OPENFLOW

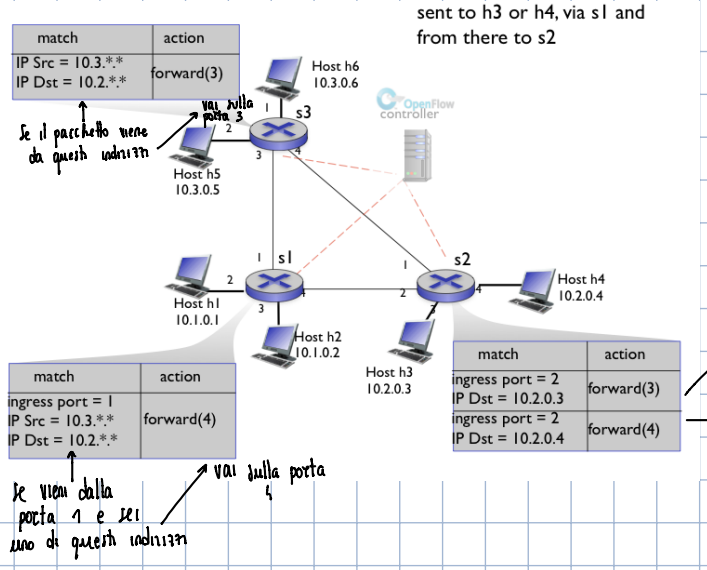
SDN risolve diversi problemi tra cui l'instradamento, le tabelle di inoltro.
Il protocollo utilizzato per inviare notifiche in un senso e integrare e configurare nel senso opposto è OPENFLOW.
Con l'inoltro generalizzato che consiste nel fatto che la tabella di inoltro si struttura in tre porzioni: ogni riga contiene tre parti (regola,azione e statistiche). Le regole sono una serie di selettori, tutta una serie di parametri che si applica a ciascun pacchetto; l'azione è quello che si fa sul pacchetto (inoltro, scarto, modifica ecc.); statistica sono dei contatori dei quanti pacchetti impattano una determinata regola.
Le tabelle di inoltro hanno una serie di righe e il loro ordine è significativo perché vengono scandite una dopo l'altra alla ricerca del match.

Match+azione : determinano diversi tipi di dispositivi. Si possono programmare dei dispositivi in grado di leggere le tabelle e poi essere un router, uno switch, un firewall o un NAT.

- Router
  - *match:* longest destination IP prefix
  - *action:* forward out a link
- Switch
  - *match:* destination MAC address
  - *action:* forward or flood
- Firewall
  - *match*: IP addresses and TCP/UDP port numbers
  - *action:* permit or deny
- NAT
  - *match:* IP address and port
  - *action:* rewrite address and port

Quello che si è tentato di fare ultimamente è di creare delle piattaforme uniformi che poi possono essere programmate e adattate a svolgere quello che serve.

Esempio OPENFLOW:



Piano di controllo

Sono tutte le funzioni che determinano il funzionamento della rete, in particolare l'instradamento.

ICMP: Internet Control Message Protocol

Architetturalmente sta sopra IP ma lo chiama direttamente, è implementato insieme a IP quindi viene considerato al livello di rete.

Nel payload viene inserito il messaggio ICMP, come fa a sapere che è proprio un messaggio di questo tipo? È un'informazione che si deve trovare nell'intestazione, è il campo protocol type.

I messaggi di ICMP sono notifica e diagnostica. In particolare contengono due campi che sono tipo e codice. C'è una lunga tabella che elenca tutti i possibili significati di questi due campi.

Type	Code	description
0	0	echo reply (ping)
3	0	dest network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

Internet non è temporalmente trasparente, i tempi cambiano sempre.

I messaggi ICMP servono anche per le notifiche, quando un host e un router scartano un pacchetto di solito fanno un messaggio ICMP in cui spiegano perché è stato scartato con un opportuno tipo e codice, si copia un'opportuna parte del pacchetto e la mandano alla sorgente del pacchetto perso.

Fare Ping su un host e si riceve risposta vuol dire che è possibile comunicare con quell'host e ci dice anche l'RTT (ma è un'informazione aggiuntiva, è stato principalmente pensato per verificare la connettività).

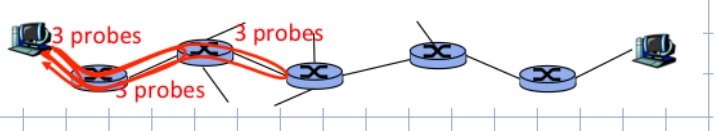
Traceroute

Cerca di esplorare e rendere noto qual'è il percorso router attraversato. Questo applicativo si trova solo sul nostro host e vuole scoprire chi è il prossimo router, per scoprirlo deve costringere gli altri router a mandare un pacchetto.

