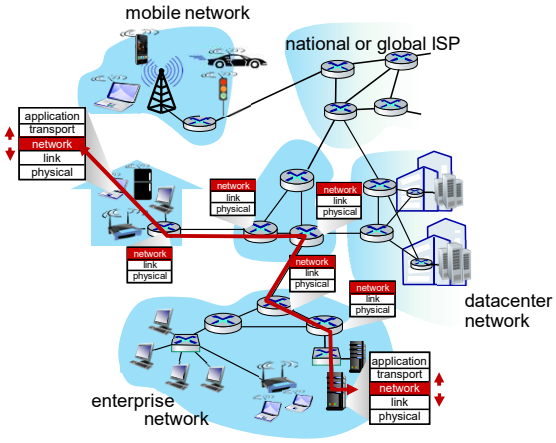
The Network layer

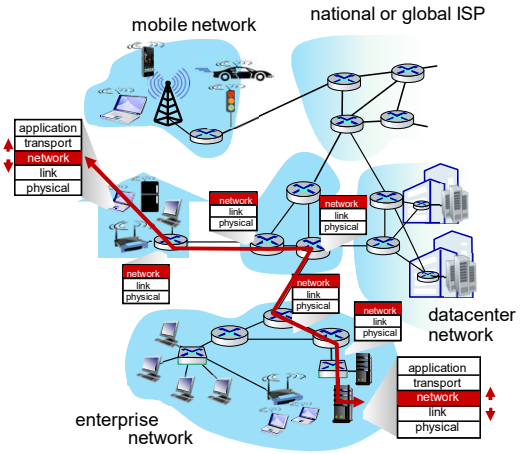


Network-layer services and protocols

The network layer protocols is in *every Internet device*: hosts, routers

Router:

- examines header fields in all IP datagrams passing through it
- moves datagrams from input ports to output ports to transfer datagrams along end-end path



Two key network-layer functions

network-layer functions:

- *forwarding:* move packets from a router's input link to appropriate router output link
- *routing:* determine route taken by packets from source to destination
  - *routing algorithms*

analogy: taking a trip

- *forwarding:* process of getting through single interchange
- *routing:* process of planning trip from source to destination

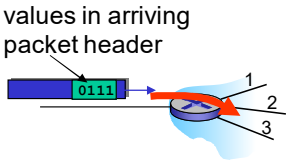
forwarding

routing

Network layer: data plane, control plane

Data plane:

- *local*, per-router function
- determines how datagram arriving on router input port is forwarded to router output port



Control plane

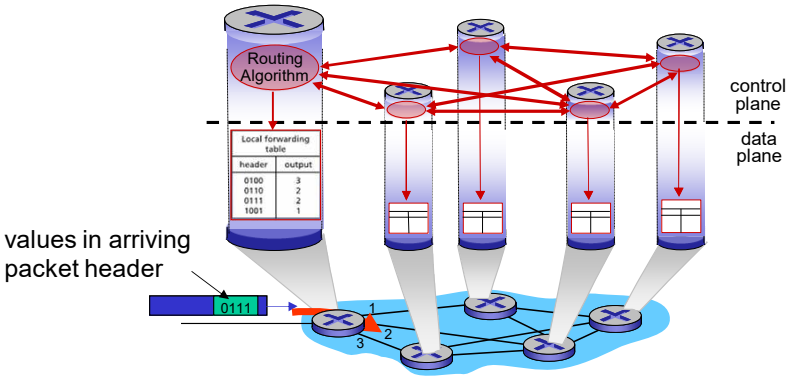
- *network-wide* logic
- determines how datagram is routed among routers along end-end path from source host to destination host

Two control-plane approaches:

- *traditional routing algorithms:* implemented in routers
- *software-defined networking (SDN)*: implemented in (remote) servers

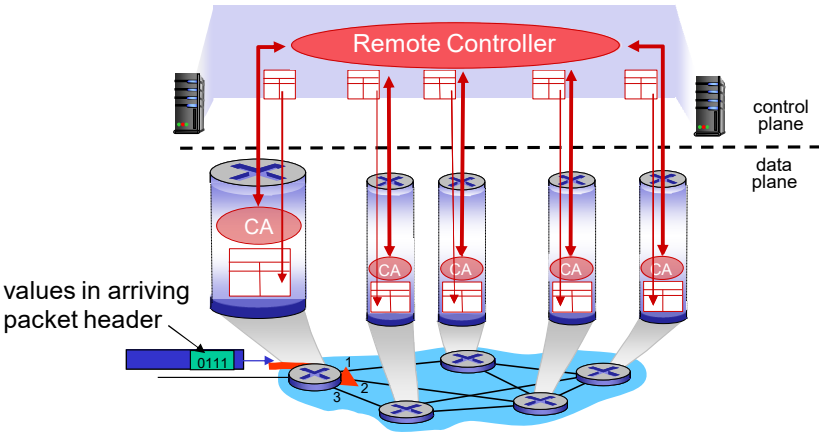
Per-router control plane

Individual routing algorithm components in each and every router interact in the control plane



Software-Defined Networking (SDN) control plane

Remote controller computes, installs forwarding tables in routers



Network service model

Q: What *service model* for "channel" transporting datagrams from sender to receiver?

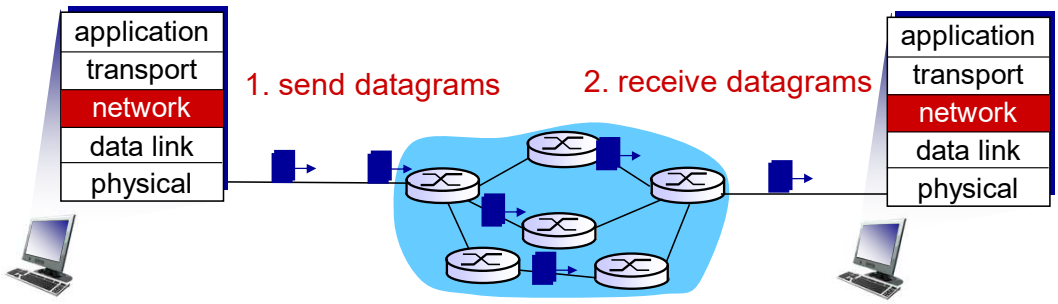
example services for *individual* datagrams:

- guaranteed delivery
- guaranteed delivery with less than 40 msec delay

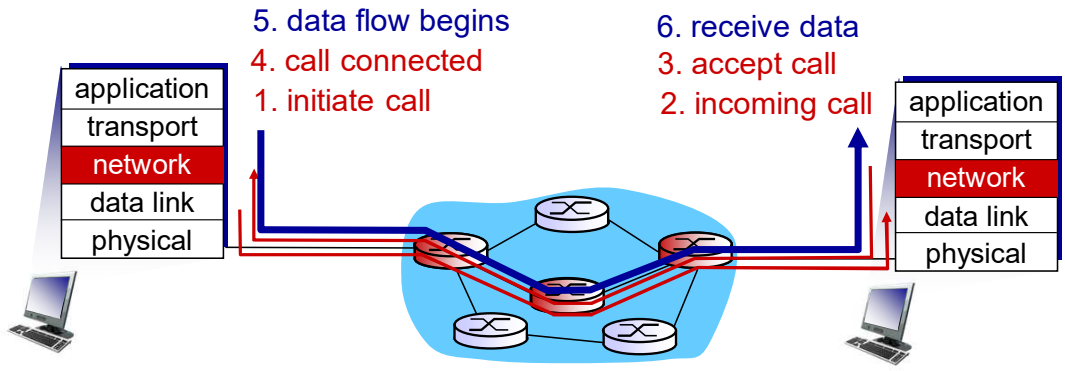
example services for a *flow* of datagrams:

- in-order datagram delivery
- guaranteed minimum bandwidth to flow
- restrictions on changes in inter-packet spacing

Datagram service



Virtual Circuit service



Virtual circuit vs Datagram

	Datagram	Virtual circuit
Circuit creation	No	Yes
State information	Address in every packet	VC address
Routing	No information	Every VC must be identified
Effects of router faults	No	Fault for every VC involved
Congestion control	Complex	Easy

Network-layer service model

Network Architecture	Service Model	Quality of Service (QoS) Guarantees ?			
		Bandwidth	Loss	Order	Timing
Internet	best effort	none	no	no	no

Internet "best effort" service model

No guarantees on*:*

1) successful datagram delivery to destination
2) timing or order of delivery
3) bandwidth available to end-end flow

Network-layer service model

Network Architecture	Service Model	Quality of Service (QoS) Guarantees ?			
		Bandwidth	Loss	Order	Timing
Internet	best effort	none	no	no	no
ATM	Constant Bit Rate	Constant rate	yes	yes	yes
ATM	Available Bit Rate	Guaranteed min	no	yes	no
Internet	Intserv Guaranteed (RFC 1633)	yes	yes	yes	yes
Internet	Diffserv (RFC 2475)	possible	possibly	possibly	no

Reflections on best-effort service:

- simplicity of mechanism has allowed Internet to be widely deployed adopted
- sufficient provisioning of bandwidth allows performance of real-time applications (e.g., interactive voice, video) to be "good enough" for "most of the time"
- replicated, application-layer distributed services (datacenters, content distribution networks) connecting close to clients' networks, allow services to be provided from multiple locations
- congestion control of "elastic" services helps

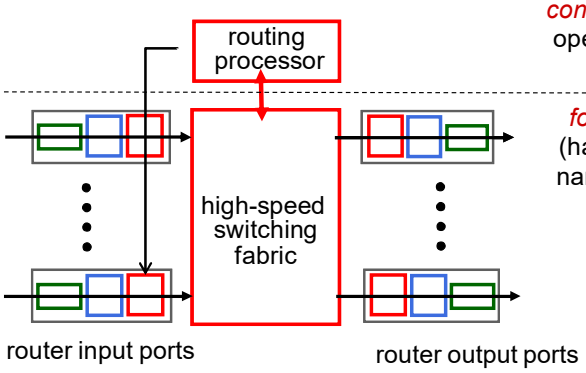
It's hard to argue with success of best-effort service model

Network layer: “data plane” roadmap

- Network layer: overview
  - data plane
  - control plane
- What’s inside a router
  - input ports, switching, output ports
  - buffer management, scheduling
- IP: the Internet Protocol
  - datagram format
  - addressing
  - network address translation
  - IPv6

Router architecture overview

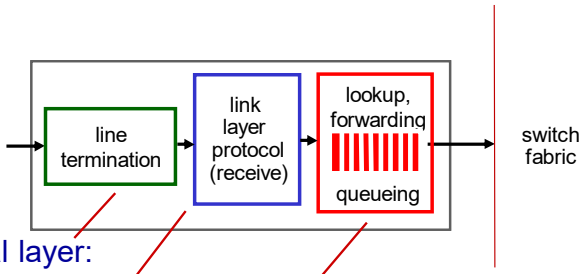
High-level view of generic router architecture:



routing, management control plane (software) operates in millisecond time frame

forwarding data plane (hardware) operates in nanosecond timeframe

Input port functions



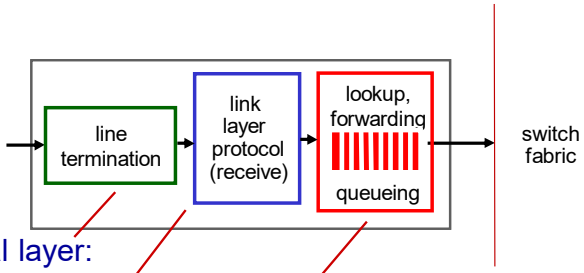
physical layer:
bit-level reception

link layer:
e.g., Ethernet

decentralized switching:

- using header field values, lookup output port using forwarding table in input port memory *("match plus action")*
- goal: complete input port processing at 'line speed'
- input port queuing: if datagrams arrive faster than forwarding rate into switch fabric

Input port functions



physical layer:
bit-level reception

link layer:
e.g., Ethernet

decentralized switching:

- using header field values, lookup output port using forwarding table in input port memory *("match plus action")*
- destination-based forwarding: forward based only on destination IP address (traditional)
- generalized forwarding: forward based on any set of header field values

Destination-based forwarding

forwarding table

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Q: but what happens if ranges don't divide up so nicely?

Destination-based forwarding

forwarding table

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00010000 00000100 through 11001000 00010111 00010000 00000111	3
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Longest prefix matching

longest prefix match

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

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 matching

longest prefix match

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

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 matching

longest prefix match

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

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 matching

longest prefix match

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

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 matching

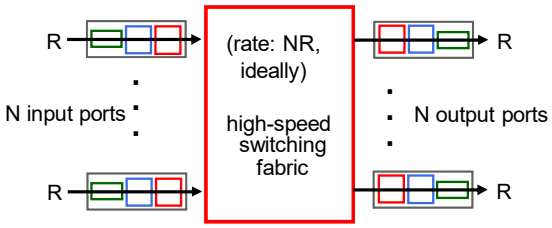
We'll see *why* longest prefix matching is used shortly, when we study addressing.

Longest prefix matching: often performed using Ternary Content Addressable Memories (TCAMs)

- *content addressable:* present address to TCAM: retrieve address in one clock cycle, regardless of table size
- Cisco Catalyst: ~1M routing table entries in TCAM

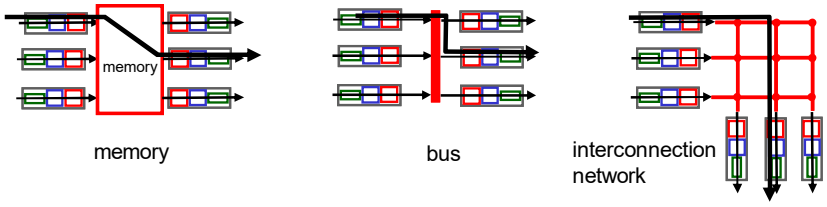
Switching fabrics

- transfer every packet from input link to appropriate output link
- switching rate: rate at which packets can be transfer from inputs to outputs
  - often measured as multiple of input/output line rate
  - N inputs: switching rate N times line rate desirable



Switching fabrics

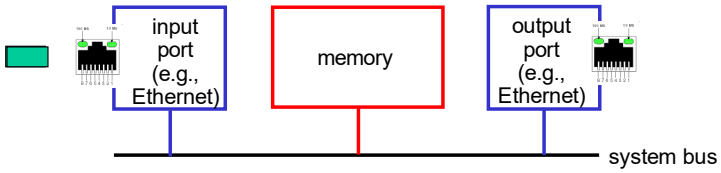
- transfer packet from input link to appropriate output link
- switching rate: rate at which packets can be transfer from inputs to outputs
  - often measured as multiple of input/output line rate
  - N inputs: switching rate N times line rate desirable
- three major types of switching fabrics:



Switching via memory

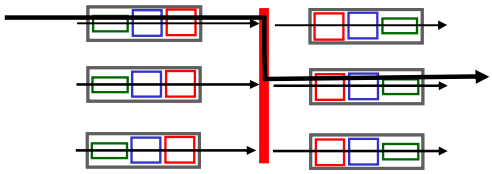
First generation routers:

- traditional computers with switching under direct control of CPU
- packet copied to system's memory
- speed limited by memory bandwidth (2 bus crossings per datagram)



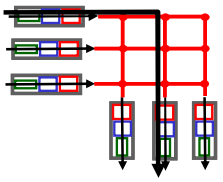
Switching via a bus

- datagram from input port memory to output port memory via a shared bus
- *bus contention:* switching speed limited by bus bandwidth
- 32 Gbps bus, Cisco 5600: sufficient speed for access routers

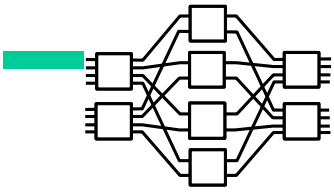


Switching via interconnection network

- Crossbar, Clos networks, other interconnection nets initially developed to connect processors in multiprocessor
- multistage switch: *nxn* switch from multiple stages of smaller switches
- exploiting parallelism:
  - fragment datagram into fixed length cells on entry
  - switch cells through the fabric, reassemble datagram at exit



3x3 crossbar



8x8 multistage switch
built from smaller-sized switches

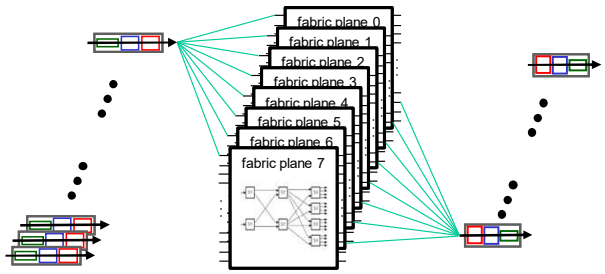
Switching via interconnection network

scaling, using multiple switching “planes” in parallel:

- speedup, scaleup via parallelism

Cisco CRS router:

- basic unit: 8 switching planes
- each plane: 3-stage interconnection network
- up to 100’s Tbps switching capacity

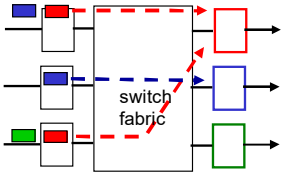


Input port queuing

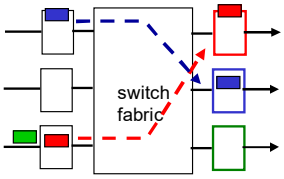
If switch fabric is slower than input ports combined -> queueing may occur at input queues

- queueing delay and loss due to input buffer overflow!

Head-of-the-Line (HOL) blocking: queued datagram at front of queue prevents others in queue from moving forward

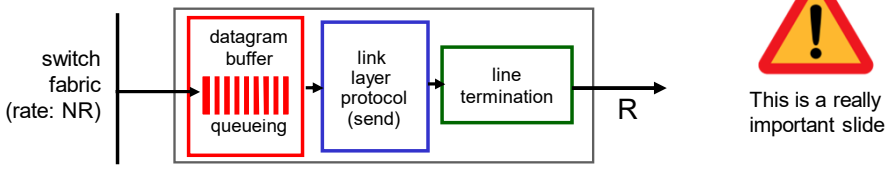


output port contention: only one red datagram can be transferred. lower red packet is *blocked*



one packet time later: green packet experiences HOL blocking

Output port queuing



This is a really important slide

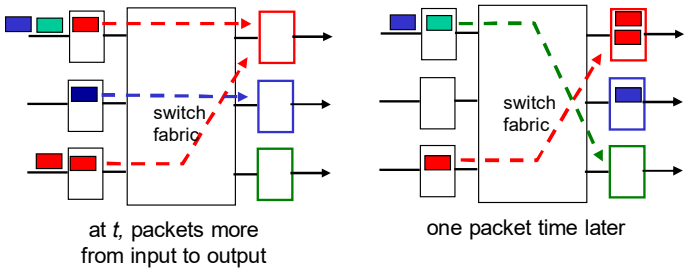
Buffering required when datagrams arrive from fabric faster than link transmission rate. *Drop policy:* which datagrams to drop if no free buffers?

→ Datagrams can be lost due to congestion, lack of buffers

Scheduling discipline chooses among queued datagrams for transmission

→ Priority scheduling – who gets best performance, network neutrality

Output port queuing



at *t*, packets more from input to output

one packet time later

- buffering when arrival rate via switch exceeds output line speed
- *queueing (delay) and loss due to output port buffer overflow!*

How much buffering?

RFC 3439 rule of thumb: average buffering equal to “typical” RTT (say 250 msec) times link capacity C

- e.g., C = 10 Gbps link: 2.5 Gbit buffer

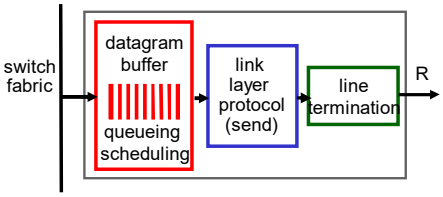
more recent recommendation: with *N* flows, buffering equal to

$$\frac{RTT \cdot C}{\sqrt{N}}$$

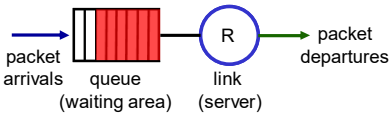
but *too* much buffering can increase delays (particularly in home routers)

- long RTTs: poor performance for realtime apps, sluggish TCP response
- recall delay-based congestion control: “keep bottleneck link just full enough (busy) but no fuller”

Buffer Management



Abstraction: queue



buffer management:

- drop: which packet to add, drop when buffers are full
  - tail drop: drop arriving packet
  - priority: drop/remove on priority basis
- marking: which packets to mark to signal congestion (ECN, RED)

Packet Scheduling: FCFS

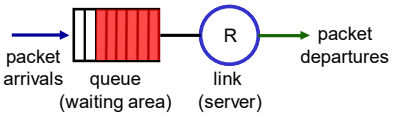
packet scheduling: deciding which packet to send next on link

- first come, first served
- priority
- round robin
- weighted fair queueing

FCFS: packets transmitted in order of arrival to output port

- also known as: First-in-first-out (FIFO)
- real world examples?

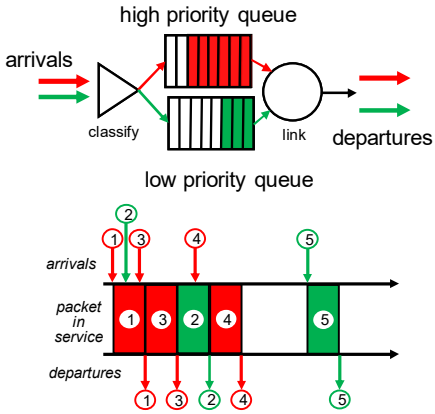
Abstraction: queue



Scheduling policies: priority

Priority scheduling:

- arriving traffic classified, queued by class
  - any header fields can be used for classification
- send packet from highest priority queue that has buffered packets
  - FCFS within priority class



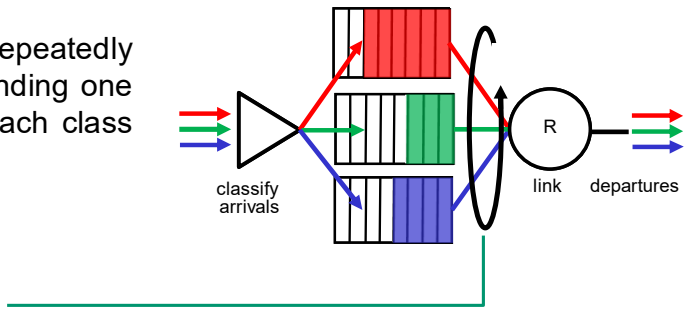
Scheduling policies: round robin

Round Robin (RR) scheduling:

arriving traffic classified, queued by class

- any header fields can be used for classification

server cyclically, repeatedly scans class queues, sending one complete packet from each class (if available) in turn



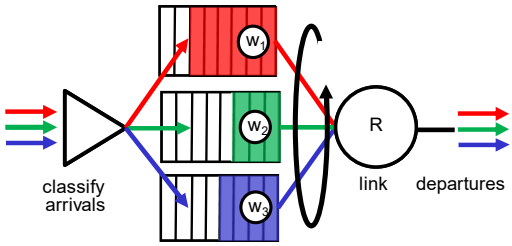
Scheduling policies: weighted fair queueing

Weighted Fair Queuing (WFQ):

- generalized Round Robin
- each class, *i,* has weight, w_i, and gets weighted amount of service in each cycle:

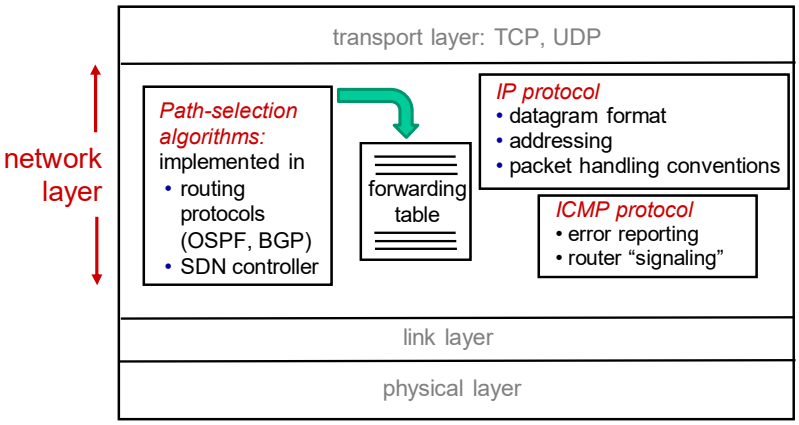
$$\frac{w_i}{\sum_j w_j}$$

- minimum bandwidth guarantee (per-traffic-class)

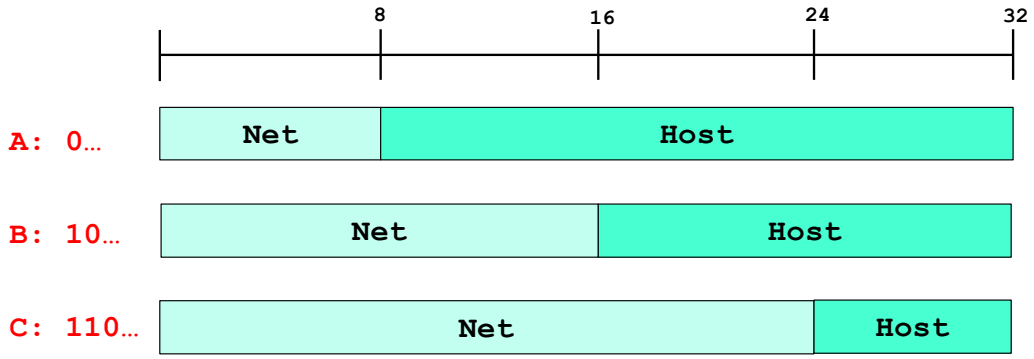


Network Layer: Internet

host, router network layer functions:



IPv4 addresses



Class	From	To
A	1.0.0.0	127.255.255.255
B	128.0.0.0	191.255.255.255
C	192.0.0.0	223.255.255.255
D (multicast)	224.0.0.0	239.255.255.255
E (riservati)	240.0.0.0	

IPv4 addresses

`0.0.0.0`	This host
`00000….xxxxxxxxxx`	a host in the subnet
`255.255.255.255`	Broadcast
`127.0.0.0 – 127.255.255.255`	Loopback
`10.0.0.0 – 10.255.255.255`	Private IP
`169.254.0.0 – 169.254.255.255`	Zero conf. net. APIPA
`172.16.0.0 – 172.31.255.255`	Private IP
`192.168.0.0 – 192.168.255.255`	Private IP

IP Address Assignment

Initially, it was carried out by the Internet Assigned Numbers Authority (IANA) (www.iana.org).

