

Routing algorithms

IP LAYER LAB

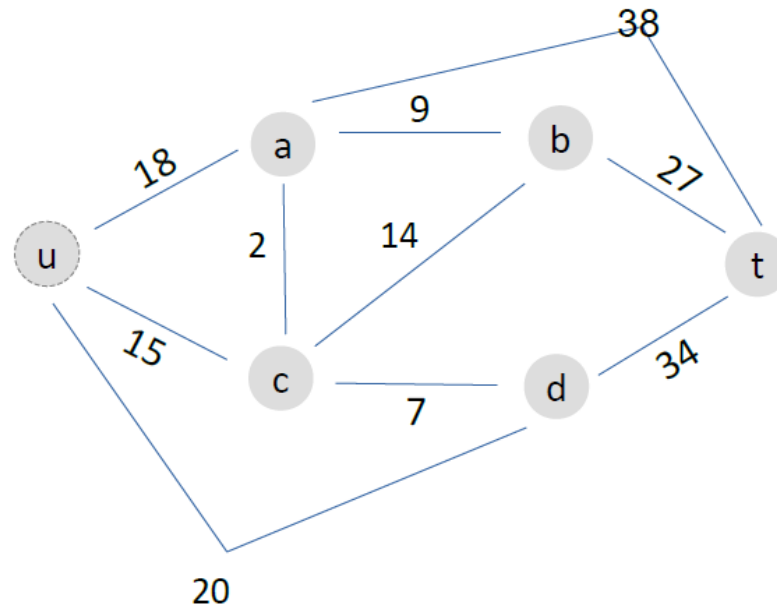
HY335A

EVRIPIDIS TZAMOUSIS

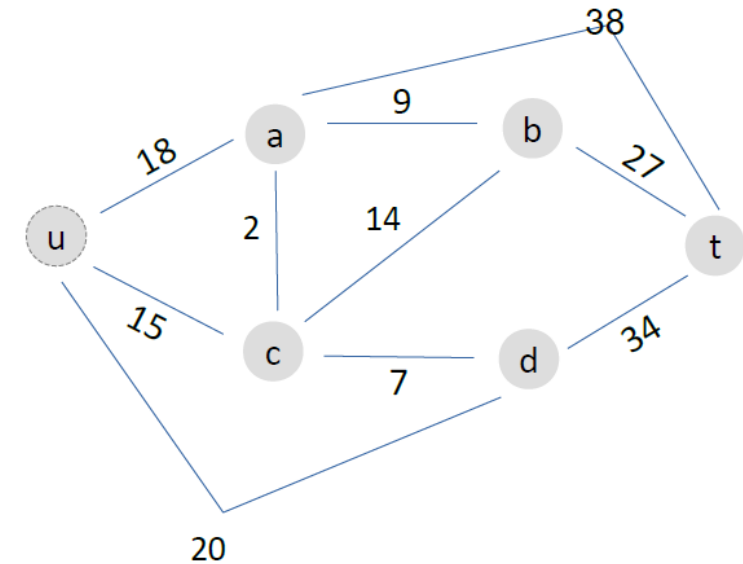
Αλγόριθμος Κατάστασης Ζεύξης

$D(v)$: κόστος της διαδρομής ελαχίστου κόστους από τον κόμβο προέλευσης στον v

$p(v)$: προηγούμενος κόμβος του v στο τρέχον μονοπάτι ελαχίστου κόστους

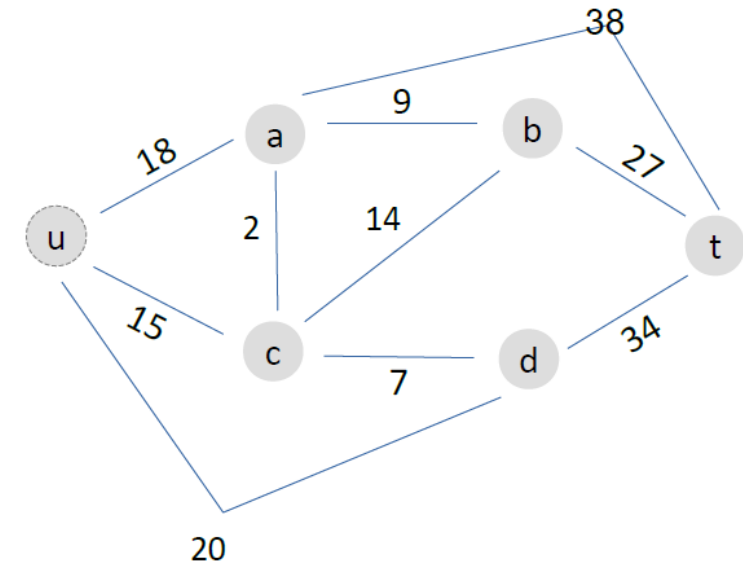


Αλγόριθμος Κατάστασης Ζεύξης



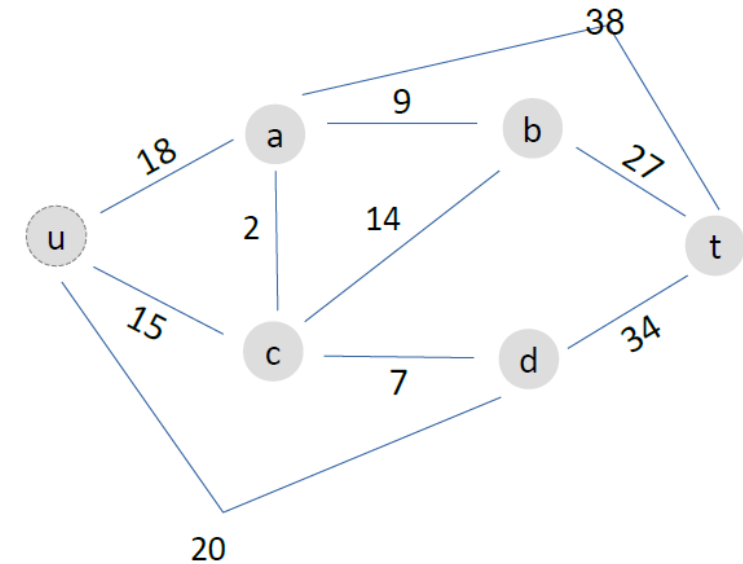
N'	D(c), p(c)	D(a), p(a)	D(d), p(d)	D(b), p(b)	D(t), p(t)
u	15, u	18, u	20, u	∞	∞

Αλγόριθμος Κατάστασης Ζεύξης



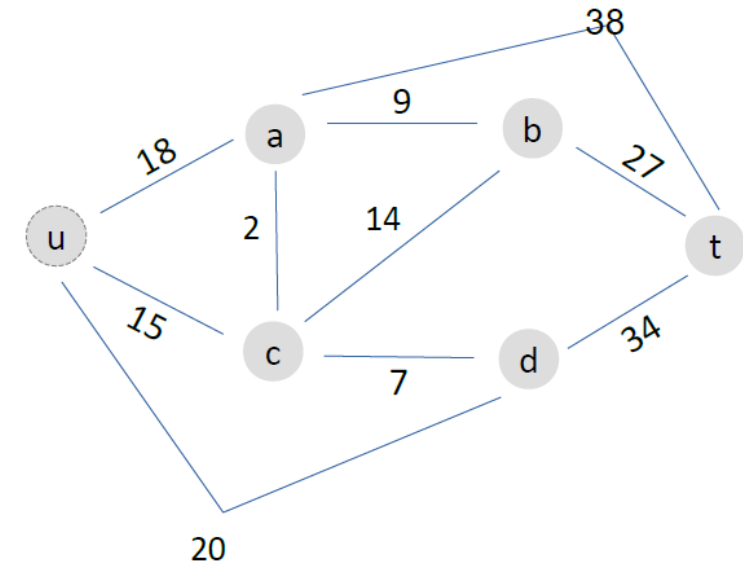
N'	D(c), p(c)	D(a), p(a)	D(d), p(d)	D(b), p(b)	D(t), p(t)
u	15, u	18, u	20, u	∞	∞
uc	-	17,c	20, u (22,c)	29,c	∞

