Routing algorithms

IP LAYER LAB

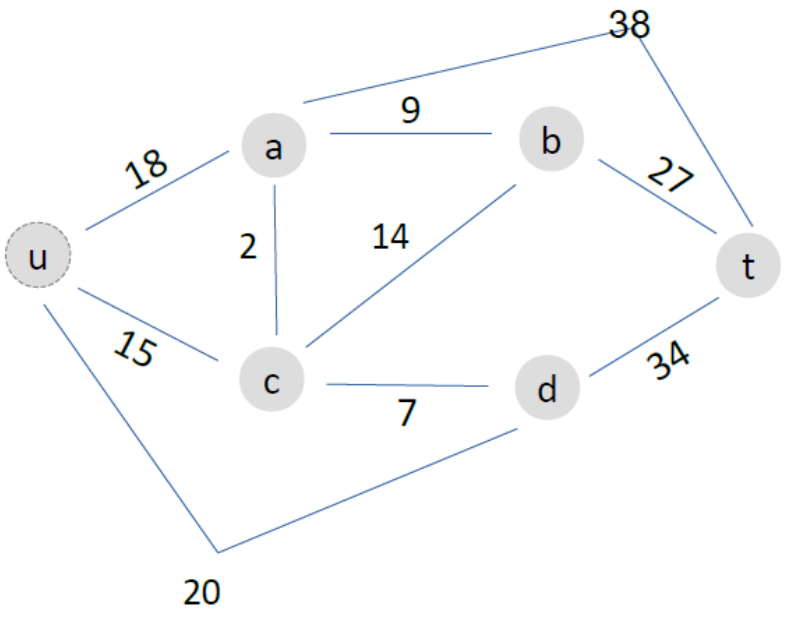
HY335A

EVRIPIDIS TZAMOUSIS

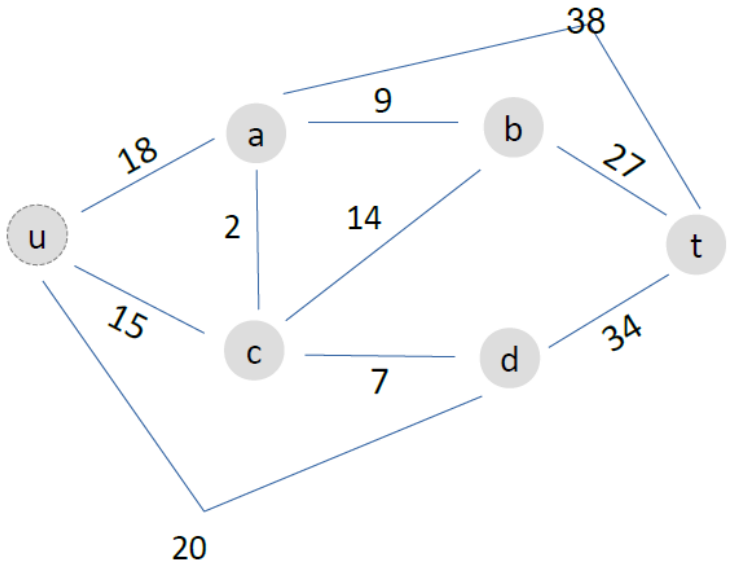
Αλγόριθμος Κατάστασης Ζεύξης

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

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

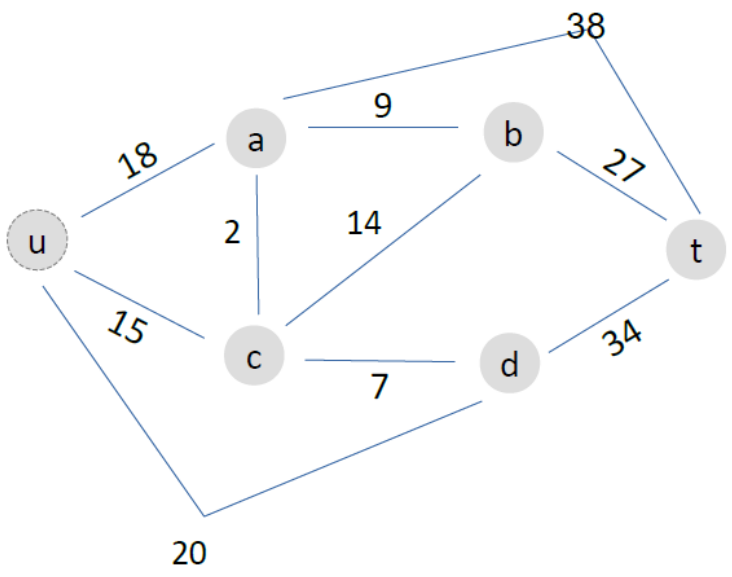


Αλγόριθμος Κατάστασης Ζεύξης



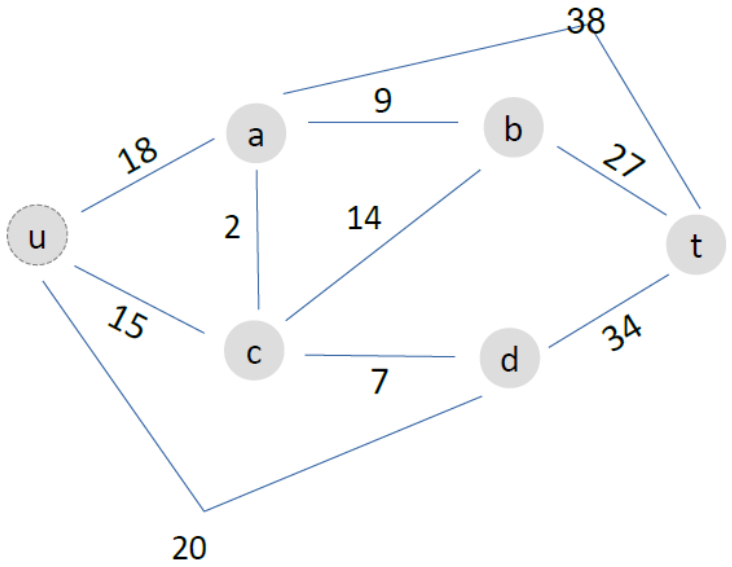
N'	D(c), p(c)	D(a), p(a)	D(d), p(d)	D(b), p(b)	D(t), p(t)
u	15, u	18, u	20, u	∞	∞

Αλγόριθμος Κατάστασης Ζεύξης



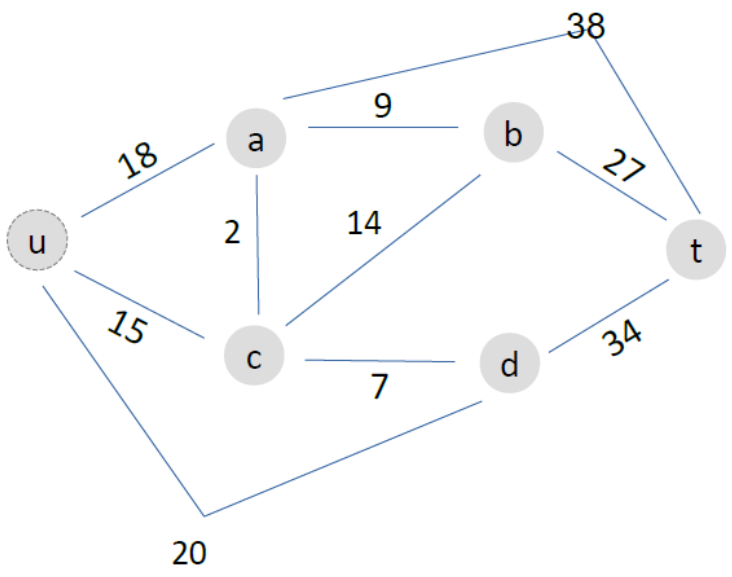
N'	D(c), p(c)	D(a), p(a)	D(d), p(d)	D(b), p(b)	D(t), p(t)
u	15, u	18, u	20, u	∞	∞
uc	-	17,c	20, u *(22,c)*	29,c	∞

Αλγόριθμος Κατάστασης Ζεύξης



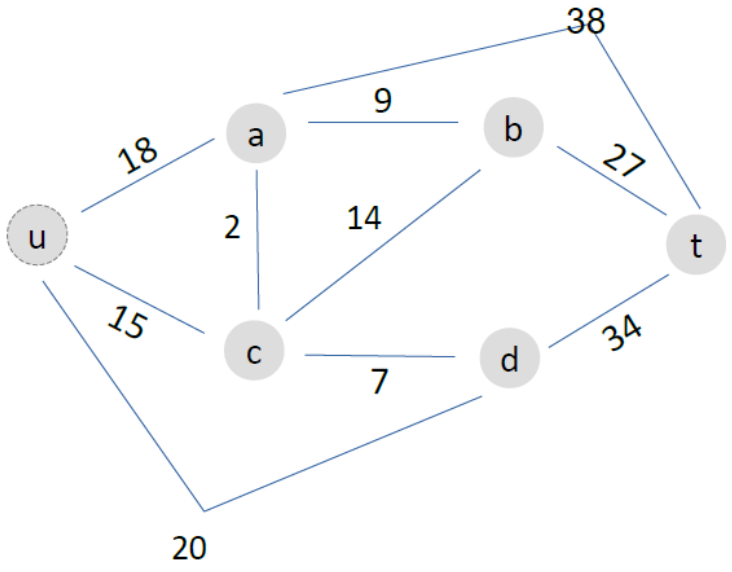
N'	D(c), p(c)	D(a), p(a)	D(d), p(d)	D(b), p(b)	D(t), p(t)
u	15, u	18, u	20, u	∞	∞
uc	-	17,c	20, u	29,c	∞
uca	-	-	20, u	26, a	55, a

Αλγόριθμος Κατάστασης Ζεύξης



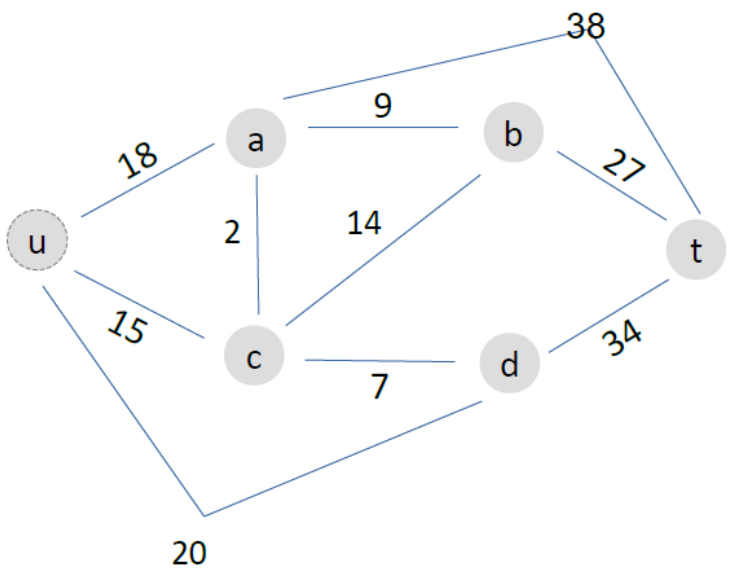
N'	D(c), p(c)	D(a), p(a)	D(d), p(d)	D(b), p(b)	D(t), p(t)
u	15, u	18, u	20, u	∞	∞
uc	-	17,c	20, u	29,c	∞
uca	-	-	20, u	26, a	55, a
ucad	-	-	-	26, a	54, d

