# The Network Layer: ICMP, IPv6 and Intradomain Routing

## COMPUTER NETWORKS A.A. 24/25

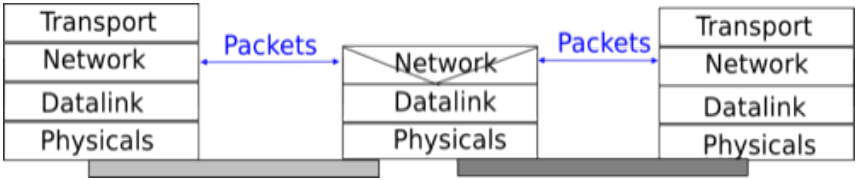

Leonardo Maccari, DAIS: Ca' Foscari University of Venice,
leonardo.maccari@unive.it

Venice, fall 2024

# License

- The IPv6 slides contain material whose copyright is of Olivier Bonaventure, Universite catholique de Louvain, Belgium https://inl.info.ucl.ac.be and thus licensed under a Creative Commons Attribution-Share Alike 3.0 Unported License.

# The network Layer in the Reference Model

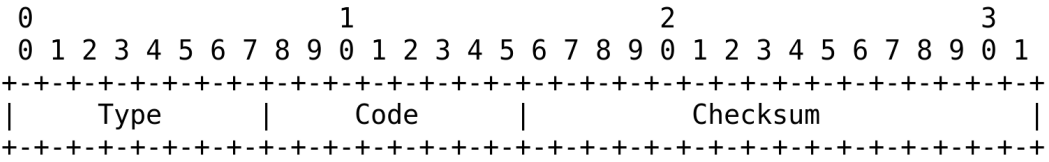


# Sect. 1 ICMP

# Internet Control Message Protocol

- ICMP is a signaling protocol: it typically does not carry user information
- It conveys error messages and messages relating to the operation and status of the network
- IP and ICMP are interdependent:
  - IP depends on ICMP for error handling
  - ICMP uses an IP datagram to convey messages
- ICMP does not use any transport layer protocols, so relies directly on the network layer

# ICMP Header




- an ICMP packet is generated when another packet produces an anomalous event.
- Type and Code: identify the type of error/event. Some are:
  - Destination (3): destination not routable/the destination port is closed, or fragmentation needed (Don't fragment is set to 1, but MTU is not enough)
  - Redirect (5): router notifies that there is a better next hop (generally ignored)
  - TTL exceeded (11): The packet got stuck in a loop
  - Echo request/reply (8/0): tell me if you exist
- subtypes can have more headers

# ICMP Datagrams

- Datagrams containing ICMP are treated like any other
- The only exception is that if an ICMP error message causes an error, then no further message is sent
- The reason is obvious: you don't want to create a ping-pong of ICMP messages that never ends.

# ICMP is Your Friend!

- ICMP is used to troubleshoot and test networks.
- Some of its uses are important for network surveying
- Examples...

# Uses of ICMP: ping

- in certain situations, you don't know if your network is working, or if the server you want to reach is on, or if DNS is working ...
- ICMP Echo request generates a response packet from the server (if it is reachable and turned on)
- Echo requests are colloquially known as *pings*
- With the `ping` command you can send requests and perform basic reachability and latency tests

# ping example

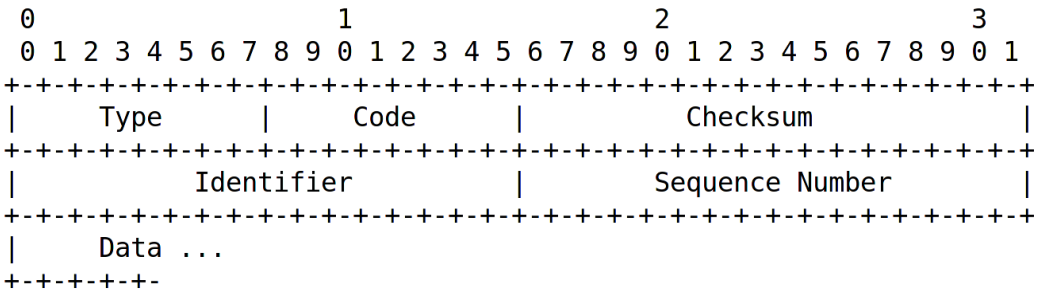
```
$ ping -c 5 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=115 time=10.3 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=115 time=10.0 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=115 time=10.5 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=115 time=10.7 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=115 time=11.1 ms

--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 10.012/10.526/11.077/0.366 ms

$ ping -c 5 202.232.2.191
PING 202.232.2.191 (202.232.2.191) 56(84) bytes of data.
64 bytes from 202.232.2.191: icmp_seq=1 ttl=50 time=278 ms
64 bytes from 202.232.2.191: icmp_seq=2 ttl=50 time=281 ms
64 bytes from 202.232.2.191: icmp_seq=3 ttl=50 time=281 ms
64 bytes from 202.232.2.191: icmp_seq=4 ttl=50 time=281 ms
64 bytes from 202.232.2.191: icmp_seq=5 ttl=50 time=280 ms

--- 202.232.2.191 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 278.040/280.064/280.985/1.113 ms
```

