

Reti di Elaboratori

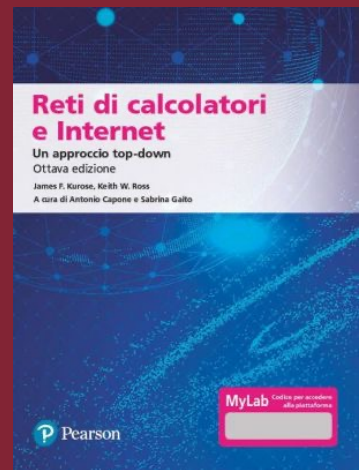
Livello di Rete: Forwarding generalizzato e middleboxes



SAPIENZA
UNIVERSITÀ DI ROMA

Alessandro Checco

alessandro.checco@uniroma1.it



Capitolo 4

Livello di rete (data plane): sommario

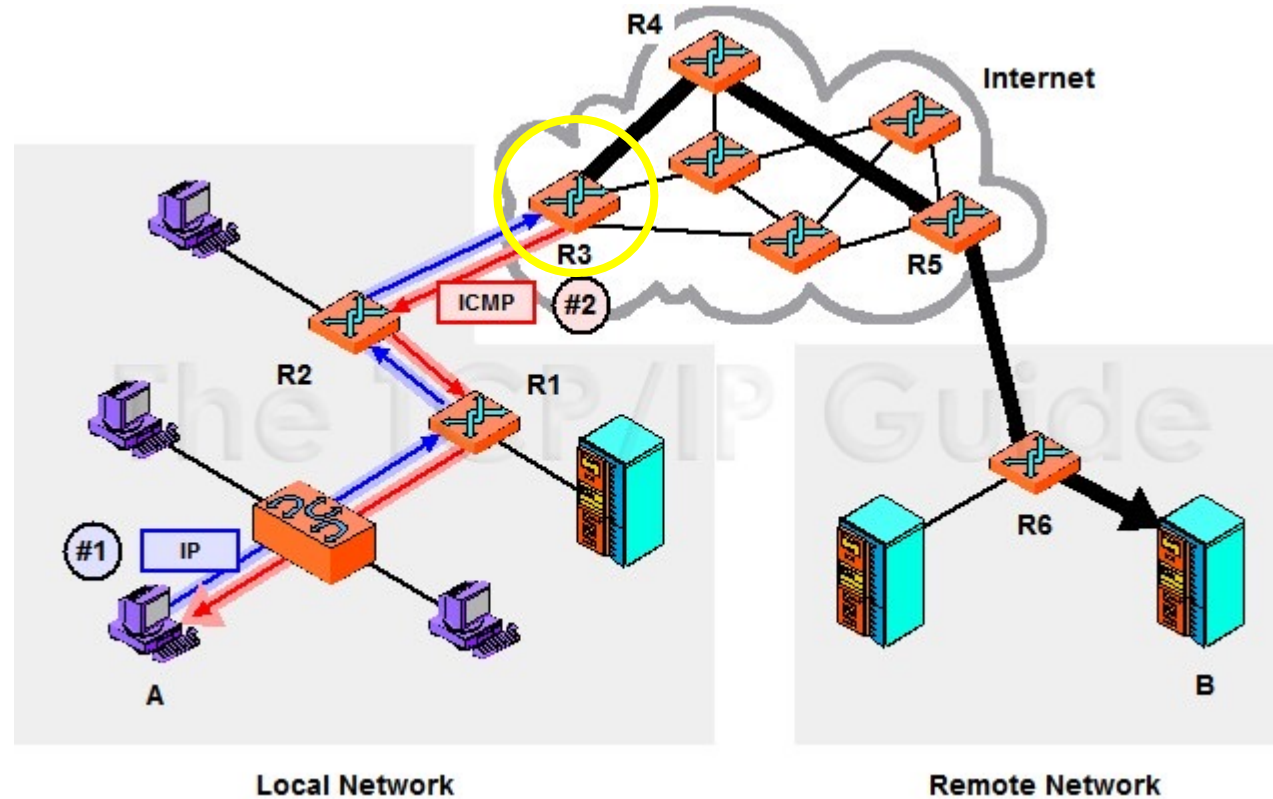
- Livello di rete: panoramica
 - piano dati
 - piano di controllo
- Dentro i router
 - porte di ingresso, commutazione, porte di uscita
 - gestione del buffer, scheduling
- **IP: il protocollo Internet**
 - formato datagramma
 - indirizzamento
 - traduzione di indirizzi di rete
 - IPv6
- Forwarding generalizzato, SDN
 - Match+action
 - OpenFlow: match+action in azione
- Middleboxes

Gestione errori?