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 → c → a → b → 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:

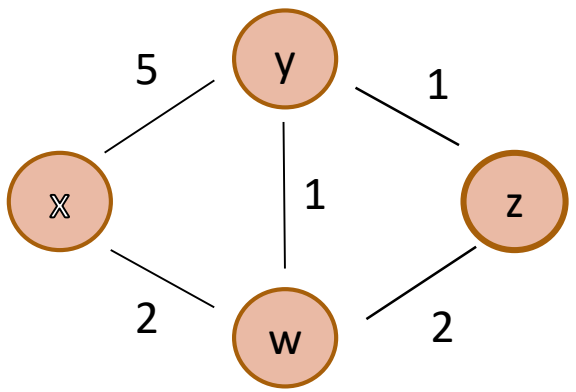
$D_x(y) = \min_v\{ c(x,v) + D_v(y)\}$

(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



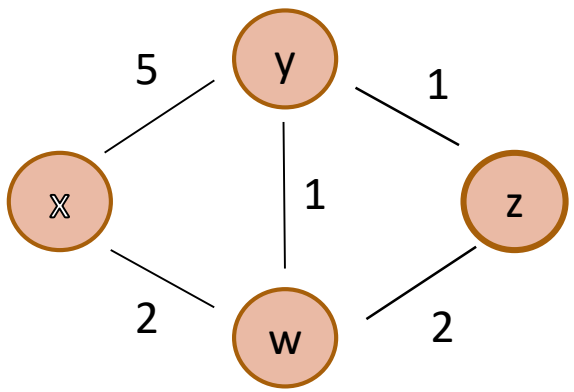
	x	y	w	z
x	0	5	2	∞
y	∞	∞	∞	∞
w	∞	∞	∞	∞

	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**

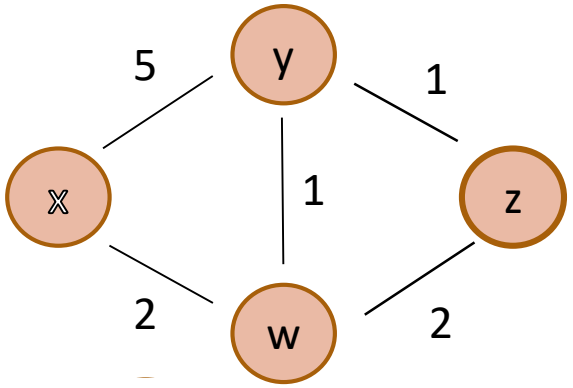
	x	y	w	z
x	0	5	2	∞
y	5	0	1	1
w	2	1	0	2
z	∞	1	2	0

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	**3**	2	**4**
y	5	0	1	1
w	2	1	0	2

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

$\mathbf{D_x(y) = c(x,w) + D_w(y) = 2 + 1 = 3}$

$\mathbf{D_x(w) = c(x,w) + D_w(w) = 2 + 0 = 2}$

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

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

$\mathbf{D_x(z) = c(x,w) + D_w(z) = 2 + 2 = 4}$

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

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

$\mathbf{D_y(x) = c(y,w) + D_w(x) = 1 + 2 = 3}$

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

$\mathbf{D_y(w) = c(y,w) + D_w(w) = 1 + 0 = 1}$

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

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

$\mathbf{D_y(z) = c(y,z) + D_z(z) = 1 + 0 = 1}$

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

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

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

$\mathbf{D_w(x) = c(w,x) + D_x(x) = 2 + 0 = 2}$

$D_w(x) = c(w,y) + D_y(x) = 1 + 5 = 3$

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

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

$\mathbf{D_w(y) = c(w,y) + D_y(y) = 1 + 0 = 1}$

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

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

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

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

	x	y	w	z
y	5	0	1	1
w	2	1	0	2
z	**4**	1	2	0

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

$\mathbf{D_z(x) = c(z,w) + D_w(x) = 2 + 2 = 4}$

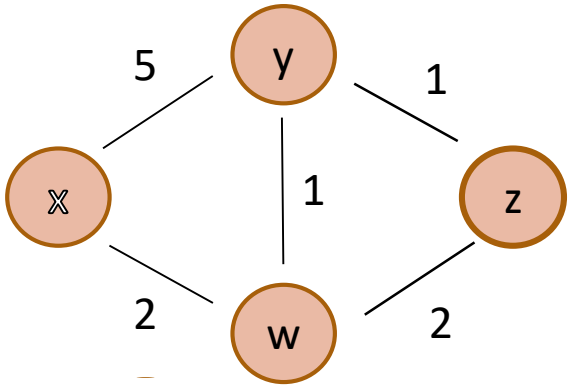
$\mathbf{D_z(y) = c(z,y) + D_y(y) = 1 + 0 = 1}$

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

$\mathbf{D_z(w) = c(z,y) + D_y(z) = 1 + 1 = 2}$

$\mathbf{D_z(w) = c(z,w) + D_w(z) = 2 + 0 = 2}$

Αποστολή ενημερώσεων



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

Ο **x** θα ενημερώσει: **y, w**

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

Ο **y** θα ενημερώσει: **x, w, z**

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

Ο **w** δεν ενημερώνει κανένα

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

Ο **z** θα ενημερώσει: **y, w**

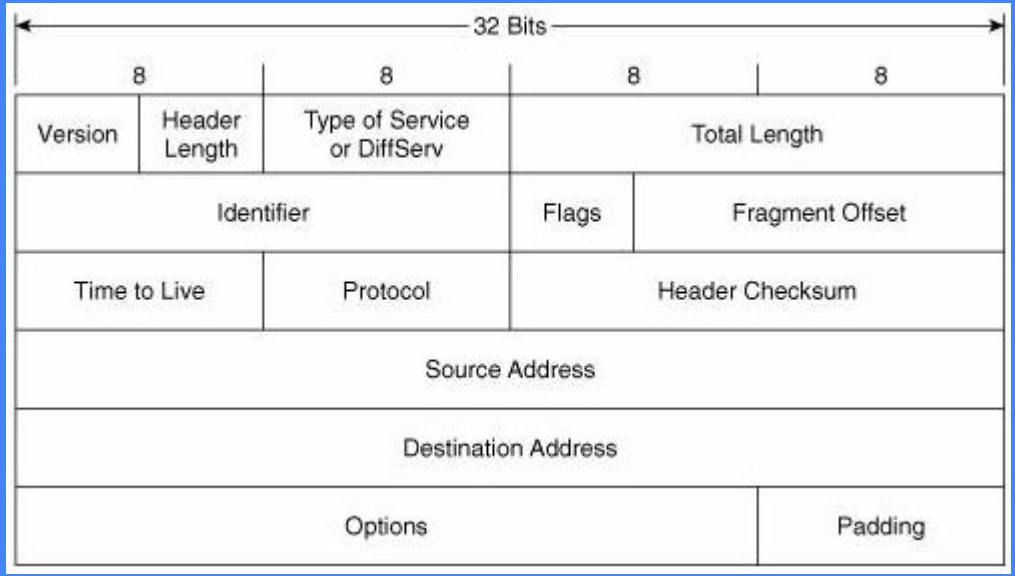
IPv4 addressing & subnetting

+ fragmentation

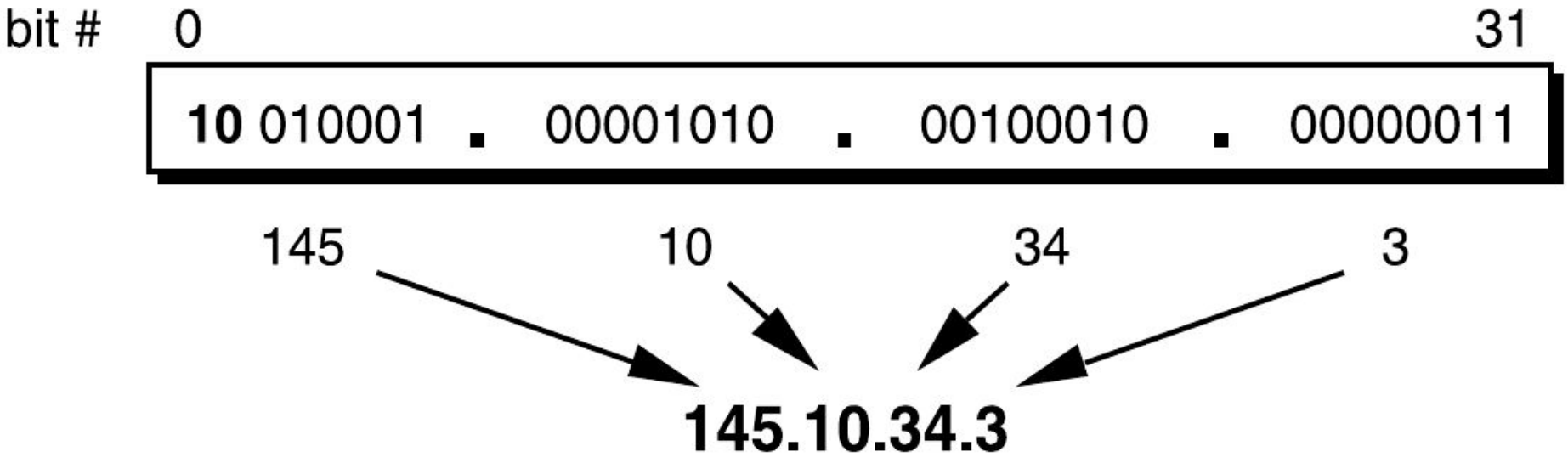
CS335a - Introduction to Computer Networks