In 1992, Regional Internet Registries (RIRs) were created to manage the addresses and domain names for each assigned area.

In 1998, a non-profit organization called ICANN (Internet Corporation for Assigned Network and Numbers, www.icann.org) was founded under the control of the U.S. Department of Commerce to coordinate all RIRs in the world.

IP Address Assignment

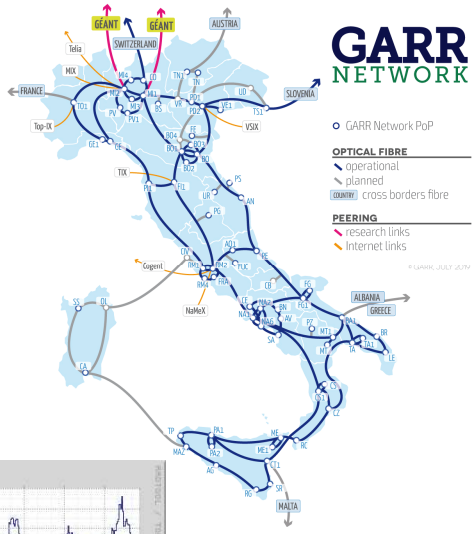
Today, there are five RIRs:

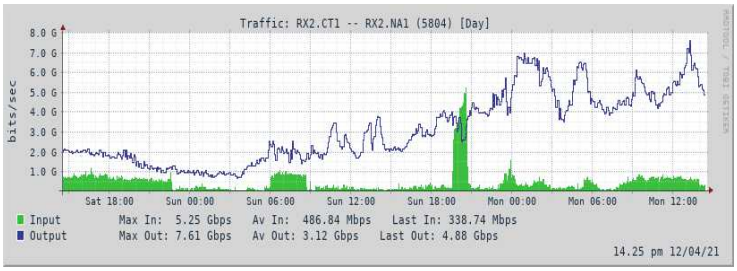
- APNIC (Asia Pacific Network Information Center): www.apnic.net.
- ARIN (American Registry for Internet Numbers): www.arin.net.
- LACNIC (Latin American and Caribbean Internet Addresses Registry): www.lacnic.net.
- RIPE NCC (Réseaux IP Européens Network Coordination Centre): www.ripe.net.
- AFRINIC (African Regional Internet Registry): www.afrinic.org.

IP Address Assignment

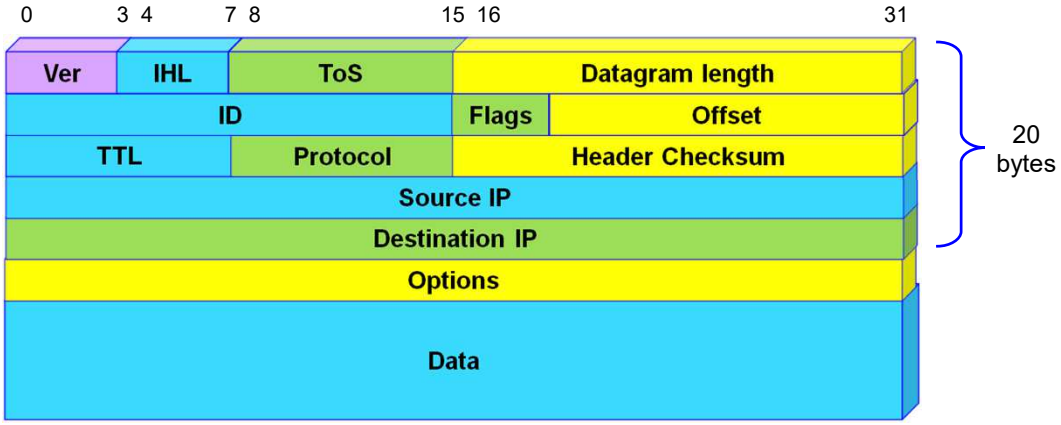
RIRs are responsible for LIRs (Local Internet Registries) to manage addresses at the local level.

GARR-LIR

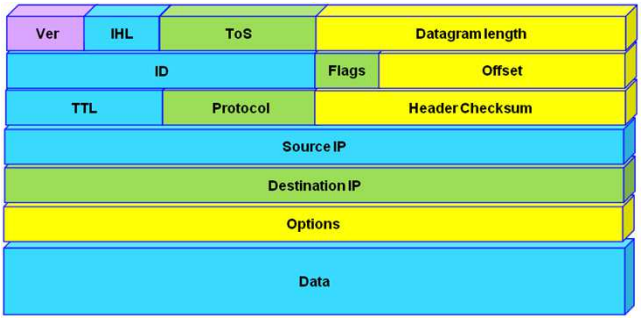




IPv4 Datagram Format



IPv4 Datagram Format



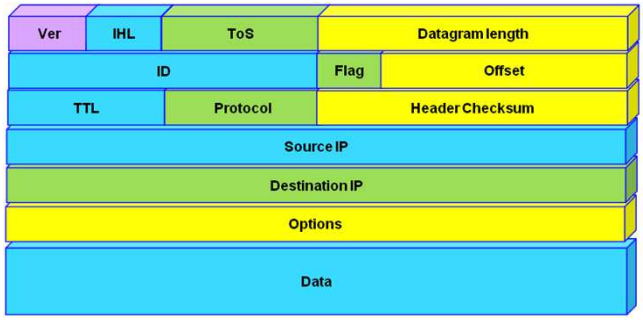
Version: 4 bits.

IHL: 4 bits - Preamble size, in 4-bytes groups

ToS: Type of service: 3 bits as priority and 3 flags. It is not used.

Datagram Length: Total length of the datagram. 16 bits => 64kB

IPv4 Datagram Format



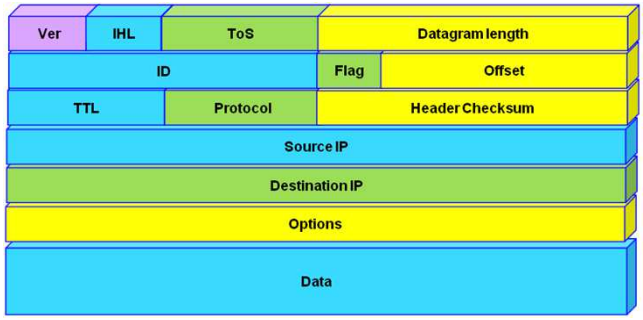
ID: Identify the datagram and its fragments.

FLAG (3 bits)

- Not used
- DF Don't fragment
- MF: More fragment

Fragment Offset: 13 bit => 8192 fragments of at least 8 bytes

IPv4 Datagram Format



TTL: Time To Live

Protocol: Specifies the transport layer protocol

Header checksum: checksum of the preamble.

Options

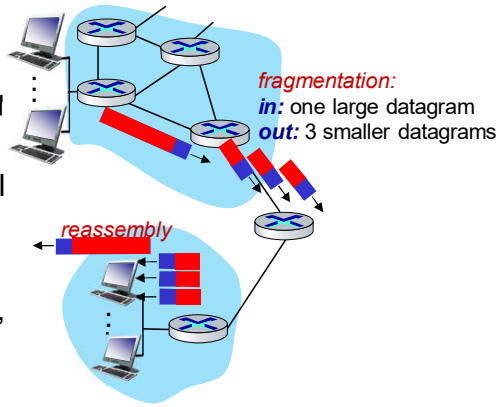
IP fragmentation/reassembly

network links have MTU (max. transfer size) - largest possible link-level frame

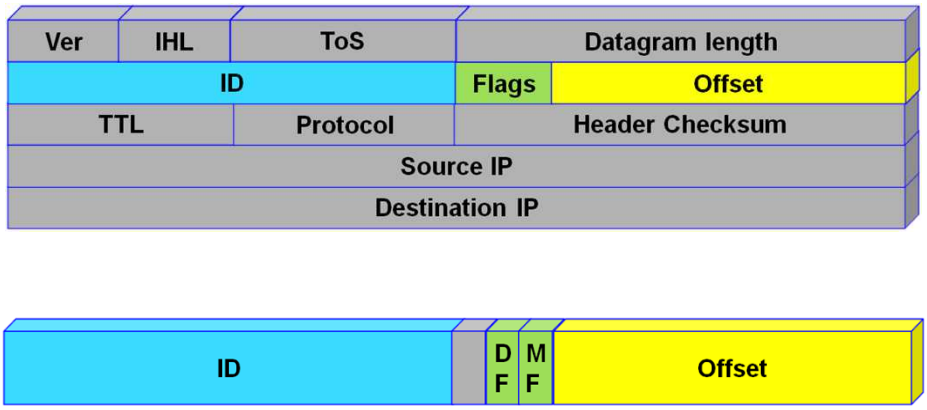
- different link types, different MTUs

large IP datagram divided ("fragmented") within net

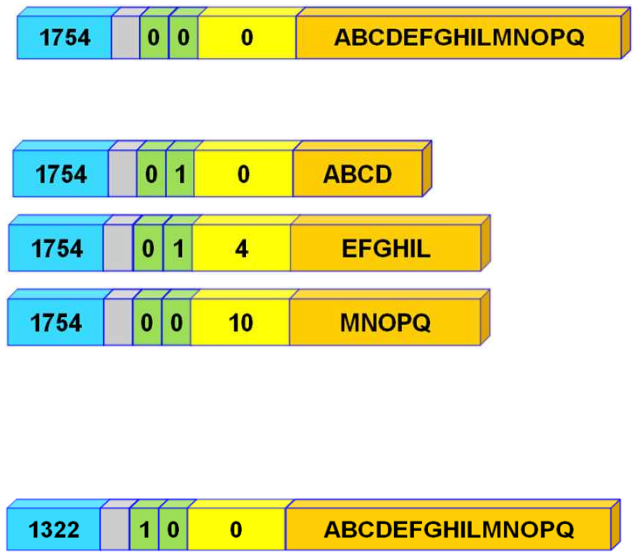
- one datagram becomes several datagrams
- "reassembled" only at *destination*
- IP header bits used to identify, order related fragments



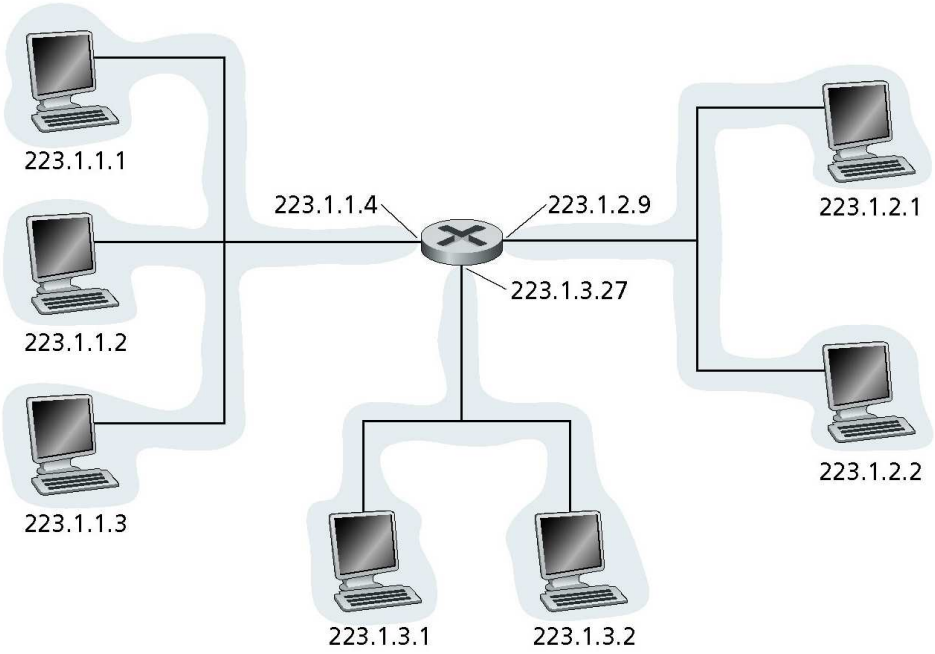
Fragmentation



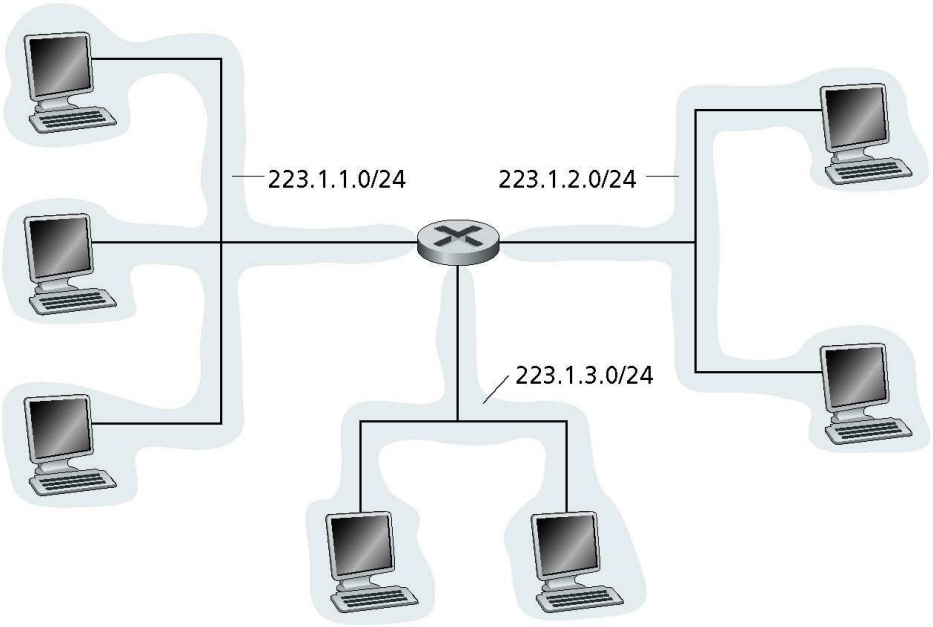
Fragmentation



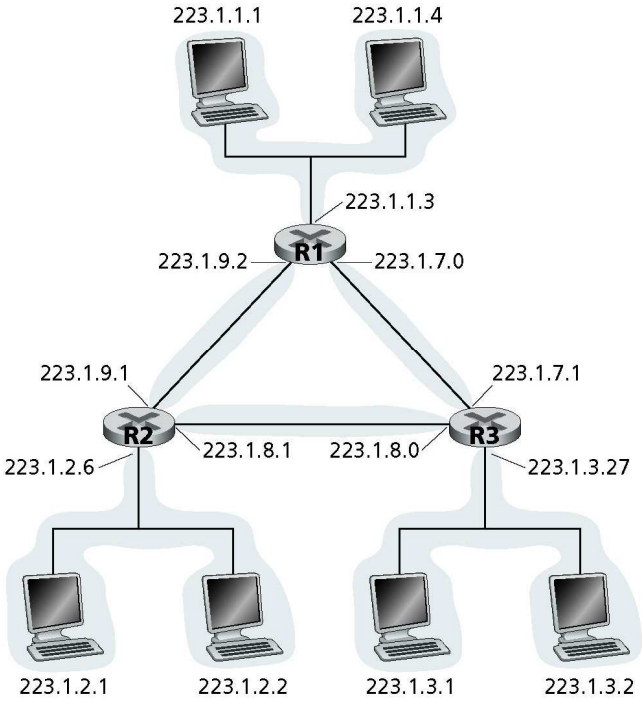
IPv4 Addressing



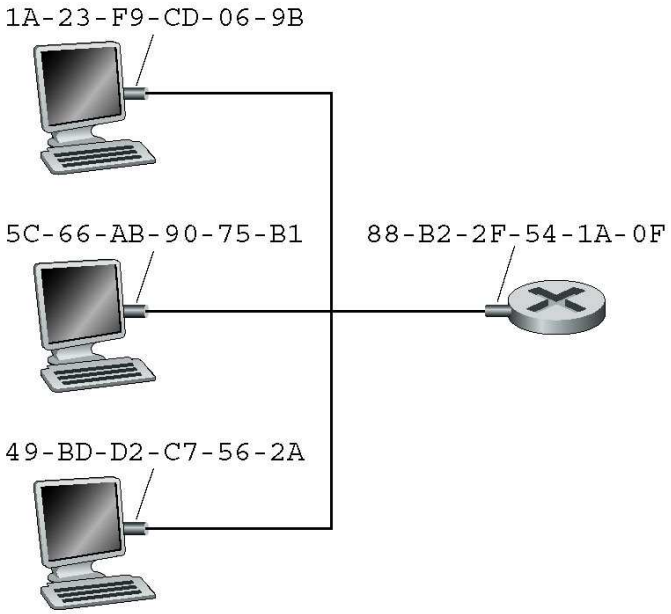
IPv4 Addressing



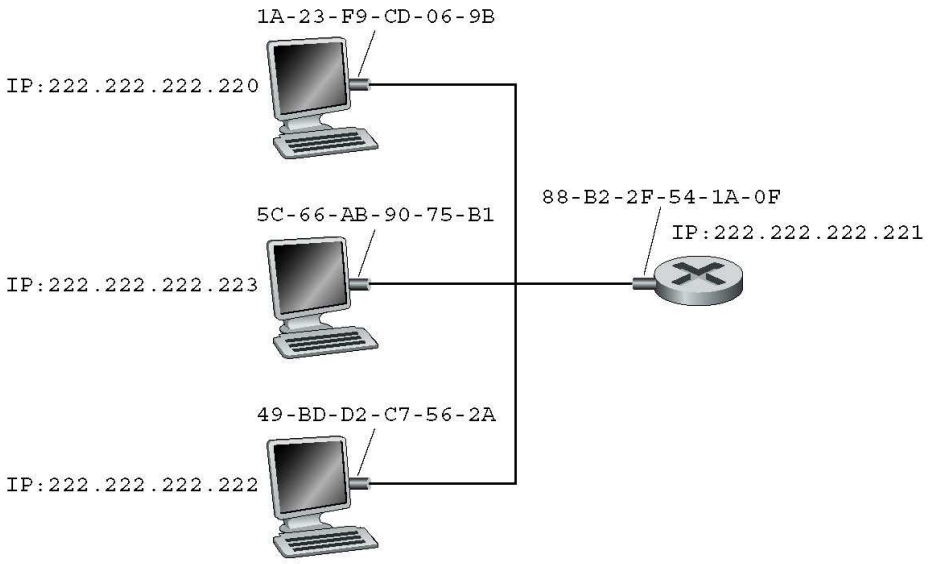
IPv4 Addressing



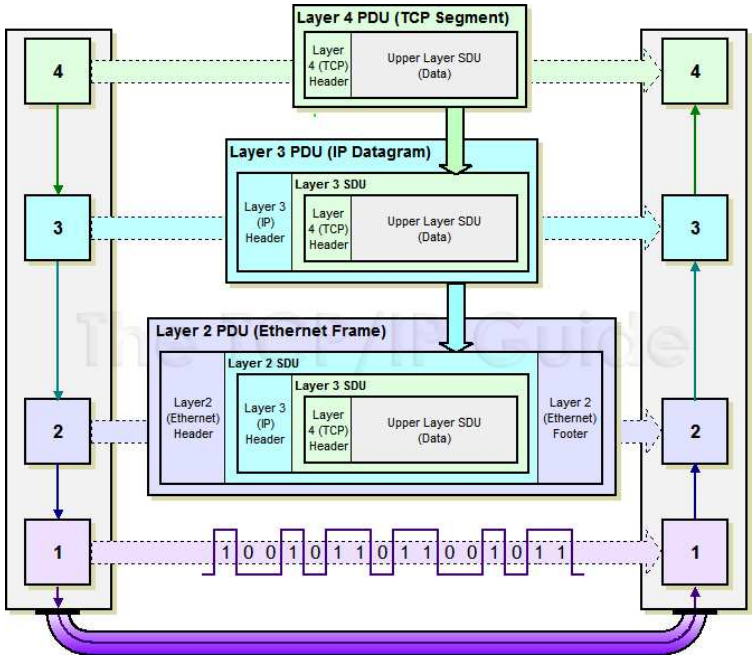
IntraLAN Addressing



IntraLAN Addressing

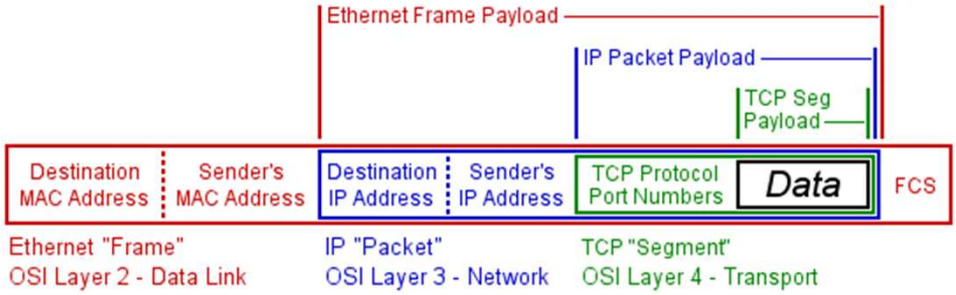


LAN Addresses and IPv4



LAN Addresses and IPv4

Encapsulation Payloads



LAN Addresses and IPv4

No.	Time	Source	Destination	Protocol	Length	Info
30	4.203963853	proxy03.mirror.garr.it	Toshiba-Z830.local	TCP	66	80 → 55492 [SYN, ACK
31	4.203985642	Toshiba-Z830.local	proxy03.mirror.garr…	TCP	54	55492 → 80 [ACK] Seq=
32	4.209664430	Toshiba-Z830.local	proxy03.mirror.garr…	HTTP	802	GET / HTTP/1.1
33	4.213469694	mil04s25-in-f74.1e100.net	Toshiba-Z830.local	TCP	74	80 → 58828 [SYN, ACK
34	4.213522703	Toshiba-Z830.local	mil04s25-in-f74.1e1…	TCP	66	58828 → 80 [ACK] Seq=
35	4.222429261	proxy03.mirror.garr.it	Toshiba-Z830.local	TCP	60	80 → 55484 [ACK] Seq=

```
Frame 32: 802 bytes on wire (6416 bits), 802 bytes captured (6416 bits) on interface 0
Ethernet II, Src: Toshiba_dc:c7:5e (e8:e0:b7:dc:c7:5e), Dst: Routerbo_13:e0:1c (4c:5e:0c:13:e0:1c)
Internet Protocol Version 4, Src: Toshiba-Z830.local (192.168.65.167), Dst: proxy03.mirror.garr.it (90.147.160.73)
Transmission Control Protocol, Src Port: 55484, Dst Port: 80, Seq: 1, Ack: 1, Len: 748
Hypertext Transfer Protocol
  ▸ GET / HTTP/1.1\r\n
    Host: mirror.garr.it\r\n
    User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:65.0) Gecko/20100101 Firefox/65.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
    Accept-Language: it-IT,it;q=0.8,en-US;q=0.5,en;q=0.3\r\n
    Accept-Encoding: gzip, deflate\r\n
    Referer: https://it.search.yahoo.com/\r\n
    Connection: keep-alive\r\n
  ▸ Cookie: __utma=225464966.2039547598.1552470368.1552470368.1552470368.1; __utmb=225464966.1.10.1552470368; __utmc=225
    Upgrade-Insecure-Requests: 1\r\n
    If-Modified-Since: Sat, 10 Nov 2018 08:45:07 GMT\r\n
```

