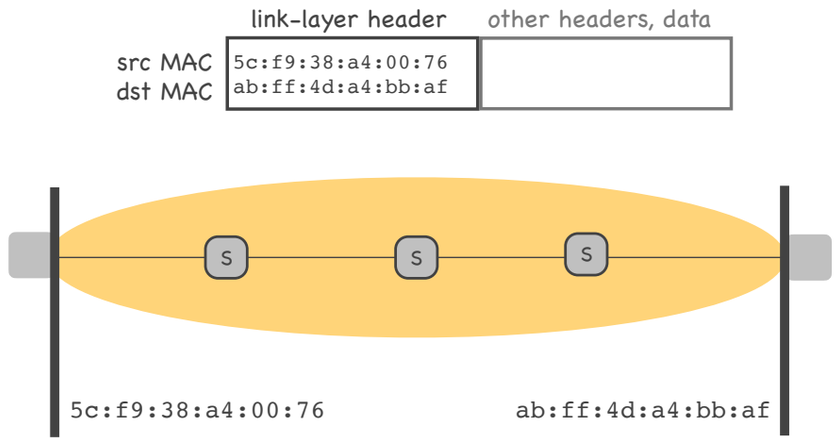
Outline

- Addressing
- Forwarding
- Learning
- Address resolution

Outline

- Addressing
- Forwarding
- Learning
- Address resolution

What type of addressing does Ethernet rely on?

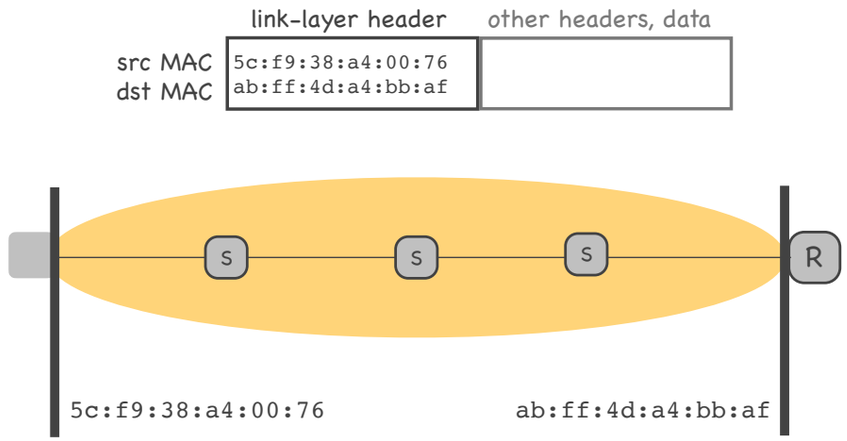


Consider a single IP subnet of type Ethernet.

Every end-system in this IP subnet has at least one network interface,
and every network interface has a link-layer address (also called MAC address, or Ethernet address, or physical address).

The thick vertical lines in this picture represent network interfaces.
5c:f9:38:a4:00:76 is the MAC address of the network interface on the left, while ab:ff:4d:a4:bb:af is the MAC address of the network interface on the right.

So, when one end-system sends a packet to another end-system, in the same IP subnet, the packet carries a link-layer header (also called MAC header, or Ethernet header), and inside this header there is a source MAC address and a destination MAC address (plus a few other fields).



What if an end-system sends a packet to another end-system that is located in a different IP subnet?
In that case, the packet will necessarily cross a router located at the border of the source end-system's IP subnet (to exit the subnet).
While the packet is traveling inside the source end-system's IP subnet, its source MAC address is the MAC address of the source end-system, while the destination MAC address is the MAC address of this router.

In general, a packet traveling inside an IP subnet always carries source and destination MAC addresses from the current subnet. The source MAC address is the MAC address of the first device (end-system or router) to forward the packet in this subnet, while the destination MAC address is the last device to receive the packet in this subnet.

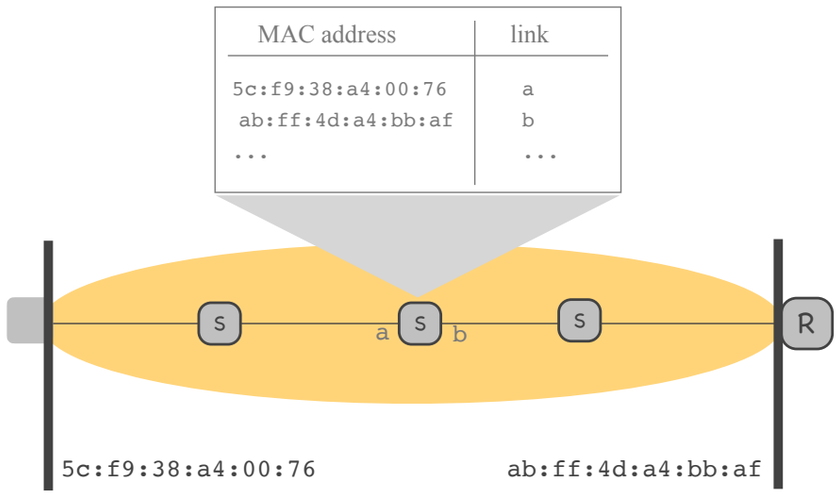
MAC address

- 48-bit number
  - *typical format: 1A-2B-DD-78-CF-CC*
  - *the value of each byte as hexadecimal*
- Flat
  - *not hierarchical like IP address*
  - *not location dependent*

Outline

- Addressing
- Forwarding
- Learning
- Address resolution

How do Ethernet switches forward packets?



