Lecture 1 Network Fundamentals

The Construction of Network

End-systems

Functions: send & receive data

Links

Functions: connect end-systems and network devices (routers, switches, and more)

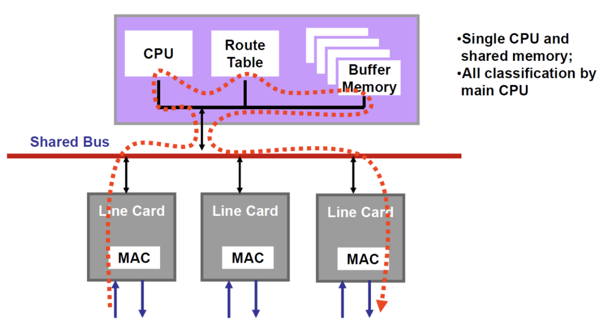
Network Devices (Switches & routers)

Functions: Switches & routers forward data to the destination

The Inside Part of a Router

> In the field of computer networks, we usually do not make a fairly clear distinction between switches and routers.

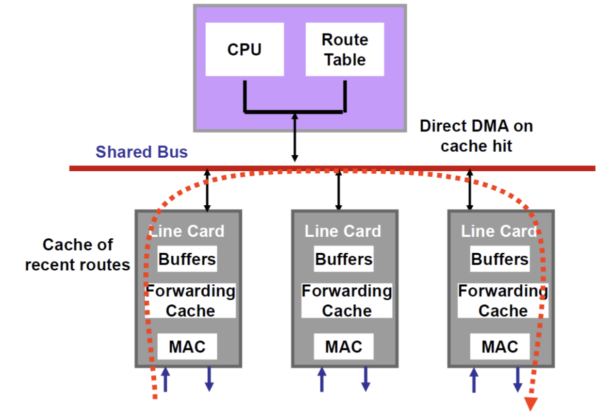
First Generation Routers



- Single CPU and shared memory
- All classification by main CPU

Since there are too many problems in this design, we offer variants and uogrades below.

Second Generation Routers



- We upgrade the Line Card which contains Buffers and Forwarding Cache now

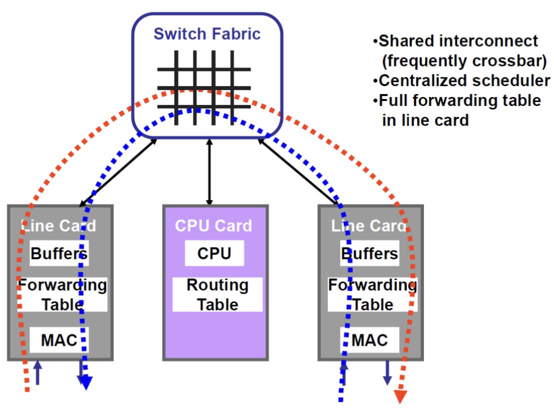
Now our strategy is listed below:

1. Normally, Line Card 1 → Line Card 2 utilizes only Line Card and Shared Bus, the CPU is not in process.
2. If the flow is exceeding the threshold, then CPU will be involved in this process.
3. 即：如果缓存命中，则给出下一跳；如没有，则仍需要访问CPU

However, there are still some problems:

1. Though we can easily decrease the participation frequency of CPU, but the overall system speed is constrained by the Shared Bus.

Third Generation Routers



- Shared interconnect (frequently crossbar) —— 共享交互
- Centralized scheduler —— 集中路由
- Full forwarding table in Line Card

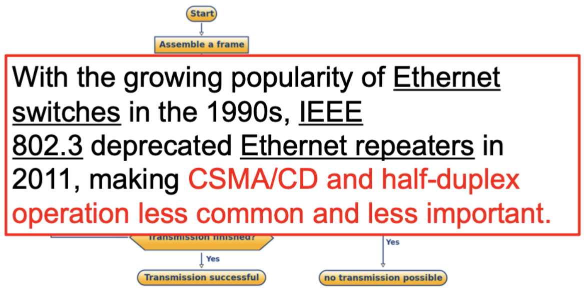
Design:

1. We use the Fabric to transport instead of Shared Bus.

2. In this way, we can indirectly expand the total "bus".

The system is nearly perfect in this way!