- Cosa accade se un router deve scartare un datagramma perché non riesce a trovare un percorso per la destinazione finale?
- Cosa accade se un datagramma ha il campo TTL pari a 0?
- E se un host di destinazione non ha ricevuto tutti i frammenti di un datagramma entro un determinato limite di tempo?
- Situazioni di errore che IP non gestisce!

Internet Control Message Protocol (ICMP)

- Viene usato da host e router per scambiarsi informazioni a livello di rete.



A typical use of ICMP is to provide a feedback mechanism when an IP message is sent. In this example, device *A* is trying to send an IP datagram to device *B*. However, when it gets to router *R3* a problem of some sort is detected that causes the datagram to be dropped. *R3* sends an ICMP message back to *A* to tell it that something happened, hopefully with enough information to let *A* correct the problem, if possible. *R3* can only send the ICMP message back to *A*, not to *R2* or *R1*.

Internet Control Message Protocol (ICMP)

- Viene usato da host e router per scambiarsi informazioni a livello di rete.
 - report degli errori: host, rete, porta, protocollo irraggiungibili.
 - **echo request/reply** (usando il programma ping).
- Livello di rete “sopra” IP:
 - ICMP è considerato parte di IP anche se **usa IP per inviare i suoi messaggi**
- **Messaggi ICMP**: hanno un campo **tipo** e un campo **codice**, e contengono l’intestazione e i primi 8 byte del datagramma IP che ha provocato la generazione del messaggio.

<u>Tipo</u>	<u>Codice</u>	<u>Descrizione</u>
0	0	Risposta eco (a ping)
3	0	rete destin. irraggiungibile
3	1	host destin. irraggiungibile
3	2	protocollo dest. irraggiungibile
3	3	porta destin. irraggiungibile
3	6	rete destin. sconosciuta
3	7	host destin. sconosciuto
4	0	riduzione (controllo di congestione)
8	0	richiesta eco
9	0	annuncio del router
10	0	scoperta del router
11	0	TTL scaduto
12	0	errata intestazione IP

Ping

- Il programma *ping* si basa sui messaggi di richiesta e risposta *echo* di ICMP

```
$ ping pads.cs.unibo.it
PING chernobog.pads.cs.unibo.it (130.136.132.11) 56(84) bytes of data.
64 bytes from chernobog.pads.cs.unibo.it (130.136.132.11): icmp_req=1 ttl=52 time=34.2 ms
64 bytes from chernobog.pads.cs.unibo.it (130.136.132.11): icmp_req=2 ttl=52 time=33.1 ms
64 bytes from chernobog.pads.cs.unibo.it (130.136.132.11): icmp_req=3 ttl=52 time=34.0 ms
64 bytes from chernobog.pads.cs.unibo.it (130.136.132.11): icmp_req=4 ttl=52 time=33.9 ms
64 bytes from chernobog.pads.cs.unibo.it (130.136.132.11): icmp_req=5 ttl=52 time=33.3 ms
--- chernobog.pads.cs.unibo.it ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 33.177/33.758/34.220/0.417 ms
```

Traceroute e ICMP

- Il programma invia una serie di datagrammi IP alla destinazione ciascuno contenente un segmento UDP con un numero di porta inutilizzata.
 - Il primo pari a TTL =1
 - Il secondo pari a TTL=2, ecc.
 - Numero di porta improbabile
 - L'origine avvia un timer per ogni datagramma
- Quando l' n -esimo datagramma arriva all' n -esimo router:
 - Il router scarta il datagramma.
 - Invia all'origine un messaggio di allerta ICMP (tipo 11, codice 0).
 - Il messaggio include il nome del router e l'indirizzo IP.