Each switch names its network interfaces (also called links, or ports),
and keeps a forwarding table that maps MAC addresses to links.

When a packet arrives, the switch reads the destination MAC address from the MAC header, looks it up in the forwarding table,
and identifies the correct output link for this packet.

L2 forwarding

- Local switch operation that determines output link for each packet
- Relies on forwarding table
  - *maps destination MAC addresses to output links*
- Similar to IP (L3) forwarding, except...

MAC address

- Flat
  - *not hierarchical like IP addresses*
  - *not location dependent*

There is an important difference between IP addresses and MAC addresses:
MAC addresses are flat, not hierarchical.

L2 vs. IP forwarding

- L2: relies on flat addresses
  - *no way to group MAC addresses in prefixes*
  - *forwarding table size = # of active destination MAC addresses in the IP subnet*
- IP (L3): relies on hierarchical addresses
  - *IP addresses grouped in IP prefixes*
  - *forwarding table size = # of local IP prefixes + foreign aggregate IP prefixes in the world*

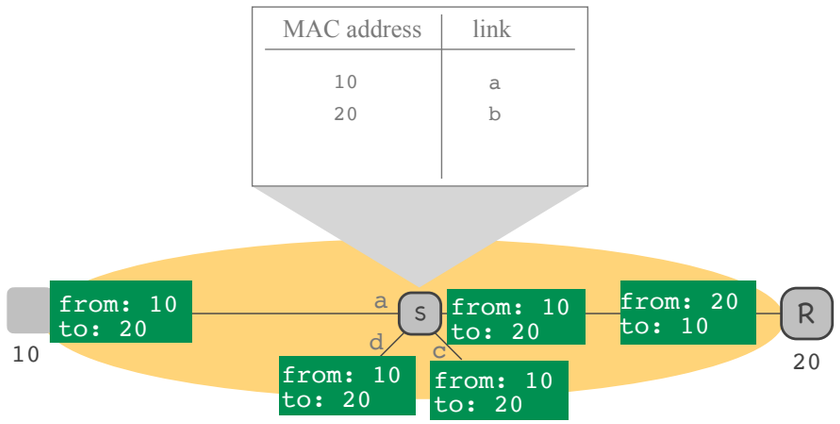
So, if we compare L2 to IP (L3) forwarding, the former relies on flat addresses, whereas the latter relies on hierarchical addresses.

The fact that MAC addresses are flat means that we cannot group MAC addresses, e.g., by prefix, as we do with IP addresses.
As a result, the forwarding table of a switch keeps an entry for every individual MAC address that is currently active in the local IP subnet.

Outline

- Addressing
- Forwarding
- Learning
- Address resolution

Who populates the forwarding tables of Ethernet switches?



The forwarding table of a link-layer switch is initially empty.
The switch fills the table based on the traffic it receives.

For example, if it receives a packet with source MAC address 10 at link (a),
it adds an entry to the forwarding table, indicating that:
if, in the future, it receives a packet with destination MAC address 10, it should forward that packet to link (a).

If the switch receives a packet with a destination MAC address for which no entry currently exists in its forwarding table,
it broadcasts that packet to all(*) links.

(*) It's not really all links, it's a subset of the links, in order to avoid loops.
Discussed in a couple of slides.

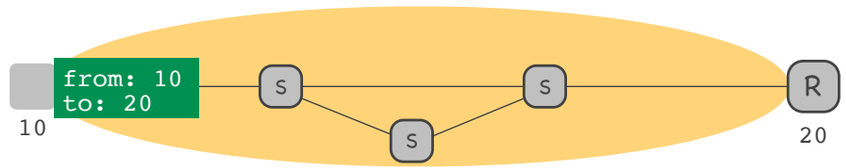
L2 learning

- Switch learns from traffic
  - *when packet with src MAC x arrives at link y, switch adds MAC x --> link y mapping to forwarding table*
- Broadcasts when it does not know
  - *when packet with unknown dst MAC arrives, switch broadcasts the packet*
- Serves similar role as IP routing, but…

L2 learning vs. IP routing

- L2 learning: relies on actual traffic
  - *switches do not exchange explicit routing information*
- IP routing: relies on routing protocol
  - *routers exchange explicit routing messages*

naïve broadcasting = forwarding loops!



We said that, when a switch receives a packet with unknown destination MAC address, the switch broadcasts the packet.
If all switches broadcast packets with unknown destination MAC addresses to all links (except the one where the packet arrived), there may be forwarding loops.