# Uses of ICMP: Echo request/reply



# Traceroute

- `traceroute` is a command that sends a packet to a destination, with TTL=1. This does not have to be an ICMP packet, and in fact UDP is normally used.
- On the first hop, the router will respond with an ICMP TTL exceeded. And you will discover the router IP (the source IP of the ICMP packet)
- Repeat with TTL=1,2,3 ... until the destination answers, and you have an estimation of the number of hops to the destination.
- Note: IP hosts are not forced to generate ICMP errors. So this is not a 100% working technique

```
$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  F32ZY.homenet.telecomitalia.it (192.168.1.1)  3.428 ms * *
 2  * * *
 3  172.17.104.116 (172.17.104.116)  10.445 ms  11.438 ms  11.401 ms
 4  172.17.104.100 (172.17.104.100)  13.361 ms 172.17.105.174 (172.17.105.174)  13.325 ms 172.17.105.70
      (172.17.105.70)  13.283 ms
 5  172.19.184.70 (172.19.184.70)  16.763 ms 172.19.184.64 (172.19.184.64)  18.310 ms  19.808 ms
 6  172.19.177.20 (172.19.177.20)  19.736 ms 172.19.177.14 (172.19.177.14)  10.250 ms 172.19.177.20
      (172.19.177.20)  11.073 ms
 7  ae48.milano11.mil.seabone.net (195.22.192.144)  10.976 ms ae48.milano50.mil.seabone.net
      (195.22.196.170)  10.852 ms ae48.milano11.mil.seabone.net (195.22.192.144)  10.835 ms
 8  72.14.216.154 (72.14.216.154)  12.783 ms 72.14.209.236 (72.14.209.236)  12.569 ms *
 9  * * *
10  dns.google (8.8.8.8)  9.721 ms  10.133 ms  10.975 ms
```