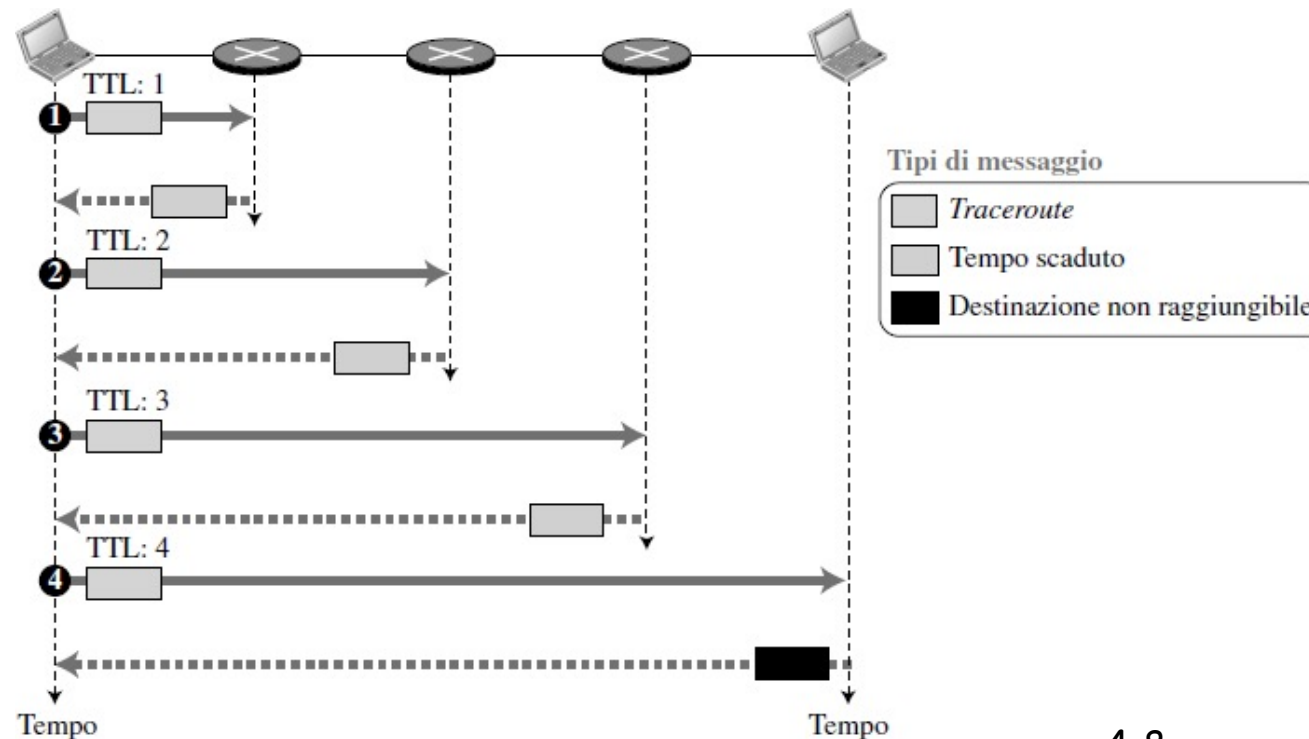
- Quando il messaggio ICMP arriva, l'origine può calcolare RTT
- Traceroute lo fa per 3 volte

Criteri di arresto dell'invio

- Quando un segmento UDP arriva all'host di destinazione.
- L'host di destinazione restituisce un messaggio ICMP di porta non raggiungibile (tipo 3, codice 3).
- Quando l'origine riceve questo messaggio ICMP, si blocca.