Questa applicazione sta al livello applicativo, arriva al livello IP e istruisce il pacchetto di mettere TTL pari a 1, esce dall'host e il prossimo router decrementa TTL che diventa 0 e quindi viene buttato. Viene quindi mandato un messaggio ICMP, questo messaggio mi consente di imparare l'indirizzo IP del primo router in cui sono finito e il tempo che ci ha messo. Quindi ricavo informazioni sugli altri router sfruttando il suo prevedibile comportamento.

Incrementando TTL posso visitare gli altri router e ricavare i loro indirizzi IP.

Finché le tabelle di inoltro non cambiano i router rimangono in quella posizione, quindi posso stare quasi certo che vengono visitati router diversi e la strada non cambia.



Fino a quando va avanti Traceroute? Una volta che arriva all'indirizzo di destinazione (che viene scritto quando mando traceroute), il TTL andrebbe a 0 ma non lo scarta perché è l'host finale, quindi si punta a un numero di porta che quasi sicuramente non risponde così che si manda un messaggio ICMP "port unreachable" . Questo messaggio fa capire che si è arrivati.

Quello che traceroute mostra è una schermata del genere:

traceroute: gaia.cs.umass.edu to www.eurecom.fr

Three delay measurements from gaia.cs.umass.edu to cs-gw.cs.umass.edu

```
1  cs-gw (128.119.240.254)  1 ms  1 ms  2 ms
2  border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145)  1 ms  1 ms  2 ms
3  cht-vbns.gw.umass.edu (128.119.3.130)  6 ms 5 ms 5 ms
4  jn1-at1-0-0-19.wor.vbns.net (204.147.132.129)  16 ms 11 ms 13 ms
5  jn1-so7-0-0-0.wae.vbns.net (204.147.136.136)  21 ms 18 ms 18 ms
6  abilene-vbns.abilene.ucaid.edu (198.32.11.9)  22 ms  18 ms  22 ms
7  nycm-wash.abilene.ucaid.edu (198.32.8.46)  22 ms  22 ms  22 ms
8  62.40.103.253 (62.40.103.253)  104 ms 109 ms 106 ms
9  de2-1.de1.de.geant.net (62.40.96.129)  109 ms 102 ms 104 ms
10 de.fr1.fr.geant.net (62.40.96.50)  113 ms 121 ms 114 ms
11 renater-gw.fr1.fr.geant.net (62.40.103.54)  112 ms  114 ms  112 ms
12 nio-n2.cssi.renater.fr (193.51.206.13)  111 ms  114 ms  116 ms
13 nice.cssi.renater.fr (195.220.98.102)  123 ms  125 ms  124 ms
14 r3t2-nice.cssi.renater.fr (195.220.98.110)  126 ms  126 ms  124 ms
15 eurecom-valbonne.r3t2.ft.net (193.48.50.54)  135 ms  128 ms  133 ms
16 194.214.211.25 (194.214.211.25)  126 ms  128 ms  126 ms
17 * * *
18 * * *
19 fantasia.eurecom.fr (193.55.113.142)  132 ms  128 ms  136 ms
```

trans-oceanic link

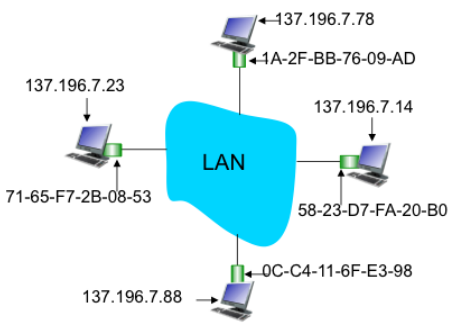
→ si vede dall'aumento improvviso del tempo

* means no response (probe lost, router not replying)

ARP

Le trame MAC hanno un indirizzo sorgente e indirizzo destinazione. Se c'è un host connesso in una sottorete (in area locale, come una LAN Ethernet) e vuole mandare un pacchetto ad un host che si trova nella stessa sottorete (quindi no router). A livello di rete l'host sa l'indirizzo di destinazione, lo manda allo strato inferiore al livello MAC a cui però deve dire a chi è destinato il pacchetto, ma per il MAC l'indirizzo IP non ha senso, può consegnare sono con un indirizzo MAC di destinazione. Quindi quando IP vuole avvalersi di questo servizio, come fa a procurarsi l'indirizzo MAC di destinazione associato a quello IP? Lo chiede usando il protocollo ARP (protocollo di traduzione degli indirizzi). Ha due soli messaggi: arp request e arp replay. Poi contiene 4 campi:l'indirizzo IP del richiedente, l'indirizzo MAC del richiedente, l'indirizzo IP di cui si vuole sapere il MAC, e tutti 0 (ossia quello che viene poi sostiuito con l'indirizzo MAC che si vuole sapere). Il messaggio ARP viene messo nella trama Ethernet e viene mandato a tutte le entità MAC della sottorete. Chi risponde manda l'ARP request solamente al Mac source address da cui proveniva la richiesta.