Kostis Triantafyllakis - ctriant@csd.uoc.gr

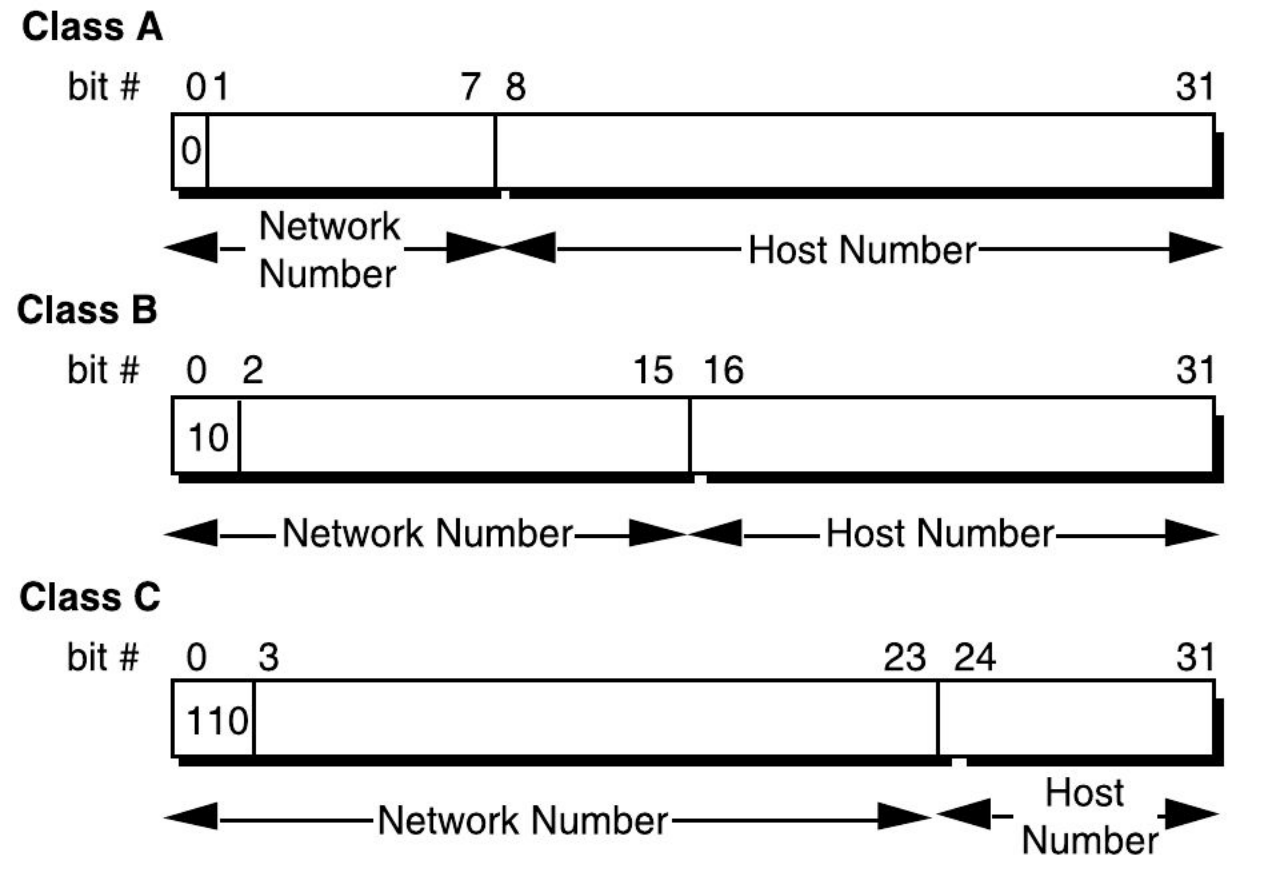
IPv4 header


32 Bits
8
8
8
8
Version
Header Length
Type of Service or DiffServ
Total Length
Identifier
Flags
Fragment Offset
Time to Live
Protocol
Header Checksum
Source Address
Destination Address
Options
Padding

Dotted-Decimal notation



Classful IP addressing



126 (2^7-2) networks
~16,000,000 hosts

~16k (2^{14}) networks
~65k (2^{16}-2)hosts

~2M (2^{21}) networks
~254 (2^8-2)hosts

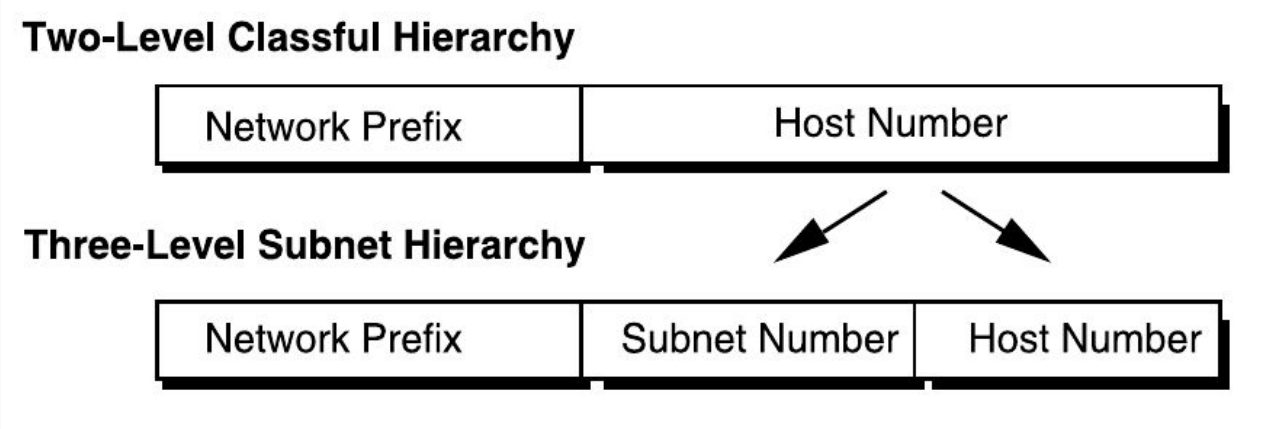
Unforeseen limitations to Classful Addressing

- The original designers never envisioned the current Internet growth
- Addresses were freely assigned to those who asked for them without concerns.
- The decision of 32-bit addresses was wrong.
  4,294,967,296 addresses are not enough
- The concept of the Classful Addressing was easy to understand and implement, but it was not efficient for a finite address space.
  - /24 supports 254 hosts that is small
  - /16 supports 65,534 hosts that is big

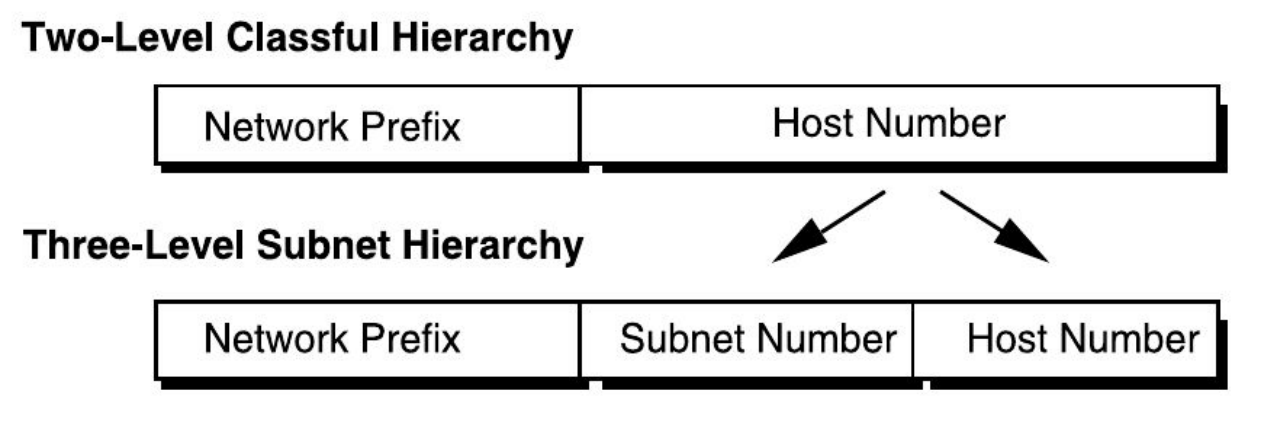
Subnetting

- The division of a single Class A, B or C network into smaller pieces

- What need led to Subnetting?
  - Internet routing tables were beginning to grow
  - Local admins had to request another network number from the Internet before a new network could be installed in their site.

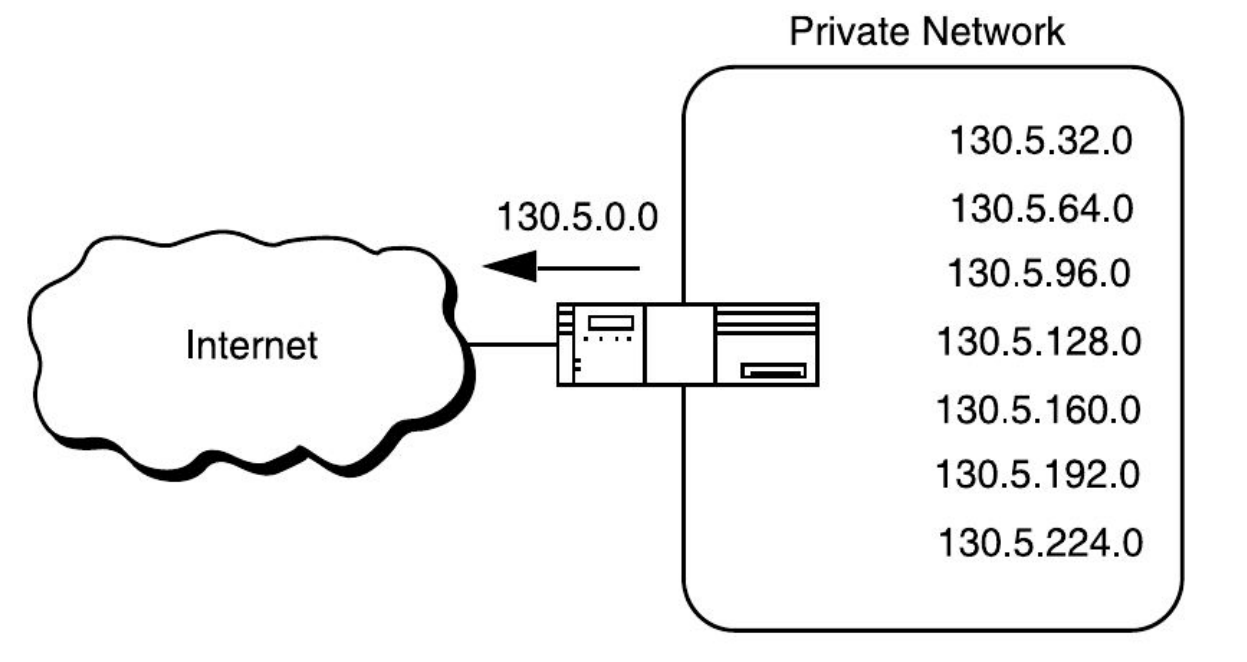


Subnetting



- Subnet structure of a network is never visible outside the organization's private network
- Each organization is assigned one (or at most a few) network addresses from the IPv4 address space.
    - The organization was free to assign a distinct sub-network number to each of its internal networks.

Subnetting



- The size of Internet routing table is not affected
- Rapid changing of routes within the private network do not affect the Internet routing table

Subnet mask

```
130.5.5.25        10000010.00000101.00000101.00011001
255.255.255.0     11111111.11111111.11111111.00000000

                              or

130.5.5.25/24     10000010.00000101.00000101.00011001
```

24 bit Extended Network Prefix

- Internet routers use only the Network prefix of the destination address to route the traffic to a subnet.
- Routers within a subnet use the Extended Network Prefix to route the traffic between the individual subnets.