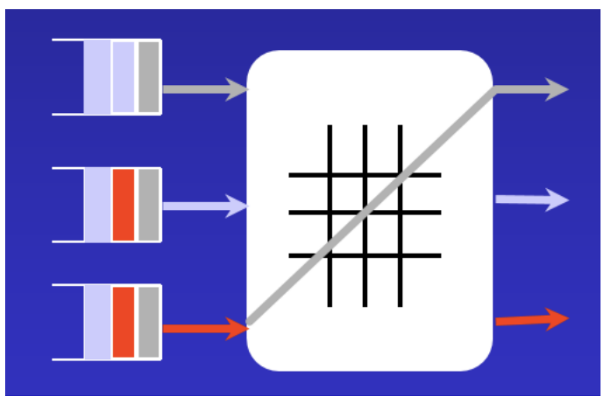
PS:
The design of Fabric Hardware guaranteed that any two signals do not collide or conflict within the Switch Fabric.

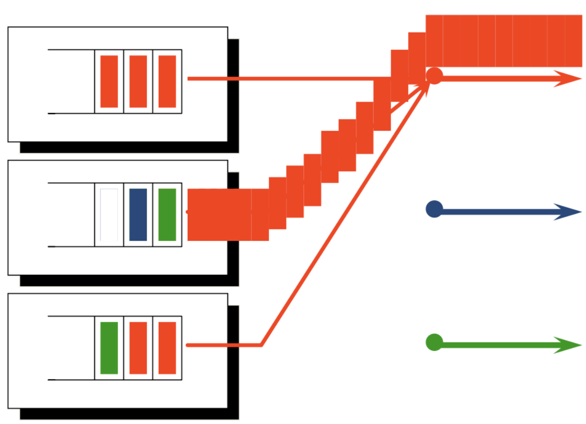


Contention

What if packets from different input ports contend for the same output port?

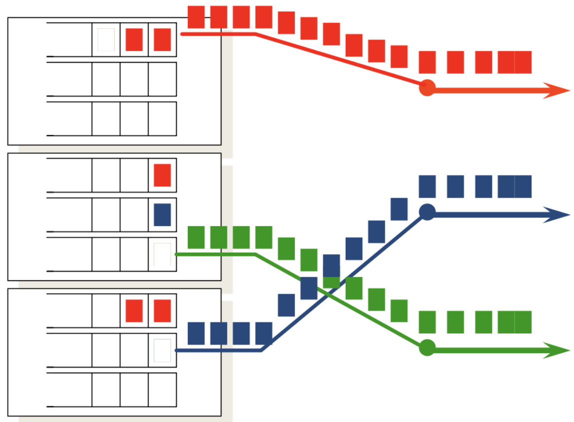
In fact, it will raise an error called *Head of Line Blocking (HOLB)*





Solution

We utilize a mechanism called *Virtual output queues (VOQs)*



⇒ Virtual Output Queues

- Each "Virtual Queue" is restricted to specified packages only.
- We use a Polling mechanism (轮询机制) to achieve this.

Control Plane vs. Data Plane

Data Plane

This plane is usually connected with *packet streaming*: table lookup, forward, filter, buffer
⇒ 表查找、转发、过滤、缓冲

Trait:

- Data reception/storage/forwarding...
- Need to be fast, the lighter, the better.

Control Plane

This plane is usually connected with *distributed algorithms*: track topology, compute routes, install forwarding rules

Trait:

- Making decisions, for example: The CPU plans a routing table based on routing protocols (OSPF/RIP).
- It doesn't need to be exceptionally fast; it's complex and implemented by the CPU at the software level.

The Classical Five Layers of Network

Services Provided by Each Layer

Num	Layer	Services Provided
Layer 5	Application	network access

Num	Layer	Services Provided
Layer 4	Transport	end-to-end delivery (reliable or not)
Layer 3	Network	global best-effort delivery
Layer 2	Link	local best-effort delivery
Layer 1	Physical	physical transfer of bits

The classical 5 layers are listed above.

