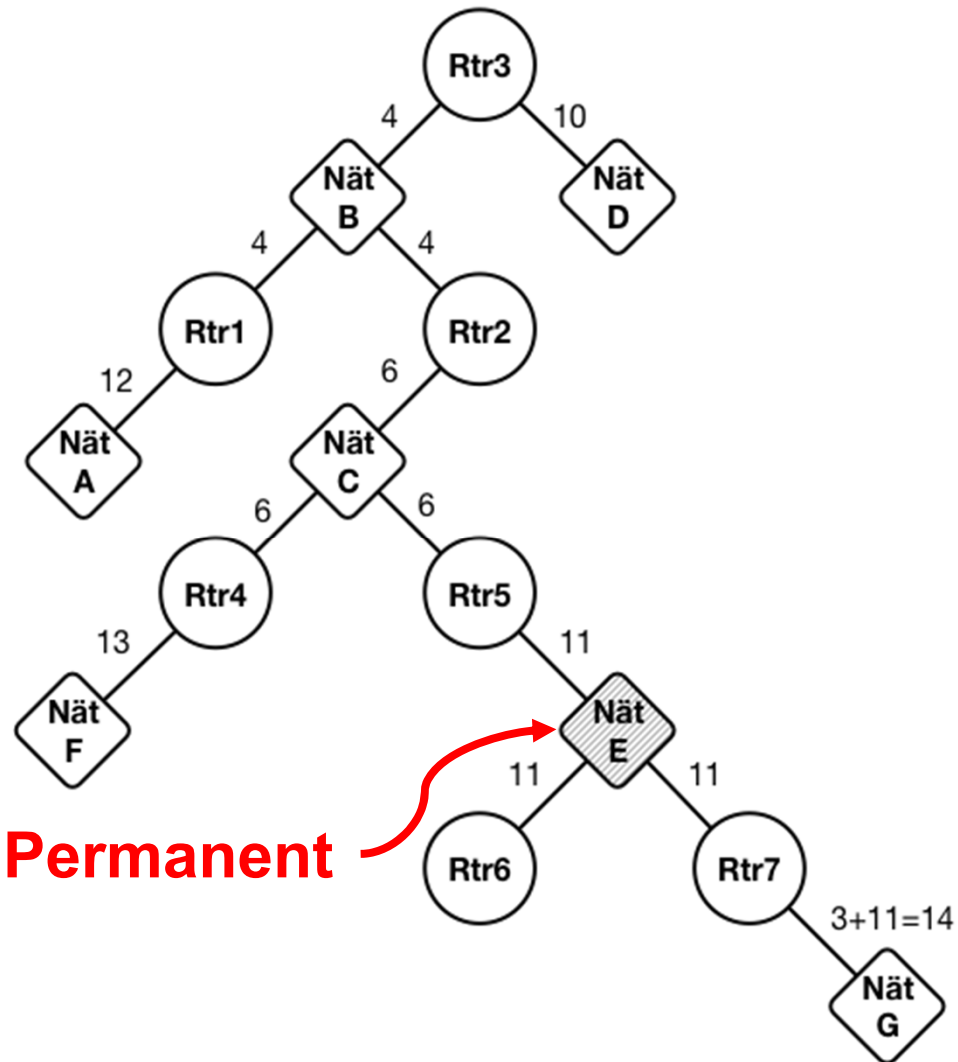
SFP Rtr 3: steg 3



“Billigast” -> Permanent



SFP Rtr 3: steg 4 (Slutlig)



“Billigast” -> Permanent



Routingtabell för rtr3

Network ID	Next-hop	Cost
Net A	Rtr1	12
Net B	-	4
Net C	Rtr2	6
Net D	-	10
Net E	Rtr2	11
Net F	Rtr2	13
Net G	Rtr2	14



Link State, funderingar

- Problem med länkar och noder som försvinner?
- Hur hitta grannar?
- Hur upptäcka att en granne försvinner?
- Periodiska uppdateringar!?

Mer i ETSF10 Internetprotokoll



LUND
UNIVERSITY

Tentafråga

NetID	Cost	Next Hop	Interface
10.0.10.0	2	17.16.12.254	r10
192.168.100.0	1	d.c.	r12
1.0.0.0	4	23.45.103.2	r13
0.0.0.0	3	17.16.12.254	r10

IP-paket med följande destination tas emot av router med denna routing-/forwarding-tabell. Vilken är nästa mottagare av paketet?

1. 1.0.0.10
2. 192.168.100.100
3. 130.235.200.53



Fråga: Kan två datorer ha samma IP-adress om de har olika nätmask?

- Alla IP-adresser måste vara unika
 - Masken enbart för att avgöra nät-id (*net id*)
- En IP adress består av nät-id och värd-id (*host id*)
- Gräns mellan nät-id och värd-id bestäms
 - Klassfullt
 - Klasslöst = nätmask
- 192.168.1.0/24 och 192.168.1.0/26 är två unika nät-id på två unika länkar
 - I routing gäller *longest match*
 - 192.168.1.0-192.168.1.63 kan därför inte dupliceras på 192.168.1.0/24



LUND
UNIVERSITY