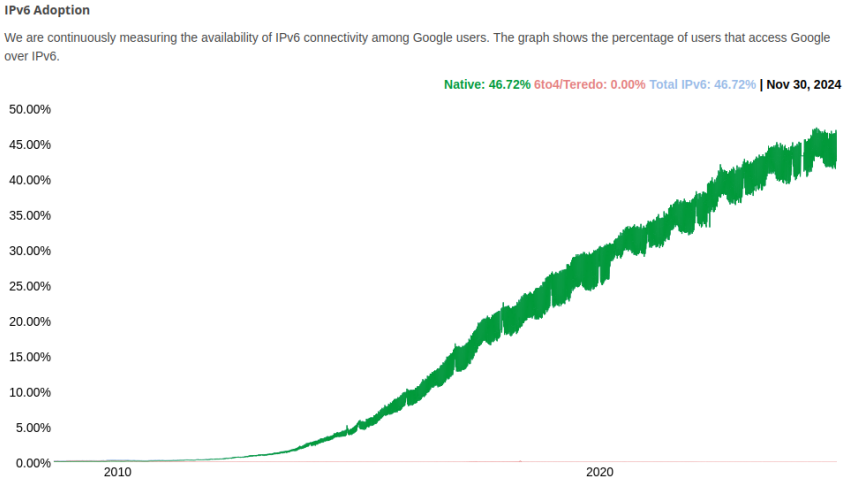
```
$ traceroute  202.232.2.191
traceroute to 202.232.2.191 (202.232.2.191), 30 hops max, 60 byte packets
 1  F32ZY.homenet.telecomitalia.it (192.168.1.1)  3.348 ms  3.472 ms  3.419 ms
 2  * * *
 3  172.17.104.116 (172.17.104.116)  9.240 ms 172.17.104.108 (172.17.104.108)  9.194 ms 172.17.104.116
    (172.17.104.116)  9.146 ms
 4  172.17.104.100 (172.17.104.100)  9.102 ms 172.17.105.70 (172.17.105.70)  11.133 ms 172.17.104.100
    (172.17.104.100)  11.038 ms
 5  172.19.184.64 (172.19.184.64)  16.417 ms 172.19.184.66 (172.19.184.66)  15.253 ms 172.19.184.68
    (172.19.184.68)  16.351 ms
 6  172.19.177.24 (172.19.177.24)  16.309 ms 172.19.177.14 (172.19.177.14)  9.969 ms  10.687 ms
 7  ae49.milano11.mil.seabone.net (195.22.205.98)  10.569 ms  10.497 ms ae49.milano50.mil.seabone.net
    (195.22.205.116)  10.458 ms
 8  ae0.milano52.mil.seabone.net (195.22.196.69)  12.362 ms  12.296 ms  11.561 ms
 9  ae-8.a00.mlanit02.it.bb.gin.ntt.net (129.250.9.41)  11.452 ms  10.644 ms  9.540 ms
10  ae-3.r20.mlanit02.it.bb.gin.ntt.net (129.250.3.120)  10.886 ms  10.842 ms  11.494 ms
11  ae-10.r21.parsfr04.fr.bb.gin.ntt.net (129.250.4.188)  26.071 ms  26.807 ms  26.747 ms
12  ae-13.r24.asbnva02.us.bb.gin.ntt.net (129.250.6.6)  108.656 ms  108.144 ms  108.058 ms
13  ae-2.r24.snjsca04.us.bb.gin.ntt.net (129.250.6.237)  162.168 ms  184.251 ms  161.641 ms
14  ae-0.a03.snjsca04.us.bb.gin.ntt.net (129.250.2.159)  162.506 ms ae-0.a02.snjsca04.us.bb.gin.ntt.net
    (129.250.2.3)  170.167 ms ae-0.a03.snjsca04.us.bb.gin.ntt.net (129.250.2.159)  165.185 ms
15  ae-0.iij.snjsca04.us.bb.gin.ntt.net (168.143.229.14)  163.668 ms  163.103 ms ae-1.iij.snjsca04.us.bb
    .gin.ntt.net (168.143.229.22)  172.307 ms
16  osk011bb00.IIJ.Net (58.138.84.177)  277.211 ms  280.954 ms osk011bb00.IIJ.Net (58.138.84.225)
    275.559 ms
17  osk008agr02.iij.net (210.130.16.19)  285.021 ms  280.732 ms  285.329 ms
```

# Sect. 2 IPv6

# IPv6

- The number of IPv4 addresses was soon understood to be too small
- NAT should have been a temporary solution, a definitive solution is IPv6
- IPv6 is now supported by all OS, but yet, after 25 years of its definition most of the Internet hosts uses IPv4.
- However, its usage is increasing with time.
- However, this does not mean IPv4 usage has decreased... So even if it is clear that IPv6 provides benefits, it is not clear it will finally replace IPv4.

# IP Adoption by Google



# IPv6

- The reason for the slow penetration is that IPv6 is not back-compatible with IPv4
- Or you use one or another
- Since there are ISPs that do not provide IPv6 to its customers, if you don't use IPv4 you can not communicate with them
- However, in 2011 we allocated the last available block of IPv4 to a RIR (the regional entity that assigns IP address to the LIR, the local entities such as ISPs)
- How are these addresses distributed?

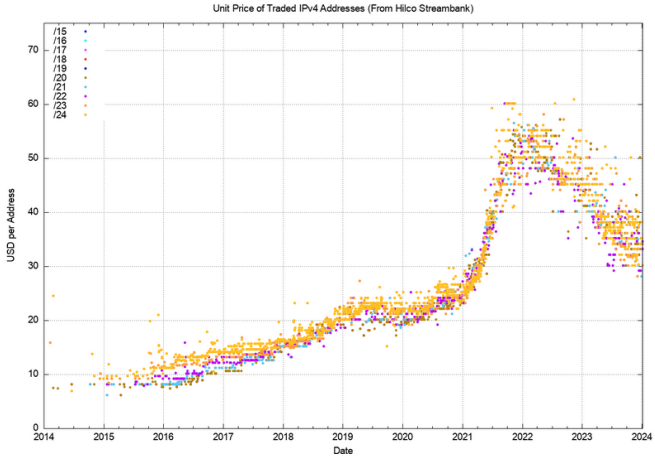
# IPv4 address distribution: top 10 countries (2013)