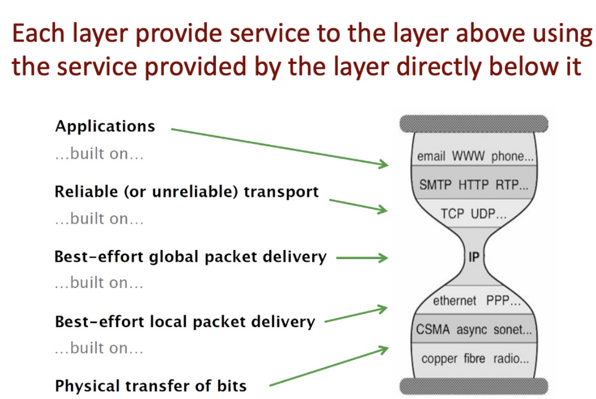
Packages Sent by Each Layer

Num	Layer	Role
Layer 5	Application	exchanges **messages** between processes
Layer 4	Transport	transports **segments** between end systems
Layer 3	Network	moves **packets** around the network
Layer 2	Link	moves **frames** across a link
Layer 1	Physical	moves **bits** across a physical medium

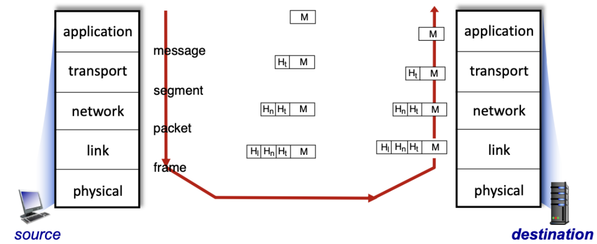
Each layer communicates with a unit of data.

Protocols on Each Layer

Num	Layer	Protocols
Layer 5	Application	HTTP, SMTP, FTP, SIP, ...
Layer 4	Transport	TCP, UDP, SCTP
Layer 3	Network	IP
Layer 2	Link	Ethernet, Wifi, (A/V)DSL, WiMAX, LTE, ...
Layer 1	Physical	Twisted pair, fiber, coaxial cable, ...

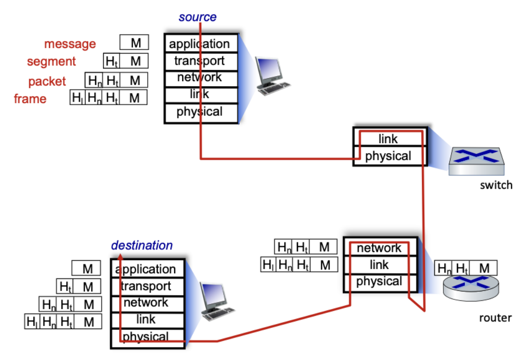


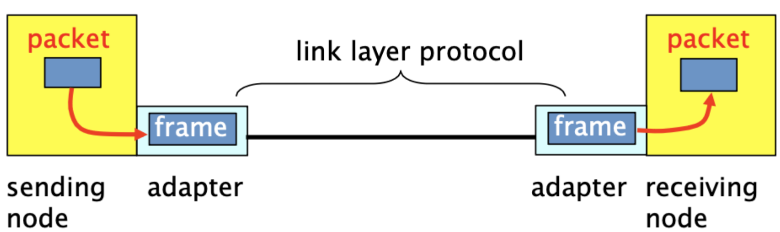
The Process of Package Transporting



For Src:
The initial information is **appended** with "control information of this layer" step by step in the top-down transmission process.

For Dst:
The information packets received at the lowest layer are **decapsulated (解封装)** step by step during the upward transmission process.





Switching

Address (Layer 2)

(1) MAC Address

- 48bits identifying the other end of link (6 bytes)
- Globally Unique
- MAC Address is **Hierarchical** (分层的)
- It is assigned with a *unique* MAC Address *around the world* with the help of **NIC**

An example:

- We give a MAC Addr called: "`34:36:3b:d2:8a:86`"`
- The segment "34:36:3b" implys the device belongs to *Apple, Inc. 1 Infinite Loop Cupertino CA 95014 US*

(2) Broadcast Address

- The address with all bits set to 1 identifies the broadcast address
- It is written as: "`ff:ff:ff:ff:ff:ff`"
- enables to send a frame to *all adapters* on the link

DHCP Service

> Since the MAC Address of one device unique, it 's really hard to utilize it to achieve worldwide connection. Therefore, we utilize IP Address to solve the problem. We use *Subnet-External Internet* to represent the connection in the real world. Now a problem arises： How do a new Host get its own IP Address?