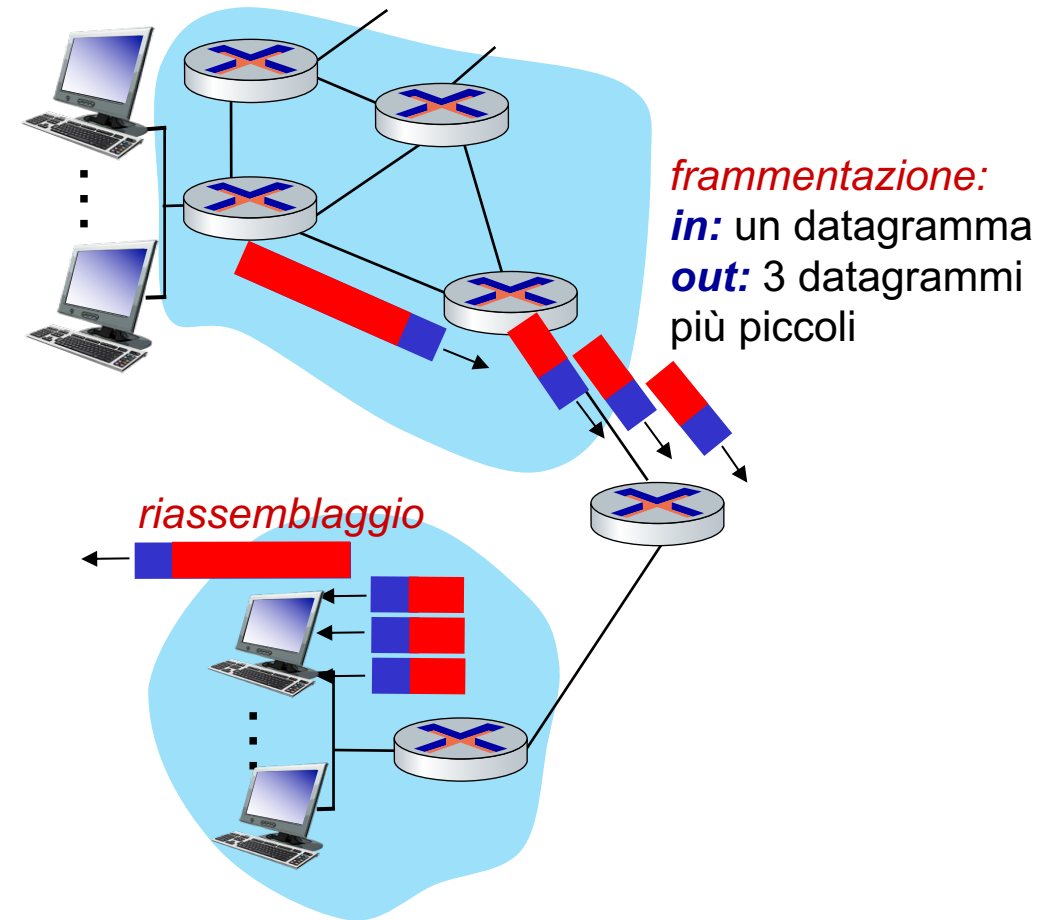
Traceroute

- Non c'è un programma server (ma solo un client)
- Le risposte arrivano da ICMP (tempo scaduto dai router intermedi e porta non raggiungibile dall'host di destinazione)



Frammentazione/riassemblaggio dell'IP

- i collegamenti di rete hanno MTU (max. transfer size): massima dimensione del frame a livello di collegamento
 - MTU varia per tipo di collegamento
 - grande datagramma IP diviso ("frammentato") lungo la rete
 - un datagramma si divide in diversi datagrammi
 - "riassemblato" solo a *destinazione*
 - Bit di intestazione IP utilizzati per identificare e ordinare i frammenti correlati



Frammentazione/riassemblaggio dell'IP

esempio:

- Datagramma da 4000 byte
- MTU = 1500 byte

1480 bytes in
data field

offset =
1480/8

	length =4000	ID =x	fragflag =0	offset =0	
--	-----------------	----------	----------------	--------------	--

*one large datagram becomes
several smaller datagrams*

	length =1500	ID =x	fragflag =1	offset =0	
--	-----------------	----------	----------------	--------------	--

	length =1500	ID =x	fragflag =1	offset =185	
--	-----------------	----------	----------------	----------------	--


	length =1040	ID =x	fragflag =0	offset =370	
--	-----------------	----------	----------------	----------------	--

IPv6 deve usare Path MTU Discovery (PMTUD) per scoprire il MTU minore in un percorso, o il pacchetto verrà scartato

Esercizio

- Quali protocolli/applicazioni sono generatori di pacchetti?