Subnet design considerations

- What is the total number of subnets that are needed today?
- What is the total number of hosts that are needed today?

- What about the future?

Example

Define the subnet and host addresses

- An organization holds the network number: **193.1.1.0 / 24**
- Needs to define **6 subnets**
- The largest subnet is required to support **25 hosts**

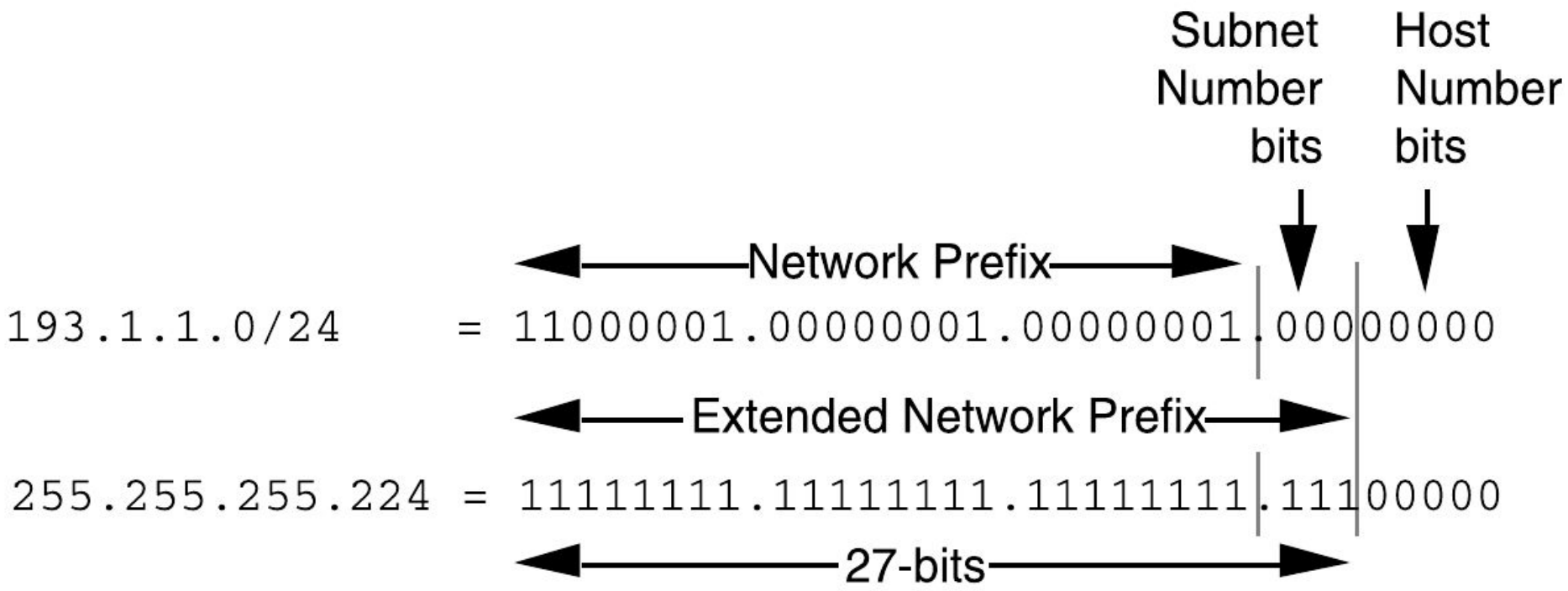
Example

Decimal: 193.1.1.0/24
Binary: 11000001.00000001.00000001.00000000

- What is the number of bits required to define 6 subnets?
  - 3bits because $2^3 = 8$ subnets
  - That leaves 2 spare subnets.

- What's the subnet mask?
  - Since the organization is subnetting a /24 it needs 3 more bits into the mask or equivalently /27

Example


Subnet Number bits
Host Number bits
Network Prefix
193.1.1.0/24 = 11000001.00000001.00000001.00000000
Extended Network Prefix
255.255.255.224 = 11111111.11111111.11111111.11100000
27-bits

Example

Base Net: 11000001.00000001.00000001 .00000000 = 193.1.1.0/24

Subnet #0: 11000001.00000001.00000001.**000** 00000 = 193.1.1.0/27

Subnet #1: 11000001.00000001.00000001.**001** 00000 = 193.1.1.32/27

Subnet #2: 11000001.00000001.00000001.**010** 00000 = 193.1.1.64/27

Subnet #3: 11000001.00000001.00000001.**011** 00000 = 193.1.1.96/27

Subnet #4: 11000001.00000001.00000001.**100** 00000 = 193.1.1.128/27

Subnet #5: 11000001.00000001.00000001.**101** 00000 = 193.1.1.160/27

Subnet #6: 11000001.00000001.00000001.**110** 00000 = 193.1.1.192/27

Subnet #7: 11000001.00000001.00000001.**111** 00000 = 193.1.1.224/27

Example

Subnet #2: 11000001.00000001.00000001.010 **00000** = 193.1.1.64/27
Host #1: 11000001.00000001.00000001.010 **00001** = 193.1.1.65/27
Host #2: 11000001.00000001.00000001.010 **00010** = 193.1.1.66/27
Host #3: 11000001.00000001.00000001.010 **00011** = 193.1.1.67/27
Host #4: 11000001.00000001.00000001.010 **00100** = 193.1.1.68/27
Host #5: 11000001.00000001.00000001.010 **00101** = 193.1.1.69/27

.

.

Host #15: 11000001.00000001.00000001.010 **01111** = 193.1.1.79/27
Host #16: 11000001.00000001.00000001.010 **10000** = 193.1.1.80/27

.

.

Host #27: 11000001.00000001.00000001.010 **11011** = 193.1.1.91/27
Host #28: 11000001.00000001.00000001.010 **11100** = 193.1.1.92/27
Host #29: 11000001.00000001.00000001.010 **11101** = 193.1.1.93/27
Host #30: 11000001.00000001.00000001.010 **11110** = 193.1.1.94/27

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

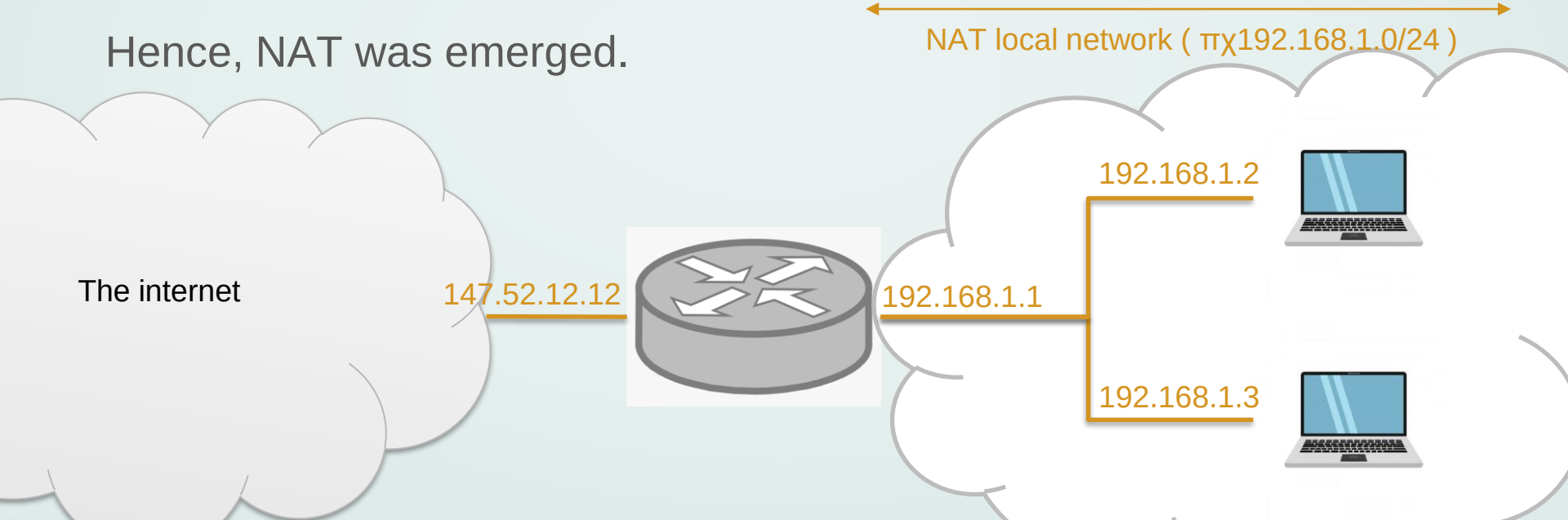
Arakadakis Konstantinos

NAT

IP addressing management within a network should be flexible.

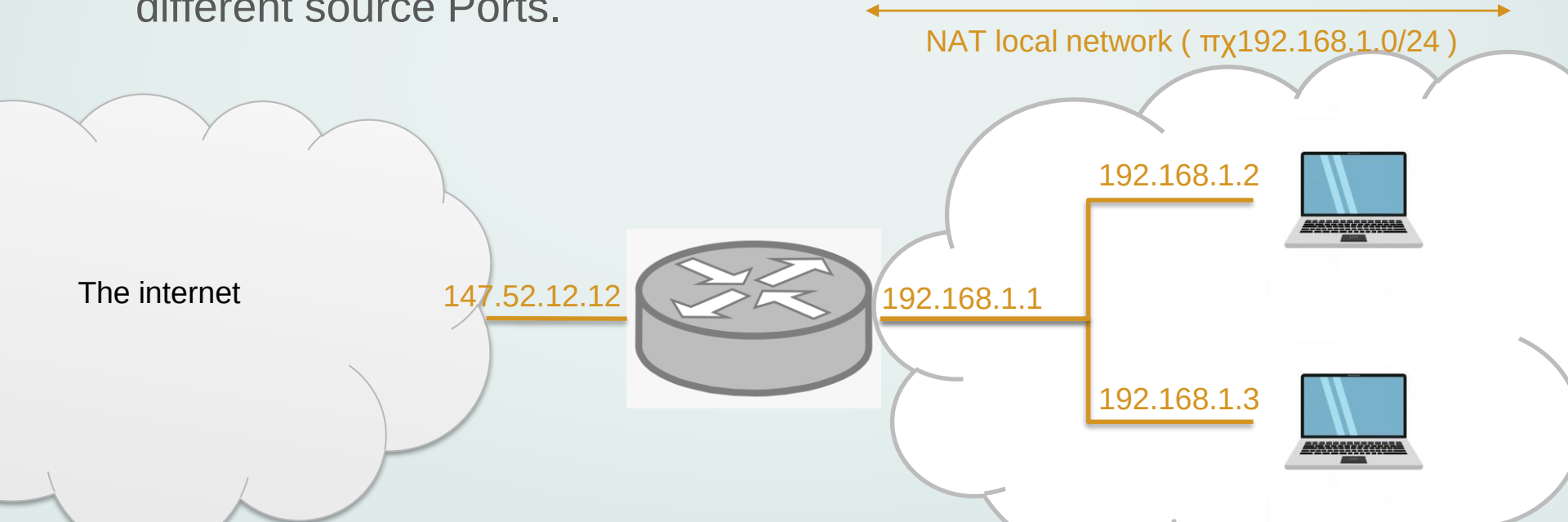
- Can support growing number of machines.
- No need to comply with global addressing standards.