Network adapters acquire an IP address using the Dynamic Host Configuration Protocol (DHCP).

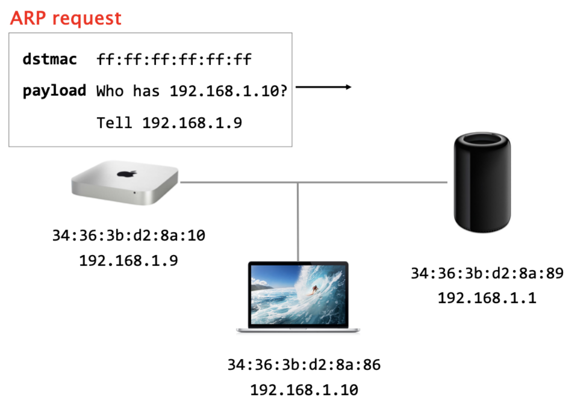
1. Host sends an “IP request” to everyone on the link using the broadcast address.
2. DHCP server (if any) answers with an IP address. (The DHCP Server will avoid IP conflicts in this subnet automatically when assigning a IP Address for response)
3. Then the host gets its own IP Address.

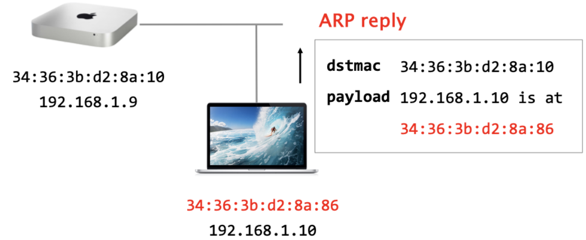
Address Resolution Protocol (ARP)

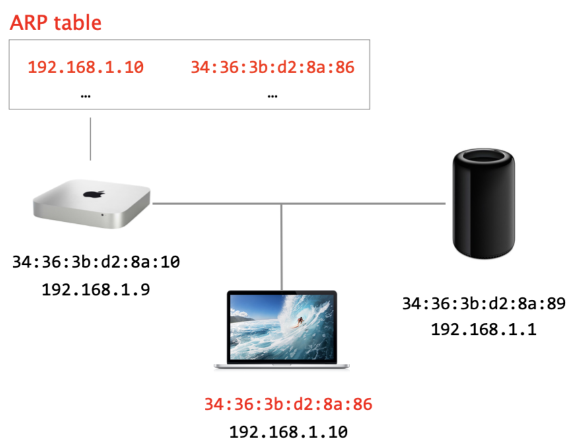
> Here we meet another problem: how do one host (Host A) send messages to another specified host (Host B). This is a *direct transmission*.

We use Address Resolution Protocol (ARP) to solve this problem.

1. Host A sends a Broadcast Message to all hosts, it contains "Who has this IP Address xxx.xxx?". **(ARP request)**
2. Every host is sent with this message and check for its own IP Address.
3. Host B finds itself suitable for this designated IP Address
4. Host B sent to Host A a message: "I am your destination!". **(ARP reply)**
5. Host A put this mapping (IP Address ↔ MAC Address) into ARP Table. **(ARP table)**







Hub & Switch

> There we meet another problem: the transmission above is direct, namely one-one mapping. However, we usually transport numerous message at a same time. Therefore, we need a **Repeater (中继器)** in the whole topology.