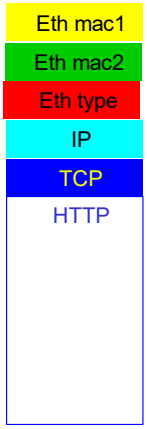
```
000  4c 5e 0c 13 e0 1c e8 e0  b7 dc c7 5e 08 00 45 00   L^······ ···^··E·
010  03 14 93 d1 40 00 40 06  a6 e6 c0 a8 41 a7 5a 93   ····@·@· ····A·Z·
020  a0 49 d8 bc 00 50 60 10  5a be ca 22 4d 02 50 18   ·I···P`· Z··"M·P·
030  00 e5 00 33 00 00 47 45  54 20 2f 20 48 54 54 50   ···3··GE T / HTTP
040  2f 31 2e 31 0d 0a 48 6f  73 74 3a 20 6d 69 72 72   /1.1··Ho st: mirr
050  6f 72 2e 67 61 72 72 2e  69 74 0d 0a 55 73 65 72   or.garr. it··User
060  2d 41 67 65 6e 74 3a 20  4d 6f 7a 69 6c 6c 61 2f   -Agent:  Mozilla/
070  35 2e 30 20 28 58 31 31  3b 20 55 62 75 6e 74 75   5.0 (X11 ; Ubuntu
080  3b 20 4c 69 6e 75 78 20  78 38 36 5f 36 34 3b 20   ; Linux  x86_64;
090  72 76 3a 36 35 2e 30 29  20 47 65 63 6b 6f 2f 32   rv:65.0)  Gecko/2
0a0  30 31 30 30 31 30 31 20  46 69 72 65 66 6f 78 2f   0100101  Firefox/
0b0  36 35 2e 30 0d 0a 41 63  63 65 70 74 3a 20 74 65   65.0··Ac cept: te
0c0  78 74 2f 68 74 6d 6c 2c  61 70 70 6c 69 63 61 74   xt/html, applicat
0d0  69 6f 6e 2f 78 68 74 6d  6c 2b 78 6d 6c 2c 61 70   ion/xhtm l+xml,ap
0e0  70 6c 69 63 61 74 69 6f  6e 2f 78 6d 6c 3b 71 3d   plicatio n/xml;q=
0f0  30 2e 39 2c 69 6d 61 67  65 2f 77 65 62 70 2c 2a   0.9,imag e/webp,*
100  2f 2a 3b 71 3d 30 2e 38  0d 0a 41 63 63 65 70 74   /*;q=0.8 ··Accept
```

LAN Addresses and IPv4

No.	Time	Source	Destination	Protocol	Length	Info
30	4.203963853	proxy03.mirror.garr.it	Toshiba-Z830.local	TCP	66	80 → 55492 [SYN, ACK
31	4.203985642	Toshiba-Z830.local	proxy03.mirror.garr…	TCP	54	55492 → 80 [ACK] Seq=
32	4.209664430	Toshiba-Z830.local	proxy03.mirror.garr…	HTTP	802	GET / HTTP/1.1
33	4.213469694	mil04s25-in-f74.1e100.net	Toshiba-Z830.local	TCP	74	80 → 58828 [SYN, ACK
34	4.213522703	Toshiba-Z830.local	mil04s25-in-f74.1e1…	TCP	66	58828 → 80 [ACK] Seq=
35	4.222429261	proxy03.mirror.garr.it	Toshiba-Z830.local	TCP	60	80 → 55484 [ACK] Seq=

```
Frame 32: 802 bytes on wire (6416 bits), 802 bytes captured (6416 bits) on interface 0
Ethernet II, Src: Toshiba_dc:c7:5e (e8:e0:b7:dc:c7:5e), Dst: Routerbo_13:e0:1c (4c:5e:0c:13:e0:1c)
Internet Protocol Version 4, Src: Toshiba-Z830.local (192.168.65.167), Dst: proxy03.mirror.garr.it (90.147.160.73)
Transmission Control Protocol, Src Port: 55484, Dst Port: 80, Seq: 1, Ack: 1, Len: 748
Hypertext Transfer Protocol
  GET / HTTP/1.1\r\n
  Host: mirror.garr.it\r\n
  User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:65.0) Gecko/20100101 Firefox/65.0\r\n
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
  Accept-Language: it-IT,it;q=0.8,en-US;q=0.5,en;q=0.3\r\n
  Accept-Encoding: gzip, deflate\r\n
  Referer: https://it.search.yahoo.com/\r\n
  Connection: keep-alive\r\n
```

```
000 4c 5e 9c 13 e0 1c e8 e0 b7 de c7 5e 08 00 45 00
010 03 14 93 d1 40 00 40 06 a6 e6 c0 a8 41 a7 5a 93
020 a0 49 d8 bc 00 50 60 10 5a be ca 22 4d 02 50 18
030 00 e5 00 33 00 00 47 45 54 20 2f 20 48 54 54 50
040 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 6d 69 72 72
050 6f 72 2e 67 61 72 72 2e 69 74 9d 0a 55 73 65 72
060 2d 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f
070 35 2e 30 20 28 58 31 31 3b 20 55 62 75 6e 74 75
080 3b 20 4c 69 6e 75 78 20 78 38 36 5f 36 34 3b 20
090 72 76 3a 36 35 2e 39 29 20 47 65 63 6b 6f 2f 32
0a0 30 31 30 30 31 30 31 20 46 69 72 65 66 6f 78 2f
0b0 36 35 2e 30 0d 0a 41 63 63 65 70 74 3a 20 74 65
0c0 78 74 2f 68 74 6d 6c 2c 61 70 70 6c 69 63 61 74
0d0 69 6f 6e 2f 78 68 74 6d 6c 2b 78 6d 6c 2c 61 70
0e0 70 6c 69 63 61 74 69 6f 6e 2f 78 6d 6c 3b 71 3d
0f0 39 2e 39 2c 69 6d 61 67 65 2f 77 65 62 70 2c 2a
100 2f 2a 3b 71 3d 30 2e 38 0d 0a 41 63 63 65 70 74
```

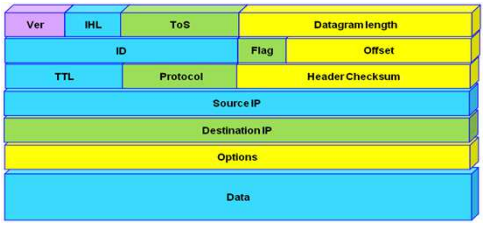


LAN Addresses and IPv4

No.	Time	Source	Destination	Protocol	Length	Info
30	4.203963853	proxy03.mirror.garr.it	Toshiba-Z830.local	TCP	66	80 → 55492 [SYN, ACK
31	4.203985642	Toshiba-Z830.local	proxy03.mirror.garr…	TCP	54	55492 → 80 [ACK] Seq=
32	4.209664430	Toshiba-Z830.local	proxy03.mirror.garr…	HTTP	802	GET / HTTP/1.1
33	4.213469694	mil04s25-in-f74.1e100.net	Toshiba-Z830.local	TCP	74	80 → 58828 [SYN, ACK
34	4.213522703	Toshiba-Z830.local	mil04s25-in-f74.1e1…	TCP	66	58828 → 80 [ACK] Seq=
35	4.222429261	proxy03.mirror.garr.it	Toshiba-Z830.local	TCP	60	80 → 55484 [ACK] Seq=

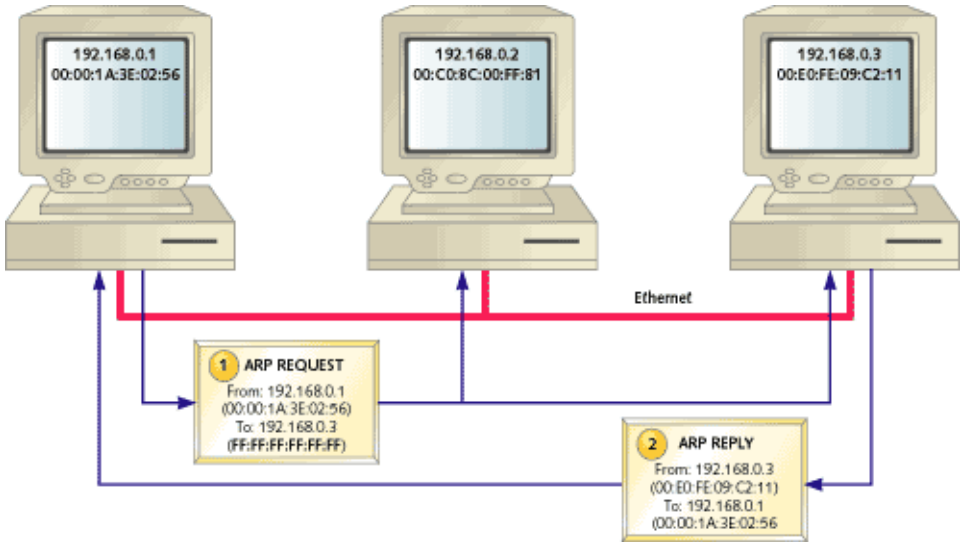
```
Frame 32: 802 bytes on wire (6416 bits), 802 bytes captured (6416 bits) on interface 0
Ethernet II, Src: Toshiba_dc:c7:5e (e8:e0:b7:dc:c7:5e), Dst: Routerbo_13:e0:1c (4c:5e:0c:13:e0:1c)
Internet Protocol Version 4, Src: Toshiba-Z830.local (192.168.65.167), Dst: proxy03.mirror.garr.it (90.147.160.73)
Transmission Control Protocol, Src Port: 55484, Dst Port: 80, Seq: 1, Ack: 1, Len: 748
```

```
45 00 03 14
93 d1 40 00
40 06 a6 e6
c0 a8 41 a7
5a 93 a0 49
d8 bc 00 50
60 10 5a be
ca 22 4d 02
50 18 00 e5
00 33 00 00
```



Ver 4 IHL 5 ToS 00 DL 0314 (788 = 802-6-6-2)
ID 93D1 Flag + Offset 010 0000000000000
TTL 40 (64) Protocol 06 (TCP) HC A6E6
Source IP C0 (192) A8 (168) 41 (065) A7 (167)
Dest IP 5A (090) 93 (147) A0 (160) 49 (073)

The ARP protocol

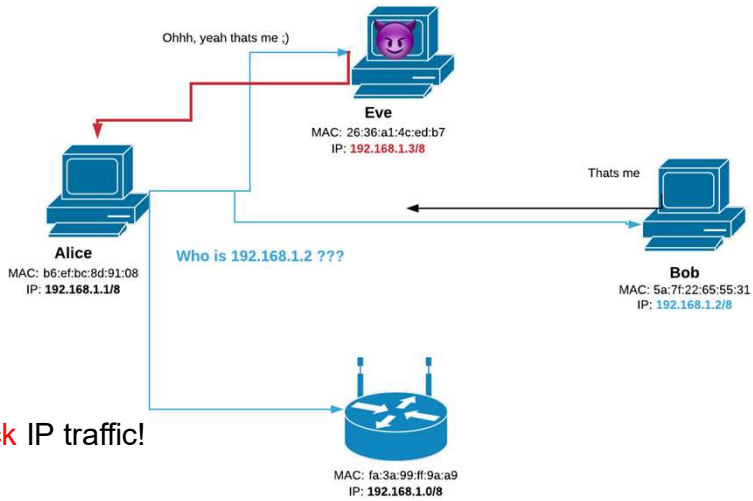


ARP table

Interface	IP Address	Physical Address	Type	Expiry
eth0	192.168.1.1	00:21:29:CC:A7:31	Static	-
eth0	192.168.1.24	00:17:31:BA:5C:A4	Dynamic	2m 27s
eth1	10.1.2.66	00:1A:70:3C:A6:3D	Dynamic	35s
wlan0	210.210.3.1	90:6C:AC:2D:DE:9A	Dynamic	1m 14s

Vulnerability

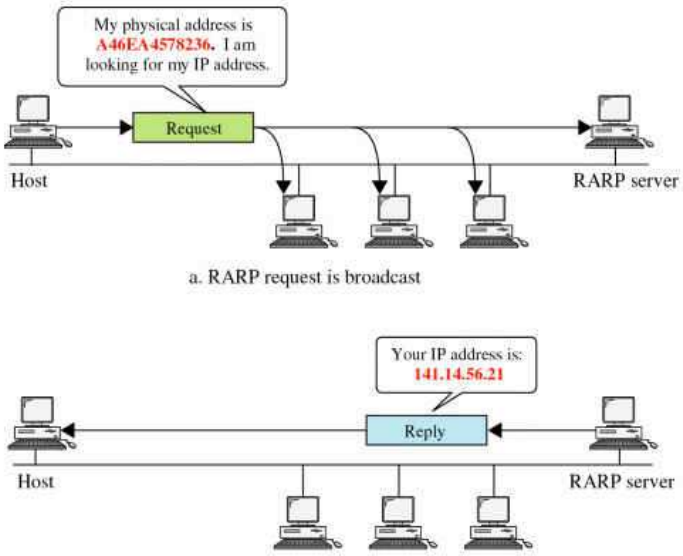
- ARP does not require authentication.
- ARP is a stateless protocol: an ARP Reply can be sent without a previous ARP Request.
- A host that receives an ARP packet has to update its ARP cache.



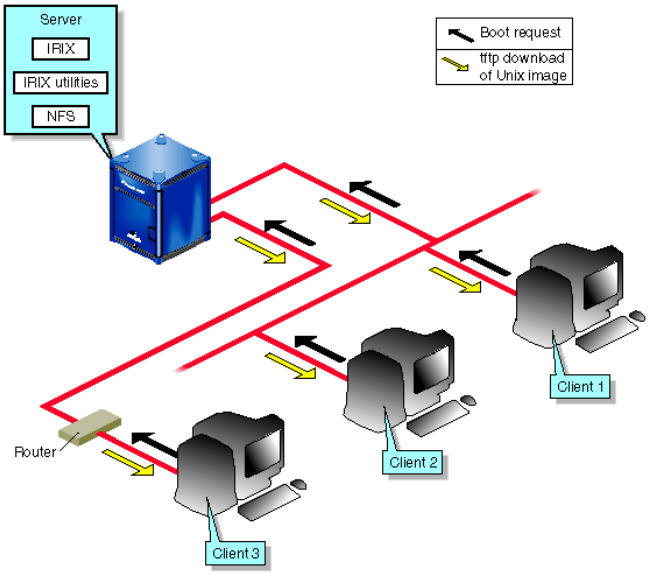
Aftermath:

It is possible to hijack IP traffic!

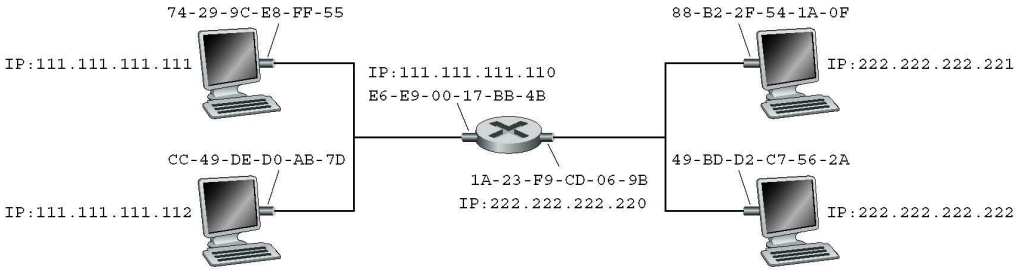
RARP


My physical address is A46EA4578236. I am looking for my IP address.
Request
Host
RARP server
a. RARP request is broadcast
Your IP address is: 141.14.56.21
Reply
Host
RARP server

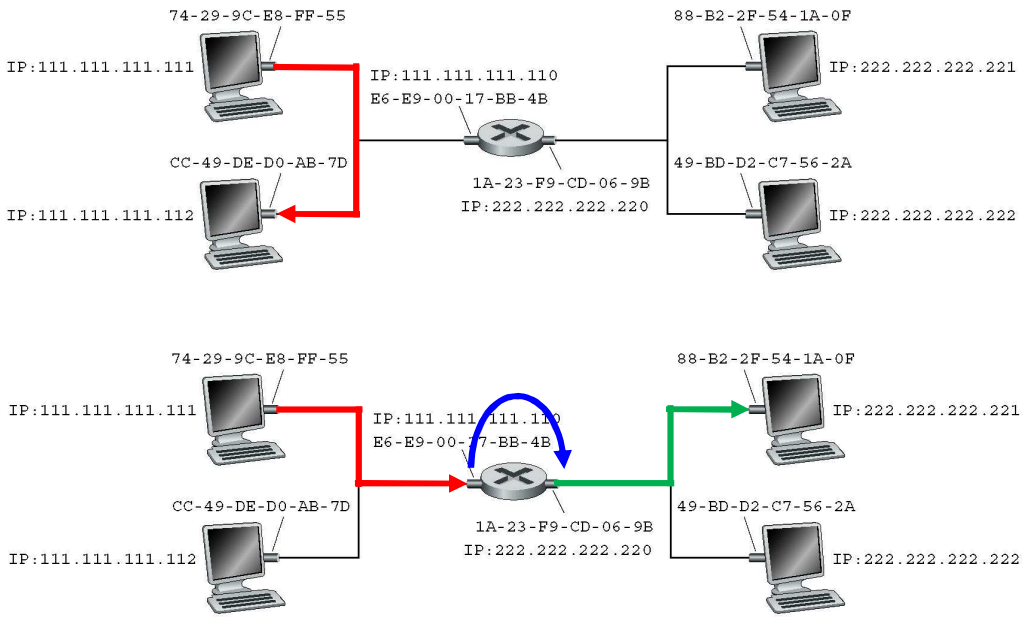
BOOTP


Server
IRIX
IRIX utilities
NFS
Boot request
tftp download of Unix image
Router
Client 1
Client 2
Client 3

Interconnected LAN



Interconnected LAN



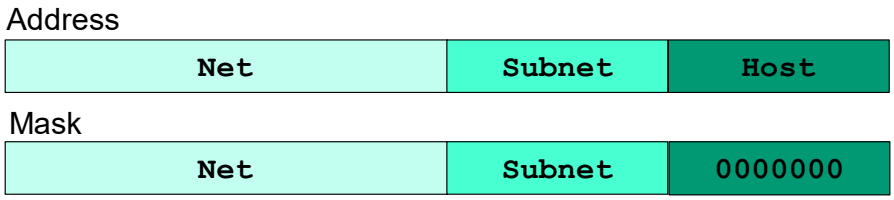
Subnet masks

CIDR (Classless Inter-Domain Routing)

VLSM (Variable Length Subnet Masking)

a.b.c.d/x

Subnet masks




```
10010111.01100001.11111100.01000010   151.097.252.066
11111111.11111111.11111110.00000000   255.255.254.000
```

AND

```
10010111.01100001.11111100.00000000   151.097.252.000
```

Subnet masks


```
151.097.252.000/23
10010111.01100001.1111110 0.00000000

10010111.01100001.1111110 0.00000001 151.097.252.001
10010111.01100001.1111110 0.00000010 151.097.252.002
…
10010111.01100001.1111110 1.11111110 151.097.253.254
```

151.97.252.0/23 All the subnet

151.97.253.255 Broadcast in the subnet

Subnet masks


```
191.15.12.24/30
10111111.00001111.00001100.00011000    network

191.15.12.24    10111111.00001111.00001100.00011000
191.15.12.25    10111111.00001111.00001100.00011001
191.15.12.26    10111111.00001111.00001100.00011010
191.15.12.27    10111111.00001111.00001100.00011011
```

Subnet masks

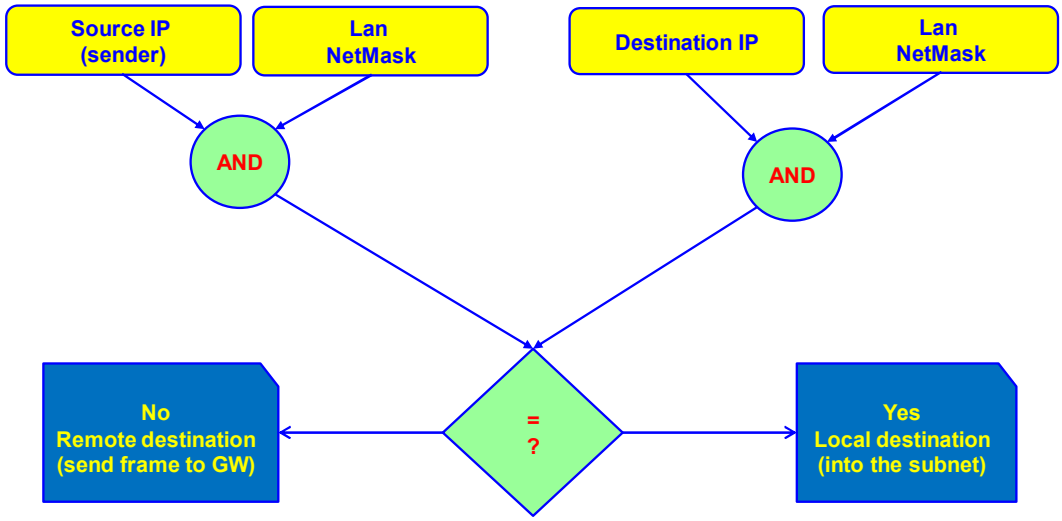
/31

A 31-bit subnet mask is often used for an interface that is the endpoint of a point-to-point network. The use of 31-bit subnet masks for IPv4 point-to-point links is described in RFC 3021.

/32

A 32-bit subnet mask defines a network with only one IP address.

Use of Netmasks



Host Table

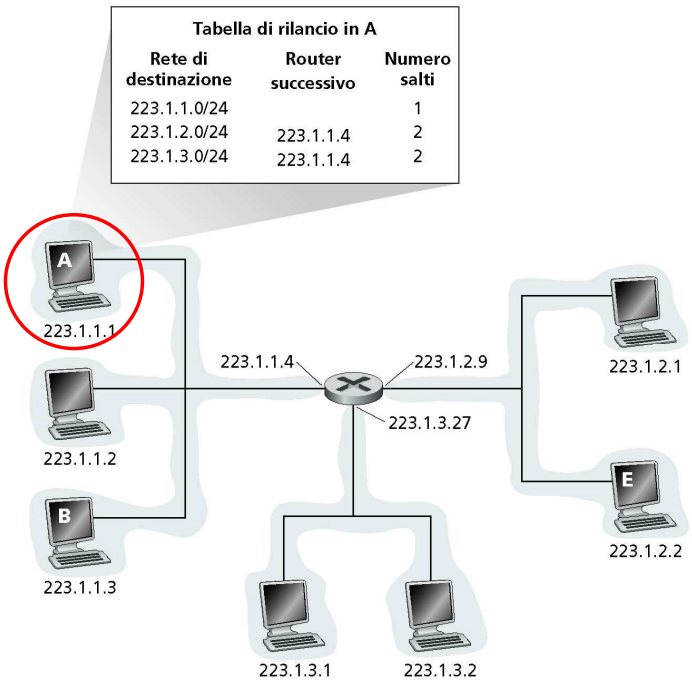
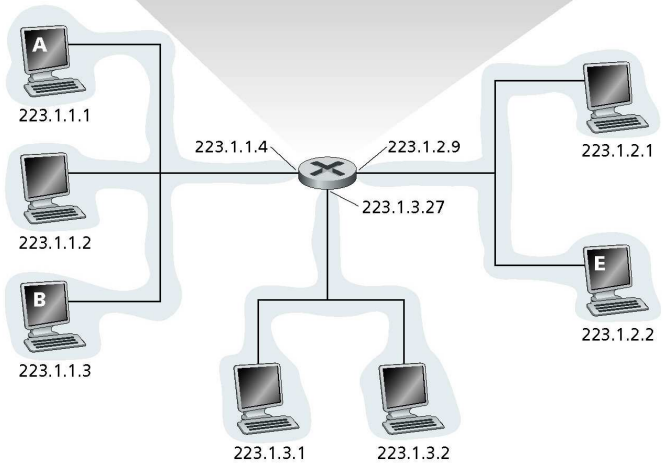


Tabella di rilancio in A
Rete di destinazione
Router successivo
Numero salti
223.1.1.0/24
1
223.1.2.0/24
223.1.1.4
2
223.1.3.0/24
223.1.1.4
2
A
223.1.1.1
223.1.1.2
B
223.1.1.3
223.1.1.4
223.1.2.9
223.1.3.27
223.1.2.1
E
223.1.2.2
223.1.3.1
223.1.3.2

Routing Table

Tabella di rilancio nel router

Rete di destinaz.	Router success.	N. hop	Interfaccia
223.1.1.0/24	—	1	223.1.1.4
223.1.2.0/24	—	1	223.1.2.9
223.1.3.0/24	—	1	223.1.3.27



Example: routing table

```
Scheda Ethernet Ethernet:

   Suffisso DNS specifico per connessione: dmi.unict.it
   Indirizzo IPv6 locale rispetto al collegamento : fe80::2430:ee60:719:1e6c%6
   Indirizzo IPv4. . . . . . . . . . . . : 192.168.65.156
   Subnet mask . . . . . . . . . . . . . : 255.255.255.0
   Gateway predefinito . . . . . . . . . : 192.168.65.1

Scheda Ethernet Ethernet 3:

   Suffisso DNS specifico per connessione:
   Indirizzo IPv6 locale rispetto al collegamento : fe80::c5f:4e6:d1c1:c3d2%39
   Indirizzo IPv4. . . . . . . . . . . . : 192.168.56.1
   Subnet mask . . . . . . . . . . . . . : 255.255.255.0
   Gateway predefinito . . . . . . . . . :


Scheda LAN wireless Wi-Fi:

   Suffisso DNS specifico per connessione: wluct.unict.it
   Indirizzo IPv6 locale rispetto al collegamento : fe80::b176:bbcd:561:97ae%54
   Indirizzo IPv4. . . . . . . . . . . . : 151.97.62.134
   Subnet mask . . . . . . . . . . . . . : 255.255.248.0
   Gateway predefinito . . . . . . . . . : 151.97.56.1

   Stato supporto. . . . . . . . . . . . : Supporto disconnesso
   Suffisso DNS specifico per connessione: wluct.unict.it
```