Αλγόριθμος Κατάστασης Ζεύξης



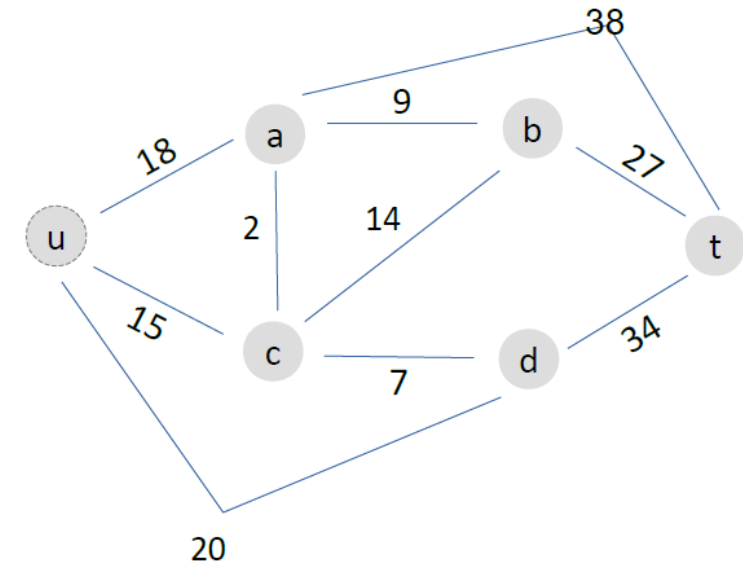
N'	D(c), p(c)	D(a), p(a)	D(d), p(d)	D(b), p(b)	D(t), p(t)
u	15, u	18, u	20, u	∞	∞
uc	-	17,c	20, u	29,c	∞
uca	-	-	20, u	26, a	55, a

Αλγόριθμος Κατάστασης Ζεύξης



N'	D(c), p(c)	D(a), p(a)	D(d), p(d)	D(b), p(b)	D(t), p(t)
u	15, u	18, u	20, u	∞	∞
uc	-	17,c	20, u	29,c	∞
uca	-	-	20, u	26, a	55, a
ucad	-	-	-	26, a	54, d

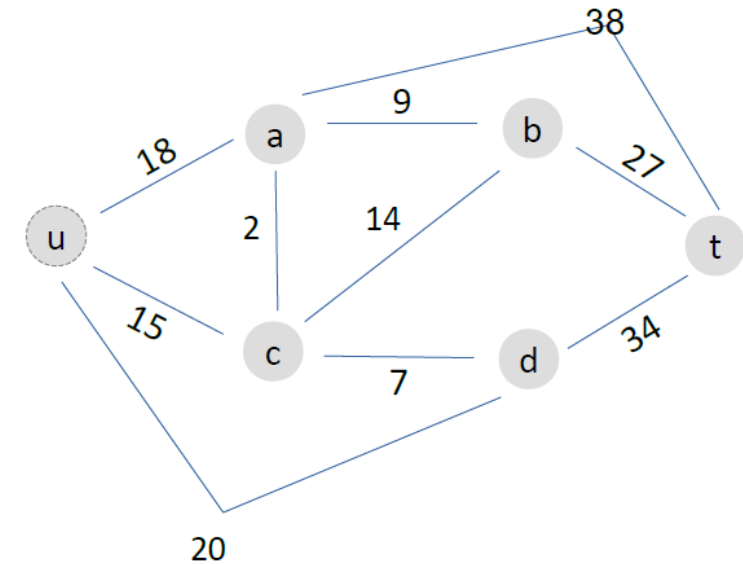
Αλγόριθμος Κατάστασης Ζεύξης



N'	D(c), p(c)	D(a), p(a)	D(d), p(d)	D(b), p(b)	D(t), p(t)
u	15, u	18, u	20, u	∞	∞
uc	-	17,c	20, u	29,c	∞
uca	-	-	20, u	26, a	55, a
ucad	-	-	-	26, a	54, d
ucadb	-	-	-	-	53, b
ucadbt	-	-	-	-	-

Αλγόριθμος Κατάστασης Ζεύξης

Μονοπάτι για t: $u \rightarrow c \rightarrow a \rightarrow b \rightarrow t$



N'	D(c), p(c)	D(a), p(a)	D(d), p(d)	D(b), p(b)	D(t), p(t)
u	15, u	18, u	20, u	∞	∞
uc	-	17,c	20, u	29,c	∞
uca	-	-	20, u	26, a	55, a
ucad	-	-	-	26, a	54, d
ucadb	-	-	-	-	53, b
ucadbt	-	-	-	-	-

Αλγόριθμος Απόστασης Διανύσματος

Τα χαρακτηριστικά του:

- Επαναληπτικός
- Ασύγχρονος
- Κατανεμημένος

Οι πληροφορίες που διατηρεί ο κάθε κόμβος x :

- $c(x,v)$, για κάθε γείτονα v
- D_x με τα εκτιμώμενα κόστη προς όλους τους κόμβους του δικτύου
- D_v , για κάθε γείτονα v

Στάδιο αρχικοποίησης για κάθε κόμβο x

Αρχικοποίηση:

Για όλους τους κόμβους y μέσα στο N :

Αν y είναι γείτονας

$$D_x(y) = c(x, y)$$

Αλλιώς

$$D_x(y) = \infty$$

Για όλους τους γείτονες v του x :

$D_v(y) = \infty$ για όλους τους προορισμούς y

Για όλους τους γείτονες v του x :

στείλε το διάνυσμα απόστασης D_x

Επαναληπτική διαδικασία για κάθε κόμβο x

Επαναληπτικά:

Περίμενε μέχρι να λάβεις μία ενημέρωση από ένα γείτονα)

Για όλους τους κόμβους y μέσα στο N :

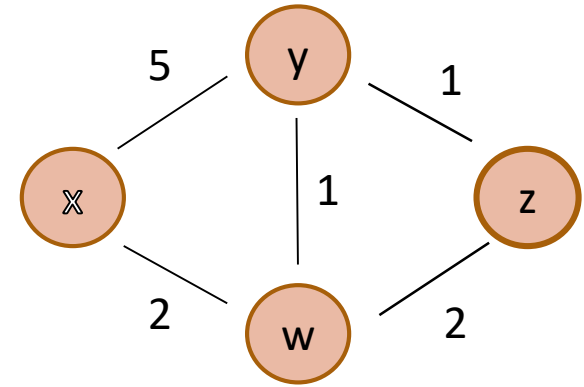
$$D_x(y) = \min_v \{ c(x,v) + D_v(y) \}$$

(βρες τον γείτονα μέσω του οποίου θα πας στο y πιο γρήγορα)

Αν το D_x άλλαξε:

Στείλε το νέο D_x σε όλους τους γείτονες

Στάδιο αρχικοποίησης



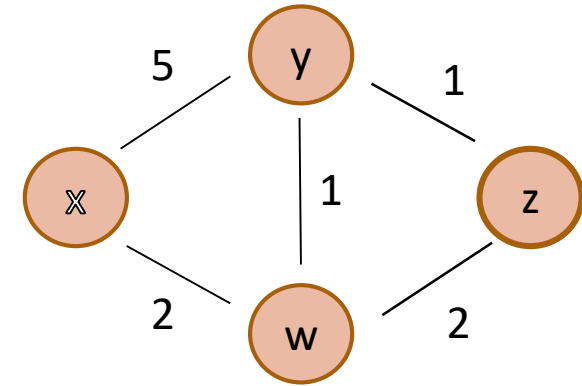
	x	y	w	z
x	0	5	2	∞
y	∞	∞	∞	∞
w	∞	∞	∞	∞

	x	y	w	z
x	∞	∞	∞	∞
y	5	0	1	1
w	∞	∞	∞	∞
z	∞	∞	∞	∞

	x	y	w	z
x	∞	∞	∞	∞
y	∞	∞	∞	∞
w	2	1	0	2
z	∞	∞	∞	∞

	x	y	w	z
y	∞	∞	∞	∞
w	∞	∞	∞	∞
z	∞	1	2	0

Στάδιο αρχικοποίησης



	x	y	w	z
x	0	5	2	∞
y	5	0	1	1
w	2	1	0	2

Ο **x** θα ενημερώσει: **y, w**

	x	y	w	z
x	0	5	2	∞
y	5	0	1	1
w	2	1	0	2
z	∞	1	2	0

Ο **y** θα ενημερώσει: **x, w, z**

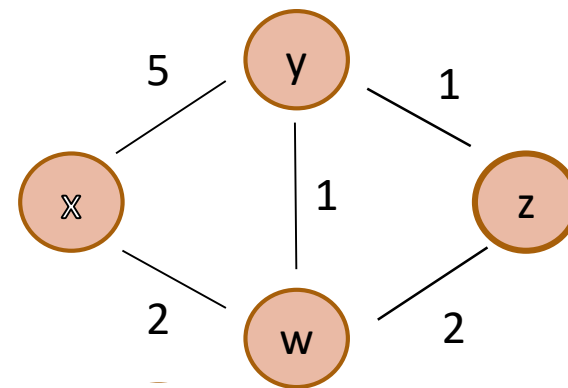
	x	y	w	z
x	0	5	2	∞
y	5	0	1	1
w	2	1	0	2
z	∞	1	2	0