| | | | | | |
|---|---|---|---|---|---|
| 1 | US | 1,617,753,952 | 43.9% | 4.78 | USA |
| 2 | CN | 343,277,568 | 9.3% | 0.24 | China |
| 3 | JP | 190,439,680 | 5.2% | 1.54 | Japan |
| 4 | DE | 123,757,440 | 3.4% | 1.48 | Germany |
| 5 | GB | 119,183,240 | 3.2% | 1.76 | United Kingdom |
| 6 | KR | 112,498,944 | 3.1% | 2.17 | Korea |
| 7 | BR | 87,198,720 | 2.4% | 0.40 | Brazil |
| 8 | FR | 82,354,800 | 2.2% | 1.27 | France |
| 9 | CA | 69,177,344 | 1.9% | 1.80 | Canada |
| 10 | IT | 54,729,024 | 1.5% | 0.93 | Italy |

76% of the allocated IP space is the top 10 countries. None of them is in Africa.

# Trading IP addresses

IP addresses can be traded, how are the price dynamics[1]?




[1] See https://labs.apnic.net/index.php/2023/01/22/ip-addresses-through-2022/

# IPv4 Vs IPv6

- Bottom line is, if we want to open up the access to IP resources, we need IPv6

# IPv6 Main Features

We don't have time to get into the details of IPv6, but we will look at some of its main features:

- Longer IP addresses
- A new IP header
- No more router fragmentation
- A replacement for `DHCP` (more on this in future lessons)
- Mandatory support for encryption
- ICMPv6

# IPv6

## ↳2.1 IPv6 Address Format

# IPv6 Address Format

- An IPv6 address is made of 128 bit, and it is written in hexadecimal format separated by columns, like the following:

  `abcd:ef01:2345:6789:abcd:ef01:2345:6789`
- In cases when the there are block of zeroes, they can be compacted:

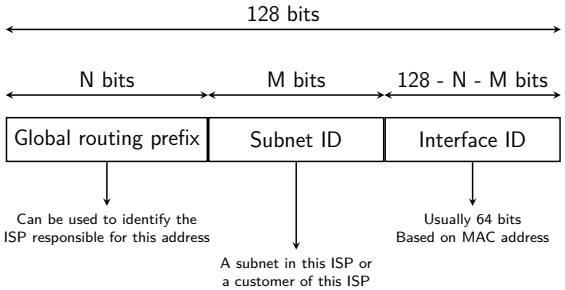
  `ff01:0:0:0:0:0:0:101` is represented as `ff01::101`
- As in IPv4 the address have a prefix of a certain length:

  `2001:0db8::cd30:0:0:0:0/60`

# IPv6 Prefixes

An IPv6 unicast address is composed of three parts :

- A global routing prefix that is assigned to the Internet Service Provider that owns this block of addresses
- A subnet identifier that identifies a customer of the ISP
- An interface identifier that identifies a particular interface on a host



# IP Availability

- In today's deployments, interface identifiers are always 64 bits wide.
- This implies that while there are $2^{128}$ different IPv6 addresses, they must be grouped in $2^{64}$ subnets.
- The current IPv6 allocations are part of the `2000::/3` subnet

# IPv6 Availability

The typical size of the IPv6 address blocks are :

- /32 for an Internet Service Provider
- /48 for a single company
- /56 for small user sites
- /64 for a single user (e.g. a home user connected via ADSL)
- /128 in the rare case when it is known that no more than one host will be attached

# IPv6 Availability

- In practice, we can have $2^{32-3}$ ISP in the world, that is about one every 15 person in the world.
- Every ISP can allocate $2^{48-32} = 65536$ big companies
- Every user that gets /64 can host $18 \times 10^{18}$ addresses in its home network...
- This is a lot, but who cares, we have $6 \times 10^{23}$ IP addresses per squared meter of the hearth surface...

# Address Assignment

- Considering those numbers, there is no need to optimize the subnet assignment anymore
- Most of the address space in a network will be unused anyway, so the kind of reasoning we did with IPv4 subnetting (how to minimize the leased addresses?) is obsoleted.