Example: routing table

```
C:\> route print
===========================================================================
Elenco interfacce
 55...........................Wintun Userspace Tunnel
  6...00 1c c2 4b 8a b3 ......Realtek USB GbE Family Controller
 39...0a 00 27 00 00 27 ......VirtualBox Host-Only Ethernet Adapter
 18...00 ff 51 02 c2 ab ......TAP-Windows Adapter V9
 35...38 fc 98 b0 b3 56 ......Microsoft Wi-Fi Direct Virtual Adapter
  4...3a fc 98 b0 b3 55 ......Microsoft Wi-Fi Direct Virtual Adapter #2
 54...38 fc 98 b0 b3 55 ......Intel(R) Wi-Fi 6 AX200 160MHz
 17...38 fc 98 b0 b3 59 ......Bluetooth Device (Personal Area Network)
  1...........................Software Loopback Interface 1
 28...94 8f 85 51 5b 53 ......Generic Mobile Broadband Adapter
===========================================================================
```

Example: routing table

```
IPv4 Tabella route
===========================================================================
Route attive:
     Indirizzo rete            Mask          Gateway       Interfaccia
Metrica
          0.0.0.0          0.0.0.0      151.97.56.1    151.97.62.134     45
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    331
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    331
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    331
      151.97.56.0    255.255.248.0         On-link     151.97.62.134    301
    151.97.62.134  255.255.255.255         On-link     151.97.62.134    301
    151.97.63.255  255.255.255.255         On-link     151.97.62.134    301
     192.168.56.0    255.255.255.0         On-link      192.168.56.1    281
     192.168.56.1  255.255.255.255         On-link      192.168.56.1    281
   192.168.56.255  255.255.255.255         On-link      192.168.56.1    281
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    331
        224.0.0.0        240.0.0.0         On-link      192.168.56.1    281
        224.0.0.0        240.0.0.0         On-link     151.97.62.134    301
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    331
  255.255.255.255  255.255.255.255         On-link      192.168.56.1    281
  255.255.255.255  255.255.255.255         On-link     151.97.62.134    301
===========================================================================
Route permanenti:
  Nessuna
```

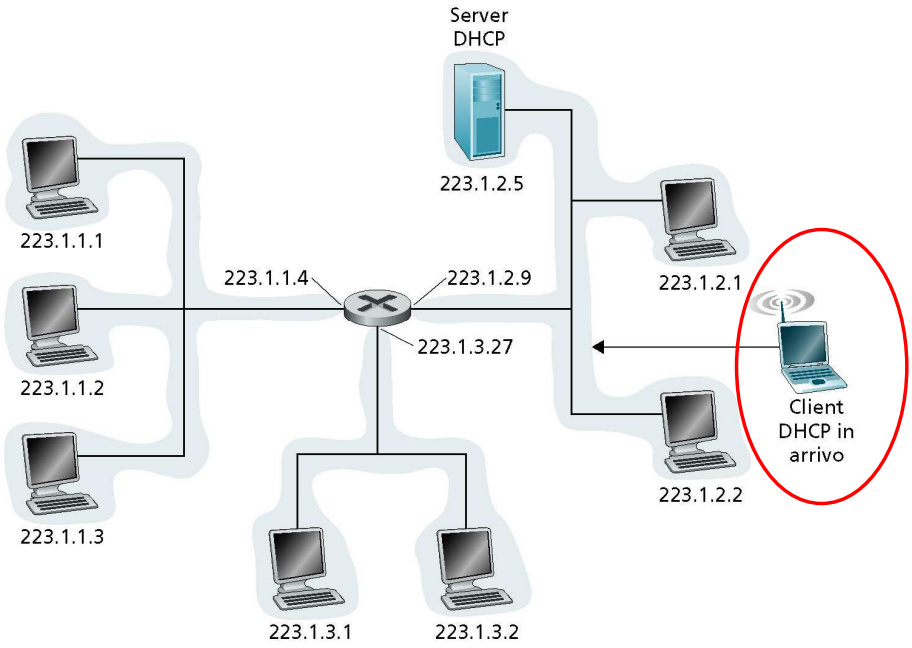
Example: routing table

```
IPv4 Tabella route
===========================================================================
Route attive:
     Indirizzo rete            Mask          Gateway      Interfaccia
Metrica
          0.0.0.0          0.0.0.0      192.168.65.1   192.168.65.156     25
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    331
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    331
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    331
     192.168.56.0    255.255.255.0         On-link       192.168.56.1    281
     192.168.56.1  255.255.255.255         On-link       192.168.56.1    281
   192.168.56.255  255.255.255.255         On-link       192.168.56.1    281
     192.168.65.0    255.255.255.0         On-link     192.168.65.156    281
   192.168.65.156  255.255.255.255         On-link     192.168.65.156    281
   192.168.65.255  255.255.255.255         On-link     192.168.65.156    281
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    331
        224.0.0.0        240.0.0.0         On-link       192.168.56.1    281
        224.0.0.0        240.0.0.0         On-link     192.168.65.156    281
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    331
  255.255.255.255  255.255.255.255         On-link       192.168.56.1    281
  255.255.255.255  255.255.255.255         On-link     192.168.65.156    281
===========================================================================
Route permanenti:
  Nessuna
```

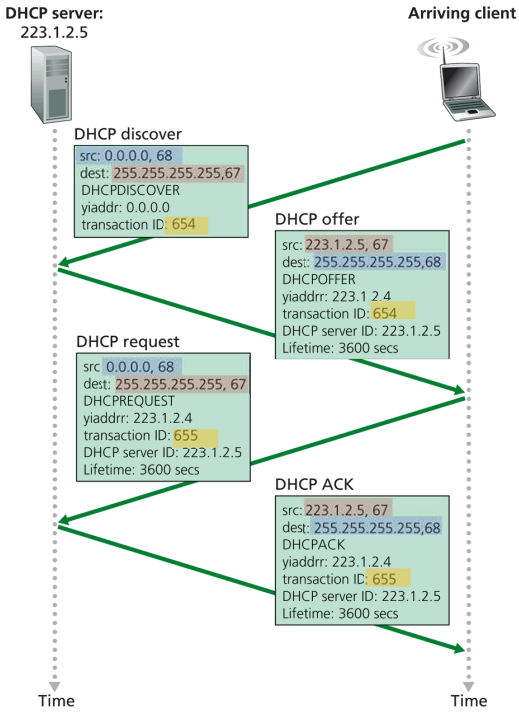
Example: routing table

```
IPv6 Tabella route
===========================================================================
Route attive:
 Interf Metrica Rete Destinazione      Gateway
  1    331 ::1/128                  On-link
 39    281 fe80::/64                On-link
  6    281 fe80::/64                On-link
 39    281 fe80::c5f:4e6:d1c1:c3d2/128
                                   On-link
  6    281 fe80::2430:ee60:719:1e6c/128
                                   On-link
  1    331 ff00::/8                 On-link
 39    281 ff00::/8                 On-link
  6    281 ff00::/8                 On-link
===========================================================================
Route permanenti:
  Nessuna
```

Address assignment: DHCPv4

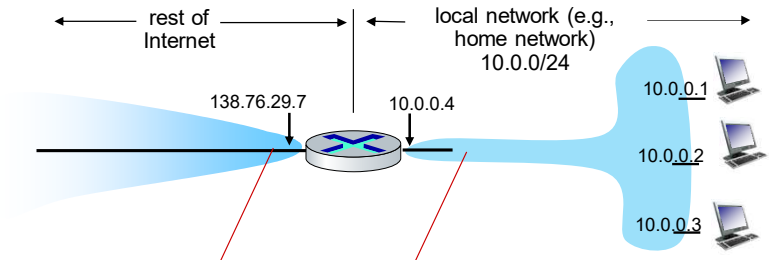


DHCPv4


DHCP server:
223.1.2.5
Arriving client
DHCP discover
src: 0.0.0.0, 68
dest: 255.255.255.255,67
DHCPDISCOVER
yiaddr: 0.0.0.0
transaction ID: 654
DHCP offer
src: 223.1.2.5, 67
dest:: 255.255.255.255,68
DHCPOFFER
yiaddrr: 223.1 2.4
transaction ID: 654
DHCP server ID: 223.1.2.5
Lifetime: 3600 secs
DHCP request
src 0.0.0.0, 68
dest: 255.255.255.255, 67
DHCPREQUEST
yiaddrr: 223.1.2.4
transaction ID: 655
DHCP server ID: 223.1.2.5
Lifetime: 3600 secs
DHCP ACK
src: 223.1.2.5, 67
dest: 255.255.255.255,68
DHCPACK
yiaddrr: 223.1.2.4
transaction ID: 655
DHCP server ID: 223.1.2.5
Lifetime: 3600 secs
Time
Time

NAT: network address translation

NAT: all devices in local network share just one IPv4 address as far as outside world is concerned



all datagrams *leaving* local network have *same* source NAT IP address: 138.76.29.7, but *different* source port numbers

datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

NAT: network address translation

All devices in local network have 32-bit addresses in a “private” IP address space (10.0.0.0/8, 172.16/12, 192.168.0.0/16 prefixes) that can only be used in local network.

Advantages:

- just one IP address needed from provider ISP for *all* devices
- can change addresses of host in local network without notifying outside world
- can change ISP without changing addresses of devices in local network
- security: devices inside local net not directly addressable, visible by outside world

NAT: network address translation

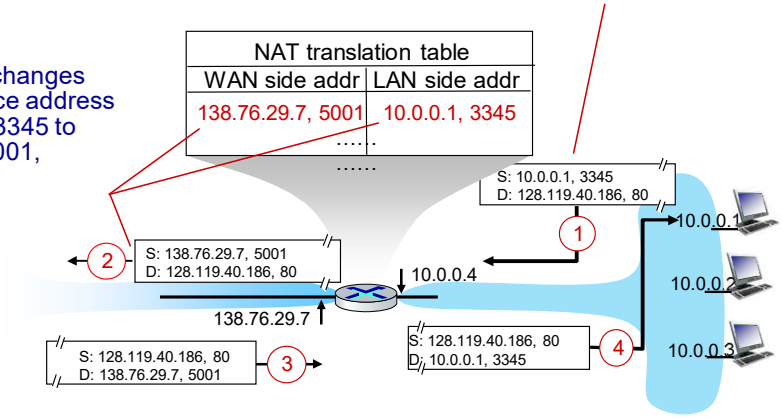
implementation: NAT router must (transparently):

- outgoing datagrams: replace (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)
  - remote clients/servers will respond using (NAT IP address, new port #) as destination address
- remember (in NAT translation table) every (source IP address, port #) to (NAT IP address, new port #) translation pair
- incoming datagrams: replace (NAT IP address, new port #) in destination fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

NAT: network address translation

1: host 10.0.0.1 sends datagram to 128.119.40.186, 80

2: NAT router changes datagram source address from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates table



3: reply arrives, destination address: 138.76.29.7, 5001

4: destination port and address are replaced: 10.0.0.1, 3345

NAT: network address translation

NAT has been controversial:

- routers “should” only process up to layer 3
- address “shortage” should be solved by IPv6
- violates end-to-end argument (port # manipulation by network-layer device)

NAT traversal: what if client wants to connect to server behind NAT?

but NAT is here to stay:

- extensively used in home and institutional nets, 4G/5G cellular nets

IPv6

"If Y2K was technology's ticking time bomb, then IPv6 is a slow, deadly gas leak."

IPv6

IPv4 limitations:

- Running out of address space
- Routing Scalability
- New services

Famous Last Words

"I think there is a world market for maybe five computers"

- Thomas Watson (*president of IBM*), 1943

"640K ought to be enough for anybody"

- Bill Gates (MicroSoft), 1981

"32 bits should be enough address space for the Internet"

- Vint Cerf (ARPA), 1977

(perhaps urban legends….)

IPv4 Address Allocation

IPv4 adresses Available in October 17, 2010 according to the RIPE

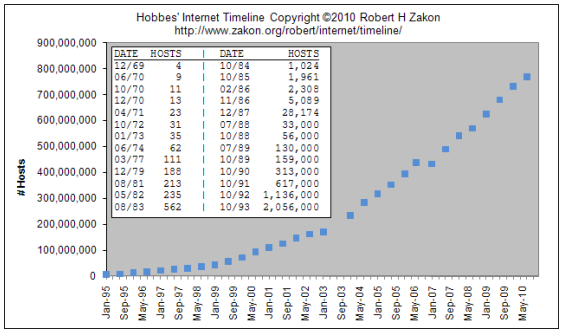
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

xx	Used
xx	Available
xx	Not useable

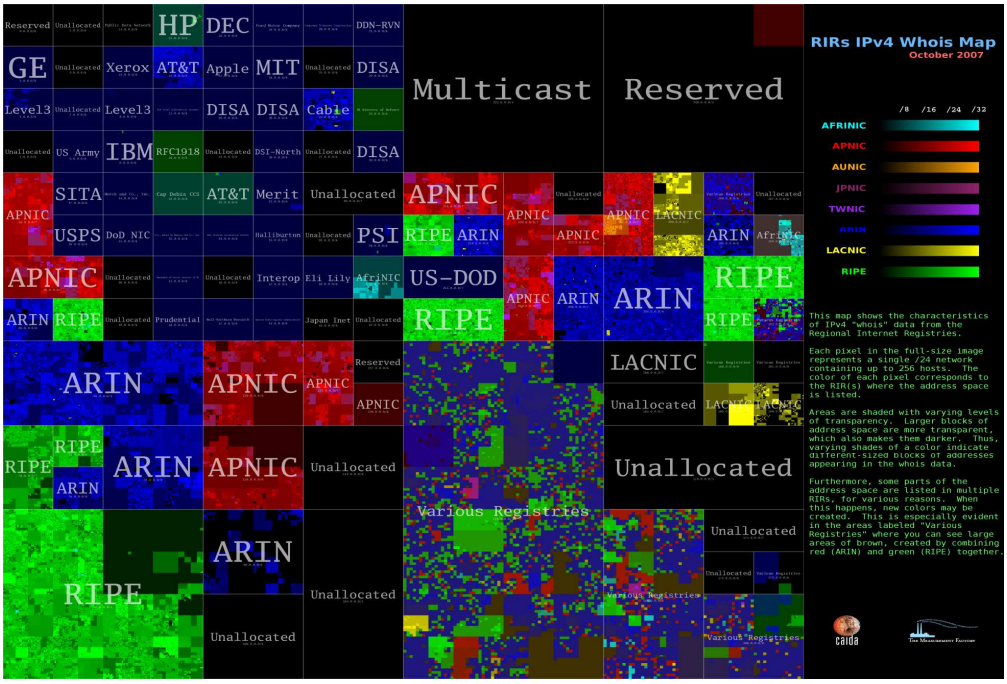
xx	Allocated in Jan. 2010	xx	Allocated in May 2010
xx	Allocated in Feb. 2010	xx	Allocated in June 2010
xx	Allocated in Apr. 2010	xx	Allocated in Aug. 2010

xx Allocated in Oct. 2010

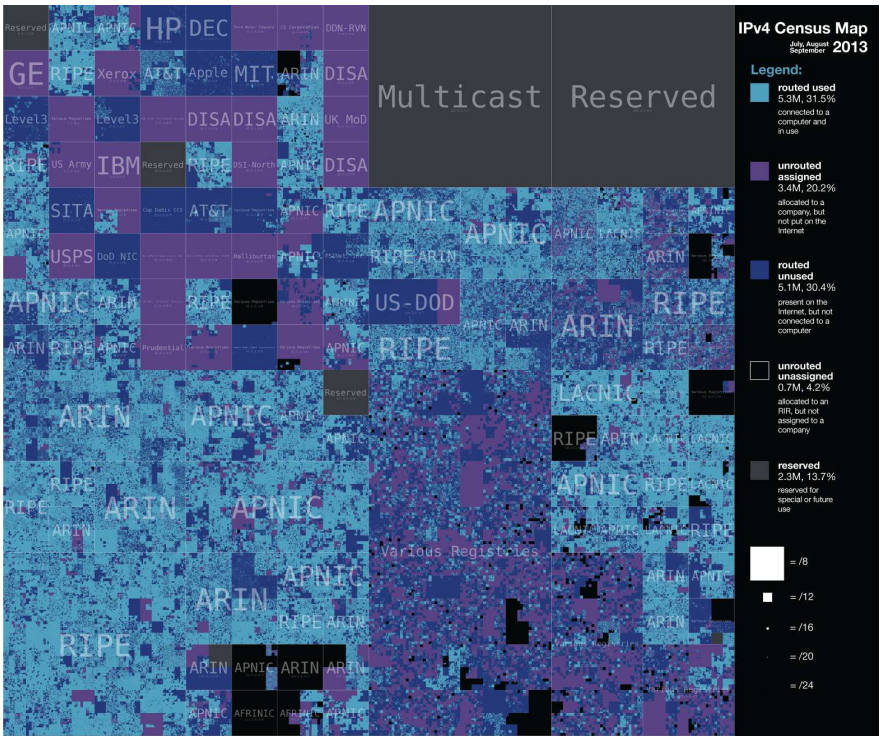
Total	256
Free	12
Percent	4,69%



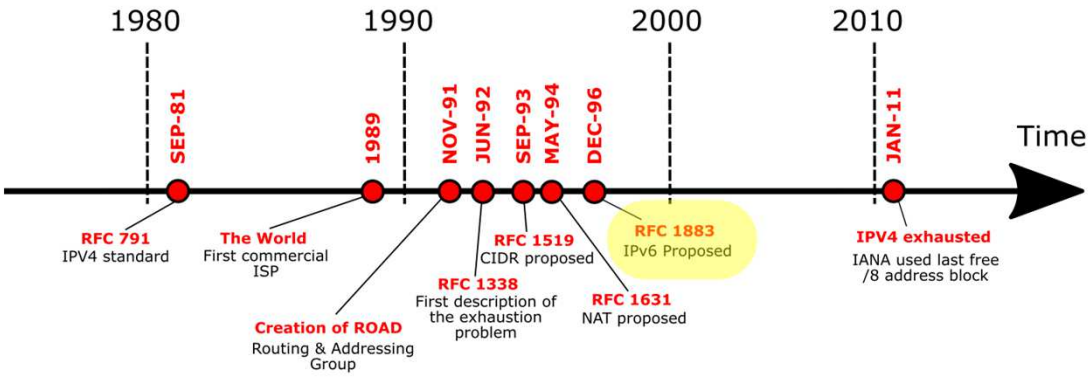
IPv4 Address Allocation



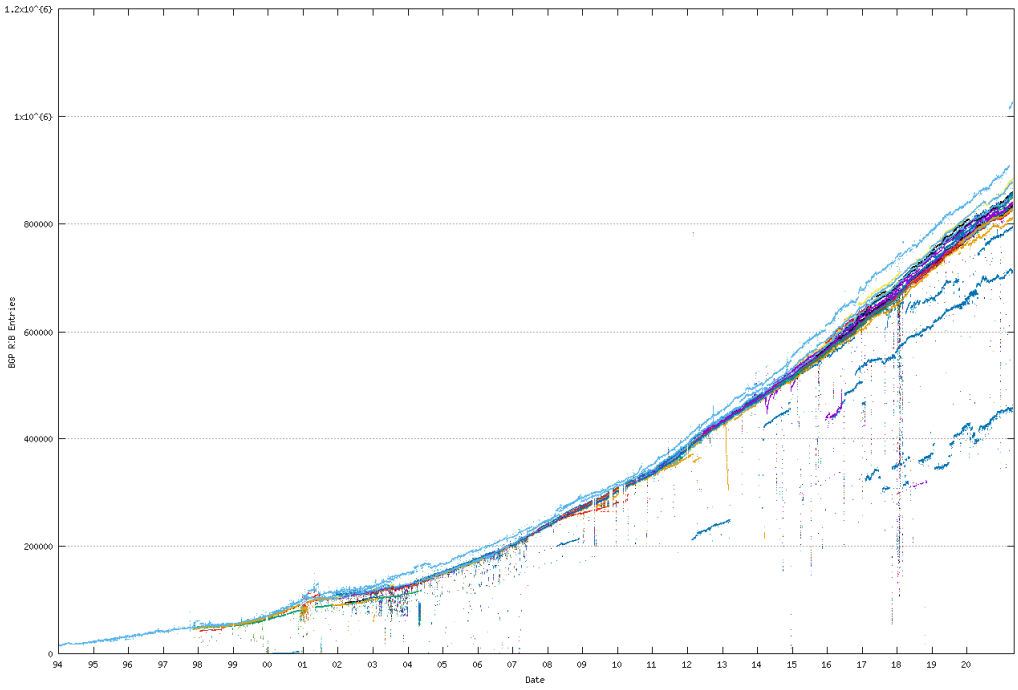
IPv4 Address Allocation



IPv4 Address Allocation



IPv4 routing table entries

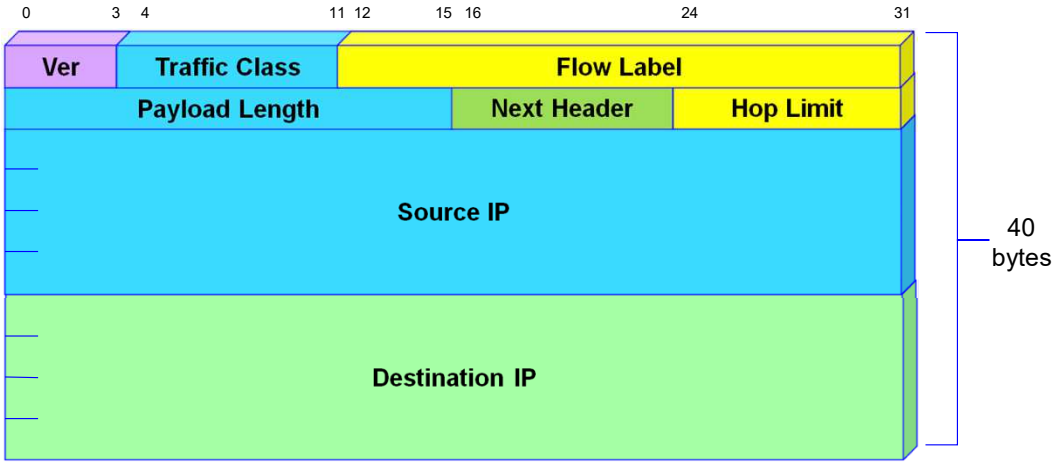


IPv6

New Services Required:

- Security
- Auto-Configuration (Plug & Play)
- Service Quality Management(QoS)
- Multicast Addressing
- Mobile Host Addressing

IPv6

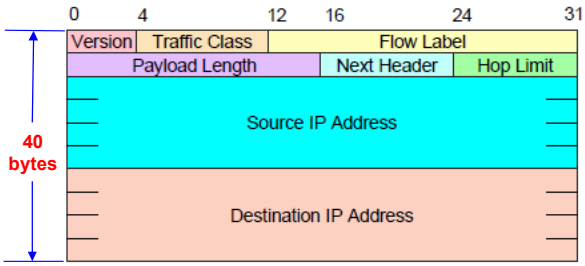


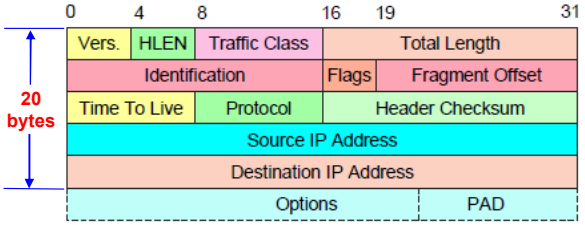
2^{128} = 340282366920938463463374607431768211456 ≈ $3x10^{38}$
2^{32} = 4294967296 ≈ $4x10^{9}$

Terrestrial Surface≈ 5.1 x 10^{14} m^2 ⇒ 6.6 x 10^{23} IPv6 addresses per m^2
Stars in the Universe (observable) ≈ 10^{24} ⇒ 10^{14} addresses per star

Header Comparison

IPv6





IPv4

Removed
- ID, Flags, Offset
- ToS, HLEN
- Header Checksum

Changed
- Total length ⇨ payload
- Protocol ⇨ next header
- TTL ⇨ hop limit

Added
- Traffic class
- Flow label

Expanded
- address: 32 ⇨ 128 bits

IPv6 addresses

0000 0000	Reserved (Include IPv4)
0000 001	OSI addresses (deprecated)
0000 010	Novell IPX addresses
001	Global unicast addresses
010	Service providers addresses
100	Geographical addresses
1111 1110 10	Link-local unicast addresses
1111 1110 11	Site-link unicast addresses (deprecated)
1111 1111	Multicast

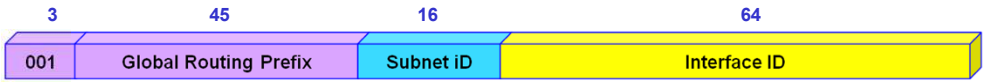
IPv6 addresses

```
8000:0000:0000:0000:0123:4567:89AB:CDEF

8000::123:4567:89AB:CDEF

2001:760:4804:0:0:0:0:105
2001:760:4804::/48
```

`::151.97.1.1`	IPv4 address mapped in IPv6
`::1`	loopback
`::`	Not specified

IPv6 addresses

Global Unicast addresses



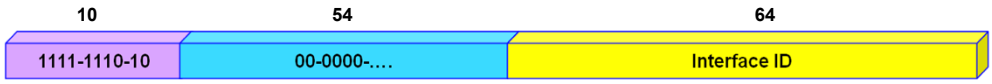
2 001:760:4804: 0: 0:0:0:105

Global Routing Prefix: Identify the site.