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.



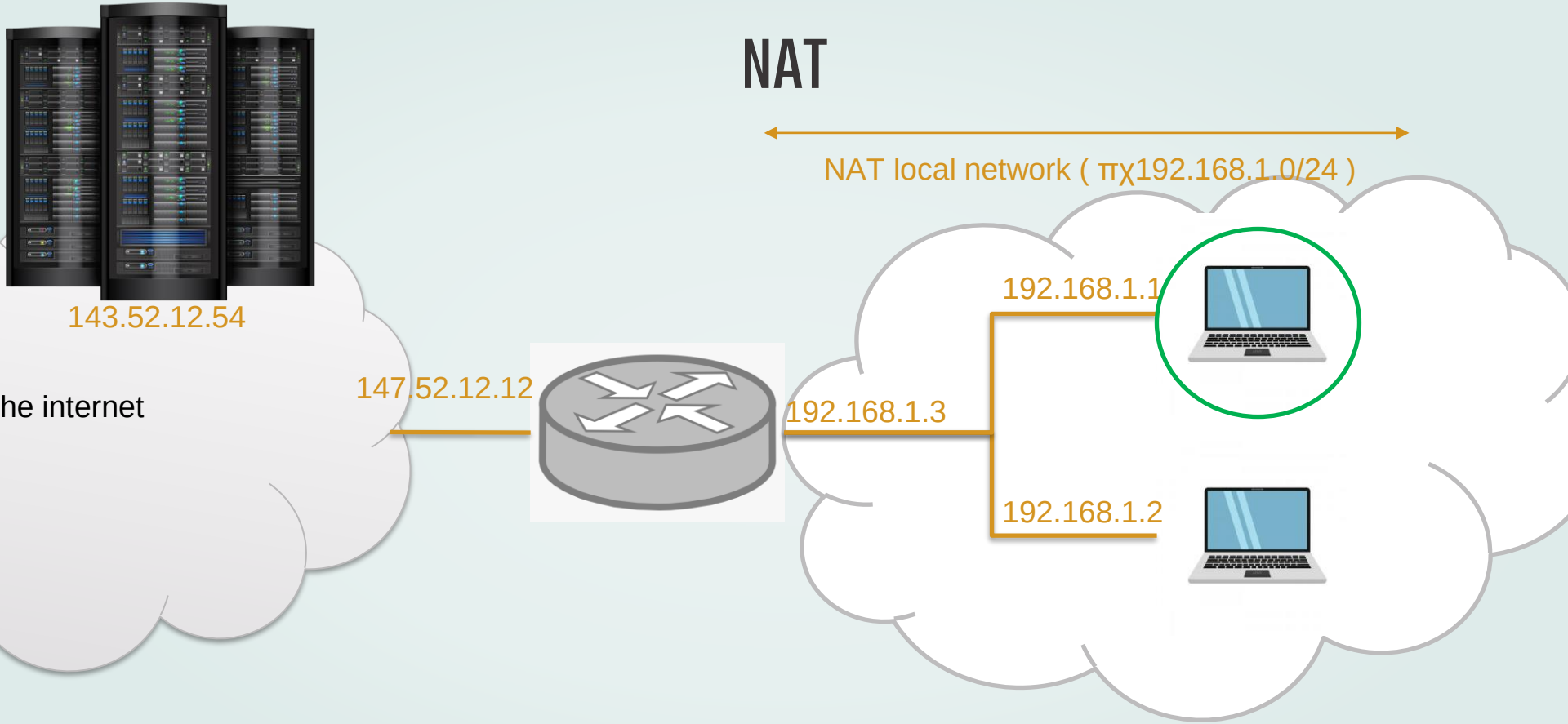
NAT

The process of a NAT enabled router:

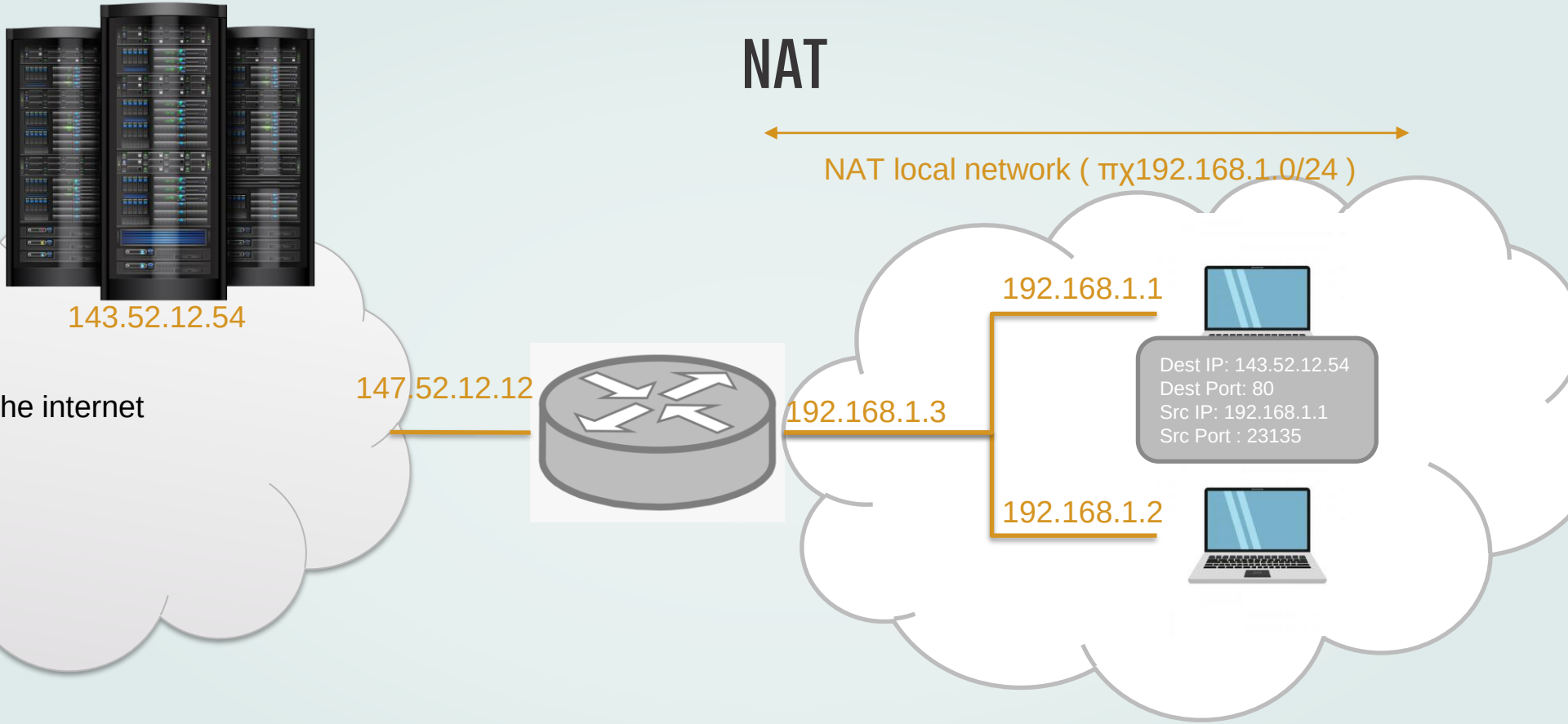
- Replace the (source IP, source Port) for **outgoing** datagrams to (NAT IP, new source Port)

- 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.

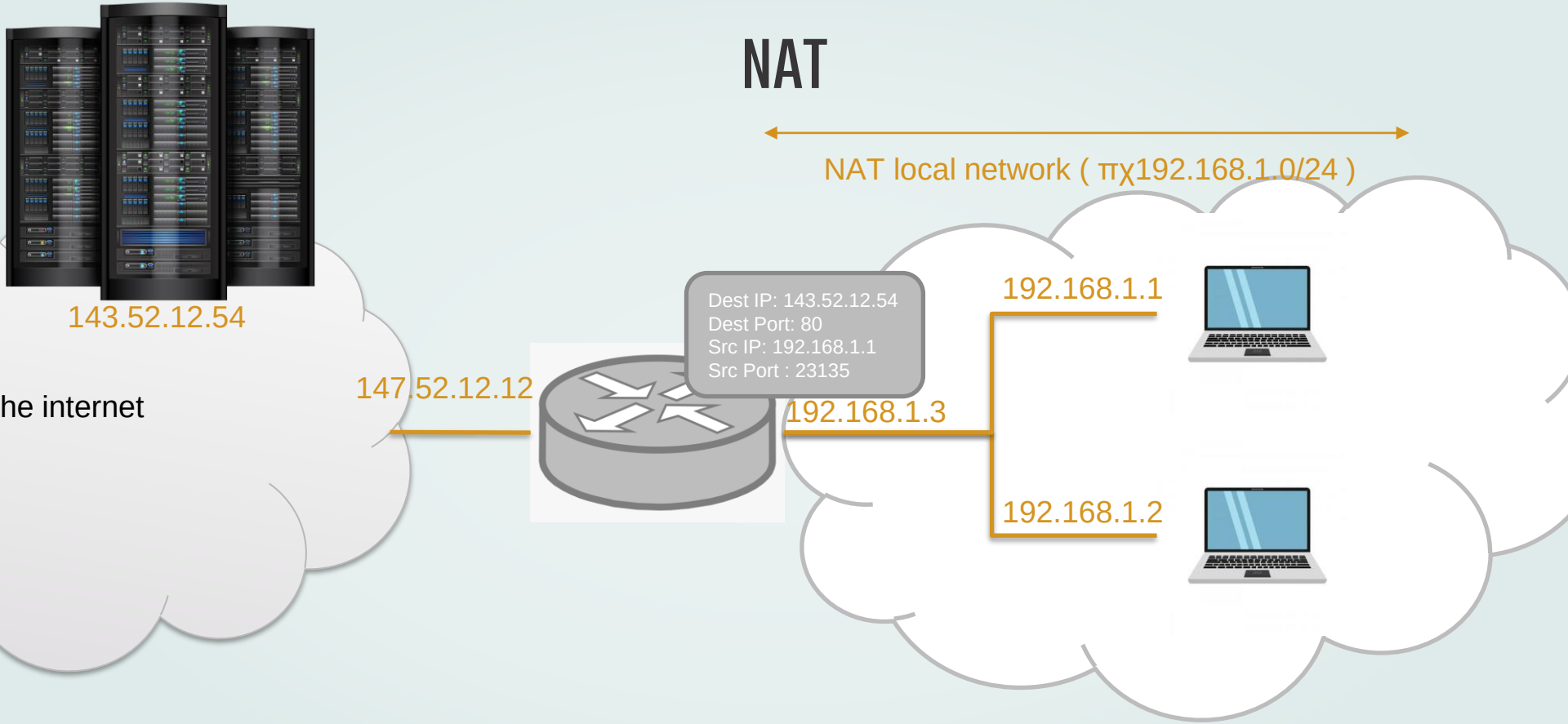
NAT


NAT local network (πχ192.168.1.0/24)
143.52.12.54
he internet
147.52.12.12
192.168.1.3
192.168.1.1
192.168.1.2