Question: how to determine interface's MAC address, knowing its IP address?



ARP table: each IP node (host, router) on LAN has table

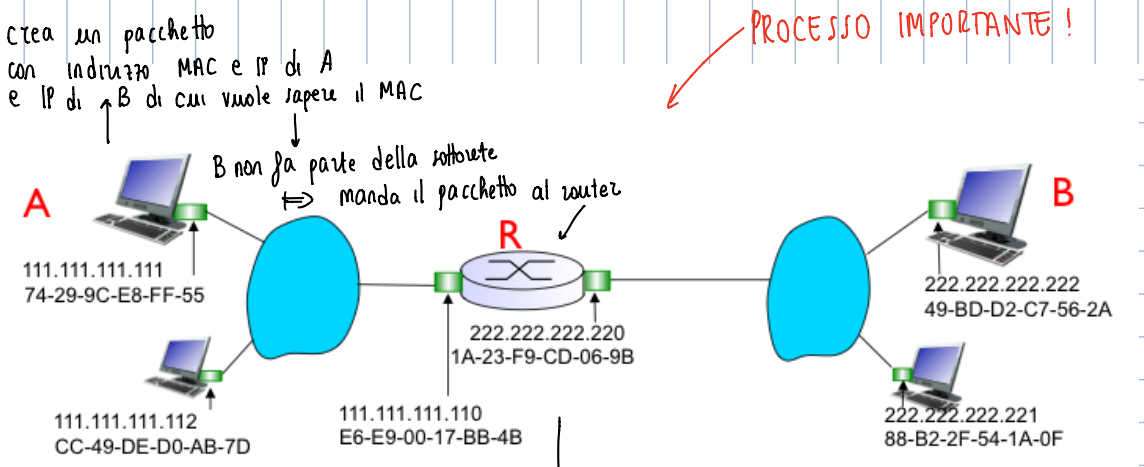
- IP/MAC address mappings for some LAN nodes:
  < IP address; MAC address; TTL>
- TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

Se l'indirizzo IP richiesto non è presente nella sottorete o è stato spento, chi richiede non aspetta all'infinito.

Una volta che ho mandato la richiesta e ottenuto la risposta, scrivo una tabella che si chiama ARP cache così che la prossima volta che mi serve nuovamente la corrispondenza non richiedo nuovamente ma consulto la tabella. Quello che viene scritto nella arp cash viene tenuto per un tempo limitato perché la situazione degli indirizzi IP potrebbe cambiare. Quindi c'è il compromesso tra efficienza (più tempo tengo l'informazione nella tabella menò richieste mando) e il fatto che le informazioni potrebbero diventare obsolete.

I router che non devono mandare l'informazione richiesta, comunque ricevono la richiesta (visto che è destinata a tutti) e apprendono l'indirizzo IP dell'host richiedente e lo mette nella propria cache.

ARP spoufing: un host può dire falsamente che ha l'indirizzo IP richiesto e quindi gli dice di mandare il traffico. Tuttavia per fare questo l'host deve stare all'interno della sottorete. È diciamo una violazione della sicurezza.



A la prima cosa che deve capire è se l'indirizzo di destinazione appartiene alla sua sottorete

↓

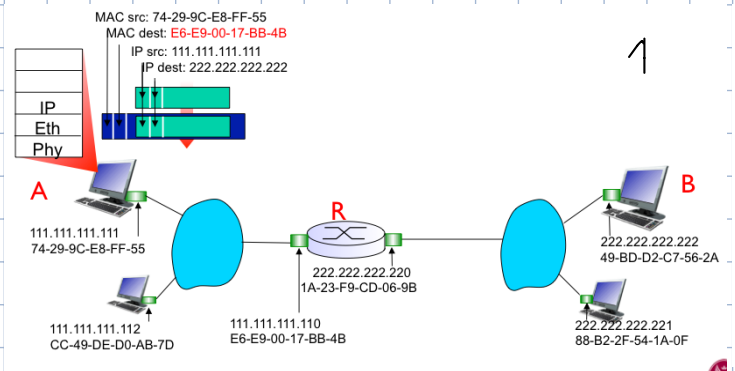
prende i primi 23 bit della sua submask e li confronta con l'indirizzo IP di destinazione