Ο **w** θα ενημερώσει: **x, y, z**

	x	y	w	z
y	5	0	1	1
w	2	1	0	2
z	∞	1	2	0

Ο **z** θα ενημερώσει: **y, w**

Υπολογισμός νέων διανυσμάτων



	x	y	w	z
x	0	<u>3</u>	2	<u>4</u>
y	5	0	1	1
w	2	1	0	2

	x	y	w	z
x	0	5	2	∞
y	<u>5</u>	0	1	1
w	2	1	0	2
z	∞	1	2	0

	x	y	w	z
x	0	5	2	∞
y	5	0	1	1
w	2	1	0	2
z	∞	1	2	0

	x	y	w	z
y	5	0	1	1
w	2	1	0	2
z	<u>4</u>	1	2	0

$$D_x(y) = c(x,y) + D_y(y) = 5 + 0 = 5$$

$$D_x(y) = c(x,w) + D_w(y) = 2 + 1 = 3$$

$$D_x(w) = c(x,w) + D_w(w) = 2 + 0 = 2$$

$$D_x(w) = c(x,y) + D_y(w) = 5 + 1 = 6$$

$$D_x(z) = c(x,y) + D_y(z) = 5 + 1 = 6$$

$$D_x(z) = c(x,w) + D_w(z) = 2 + 2 = 4$$

$$D_y(x) = c(y,x) + D_x(x) = 5 + 0 = 5$$

$$D_y(x) = c(y,w) + D_w(x) = 1 + 2 = 3$$

$$D_y(x) = c(y,z) + D_z(x) = 2 + \infty = \infty$$

$$D_y(w) = c(y,w) + D_w(w) = 1 + 0 = 1$$

$$D_y(w) = c(y,x) + D_x(w) = 5 + 2 = 7$$

$$D_y(w) = c(y,z) + D_z(w) = 1 + 2 = 3$$

$$D_y(z) = c(y,z) + D_z(z) = 1 + 0 = 1$$

$$D_y(z) = c(y,w) + D_w(z) = 1 + 2 = 3$$

$$D_y(z) = c(y,x) + D_x(z) = 1 + \infty = \infty$$

$$D_w(x) = c(w,x) + D_x(x) = 2 + 0 = 2$$

$$D_w(x) = c(w,y) + D_y(x) = 1 + 5 = 6$$

$$D_w(x) = c(w,z) + D_z(x) = 2 + \infty = \infty$$

$$D_w(y) = c(w,x) + D_x(y) = 2 + 5 = 7$$

$$D_w(y) = c(w,y) + D_y(y) = 1 + 0 = 1$$

$$D_w(y) = c(w,z) + D_z(y) = 2 + 1 = 3$$

$$D_w(z) = c(w,x) + D_x(z) = 2 + \infty = \infty$$

$$D_w(z) = c(w,y) + D_y(z) = 1 + 1 = 2$$

$$D_w(z) = c(w,z) + D_z(z) = 2 + 0 = 2$$

$$D_z(x) = c(z,y) + D_y(x) = 1 + 5 = 6$$

$$D_z(x) = c(z,w) + D_w(x) = 2 + 2 = 4$$

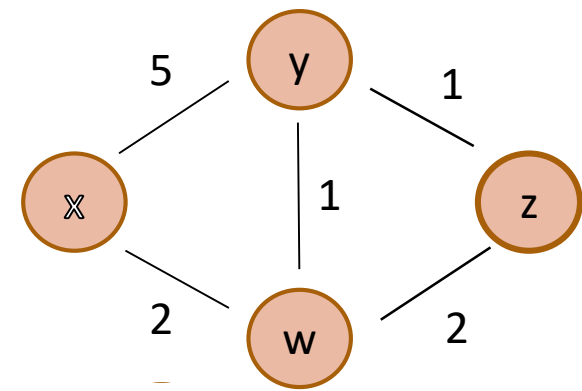
$$D_z(y) = c(z,y) + D_y(y) = 1 + 0 = 1$$

$$D_z(y) = c(z,w) + D_w(y) = 2 + 1 = 3$$

$$D_z(w) = c(z,y) + D_y(z) = 1 + 1 = 2$$

$$D_z(w) = c(z,w) + D_w(z) = 2 + 0 = 2$$

Αποστολή ενημερώσεων



	x	y	w	z
x	0	3	2	4
y	3	0	1	1
w	2	1	0	2

	x	y	w	z
x	0	3	2	4
y	3	0	1	1
w	2	1	0	2
z	4	1	2	0

	x	y	w	z
x	0	3	2	4
y	3	0	1	1
w	2	1	0	2
z	4	1	2	0

	x	y	w	z
y	3	0	1	1
w	2	1	0	2
z	4	1	2	0

Ο **x** θα ενημερώσει: **y, w**

Ο **y** θα ενημερώσει: **x, w, z**

Ο **w** δεν ενημερώνει κανένα

Ο **z** θα ενημερώσει: **y, w**

IP fragmentation

- **Maximum Transmission Unit (MTU)** defines **the largest packet size** that can traverse this path **without suffering fragmentation**
- **If an IP datagram has size larger than the MTU, then it is fragmented into smaller pieces before it is sent.**

Example: Suppose we want to transmit an IP datagram of size 3000 bytes through a link of MTU 500 bytes. How many fragments are produced and what are the values of the offset field in each of the headers?

Example

Example: Suppose we want to transmit an IP datagram of size 3000 bytes through a link of MTU 500 bytes. How many fragments are produced and what are the values of the offset field in each of the headers?

IP fragment payload = 500 bytes (MTU) - 20 bytes (min IPv4 header) = 480 bytes

IP datagram of interest payload = 3000 - 20 = 2980 bytes

Total # of segments = IP datagram of interest payload / IP fragment payload
= 2980 / 480
= 6.2
= 7 (The last packet will have smaller payload than the available 480 bytes)

Example

Example: Suppose we want to transmit an IP datagram of size 3000 bytes through a link of MTU 500 bytes. How many fragments are produced and what are the values of the offset field in each of the headers?

What about the header fields?

Segment 0:	0 - 479 bytes of original	offset = 0	more = 1
Segment 1:	480 - 959	offset = 60	more = 1
Segment 2:	960 - 1439	offset = 120	more = 1
Segment 3:	1440 - 1919	offset = 180	more = 1
Segment 4:	1920 - 2399	offset = 240	more = 1
Segment 5:	2400 - 2879	offset = 300	more = 1
Segment 6:	2880 - 2980	offset = 360	more = 0



NAT

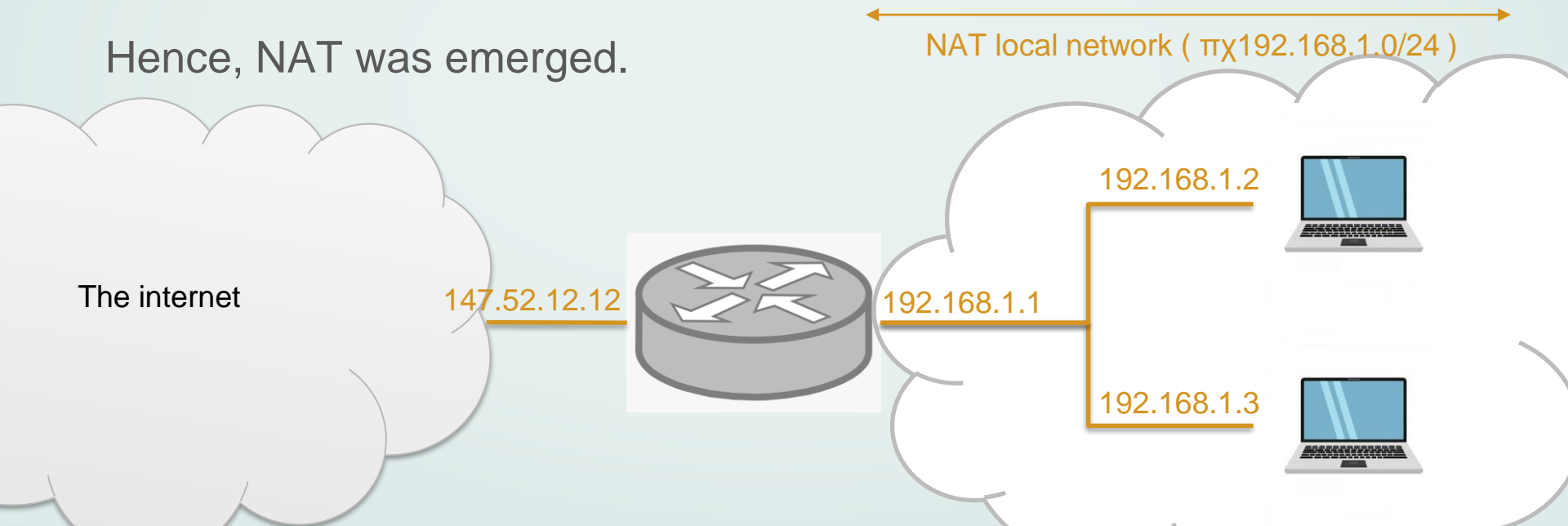
Arakadakis Konstantinos

NAT

IP addressing management within a network should be flexible.