Esercizio

- Quali protocolli/applicazioni sono generatori di pacchetti?
 - Livello applicazione **SI** (tutti)
 - TCP **SI** (solo per handshake)
 - UDP **NO**
 - ICMP **SI**
 - NAT **NO**
 - IP **NO/SI** (NO tranne per la frammentazione)
 - DHCP **SI**
 - Protocolli di routing **lo vedremo**
- Quelli **NO** si limitano ad incapsulare

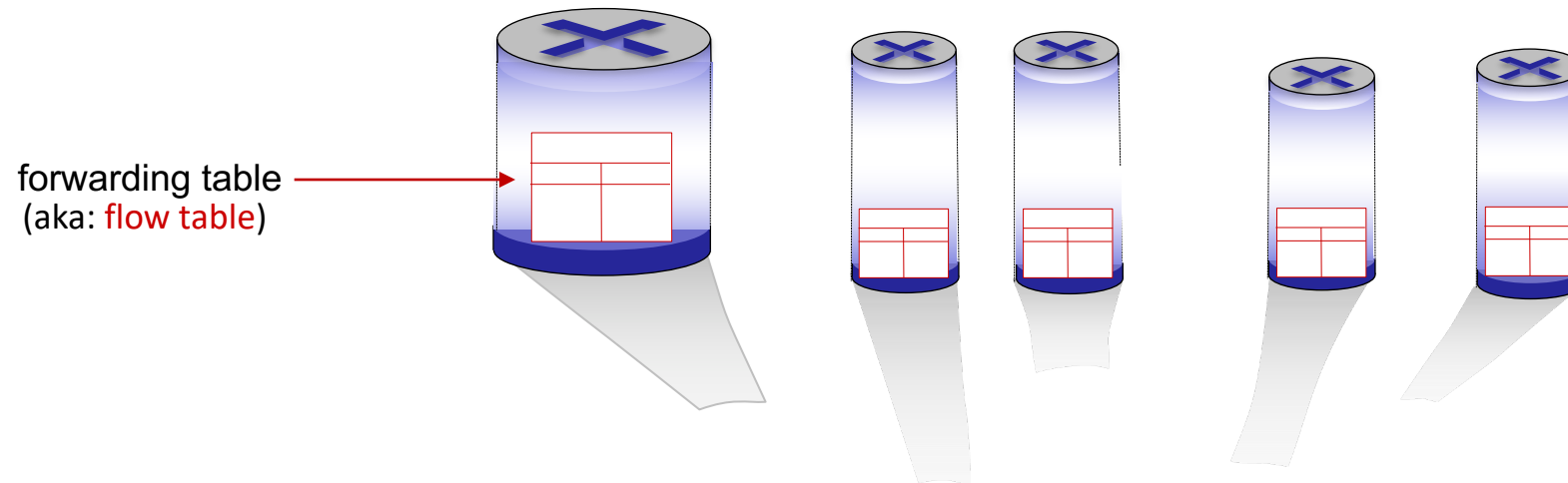
Livello di rete (data plane): sommario

- Livello di rete: panoramica
 - piano dati
 - piano di controllo
- Dentro i router
 - porte di ingresso, commutazione, porte di uscita
 - gestione del buffer, scheduling
- IP: il protocollo Internet
 - formato datagramma
 - indirizzamento
 - traduzione di indirizzi di rete
 - IPv6
- Forwarding generalizzato, SDN
 - Match+action
 - OpenFlow: match+action in azione
- Middleboxes