# IPv6 Special Addresses

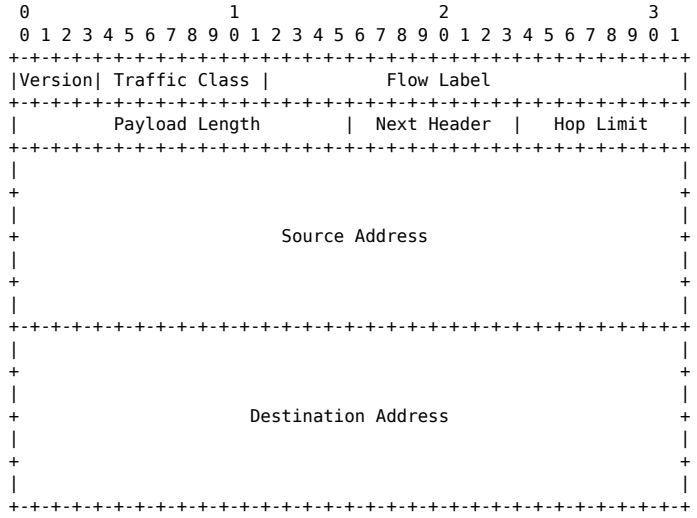
As with IPv4 there are some special addresses in IPv6:

- `fc00::/7`: Unique Local Unicast. These are like private IPs in IPv4. There is no need for them anymore, but this class can be used to make local networks to test the protocols
- `0:0:0:0:0:0:0:1`: loopback interface, *this host*.
- `ff::/8`: multicast addresses
- `fe80::/10`: Link Local Unicast (more on this later).

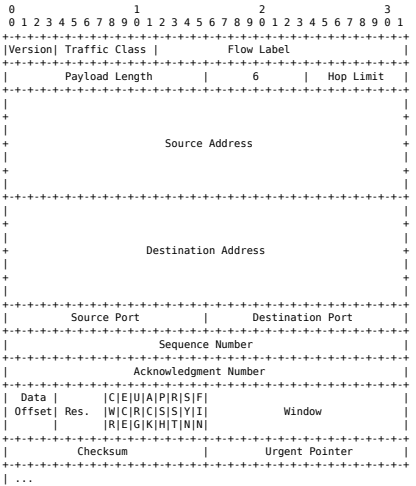
# IPv6

## ↳2.2 IPv6 Header

# A new Header


0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Version
Traffic Class
Flow Label
Payload Length
Next Header
Hop Limit
Source Address
Destination Address

# A new Header




- `Next Header`: replaces the protocol type of IPv4 (6=TCP)
- `Hop Limit`: replaces the TTL
- There is no checksum. The IP checksum was completely removed
- There is no support for fragmentation
- What follows in this case is a TCP segment. IPv6 does not impact on the transport layer, besides the change in the addresses.

# Chain of Headers

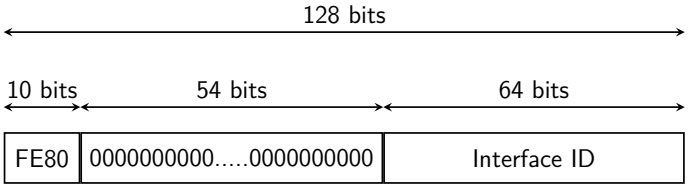
- Headers can be more than one, so that the `Next Header` will point to another header, with its own `Next Header` itself.
- This makes IPv6 extensible.

# IPv6

## ↳2.3 Address Assignment

# LLU addresses

- Each host can compute its own link local address by concatenating the `fe80::/64` prefix with the 64 bits identifier of its interface (so-called MAC address, 64 bit that are unique per NIC, more on this in next lessons).
- Link local addresses can be used when hosts that are attached to the same link (or local area network) need to exchange packets.



# LLU addresses

- They are used notably for address discovery and auto-configuration purposes, normally before the NIC receives a public IP address.
- A router cannot forward a packet whose source or destination address is a link local address.
- There are also Link Local IPv4 addresses (169.254.0.0/16), but the probability of a conflict is not so small, so they are rarely used.
- Recently, RFC7217 introduced a random but stable way of creating the interface ID using hash functions, not to reveal the MAC address

# SLAAC and DHCPv6

- Besides its LLU address, the host should receive a public IP.
- The host-id will be the MAC address, however, two protocols are used to discover the network prefix:
  - `SLAAC` (Stateless Address Autoconfiguration): the network router will simply periodically broadcast in the network a SLAAC packet that contains several parameters, among which the network prefix. The hosts will receive the SLAAC packet, compose the netid with their own MAC and this will be their own address.
  - `DHCPv6`: the v6 version of `DHCP` we will see in the next lessons

# IPv6

## ↳2.4 IPSec

# ESP and AH

- IPSec is a set of standards that implement encryption and authentication on IP packets
- IPSec was introduced in IPv4 as an optional feature
- With IPv6, the support for IPSec is mandatory.
- This does not mean every router uses it, it just means every router must support it.