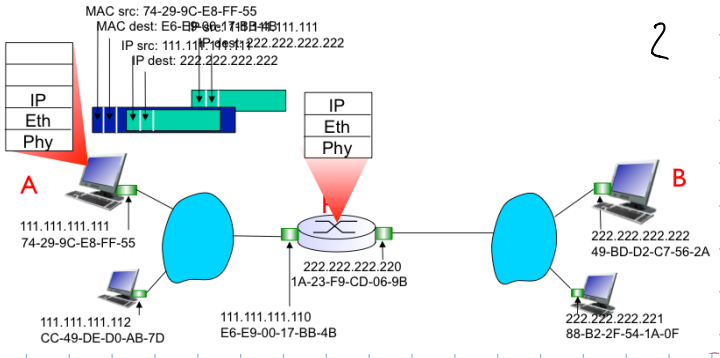
In questo caso non appartiene alla sottorete quindi A deve mandare il pacchetto al router

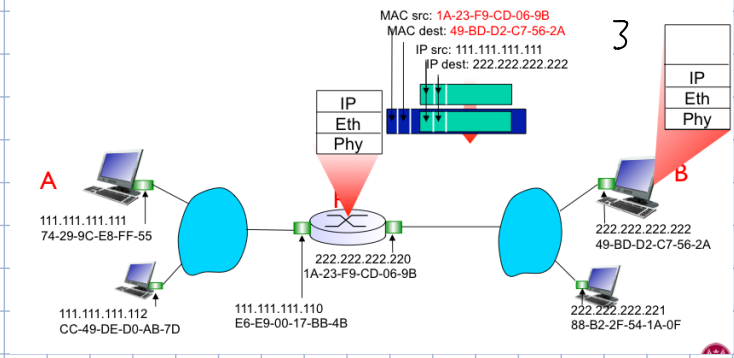
↓

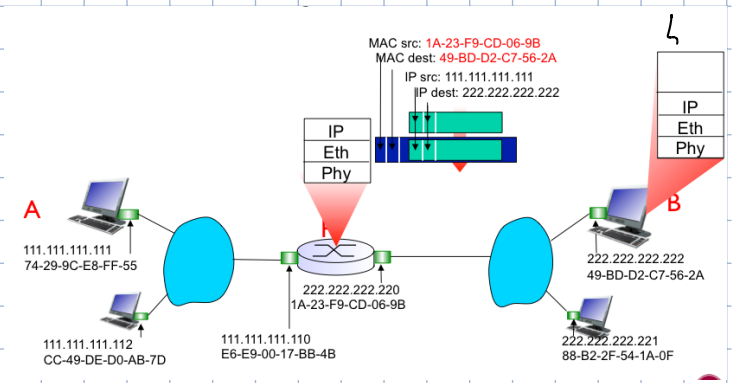
l'indirizzo di questo gli viene configurato con DHCP che oltre al suo stesso IP riceve anche l'indirizzo IP del router della sottorete

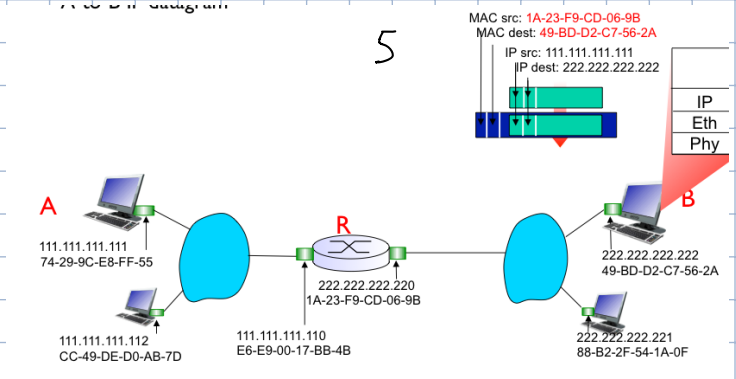
se non conosce il MAC del router lo chiede con ARP.











Instradamento

Funzione decisionale che permette di capire qual è il cammino ottimo tra sorgente e destinazione attraverso i routers. Il problema principale è che ci sono tanti percorsi.
Ci sono due approcci: classico e SDN. Nel primo non esiste un controllo centralizzato, ogni router ha un software di controllo che trova il cammino ottimo, questo software si chiama protocollo di instradamento (i router comunicano mediante messaggi).
Nell'SDN c'è un controllo centralizzato collegato a tutti i router, il protocollo è OPENFLOW e permette il dialogo tra controllo centralizzato e routers. Il controllore invia comandi o domande e i routers risposte.