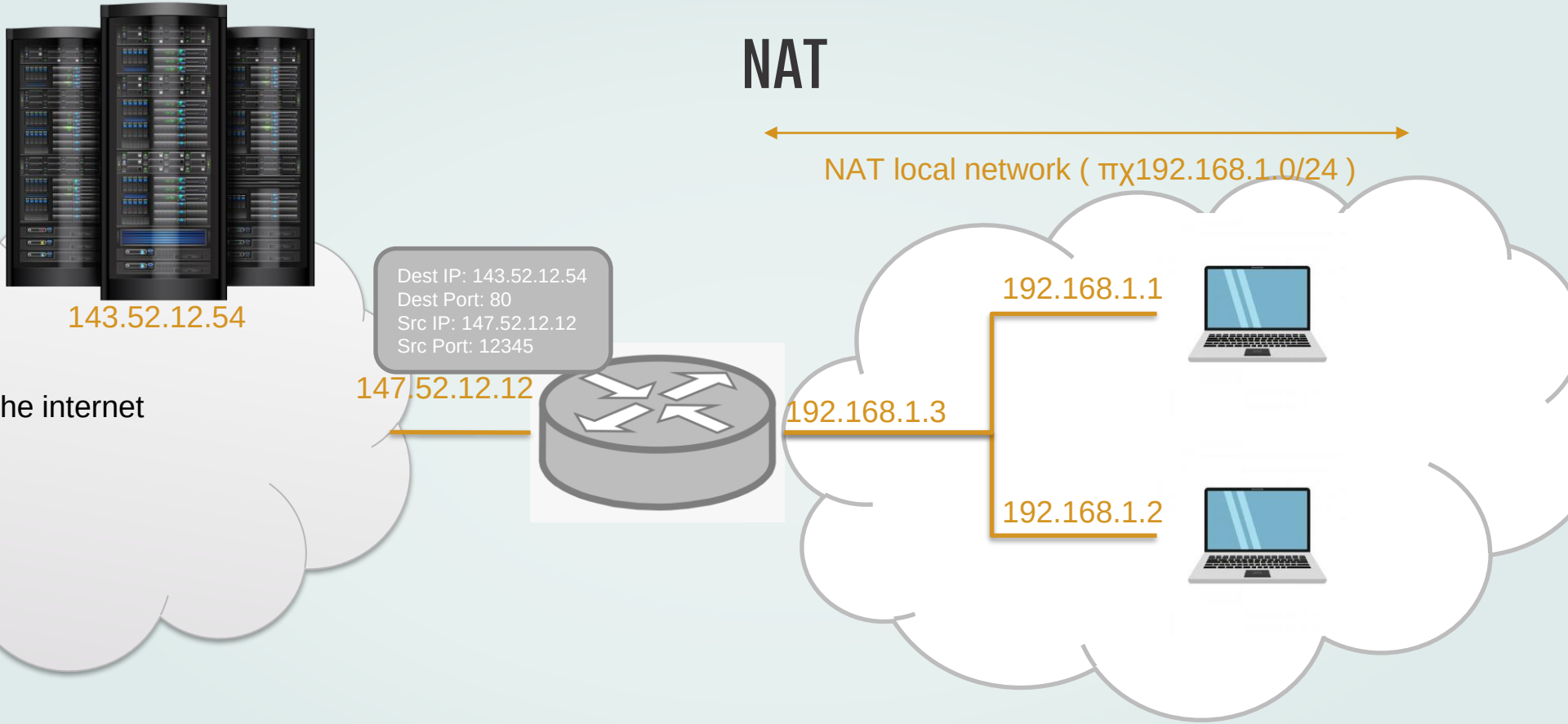
NAT


NAT local network (πχ192.168.1.0/24)
143.52.12.54
he internet
147.52.12.12
192.168.1.3
192.168.1.1
192.168.1.2
Dest IP: 143.52.12.54
Dest Port: 80
Src IP: 192.168.1.1
Src Port : 23135

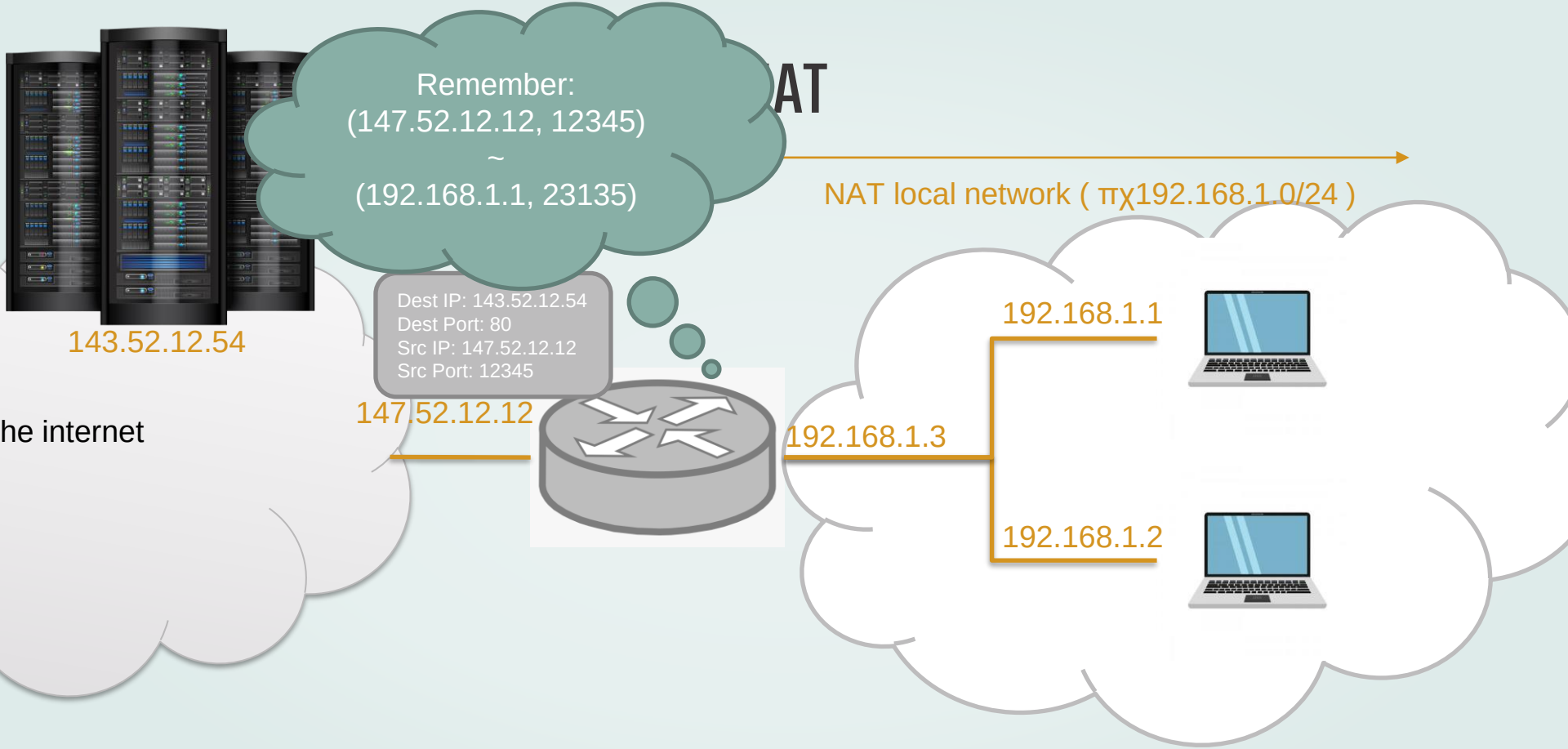
NAT


NAT local network (πχ192.168.1.0/24)
143.52.12.54
he internet
147.52.12.12
Dest IP: 143.52.12.54
Dest Port: 80
Src IP: 192.168.1.1
Src Port : 23135
192.168.1.3
192.168.1.1
192.168.1.2

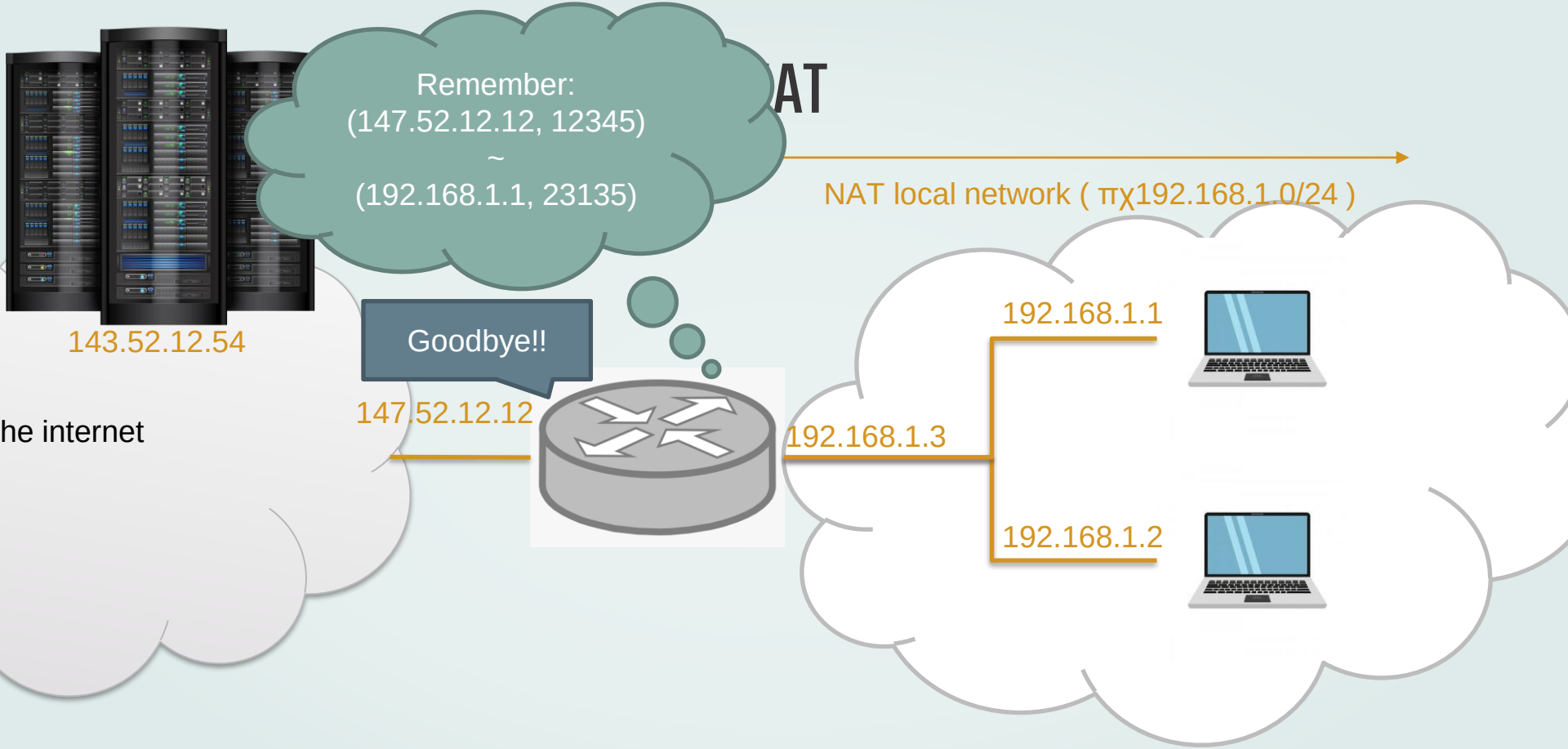
NAT


NAT local network (πχ192.168.1.0/24)
143.52.12.54
he internet
Dest IP: 143.52.12.54
Dest Port: 80
Src IP: 147.52.12.12
Src Port: 12345
147.52.12.12
192.168.1.3
192.168.1.1
192.168.1.2