# IPv6

## ↳2.5 Fragmentation

# IPv6 and Fragments

- Fragmentation on path done by routers is a pain, because it forces routers to use memory and computation
- IPv6 removed support for on-path fragmentation.
- The sender can still fragment the packets using a dedicate header, but it does not make much sense, as it is easier to just split the segments in smaller packets
- In some cases it is not convenient. An example is a large DNS response packet, UDP has no clue on the MTU used, if the packet is dropped due to MTU, you introduce delay (and at what level should you implement the re-send? UDP has no sending window). When the IPv6 layer receives the packet it fragments and send.

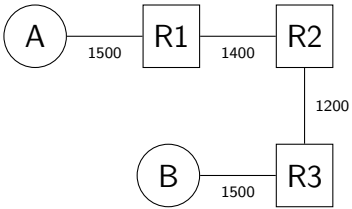
# IPv6

## ↳2.6 ICMPv6

# ICMPv6

- As the IP layer changes, also the ICMP protocol must be updated
- In most of the basic features ICMP remains the same, one relevant feature is the `Packet Too Big` ICMP code
- This replaces the `Fragmentation Needed` code of ICMPv4

# Toy Network




- Consider the network in the figure and imagine the start of a TCP connection.
- Both A and B will select MSS=1460
- Packets from A to B will be fragmented at R1, and then again at R2
- Can A and B discover the lowest MTU on the path, and use directly MSS=1160?

# Path MTU Discovery

- With IPv4 this is done by A sending a segment of size 1460 with the `Don't Fragment` bit set to True
- R1 will need to fragment the packet, but can't. So it will generate an ICMPv4 packet with code 4, this packet includes the allowed MTU.
- The IP layer will receive the ICMP packet and pass it to the TCP layer (this is breaking the layer architecture, but well, it works...), the TCP layer will reduce the MSS
- The same thing will happen again in R2, and the same on the other direction.
- After the necessary amount of failures, the minimum MTU is identified.

# Path MTU Discovery

- The same happens with IPv6, with the main difference that there is no `Don't Fragment` bit
- The principle is exactly the same, but it is more important to support it, because routers will not fragment anyway, so the MSS must be carefully chosen by A and B.

# Sect. 3 Intradomain Routing

# Intradomain Routing

- Carrier grade networks of ISPs have internal complicated topologies, as well as big organizations (like universities)
- These networks are partitioned in several subnets connected by routers, each one with its own addressing
- There are border routers that connect one network to another and need to forward the packets from one to another.
- An intradomain routing protocol implements the control plane on a network in the same administrative domain, i.e. owned by a single administrator that decides the internal routing policies on his own.

# Intradomain Routing: OSPF

- A single administrative domain is expected have a limited amount of subnets and routers connecting them. Order of tens, or hundreds at most.
- This is small enough to use a link-state routing protocol, that we know, can react to failures faster than a DV routing protocol, but produces more network overhead.
- Two of the most used protocols are Open Shortest Path First (`OSPF`, RFC 2328) and IS-IS (RFC 1195)
- `OSPF` follows the basic principles of a link-state routing protocol we already studied, with a couple of key modifications we explain the next slides.