Forwarding generalizzato: match plus action

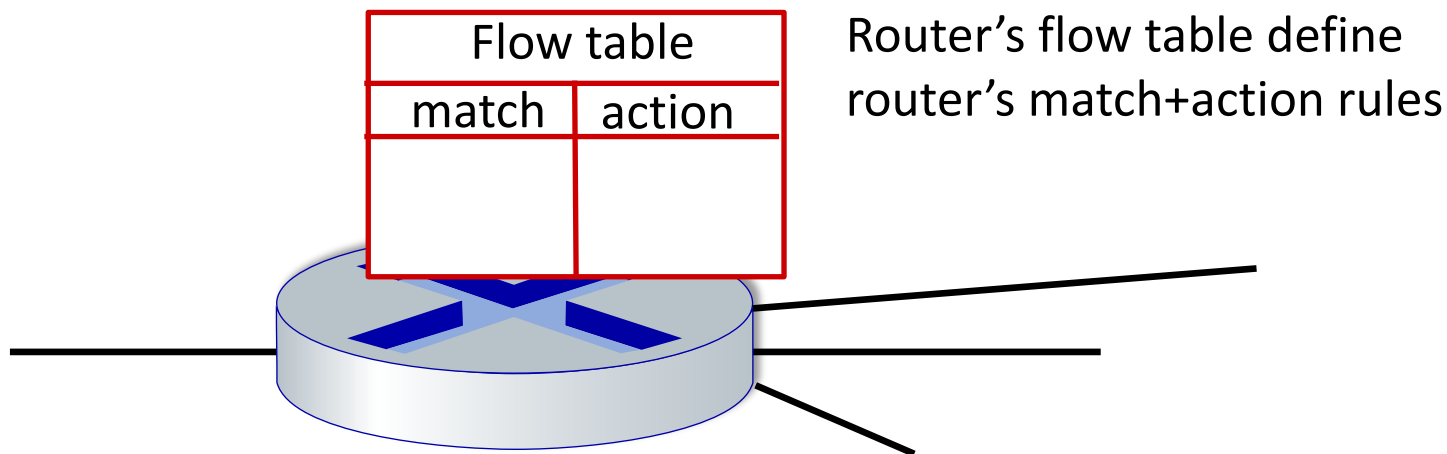
Ogni router ha una **forwarding table** (aka: **flow table**)

- astrazione “**match plus action**”: quando c’è un match (su alcuni bit) esegui un’azione
 - *destination-based forwarding*: forwarding basato solo sull’indirizzo IP di destinaz.
 - *generalized forwarding*:
 - molteplici intestazioni possono causare un match
 - più azioni possibili: drop/copy/modify/log packet



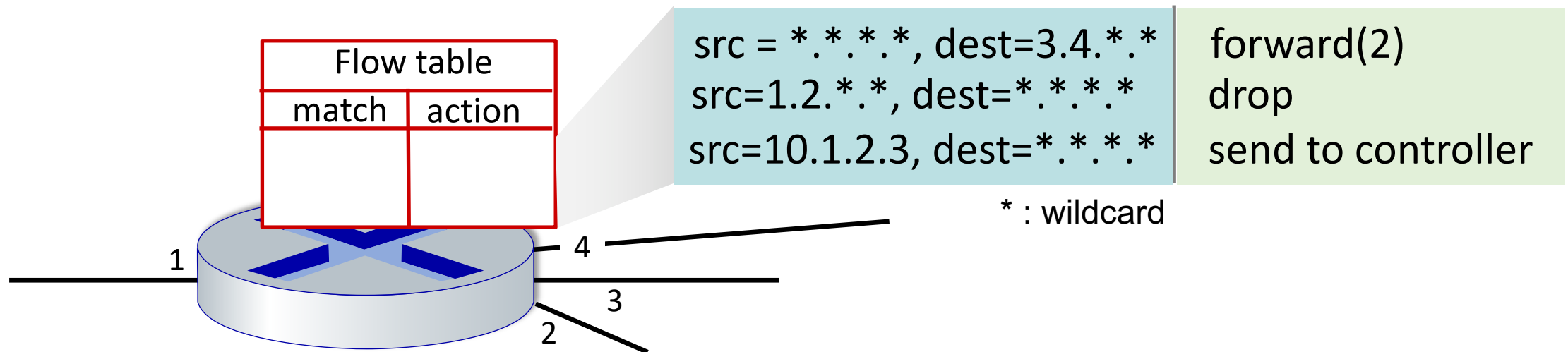
Astrazione Flow table

- **flow**: definito da campi dell'header (link, network, transport)
- **forwarding generalizzato**: regole su come gestire il pacchetto
 - **match**: identificazione di un pattern nell'header del pacchetto
 - **actions**: per i pacchetti che sono un match: drop, forward, modify, send to controller
 - **priority**: regole per disambiguare pattern sovrapposti (match multipli)
 - **counters**: #bytes and #packets