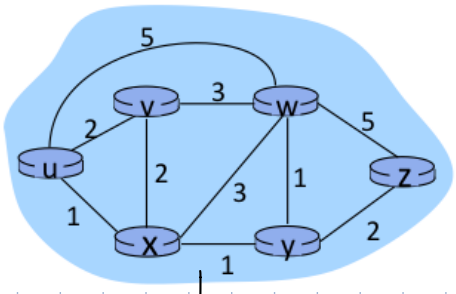
Le entità di Internet sono gli Host e i Router.

- l'host : sono sorgenti e le destinazioni delle informazioni di utente, sono normalmente connessi ad una sola sottorete. Se il pacchetto non è per lui, viene buttato.
- I router instradano i pacchetti IP tra le sottoreti, hanno interfacce verso almeno due diverse sottoreti. Se il pacchetto non è per lui ha la capacità di inoltrarlo.

Due entità IP sono direttamente connesse se appartengono alla stessa sottorete non servono router (da non confondere che il trasferimento IP è senza connessione!), sono indirettamente connesse non stanno sulla stessa sottorete e quindi inevitabilmente per

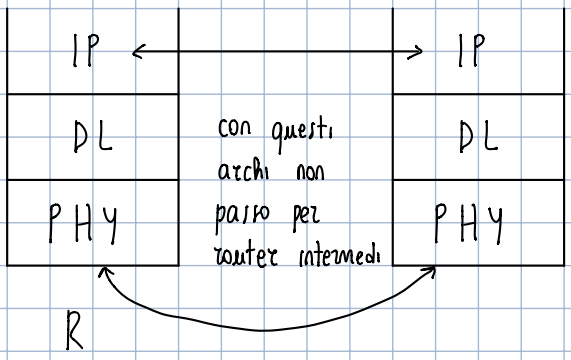
comunicare devono utilizzare uno o più router.

Per formulare il problema di ottimizzazione in cui consiste l'instradamento bisogna disegnare un modello (un grafo) della rete vista a livello IP.



non metto degli switch ethernet, o altre cose che non appartengono al livello IP, che appartengono al livello data link.

Mettere un ramo tra u e v vuol dire che è possibile mandare pacchetti senza un router.



Matrice di incidenza: ci dice i vicini dei router.
Ogni ramo rappresenta un costo per collegare i due router.
Un grafo si dice direttivo se conta il verso dei rami, bastano i grafi non direttivi perché in tlc I collegamenti sono sempre bidirezionali.

Un cammino su un grafo è una sequenza ordinata di rami e nodi che collegano sorgente e destinazione, il costo di un cammino è la somma dei costi dei rami che compongono il cammino.
Il cammino ottimo è quello che costa meno.
Se i costi sono tutti equivalenti, scelgo il cammino più corto.

Esistono due approcci per trovare i cammini ottimi: locale e globale. Materialmente si identificano in due algoritmi:

- link state algorithm (globale): il router scambia messaggi con tutti. Se i router sono n vengono scambiati n(n-1) messaggi.
- Distance vector algorithm (locale) : il router scambia messaggi solo con i suoi vicini, se si applica la regola a ogni vicino alla fine l'informazione viene condivisa su tutta la rete. (L'approccio è il passaparola).

L'aggettivo sta ad indicare con quale router l'altro router scambia messaggi.
Ambedue gli approcci sono finalizzati allo scambio di messaggi tra i router che si dicono quali sono i loro vicini e quali sono i costi. Se ognuno dà questa informazione riesco a ricostruire il grafo.

Dijkstra's algorithm

- net topology and link costs known to all nodes
  - accomplished via "link state broadcast"
  - all nodes have same info
- computes least cost paths from one node ('source") to all other nodes
  - gives forwarding table for that node

Notation:

- c(x,y): link cost from node x to y; = ∞ if not direct neighbors
- D(v): current value of cost of path from source to dest. v
- p(v): predecessor node along path from source to v
- N': set of nodes whose least cost path definitively known

```
1  Initialization:
2   N' = {a}
3   for all nodes x
4     if x adjacent to a
5        then D(x) = c(a,x) and p(x)=a
6        else D(x) = ∞
8  Loop
9    find y not in N' such that D(y) is a minimum
10   add y to N'
11   update D(z) for all z adjacent to y and not in N' :
12       D(z) = min{ D(z), D(y) + c(y,z) }
•      if D(y)+c(y,z) < D(z) then p(z)=y
•   /* new cost to z is either old cost to z or known
14      shortest path cost to y plus cost from y to z */
15  until all nodes in N'
```