# Intradomain Routing

## ↳3.1 OSPF Areas

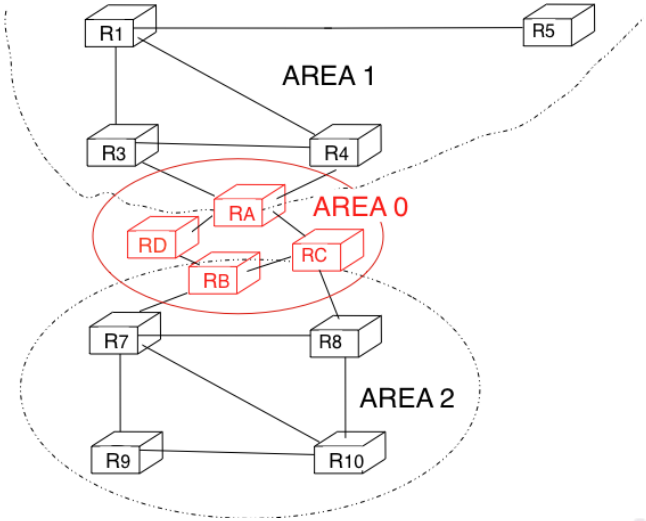
# OSPF Areas

- The network administrator partitions the network in different areas
- An area is a physically contiguous part of the network, connected to some other area by a limited number of routers.
- There are two kinds of routers for OSPF:
  - Internal routers: routers that are connected only to other routers in the same area
  - Border routers: routers that belong to more than one area
- Some of the routers export a network prefix, because they are the gateways of a subnet

# Backbone Area

- Area zero has a special meaning for OSPF, as it collects all the border routers, and eventually some router that do not belong to another area (like RD)
- The routers that do not belong to the backbone area can reach the other ones only passing across the backbone area.
- Every OSPF domain has an area zero, and only one.

# OSPF Areas




- R1, R3, R4, R5 are internal routers in area 1
- RD is an internal router in area 0
- R7, R8, R9, R10 are internal routers in area 2
- RA, RB, RC are border routers

# Link-state Behaviour

- Inside each non-backbone area, routers distribute the topology of the area by exchanging link state packets with the other routers in the area.
- The internal routers do not know the topology of other areas, but each router knows how to reach the backbone area.
- Inside an area, the routers only exchange link-state packets for all destinations that are reachable inside the area.

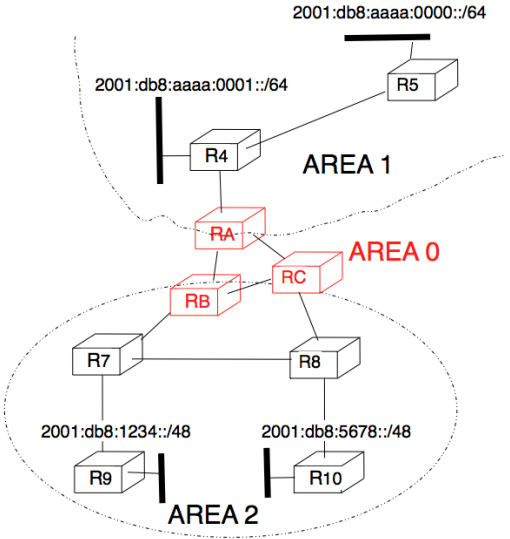
# DV Behaviour

- The inter-area routing instead is done by exchanging distance vectors between border routers.
- This helps to reduce the overhead due to link-state routing control packets.

# Routing Topology

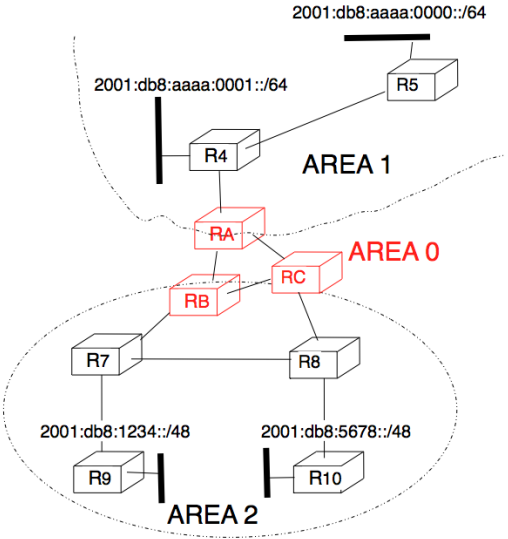
- We consider only a portion of the network, assigning IPv6 subnets to some of the routers
- In fact `OSPF` can be used with IPv6 or IPv4, and its logic does not change