(1) Ethernet hub (Layer 1)

- repeating bits from one port to **all the other ones**
- Inefficient: each bit is sent everywhere

(2) Switch (Layer 2)

- repeating bits from one port to **the correct port**
- efficient: communications between different hosts are isolated

- Switch is a plug-and-play device ! (即插即用设备)

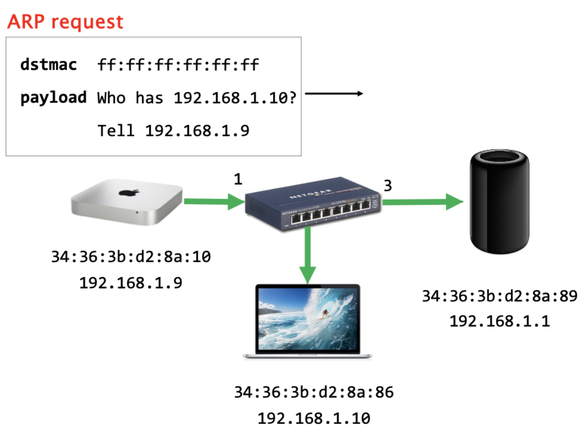
MAC Learning

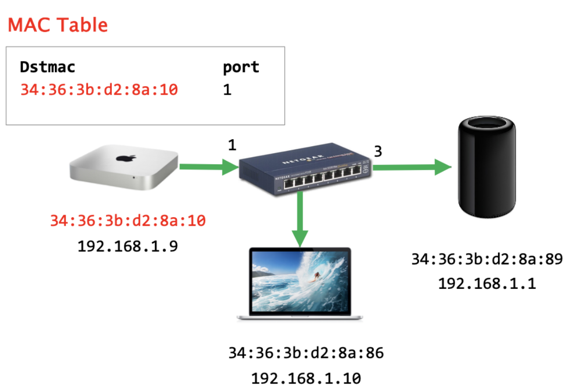
> - There comes a question now: How can Switch realizes repeating bits from one port to **the correct port**?
> - Why do switch know **the correct mapping** between port and device's MAC Address?

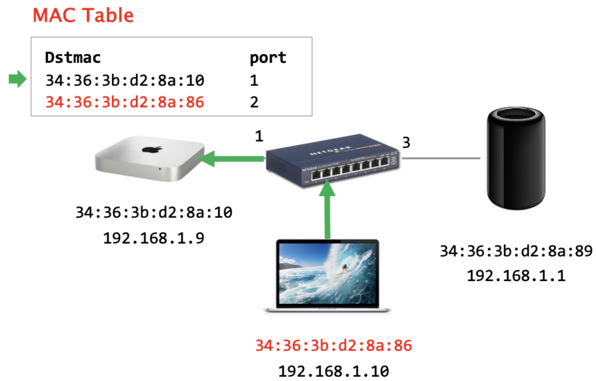
We will introduce a significant characteristic of switch to you, which is called "*MAC Learning*".

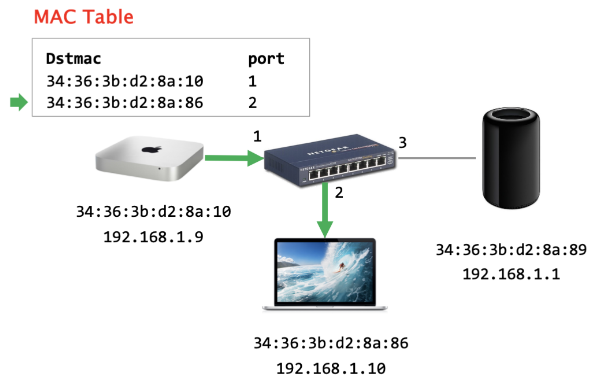
MAC Learning

1. When Host A sent an ARP request, the switch recorded its MAC and the port message sent to.
2. This mapping is recorded in MAC Table in switch.
3. For the same reason, the MAC Table will record the mapping between Dst Node's MAC Address and the port message sent back.
4. Over!