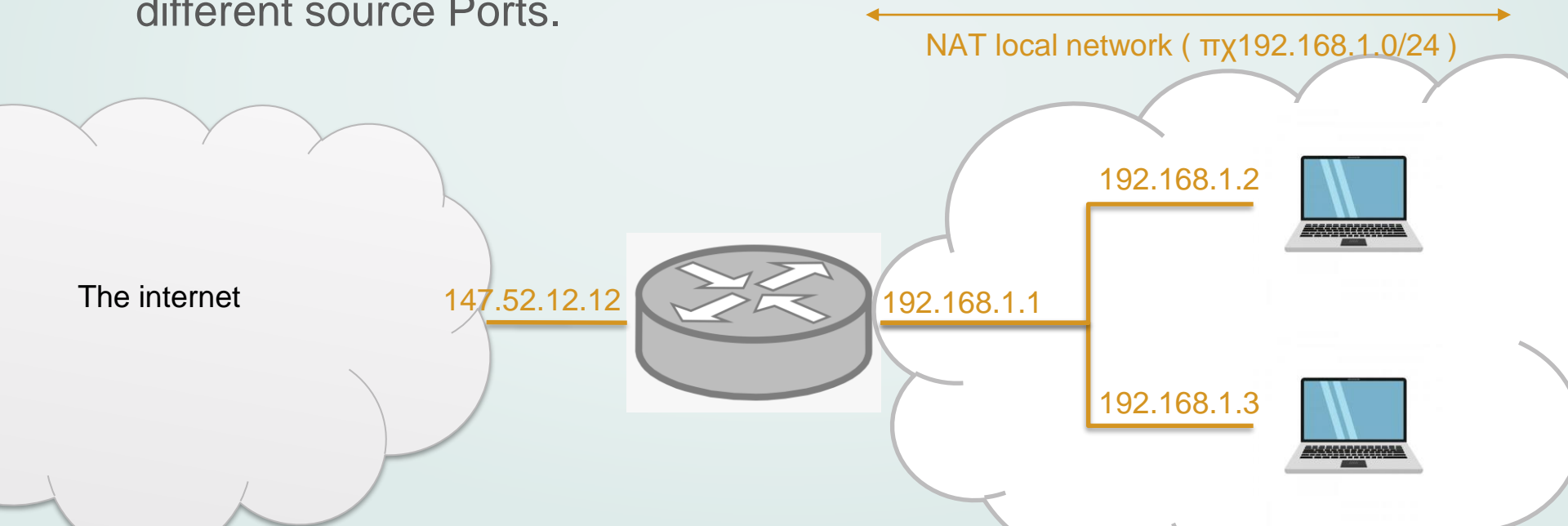
- Can support growing number of machines.
- No need to comply with global addressing standards.

Hence, NAT was emerged.



NAT

All datagrams leaving the network (originated from a host of the local network) appear to have the same IP (147.52.12.12), but with different source Ports.