Subnet ID: is managed within the site

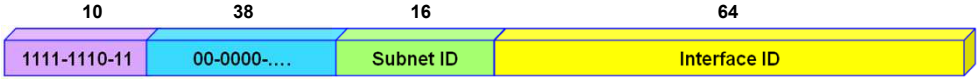
IPv6 addresses

Local Addresses: **FE80::/64** **(1111-1110-1000-0000)**



They can only be used among nodes of the same lan

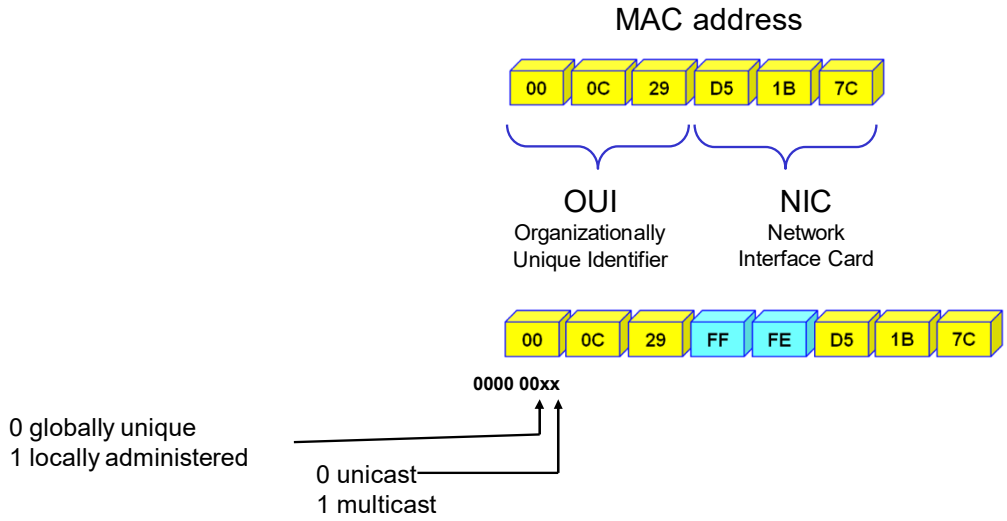
Site Addresses: **FEC0::/48** **(1111-1110-1100-0000)**



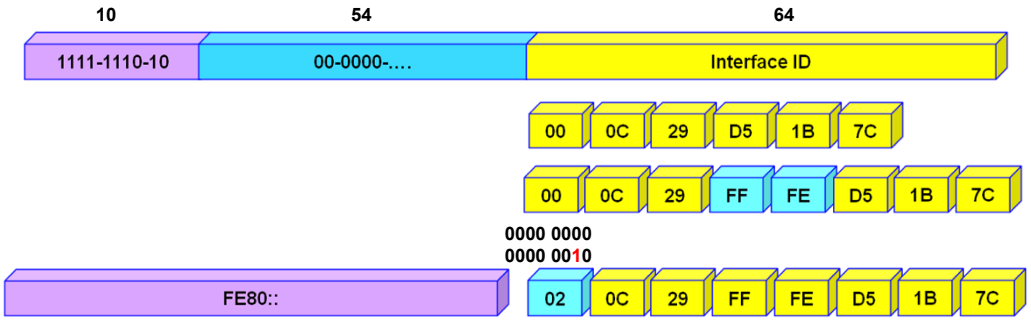
They can only be used among nodes of the same organization. They can not be routed in Internet. They are deprecated by RFC 3879.

Note that both cannot be used on the Internet.

IPv6 addresses EUI-64 (RFC 2373)



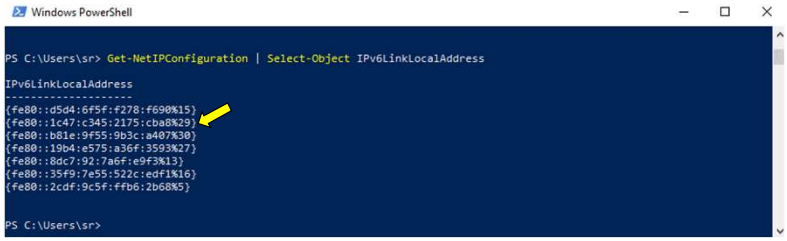
IPv6 addresses EUI-64 (RFC 2373)

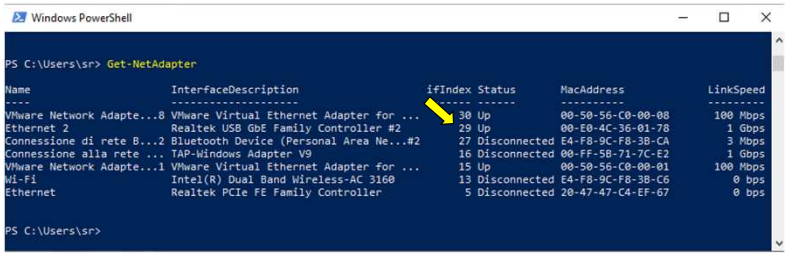



```
[root@LabReti ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:D5:1B:7C
          inet addr:192.168.1.138  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fed5:1b7c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:11 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1421 (1.3 KiB)  TX bytes:1382 (1.3 KiB)
          Interrupt:19 Base address:0x2000
```

IPv6 local link address not EUI-64

Windows OS does not use EUI-64 The IPv6 address is calculated from a random number. The FFFE group is missing in the central part.





IPv6 addresses

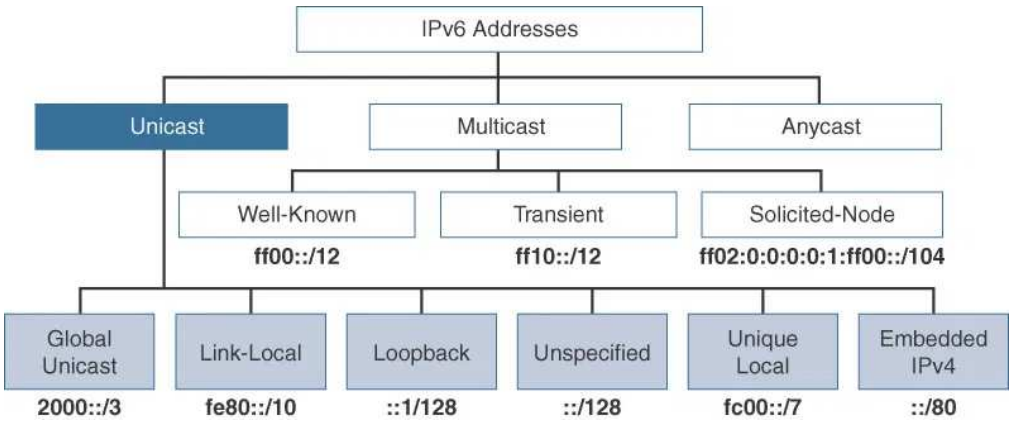
Unicast Addresses (global) **2xxx:** ..... **3xxx :** .....

Multicast Addresses **FFxx:** .....

Multicast Address	Scope	Group Within the Scope
FF01:0:0:0:0:0:0:1	Node-local	All-nodes address
FF01:0:0:0:0:0:0:2	Node-local	All-routers address
FF02:0:0:0:0:0:0:1	Link-local	All-nodes address
FF02:0:0:0:0:0:0:2	Link-local	All-routers address
FF02:0:0:0:0:0:0:5	Link-local	OSPF IGP
FF02:0:0:0:0:0:0:6	Link-local	OSPF IGP designated routers
FF02:0:0:0:0:0:0:D	Link-local	All PIM routers
FF02:0:0:0:0:0:0:16	Link-local	All MLDv2-capable routers
FF02:0:0:0:0:1:2	Link-local	All DHCP agents
FF02:0:0:0:0:1:FFXX:XX XX	Link-local	Solicited-node address
FF05:0:0:0:0:0:0:2	Site-local	All-routers address
FF05:0:0:0:0:0:1:3	Site-local	All DHCP servers

Anycast Addresses

IPv6 addresses



IPv6 over Ethernet

IPv6 Ethernet encapsulation (RFC 2464)

IPv6 packets are encapsulated in Ethernet frames just like IPv4 packets: only the Ethertype field changes (86DD instead of 0800).

Neighbour discovery (NDP) (RFC 4861)

NDP provides these services:

- Router discovery
- Prefix discovery
- Parameter discovery
- Address Autoconfiguration
- Address resolution
- Next-hop determination
- Neighbor unreachability detection
- Duplicate address detection
- Redirect messages

Neighbour Discovery Protocol

- Address Resolution Protocol (ARP),
- Internet Control Message Protocol (ICMP) router discovery,
- ICMP Redirect message

ARP is not part of ICMPv4. Having incorporated the mac address resolution mechanism into ICMPv6, makes it more independent of the type of physical medium and allows the use of security mechanisms

Neighbour Discovery Protocol

NDP provides 5 messages:

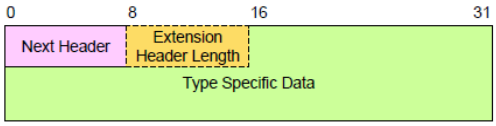
- Router Solicitation (ICMPv6 type 133)
- Router Advertisement (ICMPv6 type 134)
- Neighbor Solicitation (ICMPv6 type 135)
- Neighbor Advertisement (ICMPv6 type 136)
- Redirect (ICMPv6 type 137)

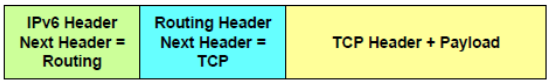
IPv6 Traffic Classes

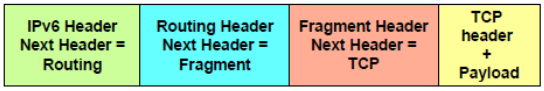
000 -111 = time insensitive (could be discarded)
1000 -1111 = priority (should not be discarded)

0 = uncharacterized
1 = filler (NetNews)
2 = unattended transfer (mail)
4 = bulk (ftp)
6 = interactive (telnet)
7 = Internet control
8 = video
15 = low quality audio

Optional headers







Optional headers

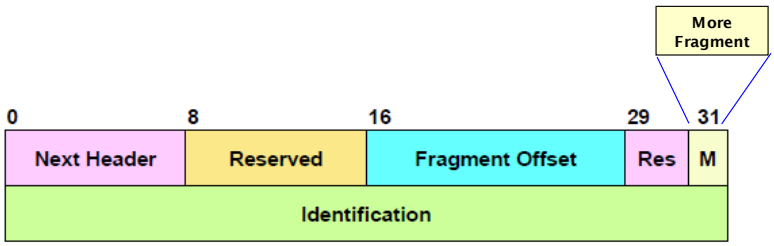
0	HBH	Hop-by-Hop option (IPv6) (Jumbogram)
1	ICMP	Internet Control Message (IPv4)
2	IGMP	Internet Group Management (IPv4)
3	GGP	Gateway-to-Gateway Protocol
4	IP	IP in IP (IPv4 encapsulation)
5	ST	Stream
6	TCP	Transmission Control
8	EGP	Exterior Gateway Protocol
9	IGP	Any private interior gateway
16	CHAOS	Chaos
17	UDP	User Datagram
29	ISO-TP4	ISO Transport Protocol Class 4
36	XTP	XTP
43	RH	Routing header (IPv6)
44	FH	Fragmentation header (IPv6)
45	IDRP	Inter-Domain Routing Protocol
46	RSVP	Reservation Protocol
50	ESP	Encapsulating Security Payload
51	AH	Authentication header (IPv6)
54	NHRP	NBMA Next Hop Resolution Protocol
58	ICMP	Internet Control Message (IPv6)
59	Null	No next header (IPv6)
60	DOH	Destination Options header (IPv6)
80	ISO-IP	ISO Internet Protocol (CLNP)
83	VINES	VINES
88	IGRP	IGRP
89	OSPF	OSPF (Open Shortest Path First)
93	AX.25	AX.25 Frames

Fragmentation

In IPv4, any node can fragment.

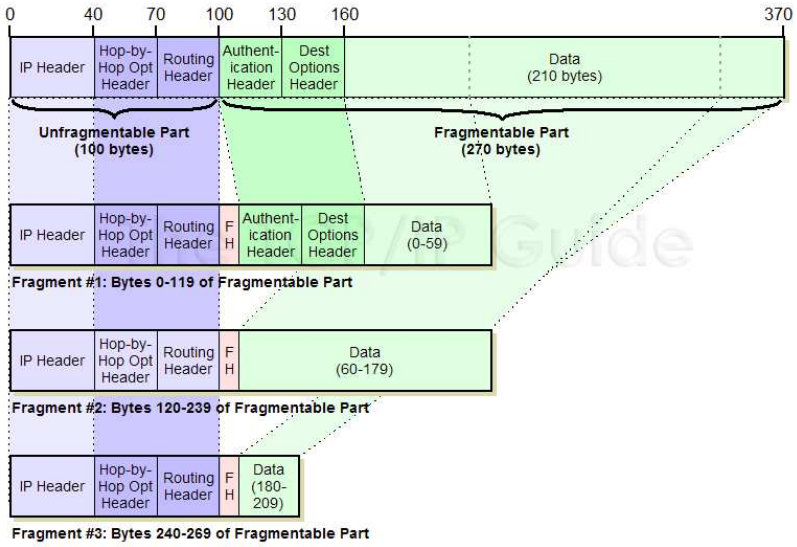
In IPv6, fragmentation is discouraged. (RFC 1981 – MTU Path discovery for IPv6). The minimum MTU is 1280 bytes, with 1500 recommended.

Only the sender can fragment, using a Fragment Header.



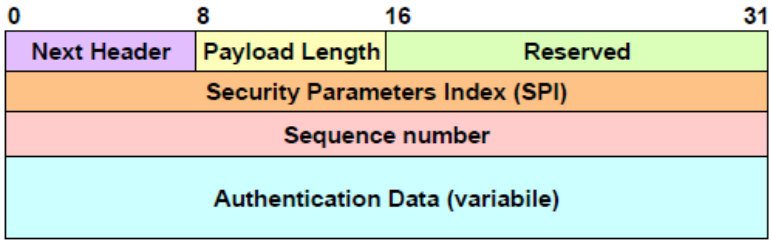
Fragmentation

The IPv6 packet can only be fragmented after the FH. Everything that comes before doesn't have to be fragmented.



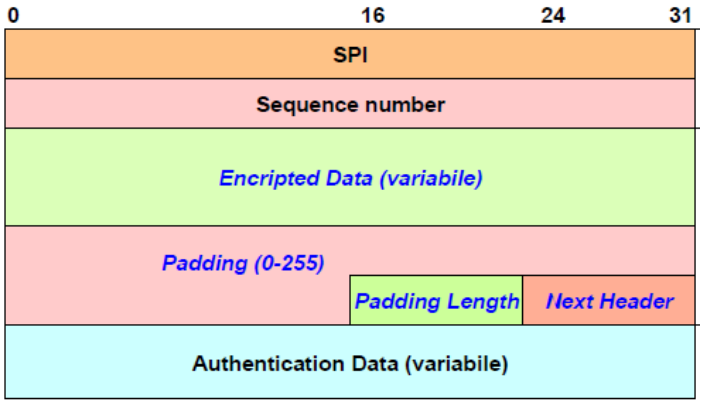
Security

Authentication Header It guarantees the authenticity and correctness of the package.

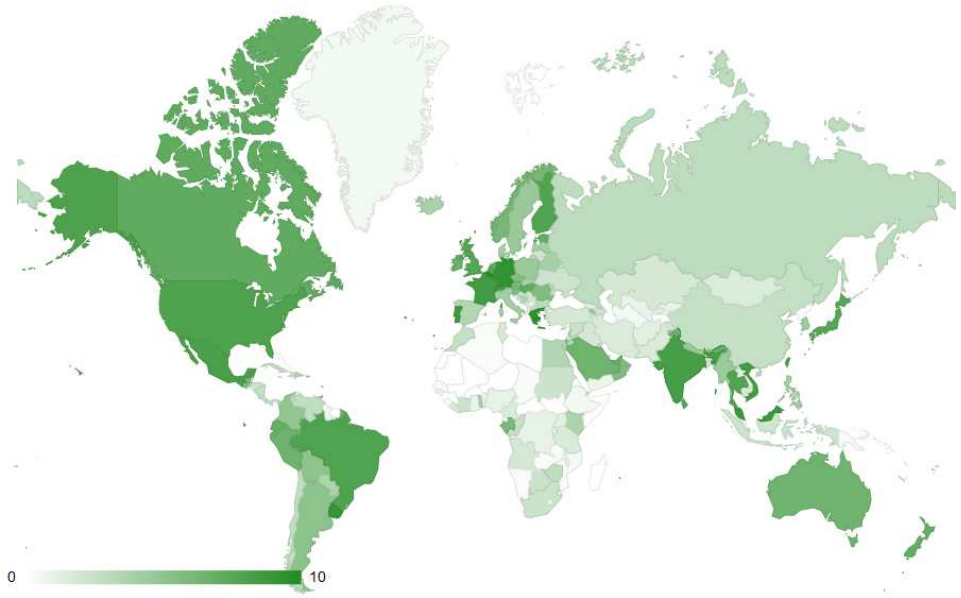


Security

Encrypted security payload Header. Provides payload encryption.

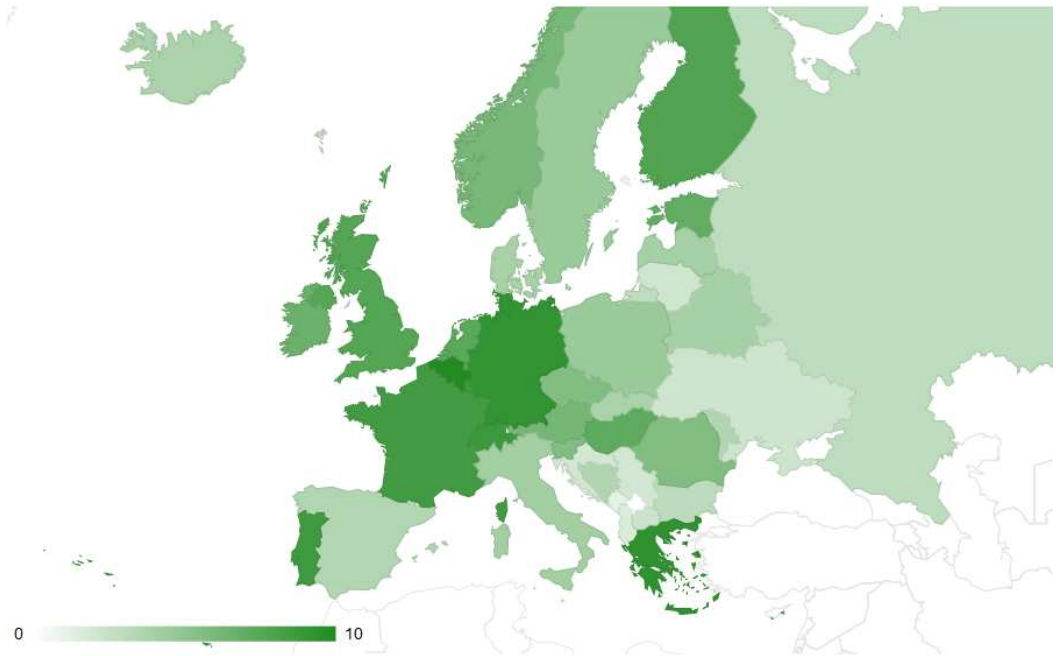


IPv6 Deployment



https://6lab.cisco.com/stats/index.php

IPv6 deployment in Europe



IPv6 diffusion in Italy

Projection of IPv6 %-age of IPv6-Enabled Web Browsers (courtesy Google) in Italy

%-age of IPv6-Enabled Web Browsers (courtesy Google)

Metric to display: allocated prefixes - announced prefixes - alive prefixes - global routing table - IPv6 web browsers (Google) - IPv6 web browsers (APnic) - IPv6 web browsers (Akamai) - IPv6 web servers

Country to plot: Italy

Predict for 365 day(s) in the future,
based on data from the 2997 last days,
using a regression: Quadratic (2nd order)
Recompute

In the last 531 days, the IPv6 metric has doubled.

Thanks to Google for their Chart tools, to Erik Kline for his suggestion and to Tom-Alexander for his polynomial regression package.

IPv6 diffusion in Italy

Projection of IPv6 %-age of IPv6-Enabled Web Browsers (courtesy Google) in Italy

%-age of IPv6-Enabled Web Browsers (courtesy Google)

Metric to display: allocated prefixes - announced prefixes - alive prefixes - global routing table - IPv6 web browsers (Google) - IPv6 web browsers (APnic) - IPv6 web browsers (Akamai) - IPv6 web servers

Country to plot: Italy

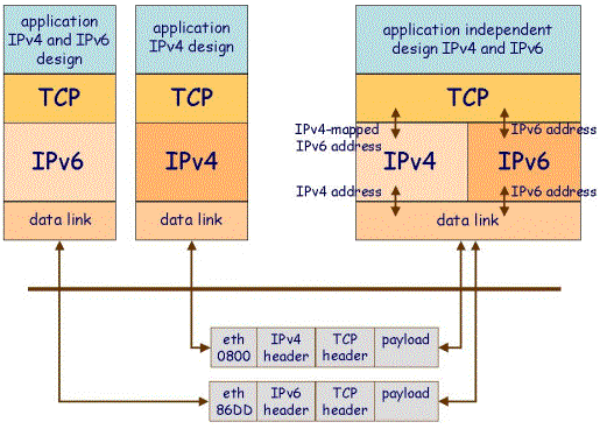
Predict for 365 day(s) in the future,
based on data from the 3338 last days,
using a regression: Quadratic (2nd order)
Recompute

In the last 891 days, the IPv6 metric has doubled.

Thanks to Google for their Chart tools, to Erik Kline for his suggestion and to Tom-Alexander for his polynomial regression package.

IPv4 IPv6 migration

The dual-stack approach

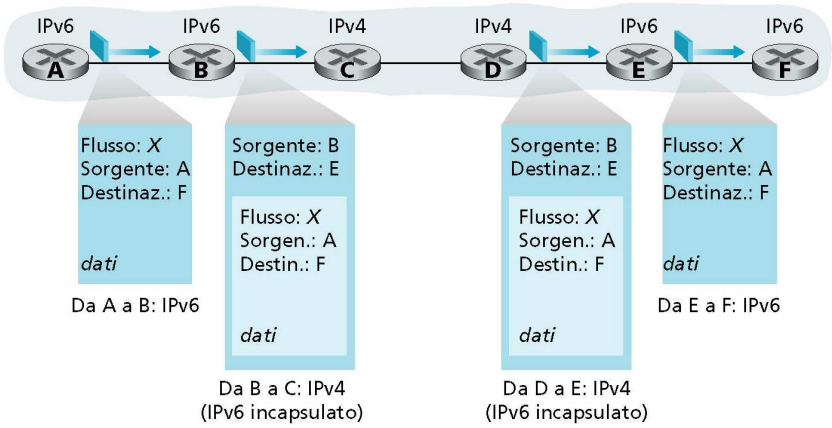


IPv4 IPv6 migration

Tunneling

Vista logica

Vista fisica



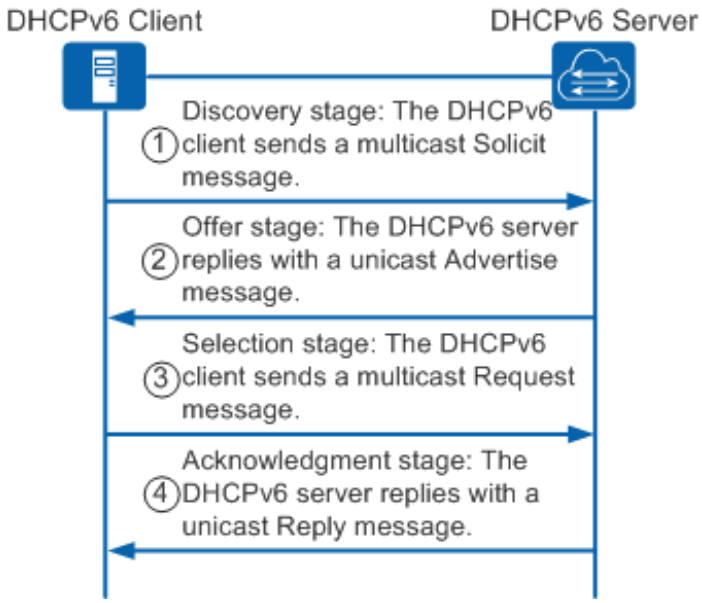
ICMPv4

Type	Code	Status	Description
0 – Echo Reply	0		Echo reply (used to ping)
3 – Destination Unreachable	0		Destination network unreachable
	1		Destination host unreachable
	2		Destination protocol unreachable
	3		Destination port unreachable
	4		Fragmentation required, and DF flag set
	5		Source route failed
5 – Redirect Message	0		Redirect Datagram for the Network
	1		Redirect Datagram for the Host
	2		Redirect Datagram for the ToS & network
	3		Redirect Datagram for the ToS & host
8 – Echo Request	0		Echo request (used to ping)
9 – Router Assignment	0		Router Advertisement
10 – Router Selection	0		Router discovery/selection/solicitation
11 – Time Exceeded	0		TTL expired in transit
	1		Fragment reassembly time exceeded
12 – Parameter Problem: Bad IP header	0		Pointer indicates the error
	1		Missing a required option
	2		Bad length
13 – Timestamp	0		Timestamp
14 – Timestamp Reply	0		Timestamp reply

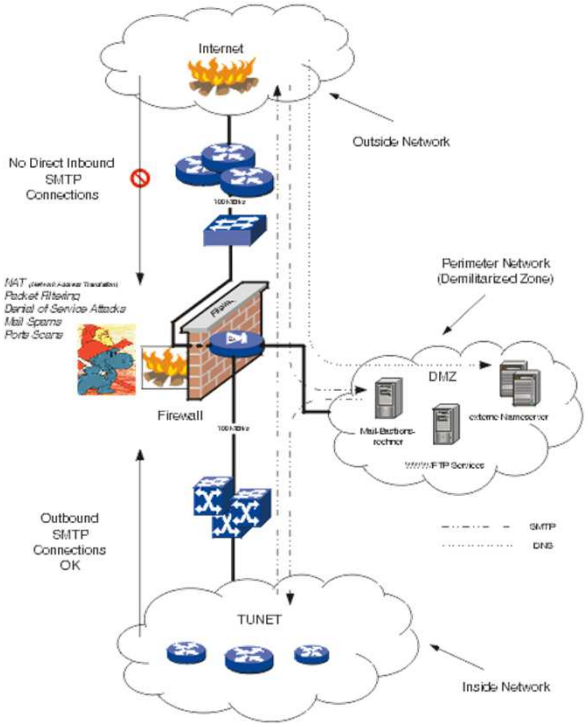
ICMPv6

Type	Meaning
1	Destination Unreachable
2	Packet Too Big
3	Time Exceeded
4	Parameter Problem
128	Echo Request
129	Echo Reply
130	Group Membership Query
131	Group Membership Report
132	Group Membership Reduction
133	Router Solicitation
134	Router Advertisement
135	Neighbor Solicitation
136	Neighbor Advertisement
137	Redirect
138	Router Renumbering

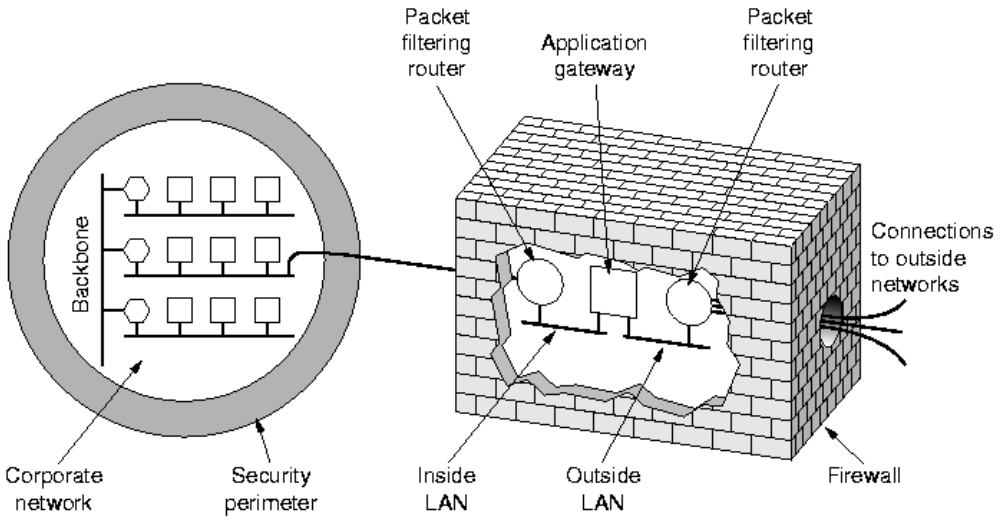
DHCPv6


DHCPv6 Client
DHCPv6 Server
Discovery stage: The DHCPv6 client sends a multicast Solicit message.
Offer stage: The DHCPv6 server replies with a unicast Advertise message.
Selection stage: The DHCPv6 client sends a multicast Request message.
Acknowledgment stage: The DHCPv6 server replies with a unicast Reply message.