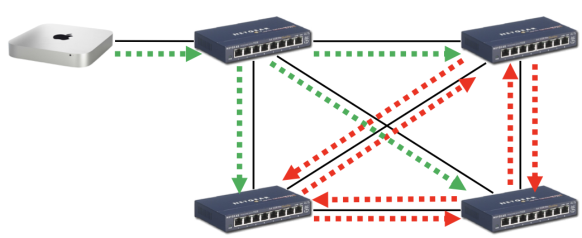


Contention

Broadcast Storm

If we meet this Closed Loop Topology(闭环拓扑结构), we will face this serious issue:

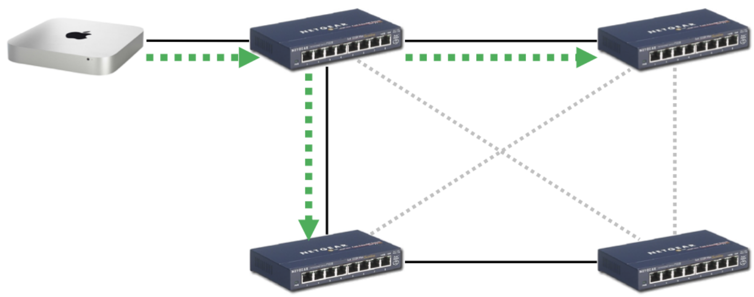
- Each frame leads to the creation of at least two new frames!
- exponential increase, with no TTL to remove looping frames...



Spanning Tree Protocol (STP)

This is the solution: Spanning Tree Protocol

⇒ Run the spanning tree protocol to disable some ports, so that the topology becomes a tree (rather than a graph)

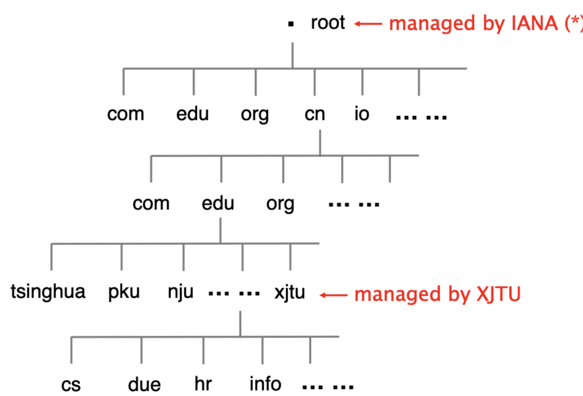


Domain Name System (DNS)

In fact, in our daily life, we can hardly use IP Address. Usually, we use Internet site (URL).

- Question: How can I know www.example.com is at 192.168.1.10?
- In fact, the *bottom-up architecture (自底向上体系结构)* is MAC Address → IP Address → URL
- The **DNS system** is a distributed data base which enables to *resolve a name into an IP address*.

Domain Name is Hierarchical



DNS Server

A DNS server stores Resource Records composed of a tuple called (name, value, type, TTL)

- TTL: Time To Leave, 需要常更新

records	name	value
A	hostname	IP **A**ddress
NS	domain	D**NS** Server name
MX	domain	**M**ail server name
CNAME	alias	**c**anonical **name** (规范名称)
PTR	IP Address	corresponding hostname

Using DIG to Resolve

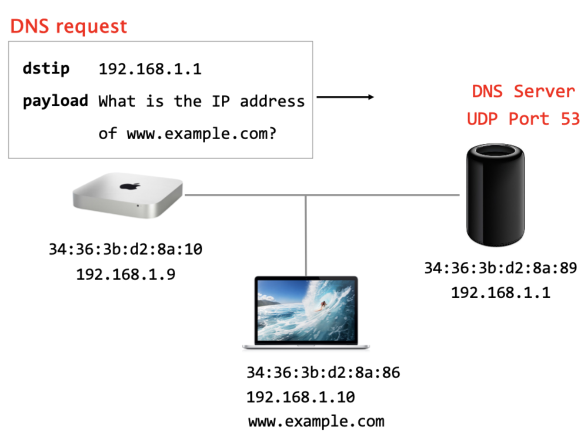
⇒ dig (domain information groper)

```bash
huluobo@ubuntu:/Users/huluobo$ dig www.google.com

; <<>> DiG 9.18.18-0ubuntu0.23.04.1-Ubuntu <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10336
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;www.google.com.        IN    A

;; ANSWER SECTION:
www.google.com.     15    IN    A 198.18.0.19

;; Query time: 3 msec
;; SERVER: 198.19.248.200#53(198.19.248.200) (UDP)
;; WHEN: Mon Mar 11 17:24:08 CST 2024
;; MSG SIZE  rcvd: 73

huluobo@ubuntu:/Users/huluobo$
```