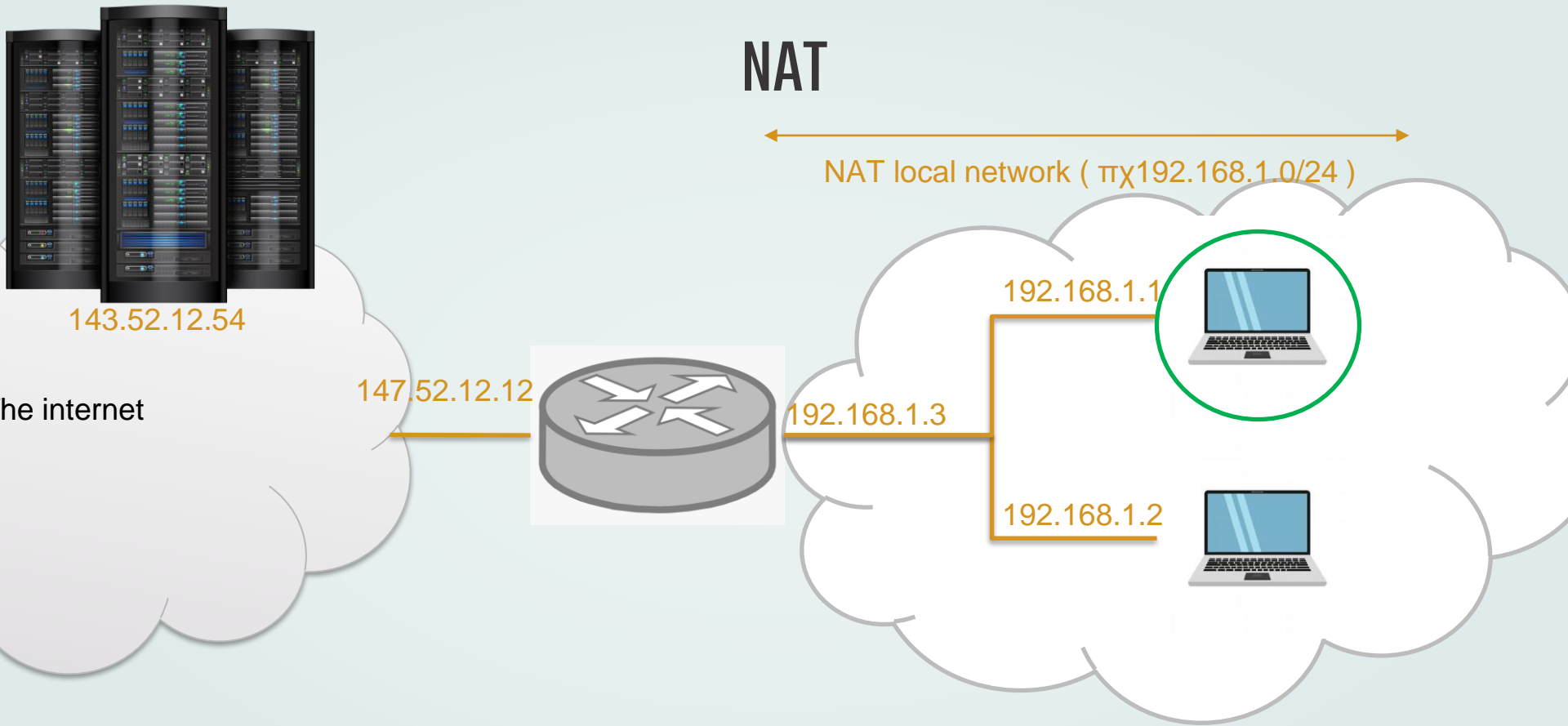


NAT

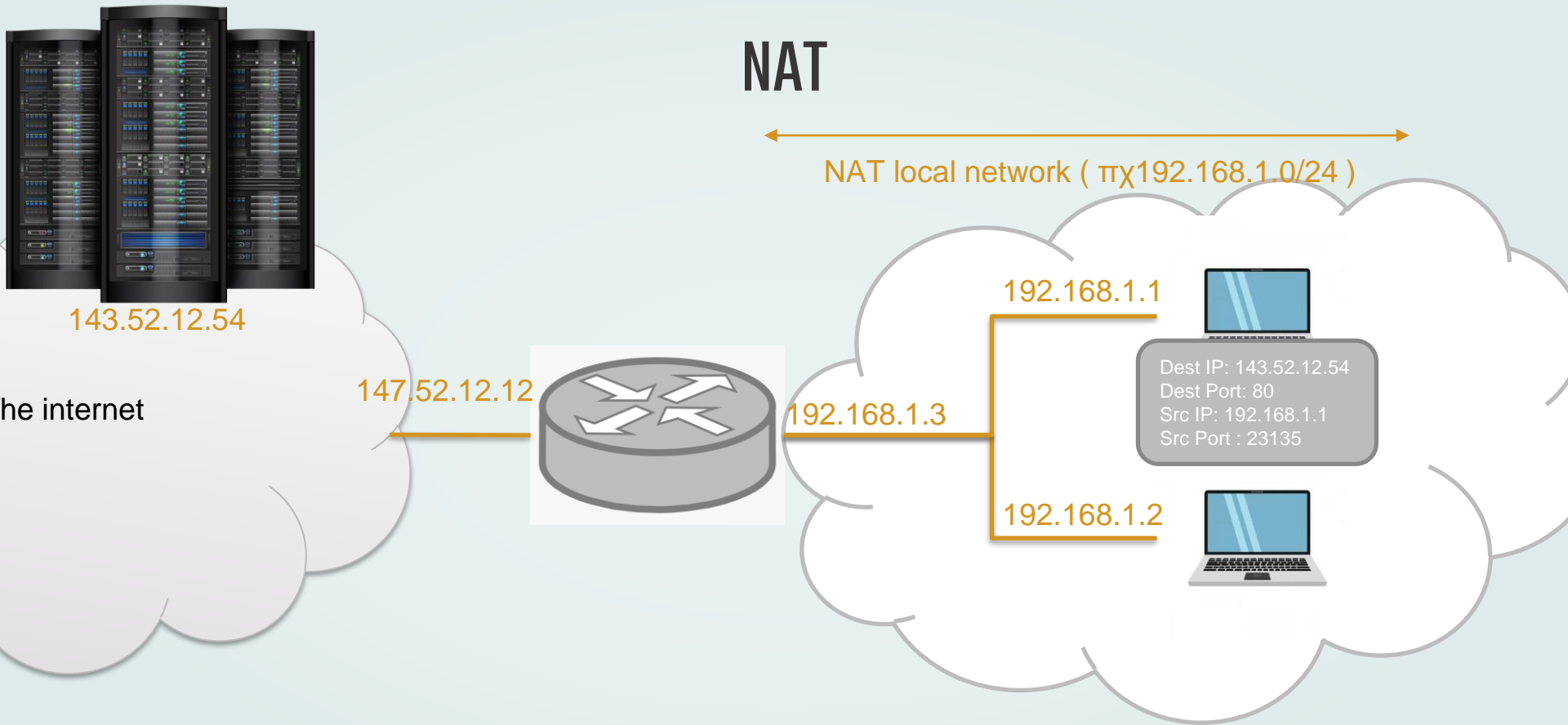
The process of a NAT enabled router:

- Replace the (source IP, source Port) for **outgoing** datagrams to (NAT IP, new source Port)
- The NAT enabled router must store this mapping of pairs.
(source IP, source Port) ~ (NAT IP, NAT Port)
- Replace the (destination IP, destination Port) for **ingoing** datagrams to the corresponding pair of source address and port.