Firewall


Internet
Outside Network
No Direct Inbound
SMTP
Connections
Perimeter Network
(Demilitarized Zone)
Packet Filtering
Denial of Service Attacks
Mail Spams
Port Scans
DMZ
Firewall
Outbound
SMTP
Connections
OK
TUNET
Inside Network

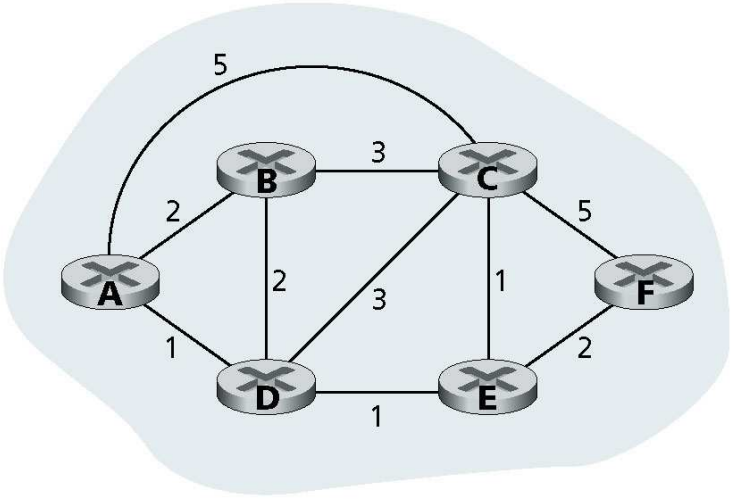
Firewall


Packet filtering router
Application gateway
Packet filtering router
Backbone
Connections to outside networks
Corporate network
Security perimeter
Inside LAN
Outside LAN
Firewall

Routing Algorithms

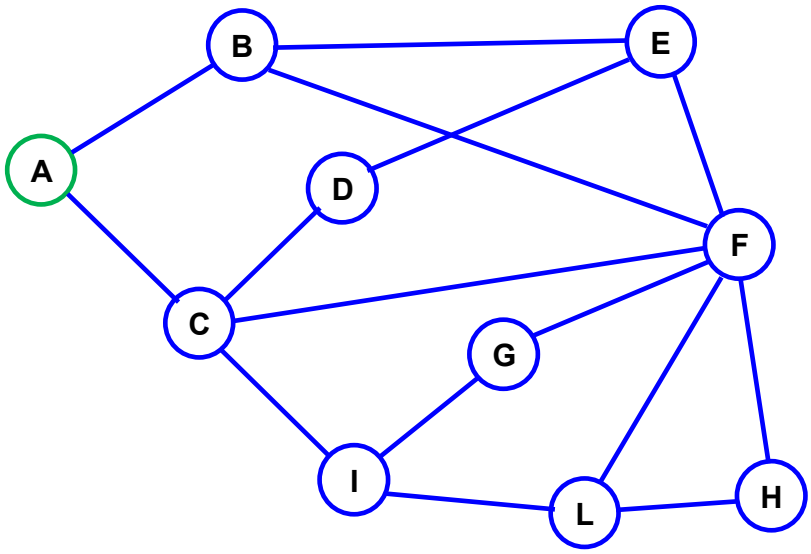
- Static
- Dynamic

- Local
- Global

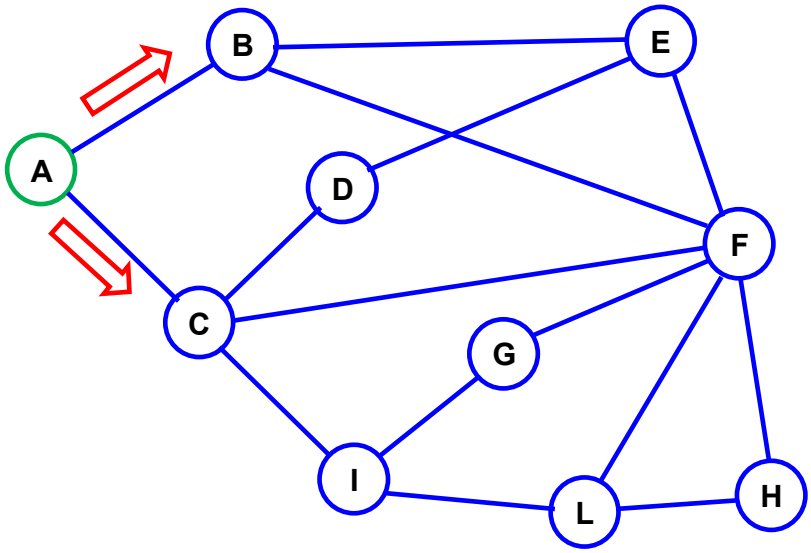


Università di Catania

Flooding

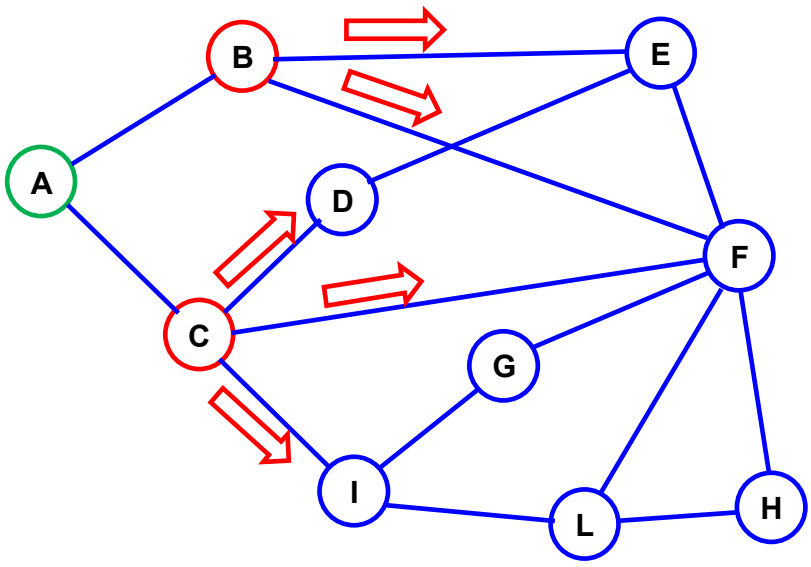


Flooding



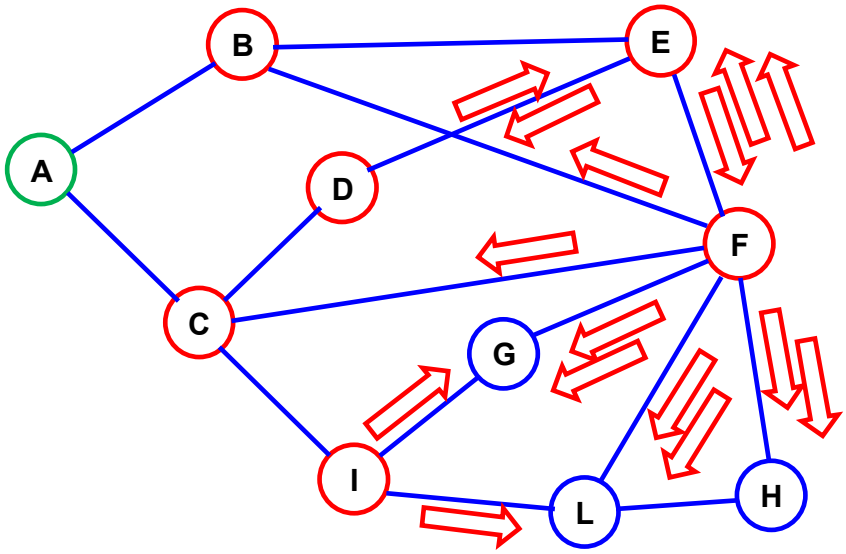
Università di Catania

Flooding



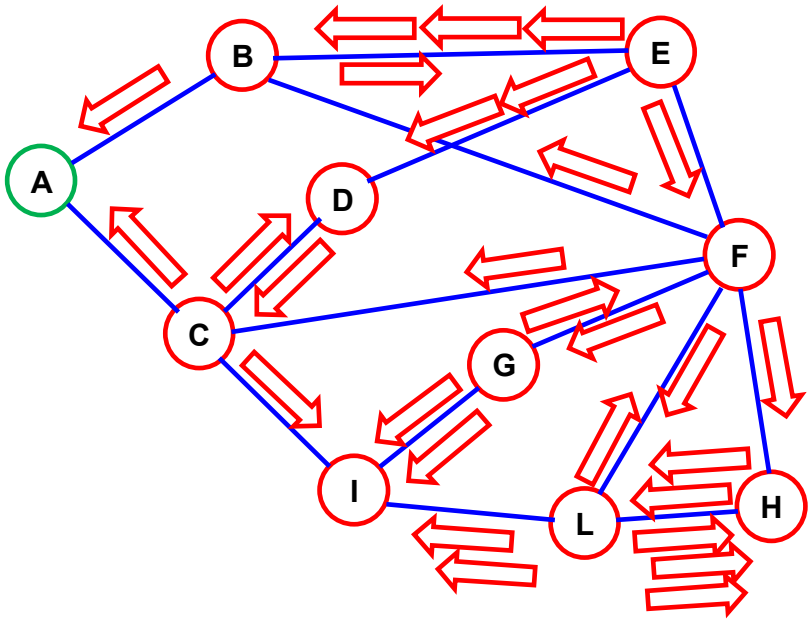
Università di Catania

Flooding

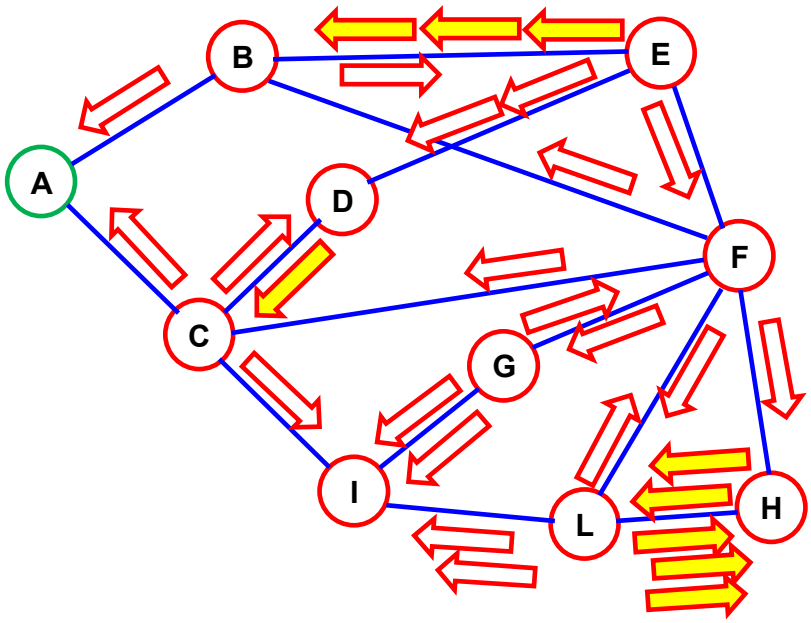


Università di Catania

Flooding



Flooding



Flooding

Pro:

- Find a route to the destination, if it exists.
- Find the best route.

Cons:

- It creates a large number of duplicate packets, saturating the network.

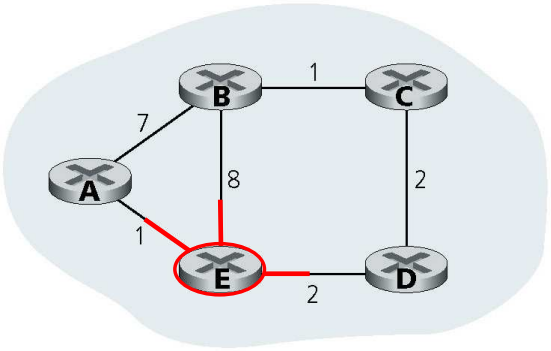
Flooding needs to be controlled in some way.

Flooding

Flood Control

1. Use an ID for packets
2. Use a HOP-limit counter
3. Delete loops from the graph
4. Forward messages only along a spanning tree

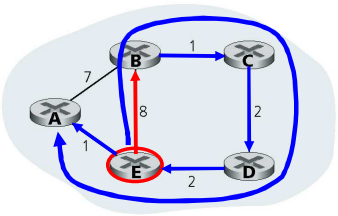
Distance Vectors



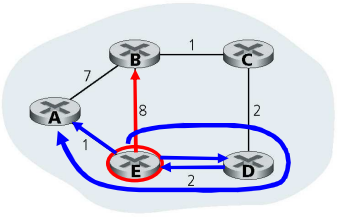
costo verso la destinazione via

destinazione	$D^E()$	A	B	D
	A	(1)	14	5
	B	7	8	(5)
	C	6	9	(4)
	D	4	11	(2)

Distance Vectors



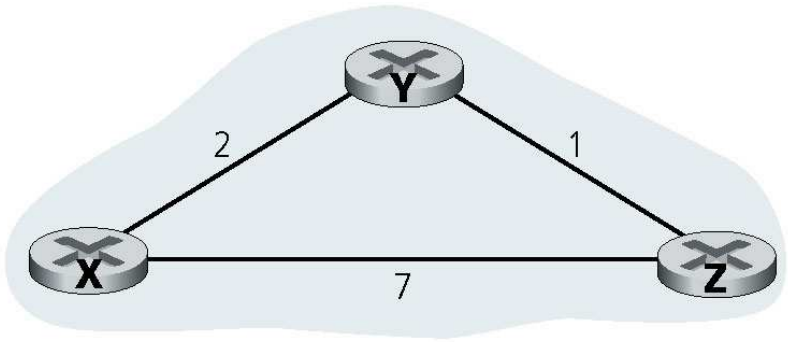
destinazione	$D^E()$ A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2



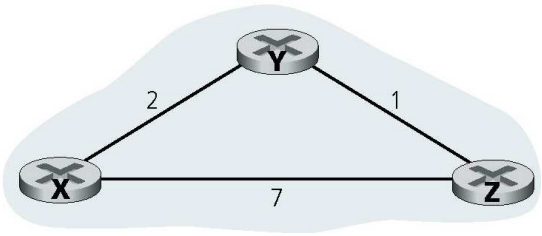
destinazione	$D^E()$ A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

Università di Catania

Distance Vectors



Distance Vectors



Destinazione / Exit

X	Y	Z
Y	∞	∞
Z	∞	∞

X	Y	Z
Y	∞	∞
Z	∞	∞

X	Y	Z
Y	2	∞
Z	∞	7

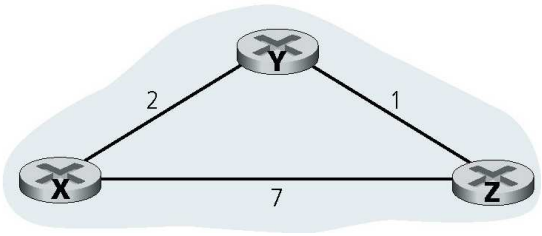
Y	X	Z
X	∞	∞
Z	∞	∞

Y	X	Z
X	2	∞
Z	∞	1

Z	X	Y
X	∞	∞
Y	∞	∞

Z	X	Y
X	7	∞
Y	∞	1

Distance Vectors



Destinazione / Uscita

	X	Y	Z
Y	∞	∞	
Z	∞	∞	

	X	Y	Z
Y	∞	∞	
Z	∞	∞	

	X	Y	Z
Y	2	∞	
Z	∞	7	

	X
Y	2
Z	7

	Y	X	Z
X	∞	∞	
Z	∞	∞	

	Y	X	Z
X	2	∞	
Z	∞	1	

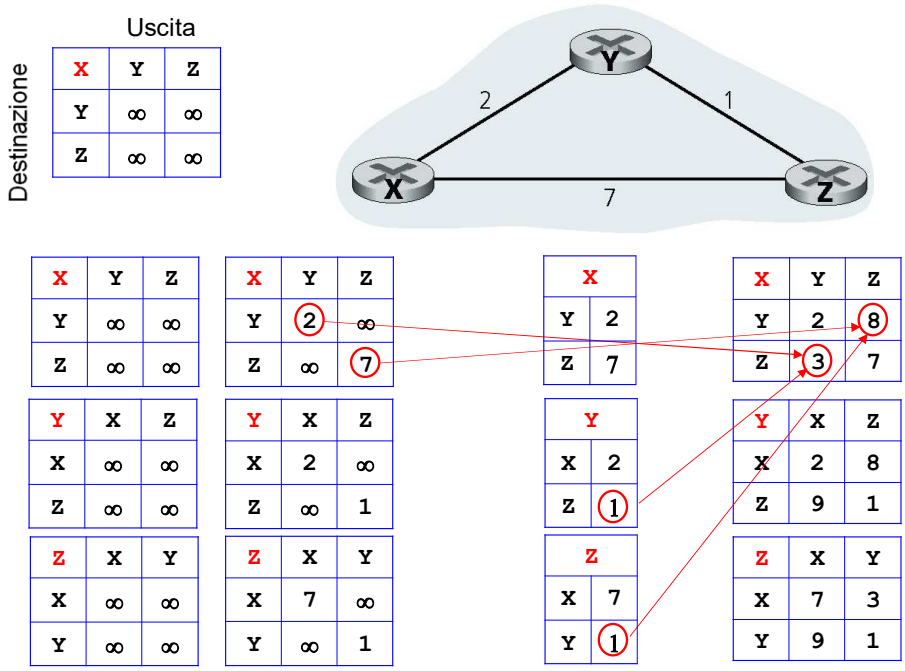
	Y
X	2
Z	1

	Z	X	Y
X	∞	∞	
Y	∞	∞	

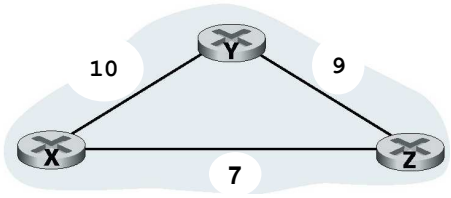
	Z	X	Y
X	7	∞	
Y	∞	1	

	Z
X	7
Y	1

Distance Vectors



Distance Vectors: Improvements



Destinazione

Uscita

X	Y	Z
Y	∞	∞
Z	∞	∞

X	Y	Z
Y	10	∞
Z	∞	7

X	Y	Z
Y	10	16
Z	19	7

Y	X	Z
X	10	∞
Z	∞	9

Y	X	Z
X	10	16
Z	17	9

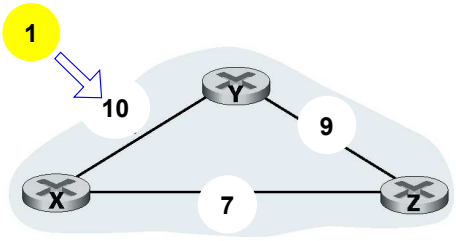
Z	X	Y
X	7	∞
Y	∞	9

Z	X	Y
X	7	19
Y	17	9

Distance Vectors: Improvements

Destinazione / Uscita

	X	Y	Z
Y	∞	∞	
Z	∞	∞	



X	Y	Z
Y	10	16
Z	19	7

X	Y	Z
Y	1	16
Z	19	7

Y	X	Z
X	10	16
Z	17	9

Y	X	Z
X	1	16
Z	17	9

Z	X	Y
X	7	19
Y	17	9

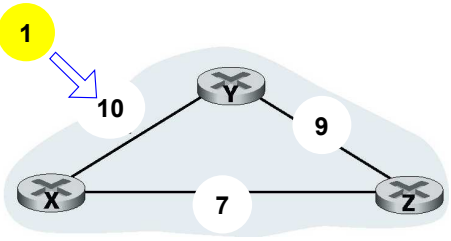
Z	X	Y
X	7	19
Y	17	9

Z	X	Y
X	7	10
Y	8	9

to Z:

exit X to Y

Distance Vectors: Improvements



Destinazione / Uscita

	X	Y	Z
Y	∞	∞	
Z	∞	∞	

	X	Y	Z
Y	10	16	
Z	19	7	

	X	Y	Z
Y	1	16	
Z	19	7	

	X
Y	1
Z	7

	X	Y	Z
Y	1	16	
Z	10	7	

	Y	X	Z
X	10	16	
Z	17	9	

	Y	X	Z
X	1	16	
Z	17	9	

	Y
X	1
Z	9

	Y	X	Z
X	1	16	
Z	8	9	

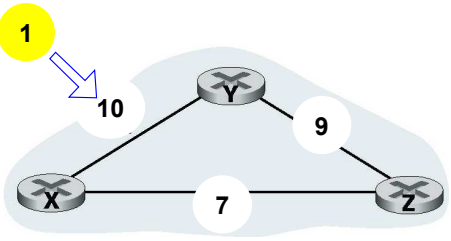
	Z	X	Y
X	7	19	
Y	17	9	

	Z	X	Y
X	7	19	
Y	17	9	

	Z
X	7
Y	9

	Z	X	Y
X	7	10	
Y	8	9	

Distance Vectors: Improvements



Destinazione / Uscita

	X	Y	Z
Y	∞	∞	
Z	∞	∞	

	X	Y	Z
Y	10	16	
Z	19	7	

	X	Y	Z
Y	1	16	
Z	19	7	

	X
Y	1
Z	7

	X	Y	Z
Y	1	16	
Z	10	7	

	Y	X	Z
X	10	16	
Z	17	9	

	Y	X	Z
X	1	16	
Z	17	9	

	Y
X	1
Z	9

	Y	X	Z
X	1	16	
Z	8	9	

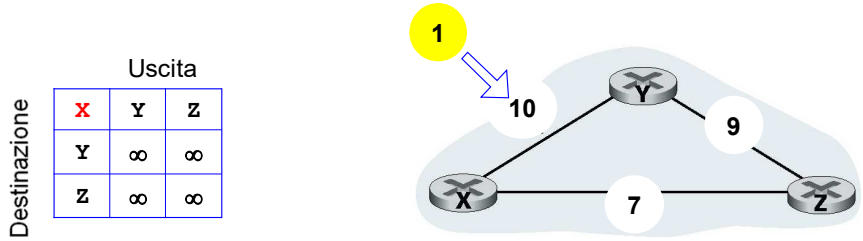
	Z	X	Y
X	7	19	
Y	17	9	

	Z	X	Y
X	7	19	
Y	17	9	

	Z
X	7
Y	9

	Z	X	Y
X	7	10	
Y	8	9	

Distance Vectors: Improvements



Destinazione / Uscita

	X	Y	Z
Y	∞	∞	
Z	∞	∞	

	X	Y	Z
Y	10	16	
Z	19	7	

	X	Y	Z
Y	1	16	
Z	19	7	

	X
Y	1
Z	7

	X	Y	Z
Y	1	16	
Z	10	7	

	X
Y	1
Z	7

	Y	X	Z
X	10	16	
Z	17	9	

	Y	X	Z
X	1	16	
Z	17	9	

	Y
X	1
Z	9

	Y	X	Z
X	1	16	
Z	8	9	

	Y
X	1
Z	8

	Z	X	Y
X	7	19	
Y	17	9	

	Z	X	Y
X	7	19	
Y	17	9	

	Z
X	7
Y	9

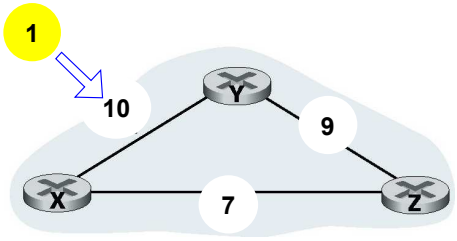
	Z	X	Y
X	7	10	
Y	8	9	

	Z
X	7
Y	8

Distance Vectors: Improvements

Uscita

Destinazione	X	Y	Z
Y	∞	∞	
Z	∞	∞	



X	Y	Z
Y	1	16
Z	10	7

X	
Y	1
Z	7

X	Y	Z
Y	1	15
Z	9	7

Y	X	Z
X	1	16
Z	8	9

Y	
X	1
Z	8

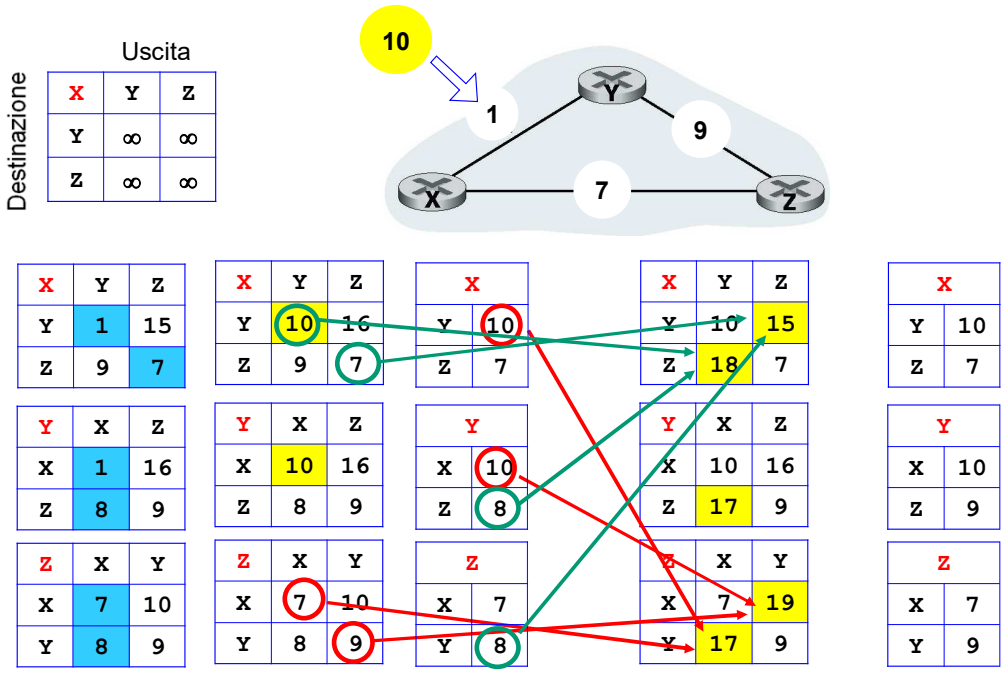
Y	X	Z
X	1	16
Z	8	9

Z	X	Y
X	7	10
Y	8	9

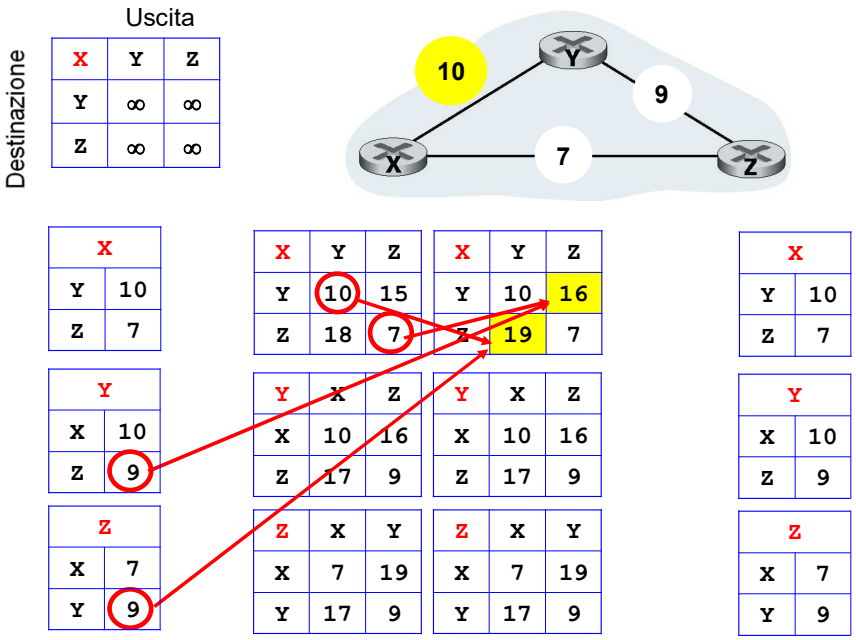
Z	
X	7
Y	8

Z	X	Y
X	7	10
Y	8	9

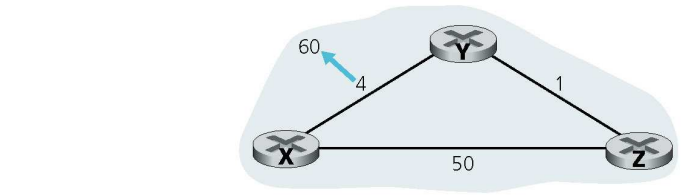
Distance Vectors: Worsening



Distance Vectors: Worsening



Distance Vectors



Best Known Route

via

D^Y	X	Z
X	4	6

D^Y	X	Z
X	60	6

Best Known Route

via

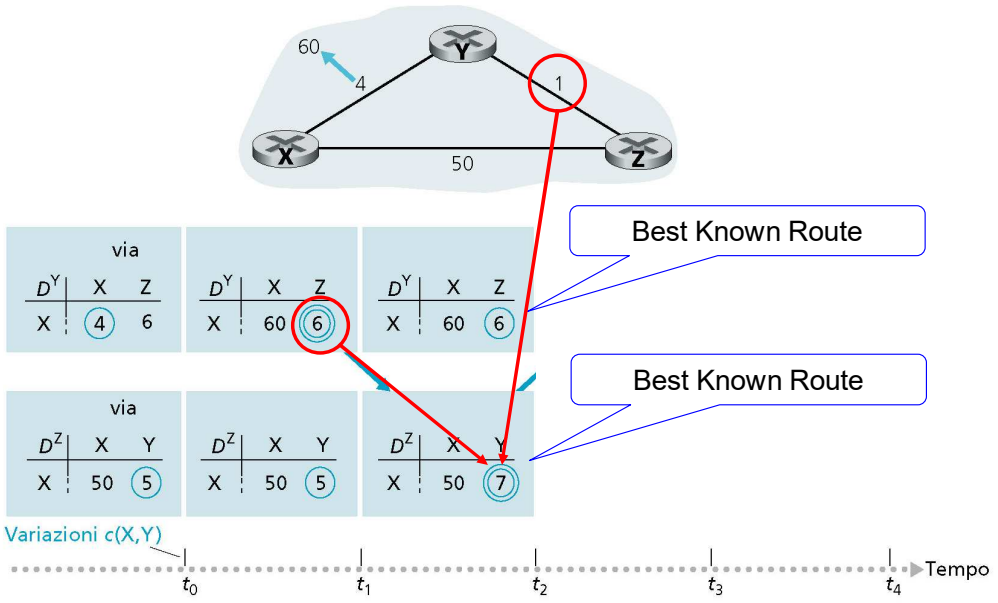
D^Z	X	Y
X	50	5

D^Z	X	Y
X	50	5

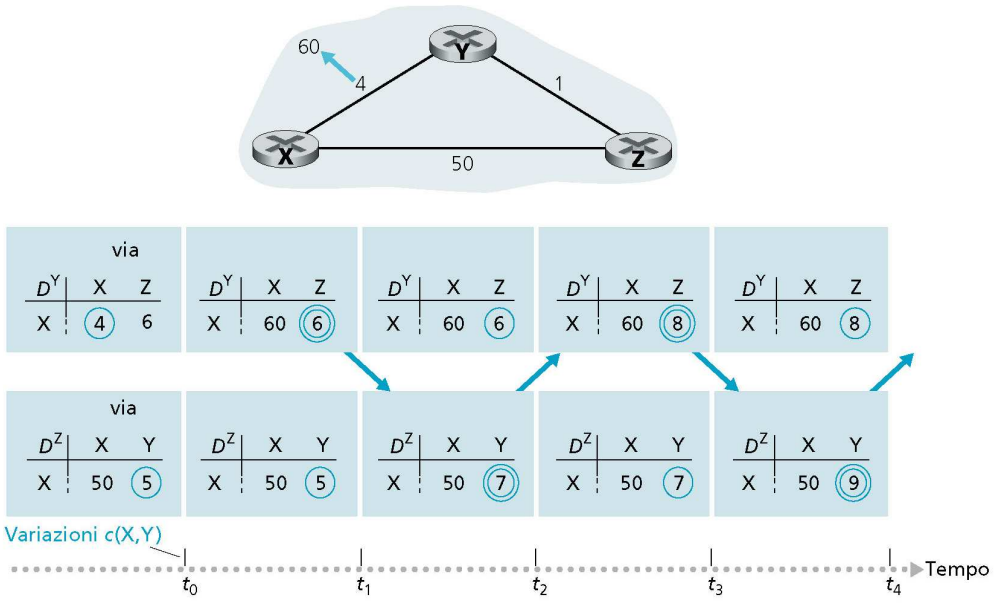
Variazioni c(X,Y)

t_0 t_1 t_2 t_3 t_4 Tempo

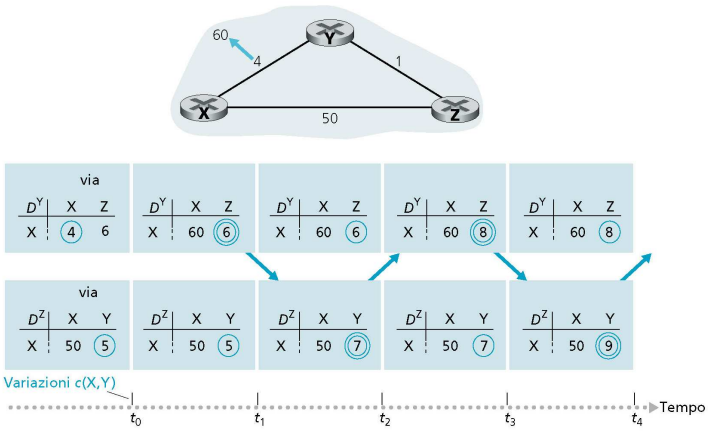
Distance Vectors



Infinity Counting



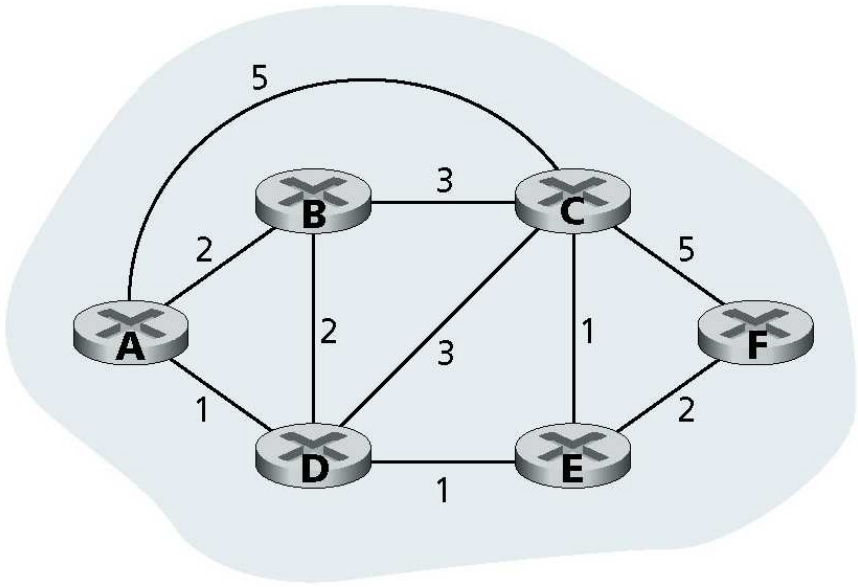
Poisoned reverse



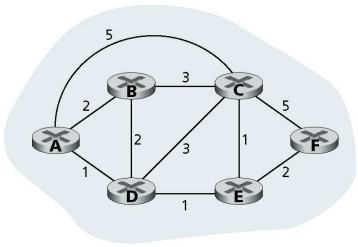
Possible solution: send ∞ to the node that has our reference as the best exit to reach another node

(e.g.: z says to y that its distance to x is ∞ if the best path to x starts from y).

Link State Routing



Link State Routing



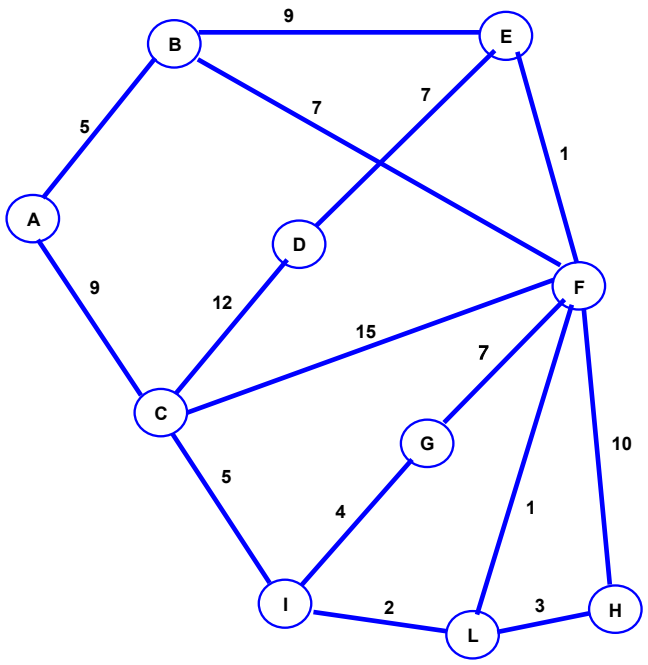
The first version of Arpanet used Distance Vector Routing.

The algorithm was replaced after 1979 with Link State Routing as the former did not take into account channel capacities but only transmission delay times.

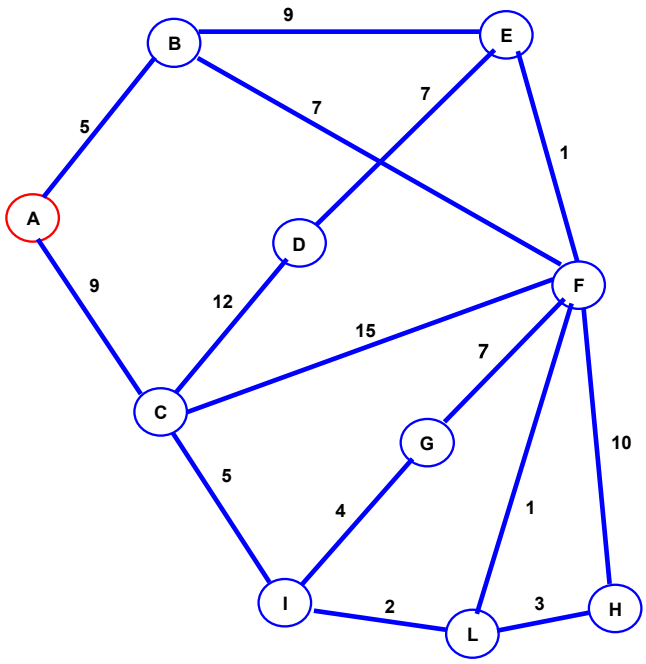
With this algorithm, each router investigates its neighbors and determines communication costs. It then sends the information in broadcast, so that each router can calculate the optimal routes.

The calculation is based on the algorithm of Dijkstra.