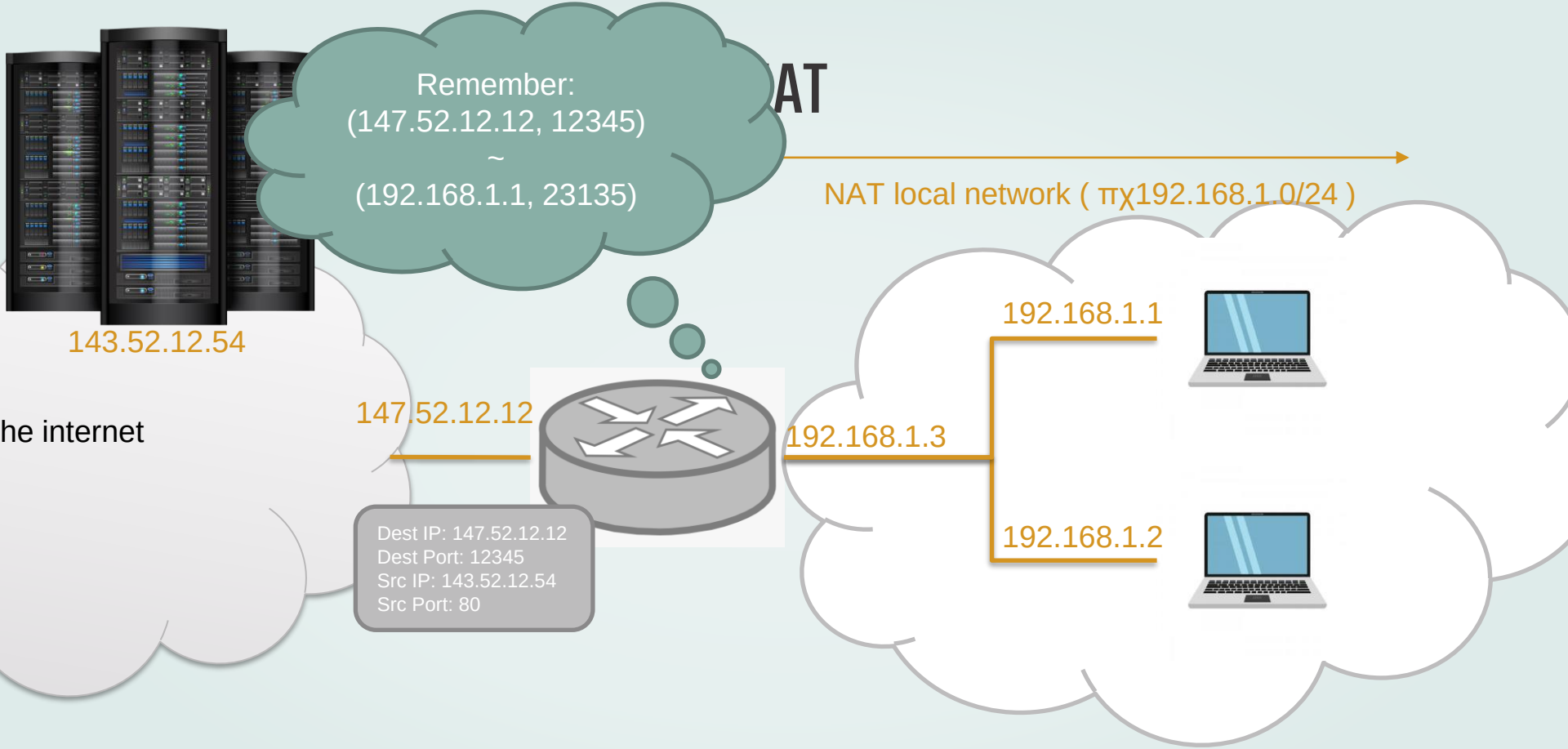
AT


Remember:
(147.52.12.12, 12345)
~
(192.168.1.1, 23135)
NAT local network (πχ192.168.1.0/24)
143.52.12.54
he internet
Dest IP: 143.52.12.54
Dest Port: 80
Src IP: 147.52.12.12
Src Port: 12345
147.52.12.12
192.168.1.3
192.168.1.1
192.168.1.2

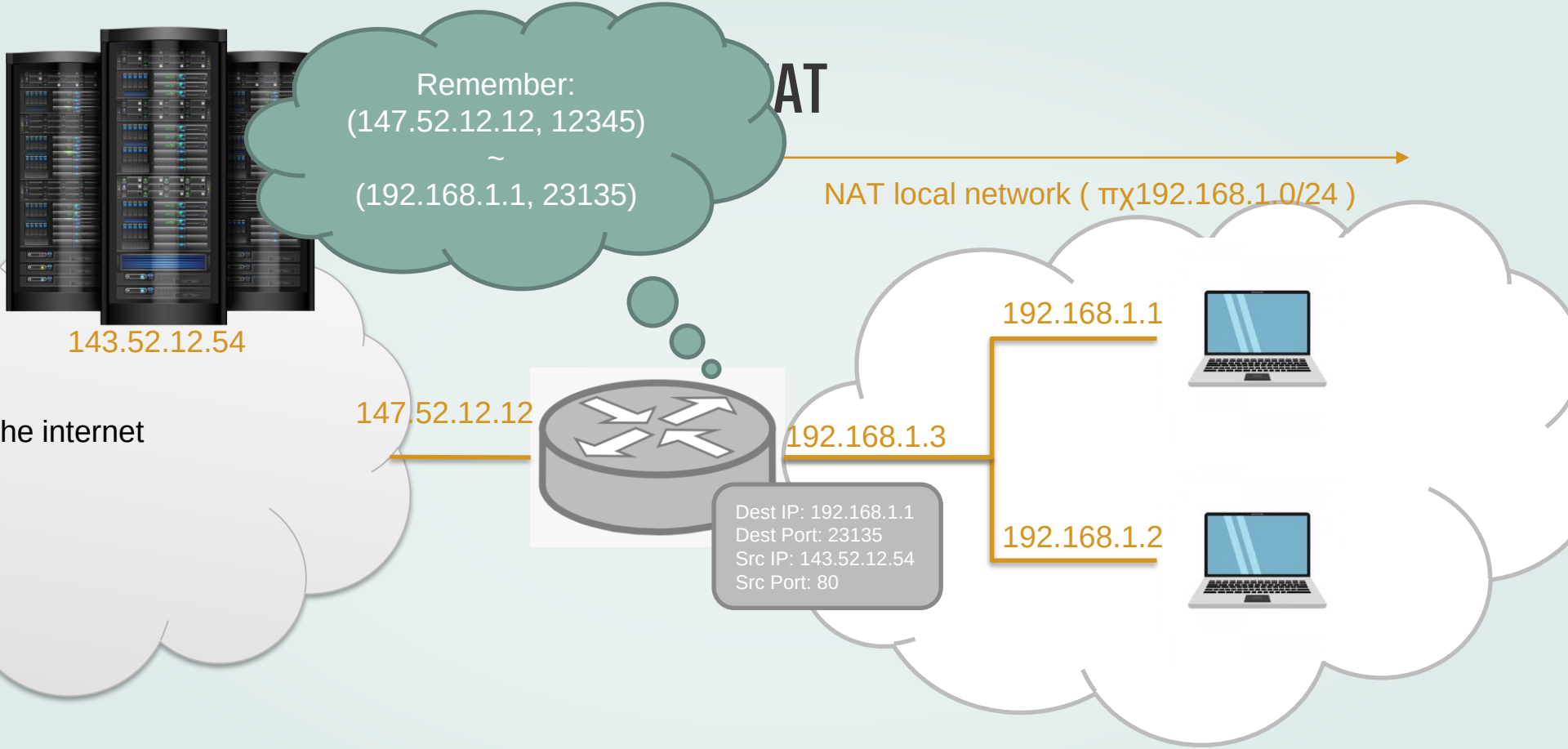
AT


Remember:
(147.52.12.12, 12345)
~
(192.168.1.1, 23135)
NAT local network (πχ192.168.1.0/24)
143.52.12.54
Goodbye!!
147.52.12.12
he internet
192.168.1.3
192.168.1.1
192.168.1.2