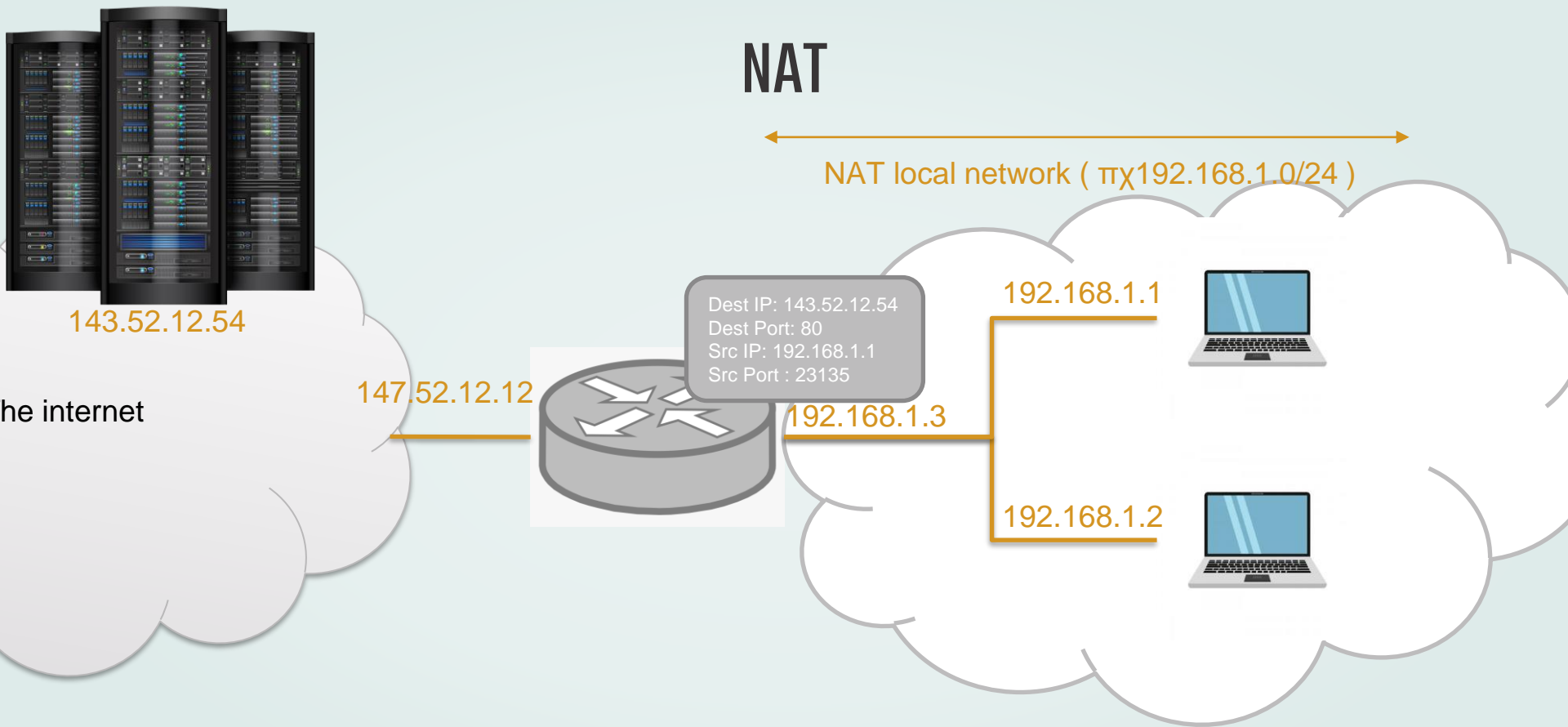
NAT



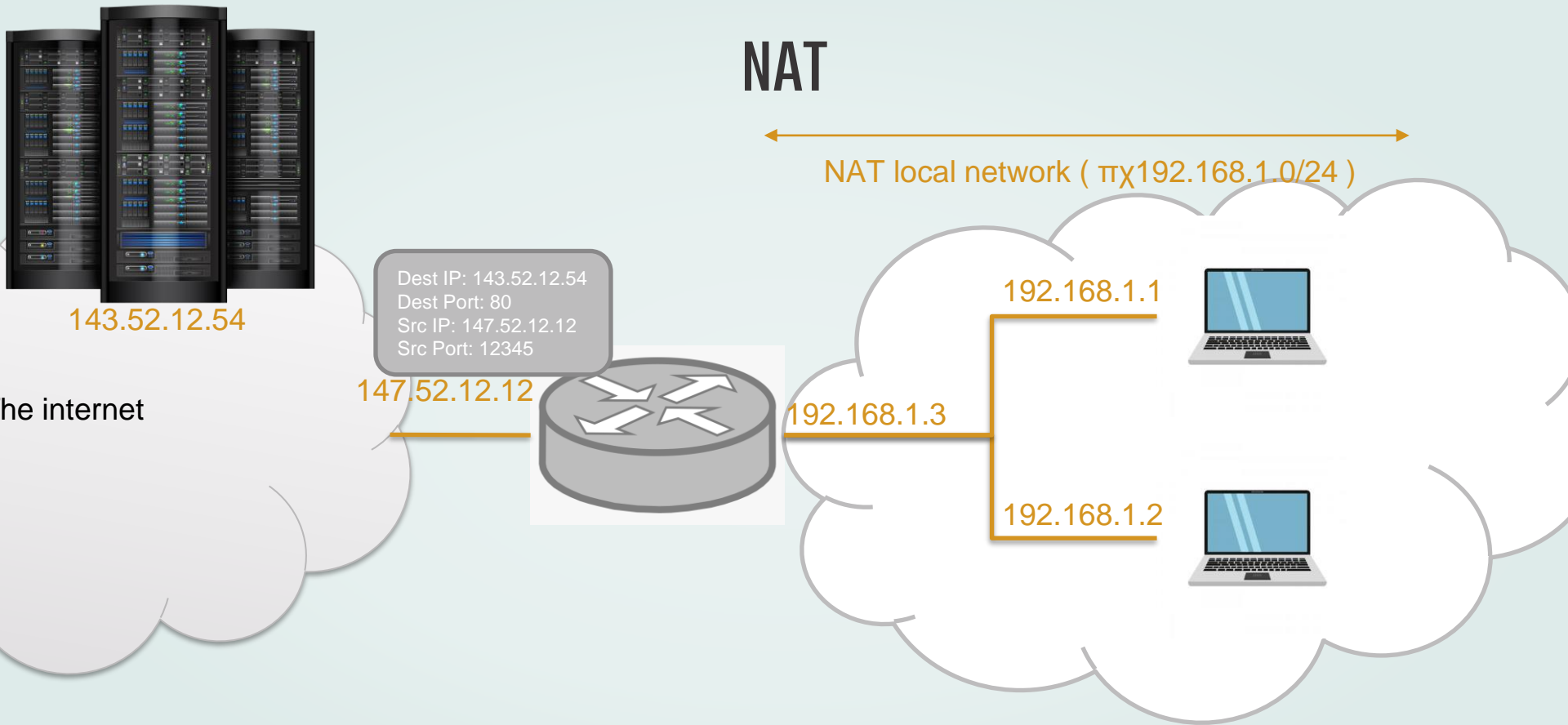
NAT

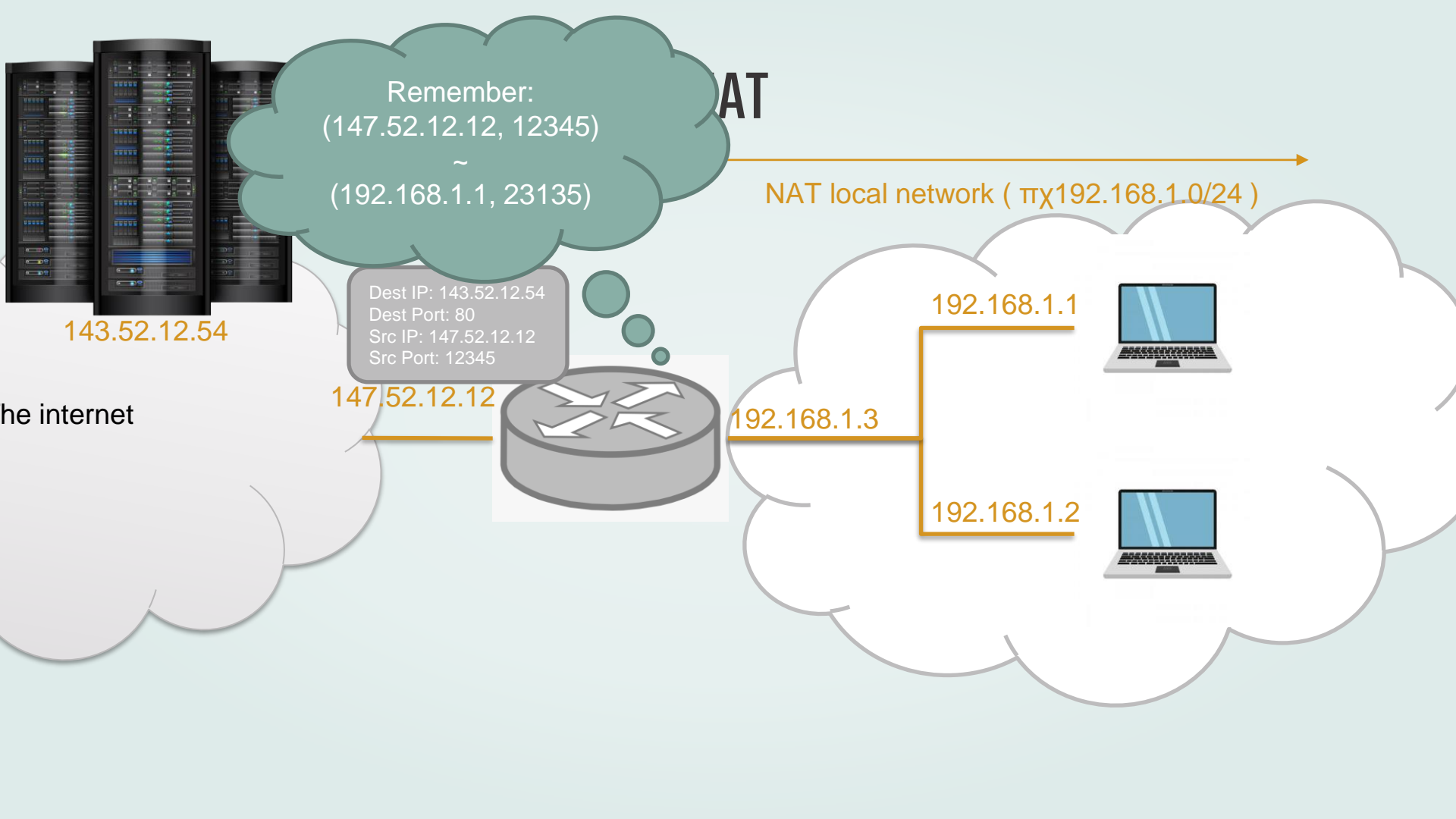


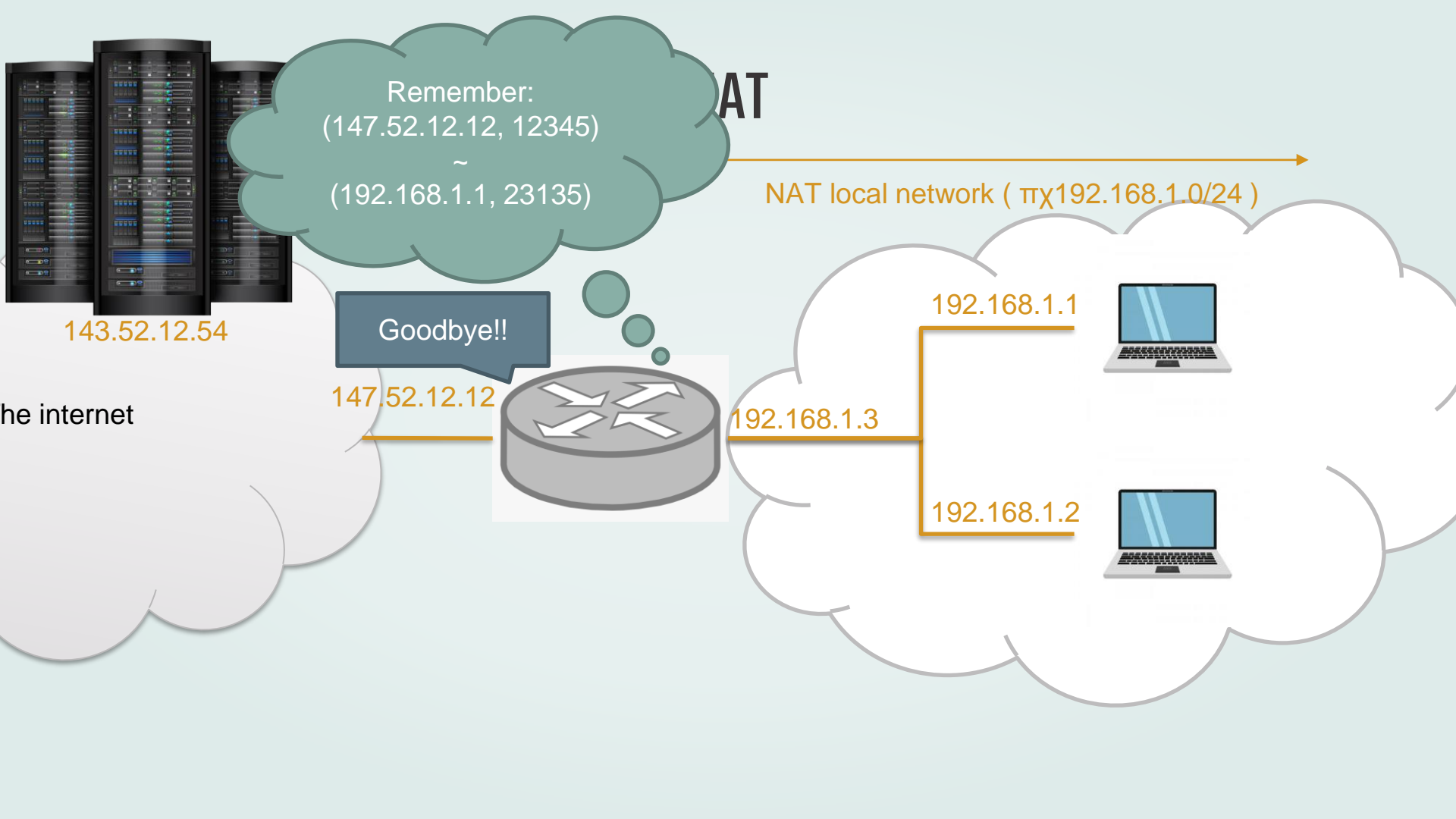
NAT

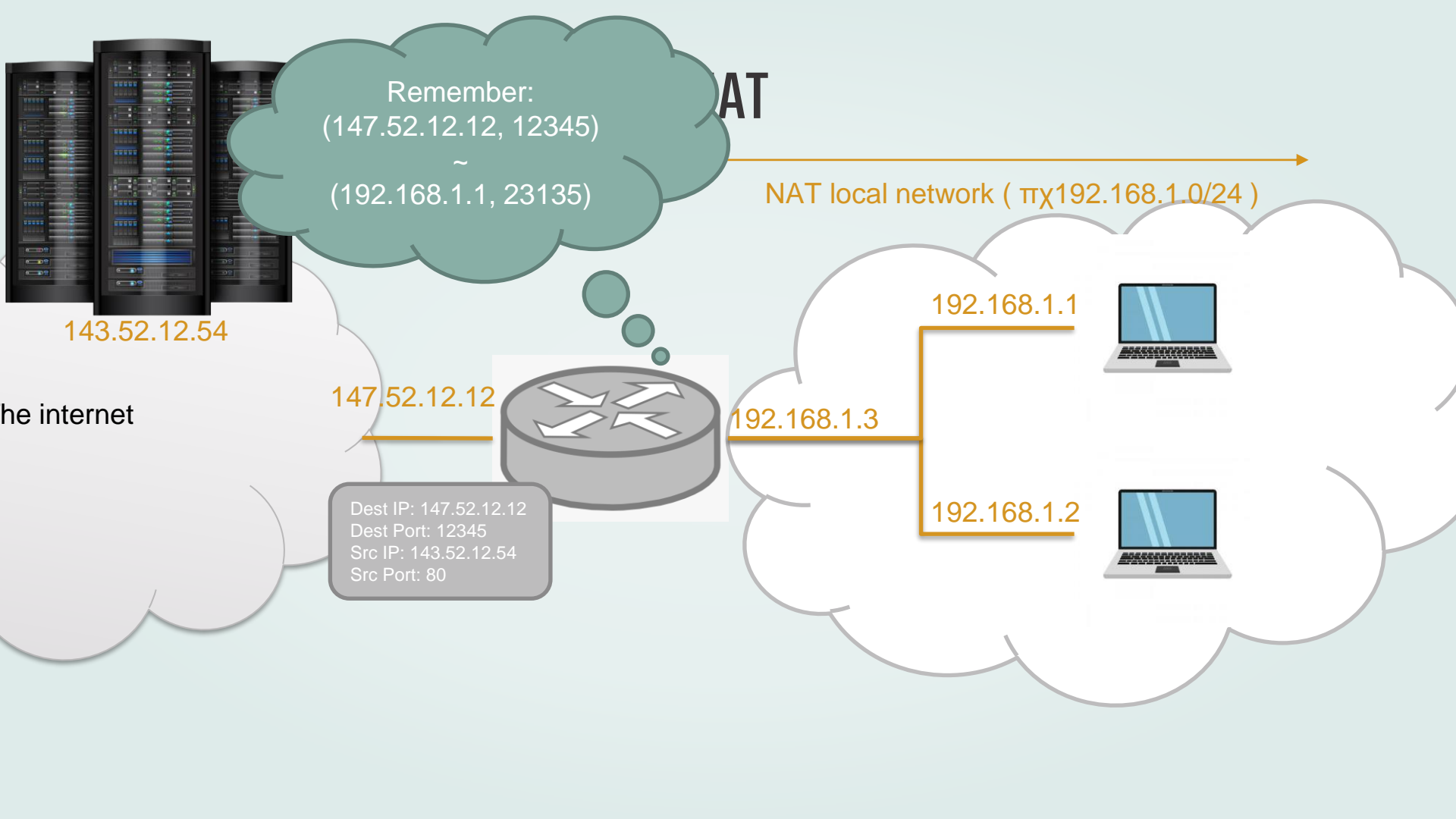


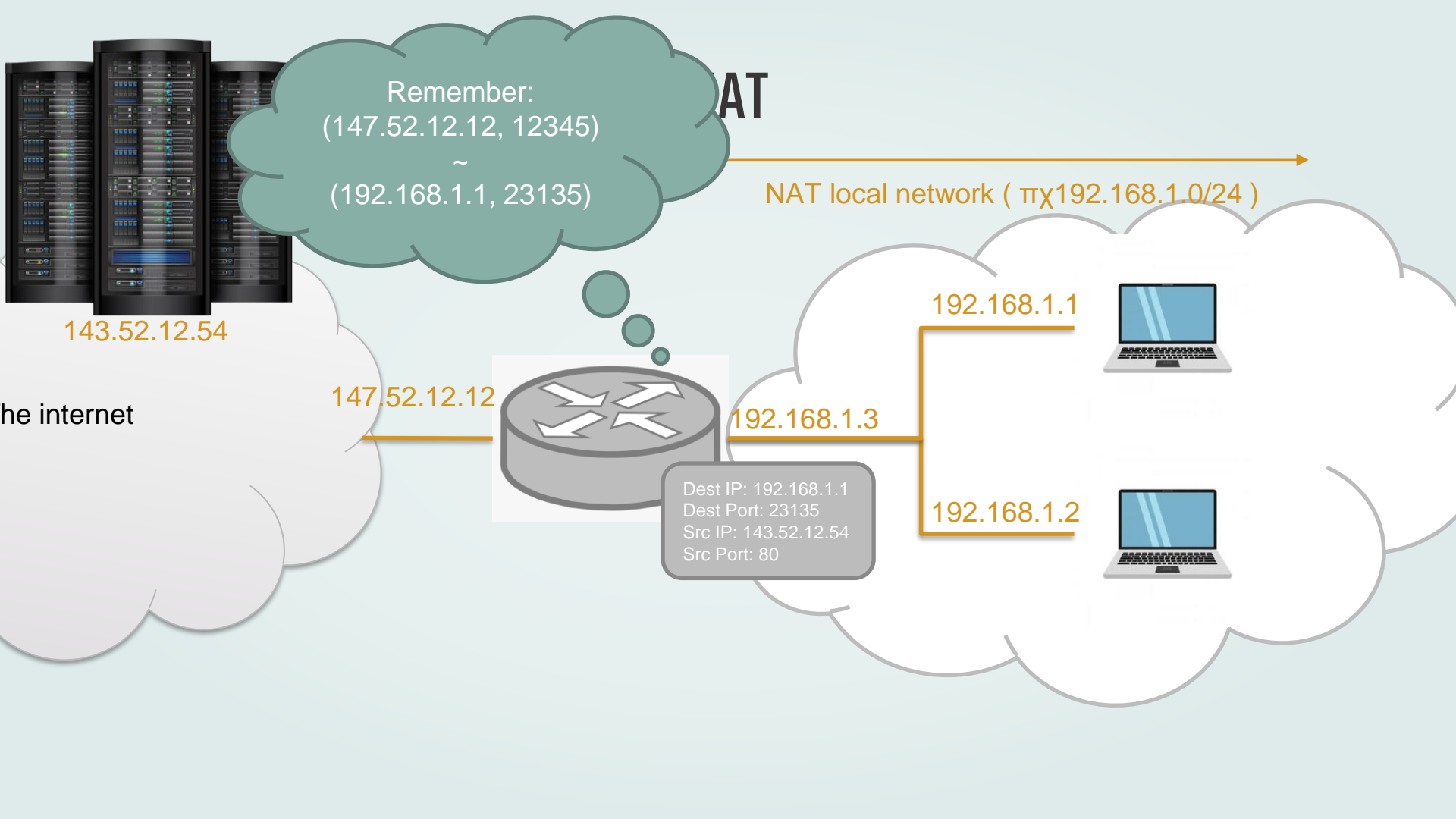
NAT



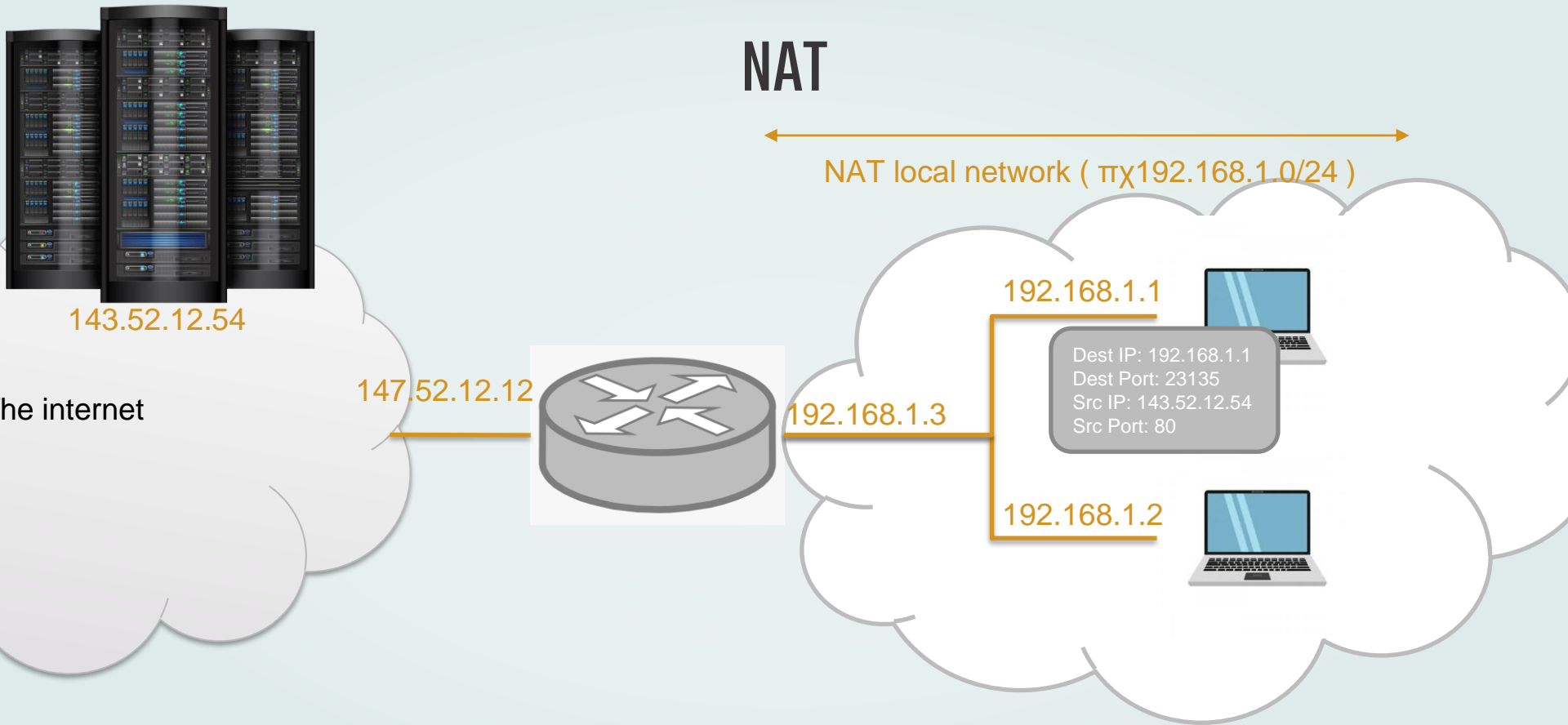




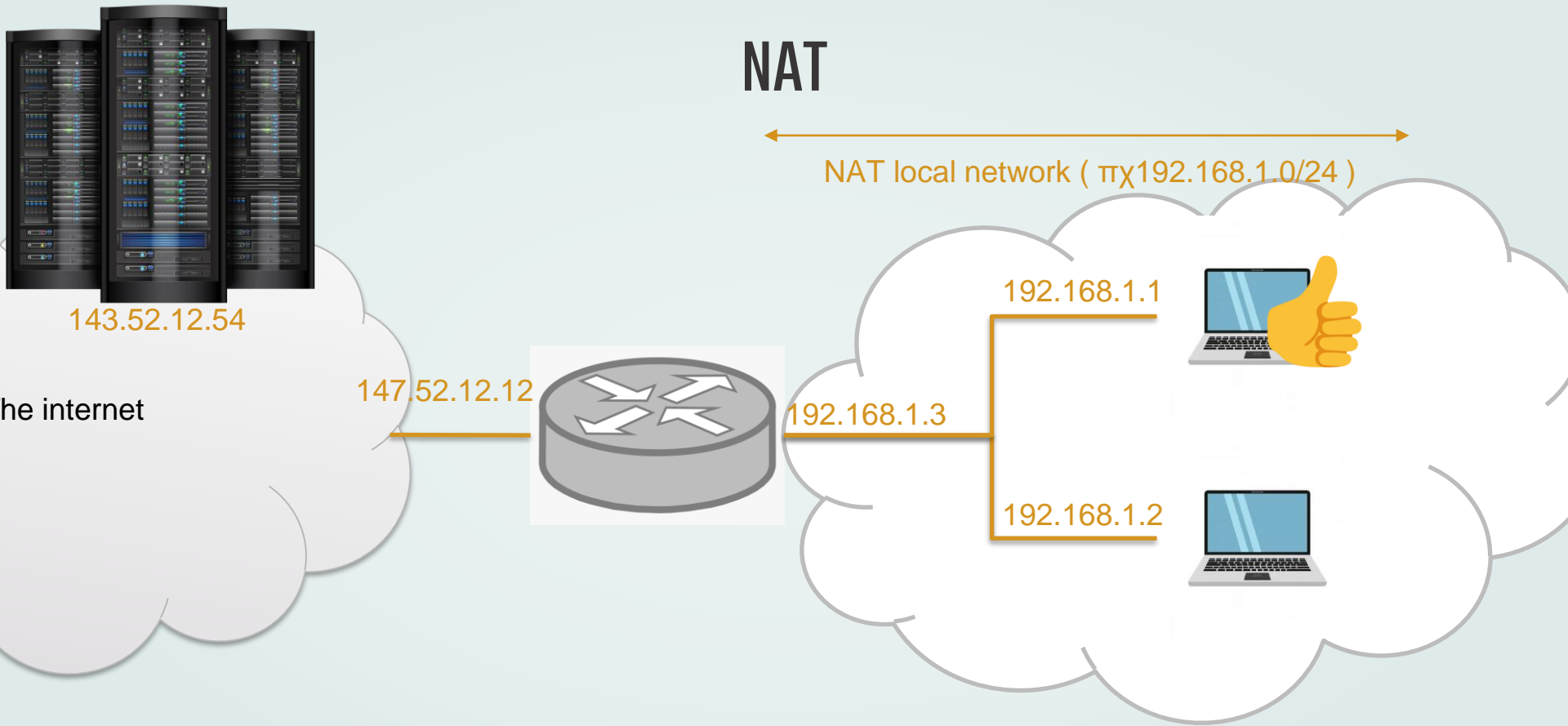




NAT



NAT



Longest prefix match

longest prefix match

όταν ψάχνουμε την καταχώρηση στον πίνακα προώθησης για μια δοθείσα διεύθυνση προορισμού, χρησιμοποιούμε το μεγαλύτερο πρόθεμα που ταιριάζει στην διεύθυνση προορισμού.

Destination Address Range				Link interface
11001000	00010111	00010***	*****	0
11001000	00010111	00011000	*****	1
11001000	00010111	00011***	*****	2
otherwise				3

examples:

11001000 00010111 00010110 10100001 which interface?
11001000 00010111 00011000 10101010 which interface?