# Routing Topology



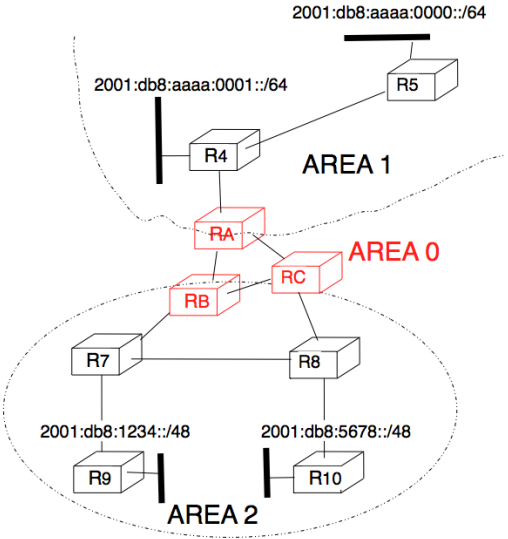

- R4 exports 2001:0db8:aaaa:0001:/64
- R5 exports 2001:0db8:aaaa:0000:/64
- R9 exports 2001:0db8:1234::/48
- R10 exports 2001:0db8:5678::/48

# Routing Topology




- RB advertises 2001:0db8:1234::/48 at a distance of 2 and 2001:0db8:5678::/48 at a distance of 3
- RC advertises 2001:0db8:5678::/48 at a distance of 2 and 2001:0db8:1234::/48 at a distance of 3

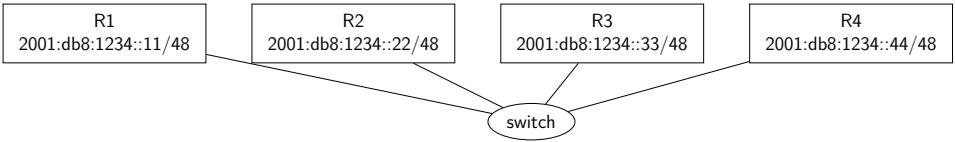
# Routing Topology




- RA advertises 2001:db8:aaaa:0000::/64 at a distance of 2 and 2001:db8:aaaa:0001::/64 at a distance of 1 from RA
- Alternatively, it advertises 2001:db8:aaaa:0000::/63 at a distance of 2 from RA
- This is called network aggregation, and reduces the number of lines in the other routers' forwarding tables.

# Sect. 4 OSPF Designated Routers

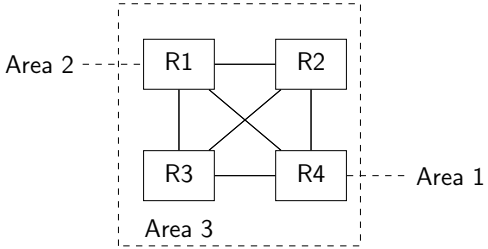
# Virtual Full Mesh Networks




- In this network, all routers are connected to the same switch
- Assume they are all in Area 3 and that R1 and R4 are border routers
- `OSPF` will exchange link-state messages, and the logical topology that is created will be a full mesh

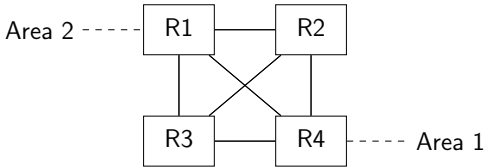
# Virtual Full Mesh Networks

- OSPF will exchange link-state messages, and the logical topology that is created will be a full mesh
- This will give the false perception that there is more than one path between the two border routers
- Router R4 for instance will believe that a packet going from Area 1 to Area 2 has three possible paths: R4-R1; R4-R2-R1; R4-R3-R1



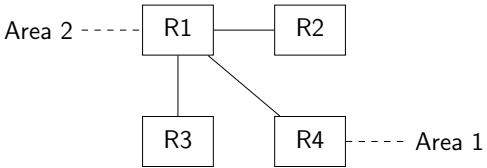
# Virtual Full Mesh Networks

- Assume the switch fails.
- R4 will detect the link failure between R1 and R4.
- It will reroute traffic to R2, or R3.
- However, all links are failed. But the order in which the failures are detected is not predictable.
- So R4 may try several routes, before it concludes that all are broken and stop exporting the routes to Area 2



# Designated Router

- In order to avoid this, OSPF allows to select a designated router
- All the other routers will export only the link to the designated router and not to the others



# Detecting Link Failure

- A final note on how to detect link failures
- The most straightforward way is to detect the loss of `HELLO` messages, however, generating them with a very high frequency can induce a very high load on routers
- Ideally, the link-layer will notify the network layer if the link fails, but this is not guaranteed by all link layers
- As an alternative, some heartbeat dedicated protocol can be used to monitor each link, sending unicast messages that are way easier to generate