Dijkstra



Dijkstra

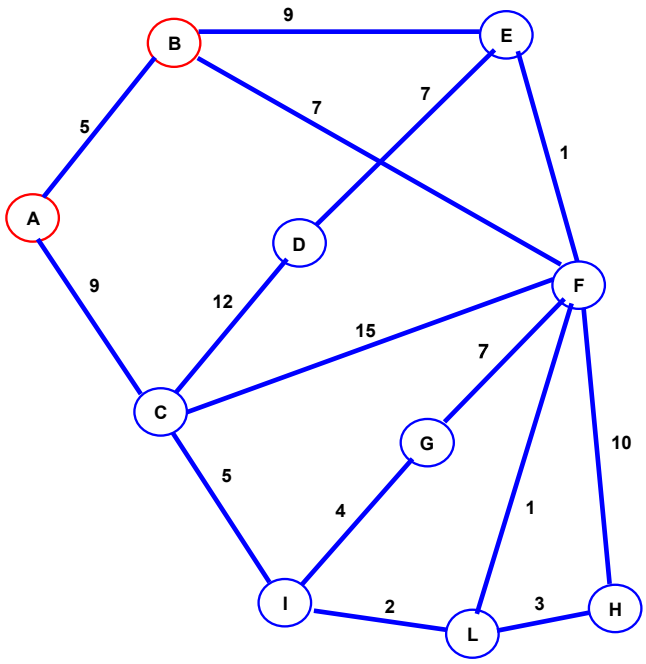


A

B = 5

C = 9

Dijkstra



A

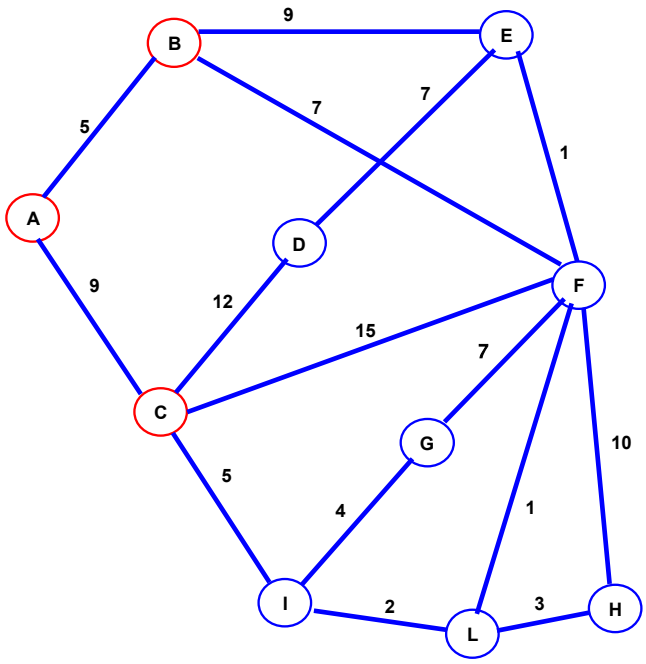
B = 5

C = 9

E = 5+9 = 14

F = 5+7= 12

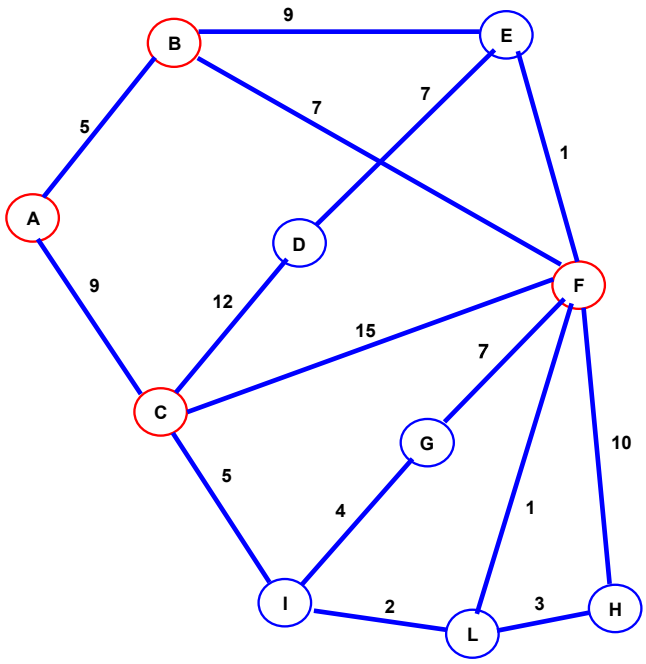
Dijkstra



A
B = 5
C = 9

E = 5+9 = 14
F = 5+7= 12
D = 9+12= 21
F = 9+15= 24
I = 9+5 = 14

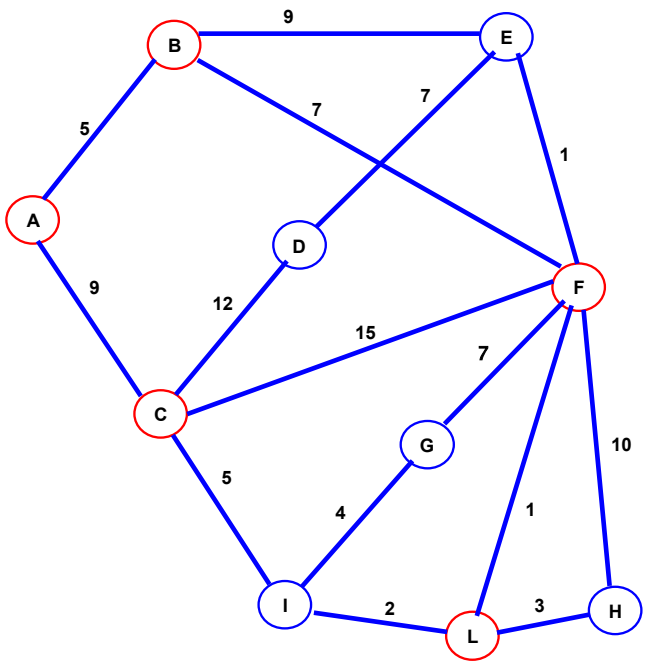
Dijkstra



A
B = 5
C = 9
F = 12

E = 5+9 = 14
D = 9+12= 21
I = 9+5 = 14
E = 12+1 = 13
G = 12+7 = 19
H = 12+10 = 22
L = 12+1= 13

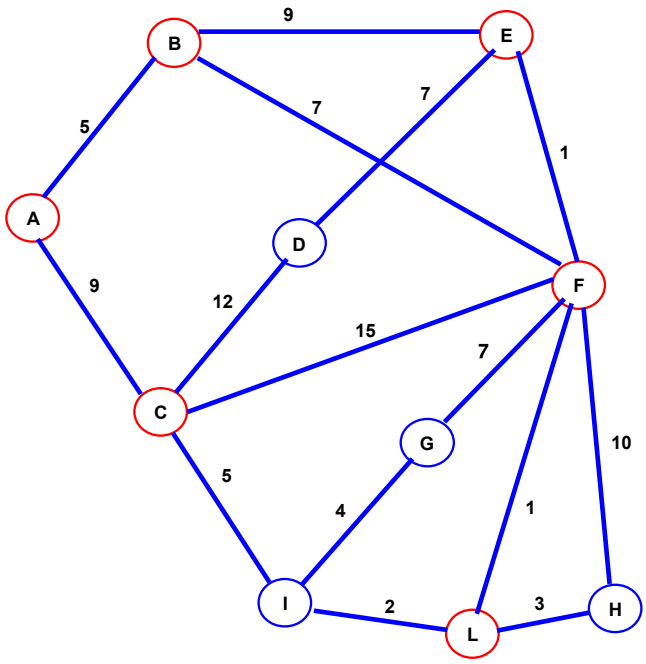
Dijkstra



A
B = 5
C = 9
F = 12
L = 13

E = 5+9 = 14
D = 9+12= 21
I = 9+5 = 14
E = 12+1 = 13
G = 12+7 = 19
H = 12+10 = 22
I = 13+2 = 15
H = 13+3 = 16

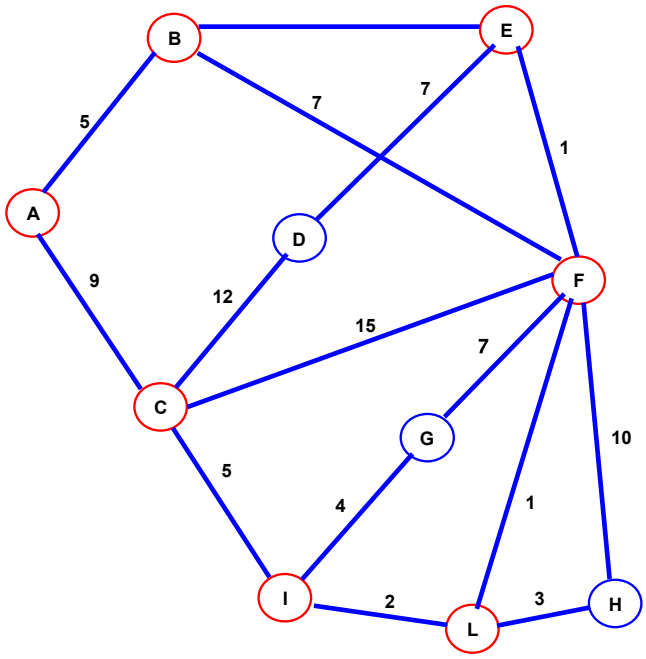
Dijkstra



A
B = 5
C = 9
F = 12
L = 13
E = 13

D = 9+12= 21
I = 9+5 = 14
G = 12+7 = 19
H = 12+10 = 22
I = 13+2 = 15
H = 13+3 = 16
D = 13+7 = 20

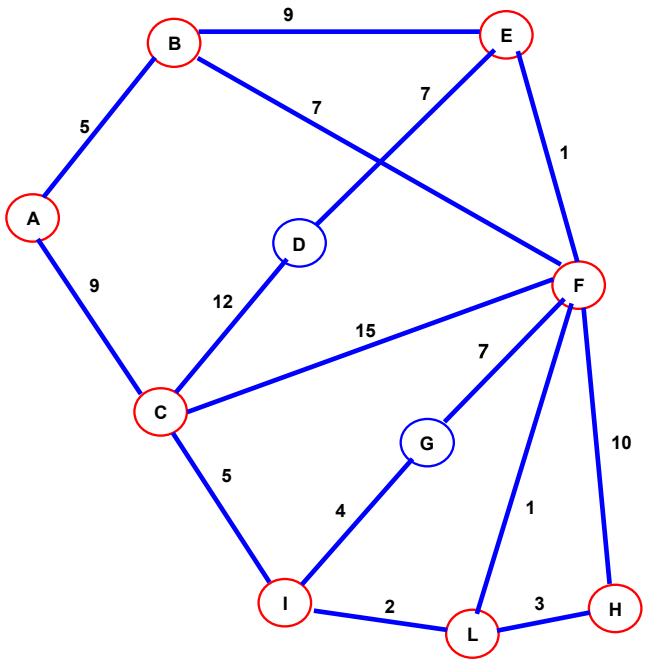
Dijkstra



A
B = 5
C = 9
F = 12
L = 13
E = 13
I = 14

D = 9+12= 21
G = 12+7 = 19
H = 12+10 = 22
H = 13+3 = 16
D = 13+7 = 20
G = 14+4 = 18

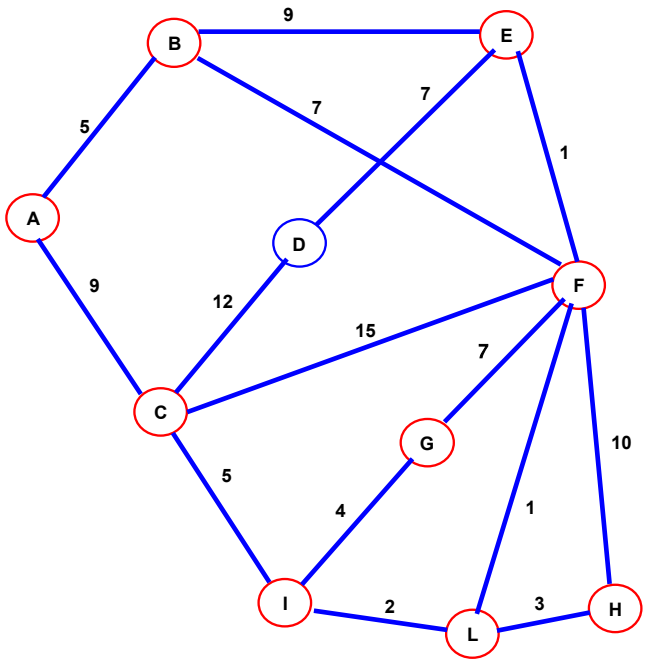
Dijkstra



A
B = 5
C = 9
F = 12
L = 13
E = 13
I = 14
H = 16

D = 9+12= 21
G = 12+7 = 19
D = 13+7 = 20
G = 14+4 = 18

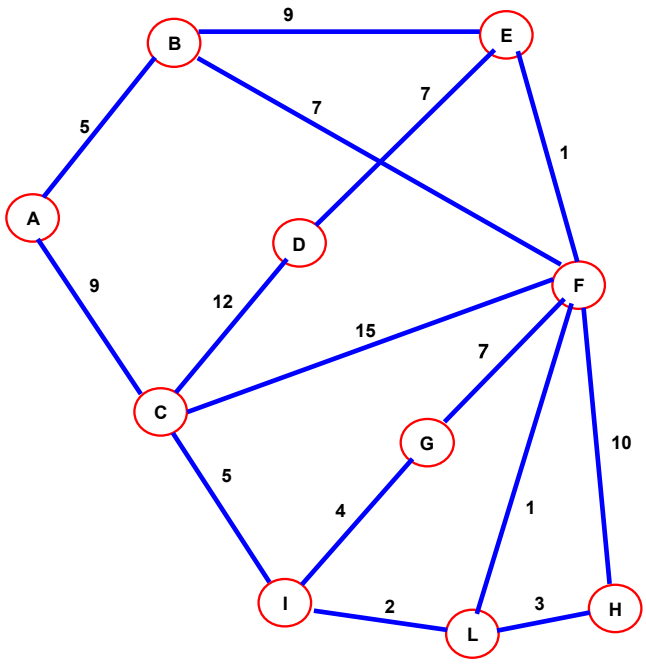
Dijkstra



A
B = 5
C = 9
F = 12
L = 13
E = 13
I = 14
H = 16
G = 18

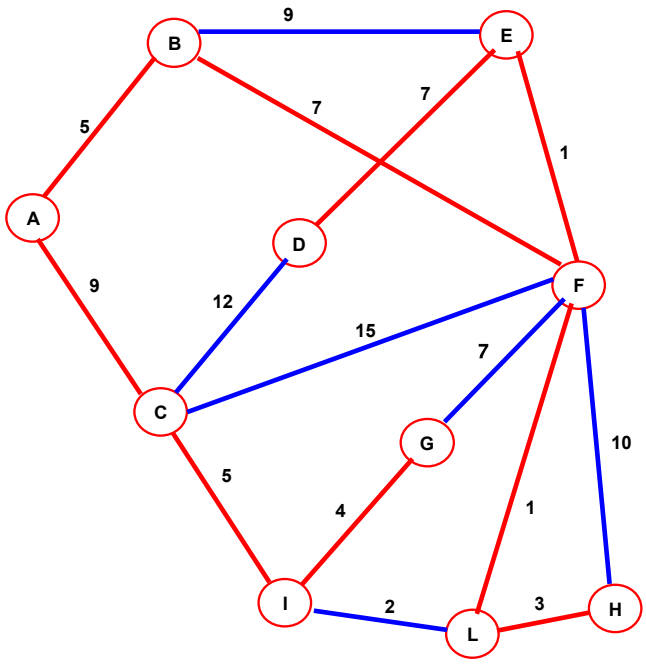
D = 9+12= 21
D = 13+7 = 20

Dijkstra



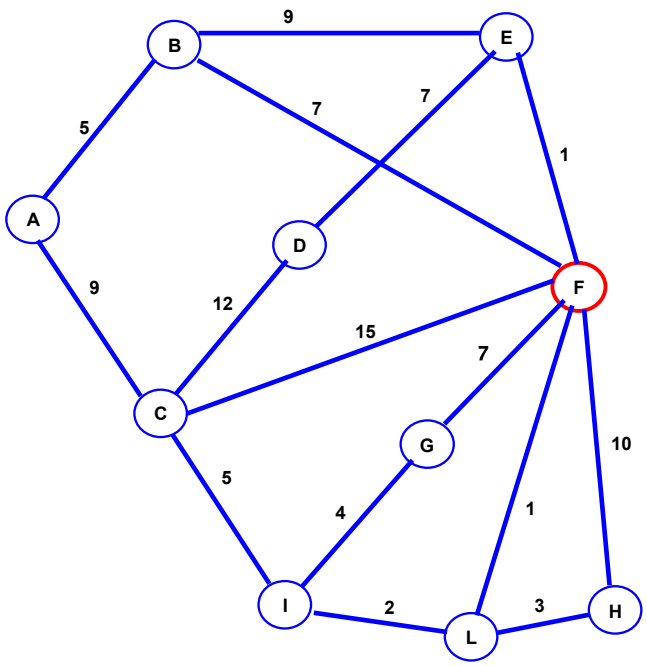
A
B = 5
C = 9
F = 12
L = 13
E = 13
I = 14
H = 16
G = 18
D = 20

Dijkstra



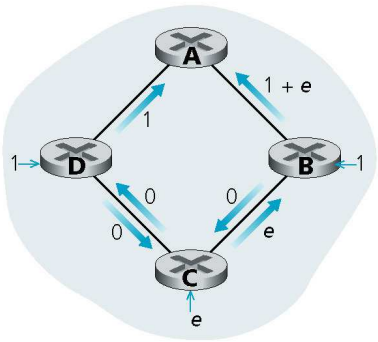
A
B = 5 A B
C = 9 A C
F = 12 A B F
L = 13 A B F L
E = 13 A B F E
I = 14 A C I
H = 16 A B F L H
G = 18 A C I G
D = 20 A B F E D

Dijkstra



Link State Routing

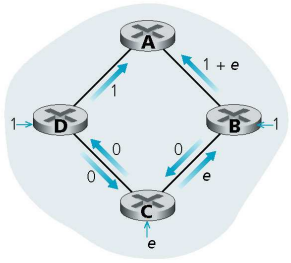
In the LSR, each router periodically informs its neighbors about its link situation.



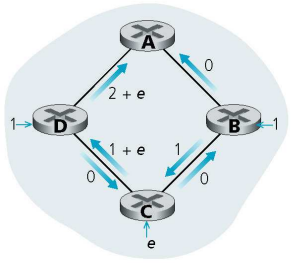
a. Routing iniziale

In the figure, nodes B, C, and D receive traffic from the outside to node A.

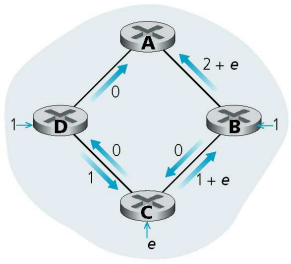
Link State Routing



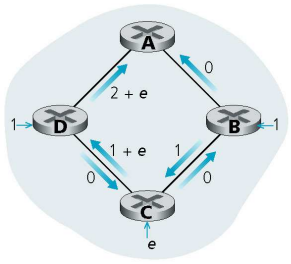
a. Routing iniziale



b. *B* e *C* rilevano il miglior percorso verso *A*, senso orario



c. *B*, *C* e *D* rilevano il miglior percorso verso *A*, senso antiorario



d. *B*, *C* e *D*, rilevano il miglior percorso verso *A*, senso orario

Comparison between DV and LSR

DV	LSR
Decentralized	Global
Messages for neighbors only	Broadcast Messages
Information regarding all destinations	Information about your links only
Infinity Counting	Synchronisation problems (oscillations)
Messages only for link changes	Increased message traffic (periodic sends)
Not very robust against malicious nodes	Robust to attacks

Internet Routing

Basic principles:

- Dynamic Routing
- Intra-AS and Inter-AS routing

The Internet is organized into "autonomous systems" (AS).

AS definition: unit of routing policy, either a single network or a group of networks that is controlled by a common network administrator on behalf of a single administrative entity (such as a university, a business enterprise, or a business division)

Each AS is internally strongly connected.

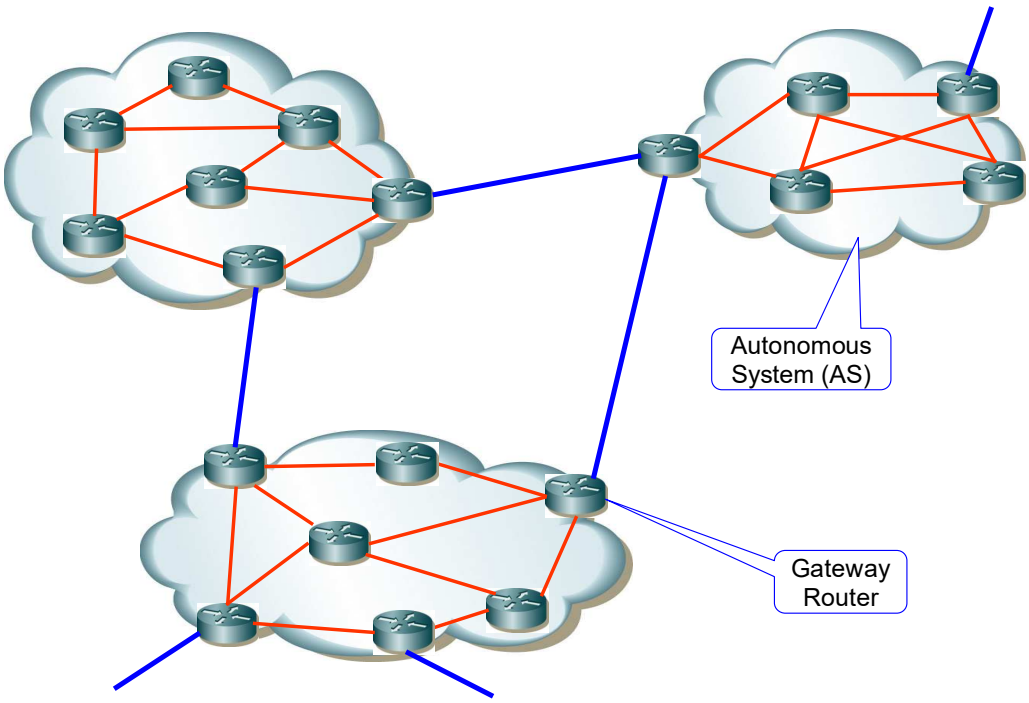
Internet Routing

Protocols:

- Interior Gateway Protocols (IGPs) within AS (RIP, OSPF, HELLO, IS-IS)
- Exterior Gateway Protocols (EGPs) among AS (EGP, BGP-4)

Università di Catania

AS



RIP

Routing Information Protocol:

- RIP v1 (RFC 1058)
- RIP v2 (RFC 1723, 2453).

- Based on Distance Vector.
- Metric used: HOP (Number of links traversed)

RIP v1

The maximum number of HOPs allowed is 15: with networks that are too large, convergence would be slow.

Beyond that value, the router is considered unreachable.

The Routing Table (RT) are exchanged every 30 seconds.

If a path for a router is not updated within 180 seconds, its distance is set to infinity.

After other 120 seconds (garbage-collection timer) the router is deleted from the RT.

RIPv1 messages

RIP v1 uses two types of messages:

- REQUEST: to ask adjacent nodes for information
- RESPONSE: to send routing information

A routing table contains:

- Destination Address
- Distance to destination(in hop)
- Next hop: neighbor router to send packets to
- Timeout
- Garbage-collection timer

RIPv1 datagrams

Variable length up to 512 bytes (max 25 target networks).

Command	Version	Must Be Zero
Address Family Identifier		Must Be Zero
IP Address		
Must Be Zero		
Must Be Zero		
Metric		

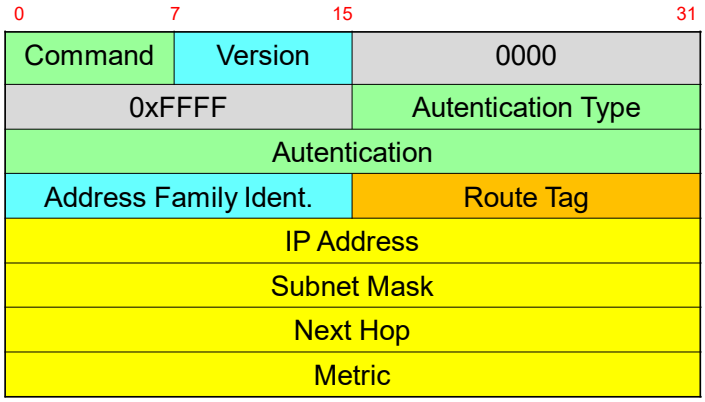
Command : 1 = request, 2= update.
Version : Protocol version.
Address Family Identifier : always 2 for IP protocol
IP Address : destination address (network or subnet)
Metric : hop count (value between 1 and 15)

RIP v2

Features of RIPv2

- CIDR and VLSM addressing
- Message authentication
- next hop identification
- Split horizon, poison reverse

RIPv2 datagrams



Command : 1 = request, 2= update.

Version : Protocol version.

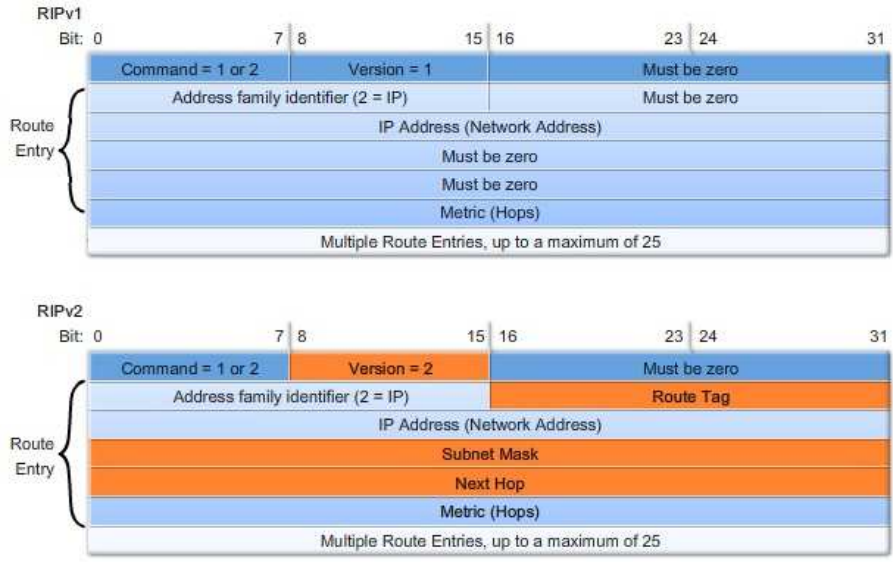
Address Family Identifier : always 2 for IP protocol

Route Tag: Identifying External Routes

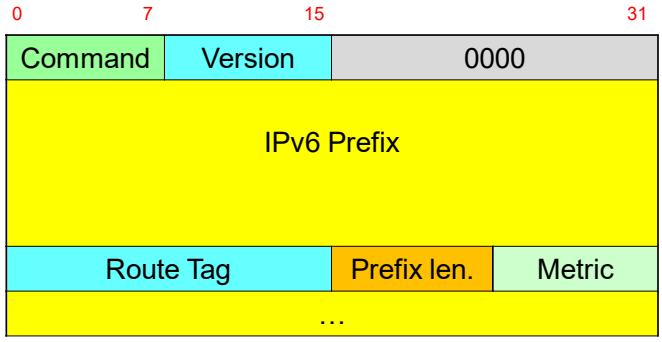
IP Address : destination address (network or subnet)

Metric : hop count (value between 1 and 15)

RIPv2 datagrams



RIPng



RIPng is based on RIPv2, but is not an extension of it. It is meant exclusively for IPv6 (does not support IPv4).

RIPng has the same features as RIPv2 except authentication

OSPF

Open Shortest Path First

- OSPFv1 (RFC 1131);
- OSPFv2 (RFC 2178, 2328).

- Designed to replace RIP
- It is based on Link State Routing
- Good for large networks
- Messages are authenticated
- Allows the use of different metrics

OSPF

The OSPF protocol consists of three parts:

- HELLO: Neighbor Discovery and Verification
- EXCHANGE: Initial DB synchronization
- FLOODING: DB update

Database in Link State

Each router has a database consisting of"link state record"

The various DBs are updated and synchronized via "Link State Advertisement" (LSA).

LSAs are issued:

- when a router encounters a new neighbor
- when a router – link breaks down
- when the cost of a link changes
- periodically, typically every 30 minutes

Link State Advertisement

LS Age	Options	Ls Type
Link State ID		
Advertisement Router		
Link State Sequence Number		
Link State Checksum	Lenght	
LS Data		

OSPF common header

Version	Type	Message Lenght
Router ID		
Area ID		
Checksum		Autentication Type
Autentication Data		
Rest of the OSPF Message		

Type:

1. Hello
2. DB Description
3. LS Request
4. LS Update
5. LS ACK

Router ID: router che ha generato il messaggio

AreaID: Ip dell'area a cui si riferisce il messaggio

Types of OSPF datagrams

Hello: periodic transmission for Neighbor Discovery

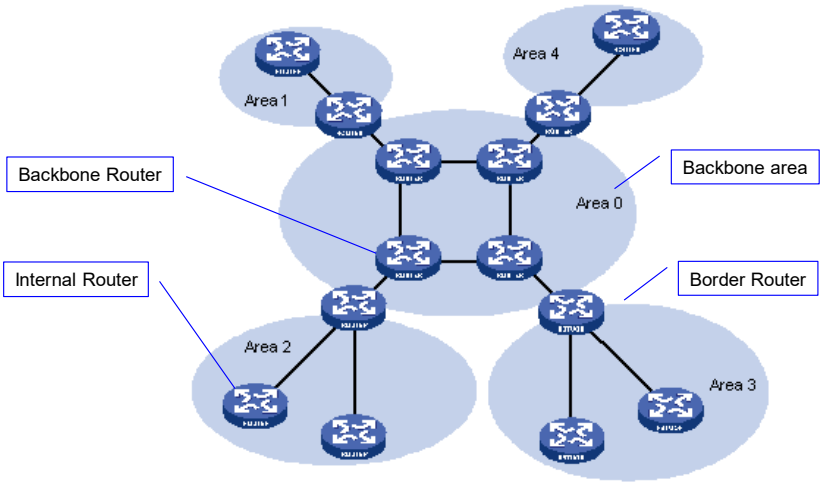
Database Description: used for the exchange of link-state information of each router

LS Request: requesting specific parts of a neighbor's link-state database

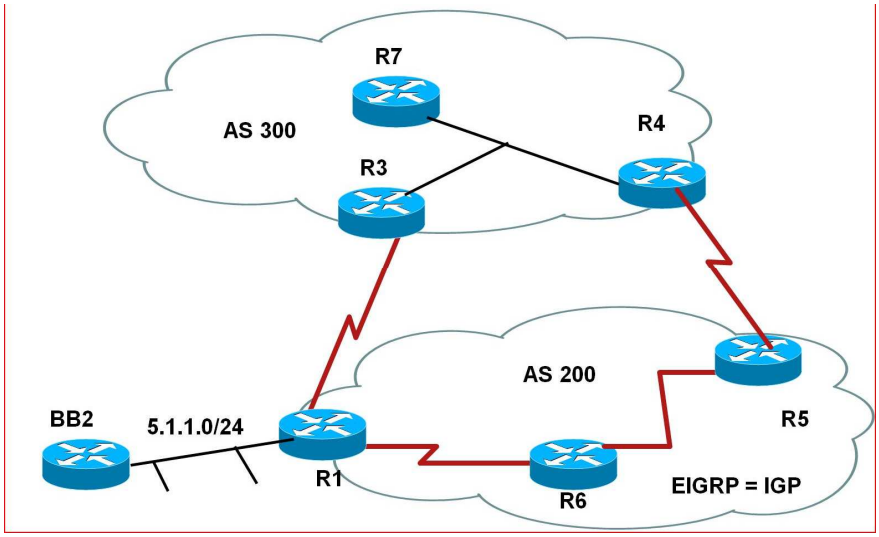
Link-State Update: Transfers link-state ads to neighbors

Link-State Acknowledgments: Send an acknowledgment of receipt of a link-state update

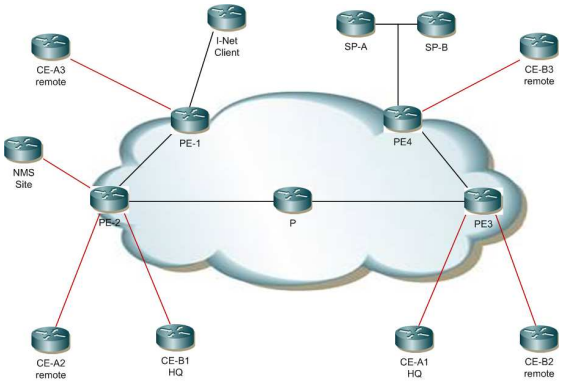
OSPF


Area 1
Area 4
Backbone Router
Backbone area
Area 0
Internal Router
Border Router
Area 2
Area 3

BGP v4



BGP



- AS Border Router (ASBR): Router connected to other stand-alone systems
- BGP speaker: router that supports the BGP protocol (a BGP speaker does not necessarily coincide with an AS border router)
- BGP Neighbors: pair of BGP speakers exchanging inter-AS routing information
  - interiors: if they belong to the same AS
  - exteriors: if they belong to different ASs