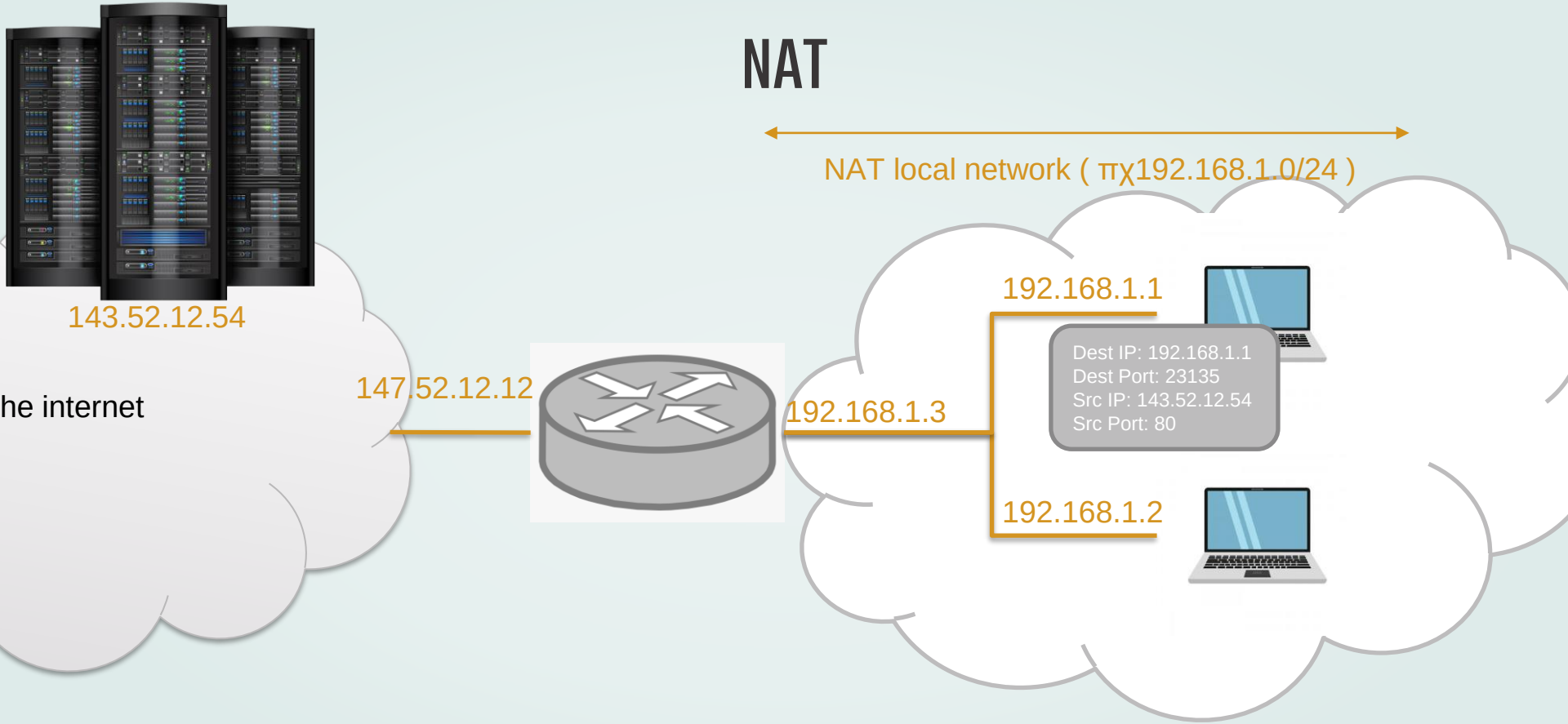
AT


Remember:
(147.52.12.12, 12345)
~
(192.168.1.1, 23135)
NAT local network (πχ192.168.1.0/24)
143.52.12.54
he internet
147.52.12.12
192.168.1.3
192.168.1.1
192.168.1.2
Dest IP: 147.52.12.12
Dest Port: 12345
Src IP: 143.52.12.54
Src Port: 80

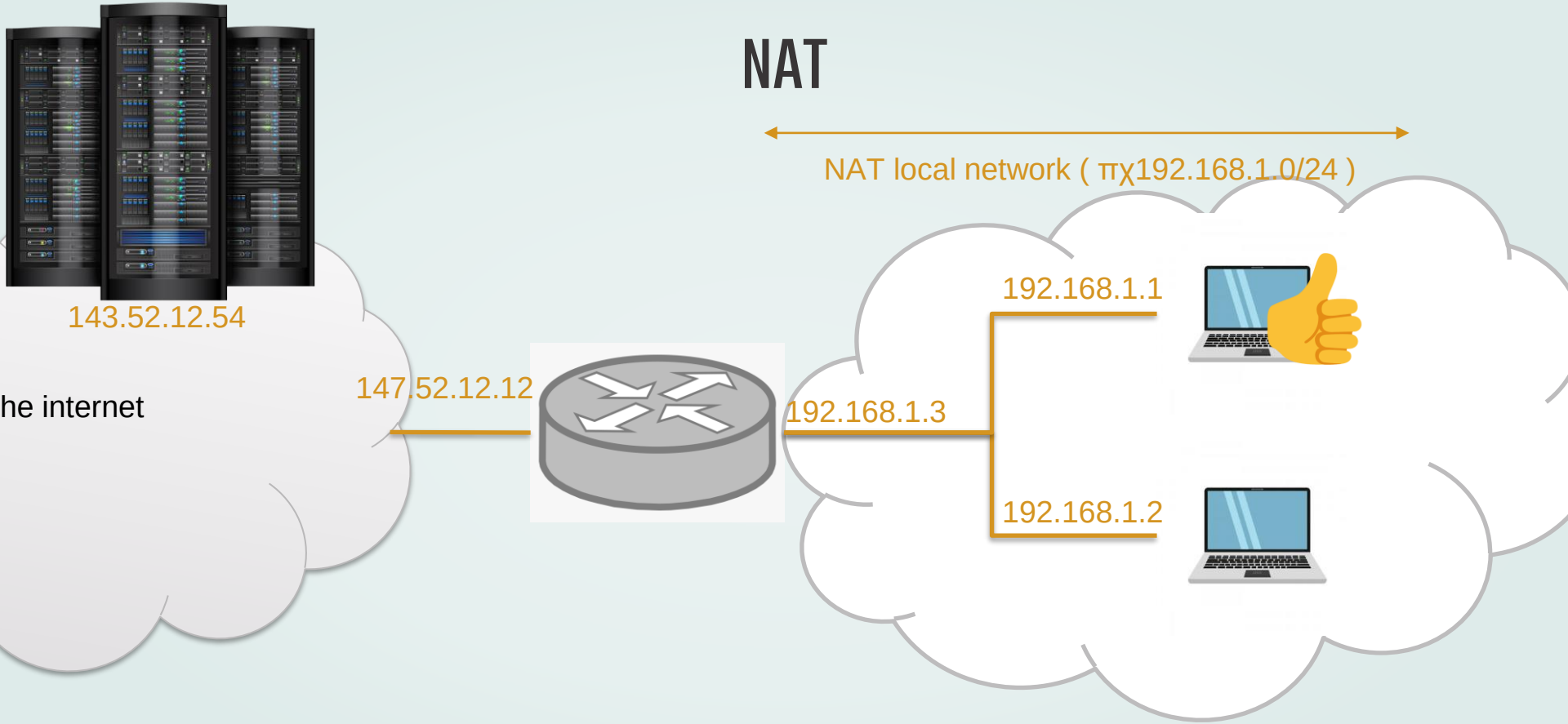
AT


Remember:
(147.52.12.12, 12345)
~
(192.168.1.1, 23135)
NAT local network (πχ192.168.1.0/24)
143.52.12.54
he internet
147.52.12.12
192.168.1.3
192.168.1.1
192.168.1.2
Dest IP: 192.168.1.1
Dest Port: 23135
Src IP: 143.52.12.54
Src Port: 80

NAT


NAT local network (πχ192.168.1.0/24)
143.52.12.54
he internet
147.52.12.12
192.168.1.3
192.168.1.1
Dest IP: 192.168.1.1
Dest Port: 23135
Src IP: 143.52.12.54
Src Port: 80
192.168.1.2

NAT


NAT local network (πχ192.168.1.0/24)
143.52.12.54
he internet
147.52.12.12
192.168.1.3
192.168.1.1
192.168.1.2

IPv6

Arakadakis Konstantinos

IPv6

IPv6 has been defined to ultimately replace IPv4.

Why?

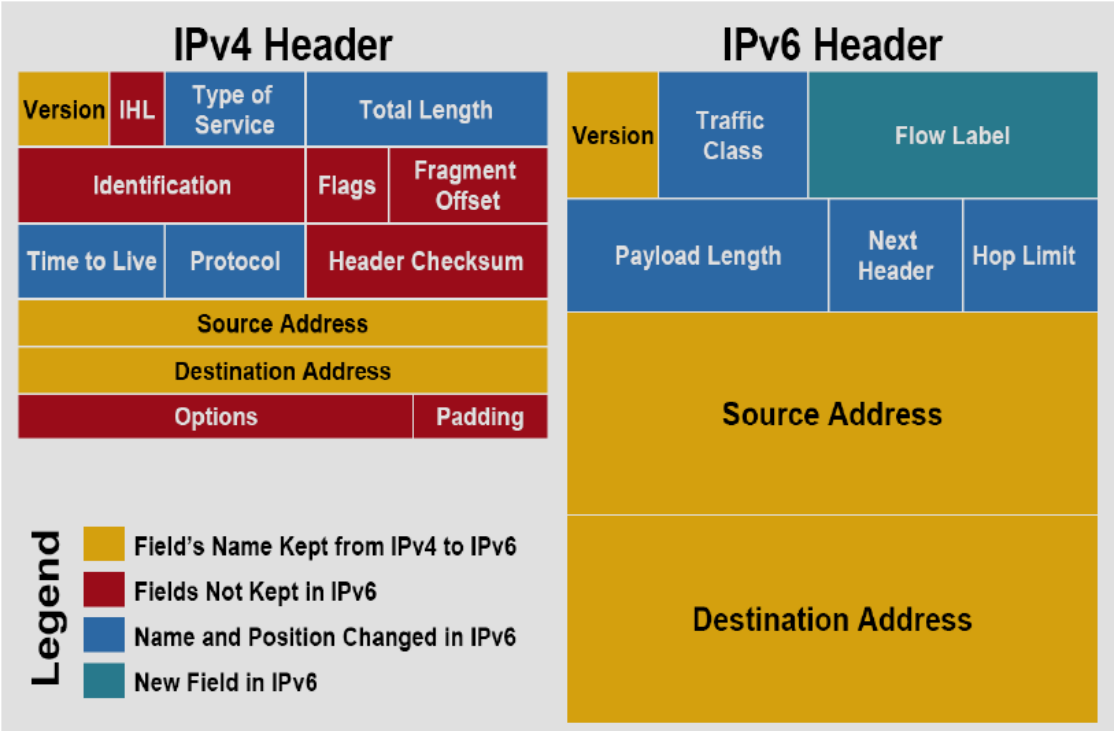
- Address space exhaustion
- Requirements for new services

IPv6

IPv6 enhancements:

- Extended address space (128 bits)
- Support for resource allocation (packet flows for QoS)
- Ipsec
- Increased addressing flexibility (anycast, No broadcast)
- NO fragmentation
- Simpler header

IPv6


IPv4 Header
Version
IHL
Type of Service
Total Length
Identification
Flags
Fragment Offset
Time to Live
Protocol
Header Checksum
Source Address
Destination Address
Options
Padding
IPv6 Header
Version
Traffic Class
Flow Label
Payload Length
Next Header
Hop Limit
Source Address
Destination Address
Legend
Field's Name Kept from IPv4 to IPv6
Fields Not Kept in IPv6
Name and Position Changed in IPv6
New Field in IPv6

ICMP

Arakadakis Konstantinos

ICMP

- Used by hosts and routers to signal information e.g. errors. (Network is unreachable)
- Technically speaking, it is not part of IP as ICMP messages are encapsulated in IP datagrams.
- **Ping** and **Traceroute** use ICMP messages.