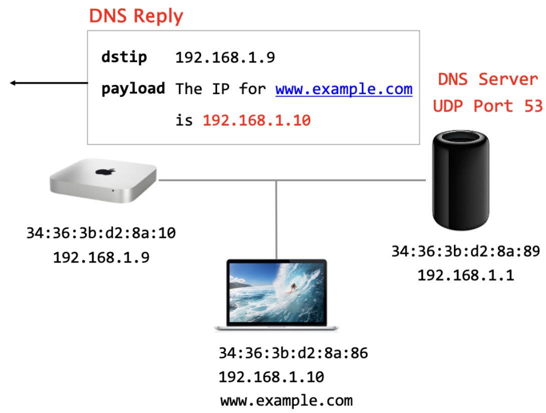
注意这里的：IN / A / 198.18.0.19

The Simple Process of DNS

The pictures shown below are only for the situation within the "same subnet":

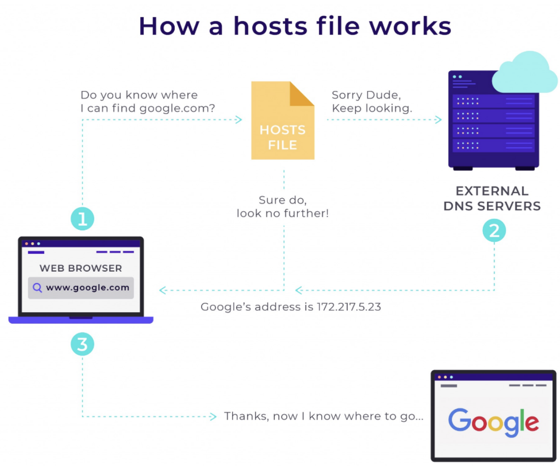


What you actually do

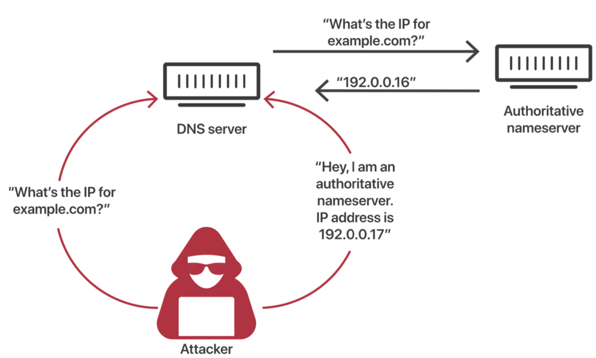


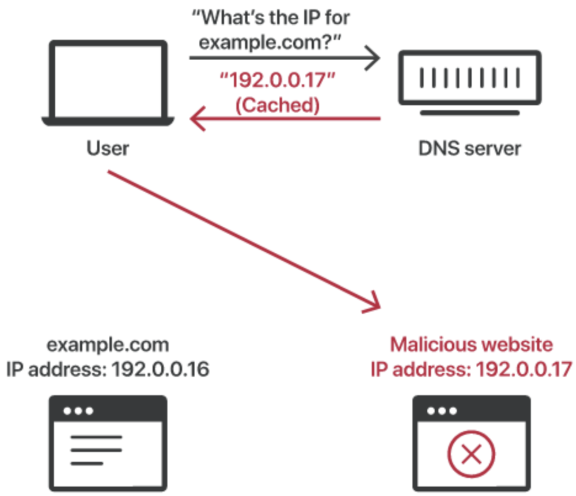
The Hosts File

1. Host file takes priority over DNS
2. It 's similar to a local cache of (IP ↔ URL)



DNS Cache Poisoning

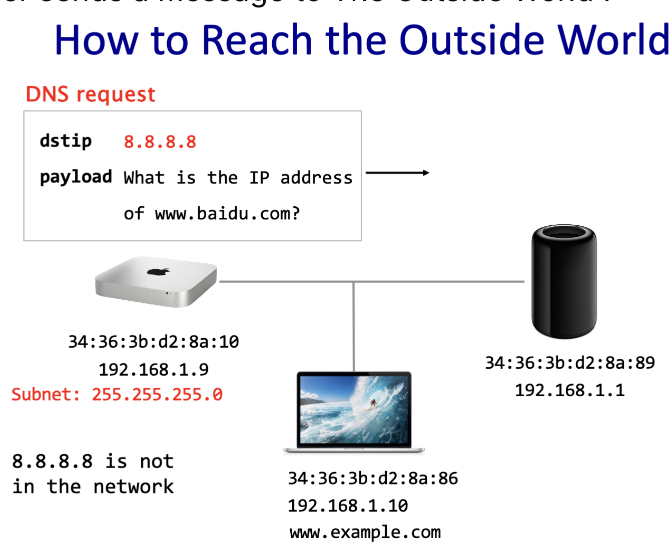


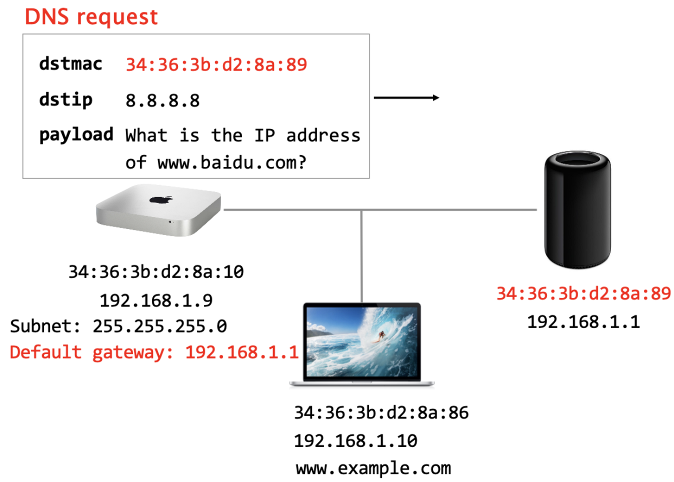


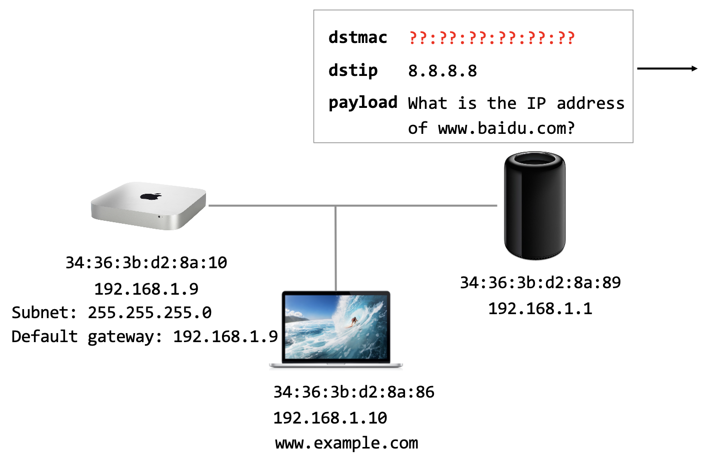
The Real Process of DNS

The pictures shown below contain the situation which is out of the "same subnet".

1. Host A wanna know whose IP Address is 8.8.8.8
2. Host A sends a Broadcast Message to all hosts.
3. The Gateway Server received this message.
4. The Gateway Server **checks** the relationship between the *target IP and the current subnet IP*, and finds out this IP Address is not belong to this subnet.
5. So the Gateway Server sends a message to The Outside World !




DNS request
dstmac 34:36:3b:d2:8a:89
dstip 8.8.8.8
payload What is the IP address of www.baidu.com?
34:36:3b:d2:8a:10
192.168.1.9
Subnet: 255.255.255.0
Default gateway: 192.168.1.1
34:36:3b:d2:8a:89
192.168.1.1
34:36:3b:d2:8a:86
192.168.1.10
www.example.com


dstmac ??:??:??:??:??:??
dstip 8.8.8.8
payload What is the IP address of www.baidu.com?
34:36:3b:d2:8a:10
192.168.1.9
Subnet: 255.255.255.0
Default gateway: 192.168.1.9
34:36:3b:d2:8a:89
192.168.1.1
34:36:3b:d2:8a:86
192.168.1.10
www.example.com