Longest prefix match (2)

longest prefix match

όταν ψάχνουμε την καταχώρηση στον πίνακα προώθησης για μια δοθείσα διεύθυνση προορισμού, χρησιμοποιούμε το μεγαλύτερο πρόθεμα που ταιριάζει στην διεύθυνση προορισμού.

Destination Address Range	Link interface
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 match! 1 00011*** *****	2
otherwise	3

examples:

11001000 00010111 00010110 10100001 which interface?
11001000 00010111 00011000 10101010 which interface?

Longest prefix match (3)

longest prefix match

όταν ψάχνουμε την καταχώρηση στον πίνακα προώθησης για μια δοθείσα διεύθυνση προορισμού, χρησιμοποιούμε το μεγαλύτερο πρόθεμα που ταιριάζει στην διεύθυνση προορισμού.

Destination Address Range	Link interface
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011*** *****	2
otherwise	3

match!

examples:

11001000 00010111 00010110 10100001	which interface?
11001000 00010111 00011000 10101010	which interface?

Longest prefix match (4)

If the router has the following forwarding table

Destination Network	Interface
147.52.0.0/16	Eth0
147.52.1.0/24	Eth1
147.52.2.0/24	Eth2
147.52.2.4/32	Eth3
147.52.2.4/30	Eth4
Default	Eth5

Longest prefix match (5)

Report the correct egress interfaces (the interface that will be used to forward the packet from) for the packets with the following destination IPs.

- 147.52.0.16
- 147.51.0.16
- 147.52.2.4
- 147.52.2.6
- 147.52.2.7
- 147.52.2.8
- 147.52.3.0

Longest prefix match (6)

Report the correct egress interfaces (the interface that will be used to forward the packet from) for the packets with the following destination IPs.

- 147.52.0.16 -> **Eth0**
- 147.51.0.16 -> **Eth5**
- 147.52.2.4 -> **Eth3**
- 147.52.2.6 -> **Eth4**
- 147.52.2.7 -> **Eth4**
- 147.52.2.8 -> **Eth2**
- 147.52.3.0 -> **Eth0**