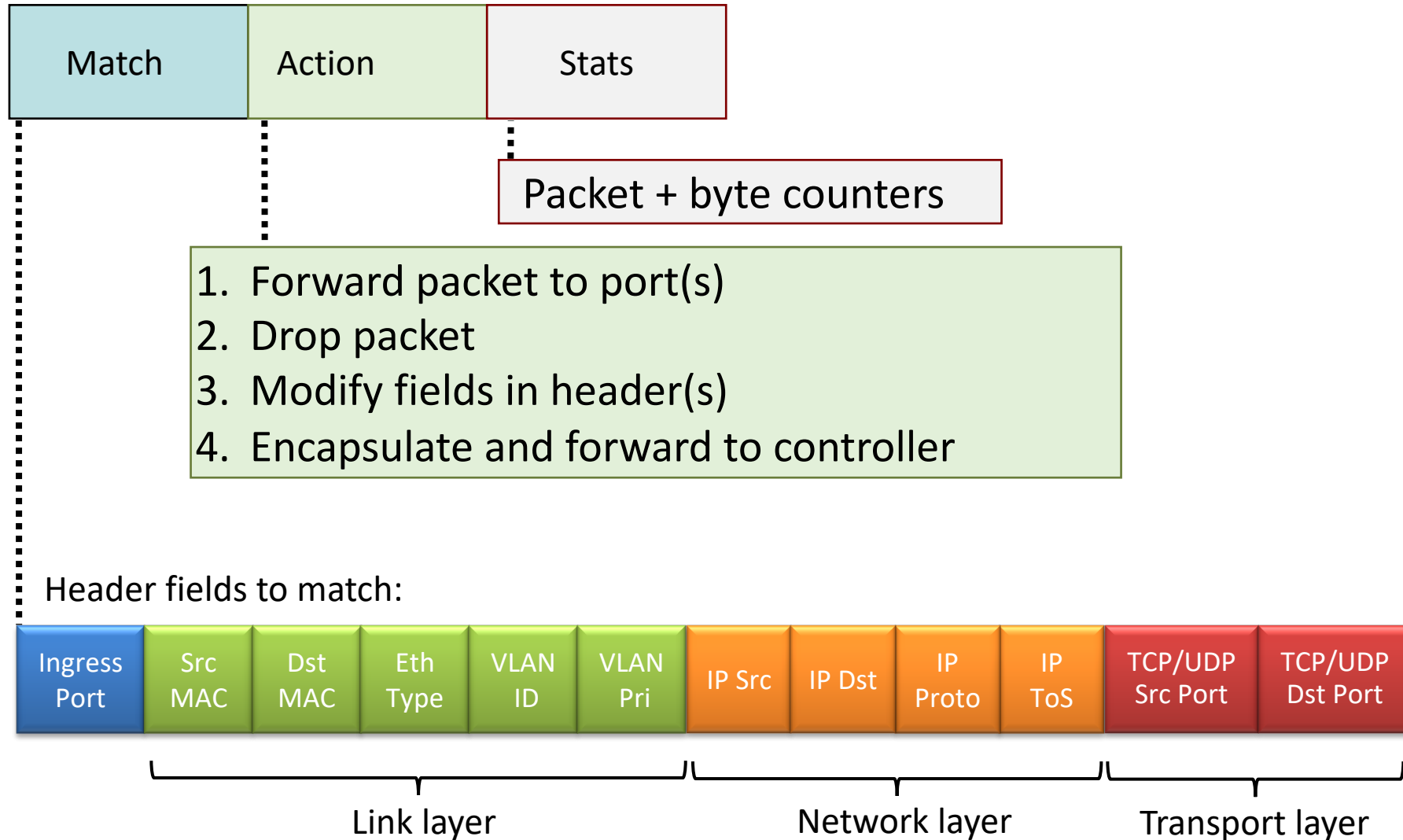


Astrazione Flow table

- **flow**: definito da campi dell'header (link, network, transport)
- **forwarding generalizzato**: regole su come gestire il pacchetto
 - **match**: identificazione di un pattern nell'header del pacchetto
 - **actions**: per i pacchetti che sono un match: drop, forward, modify, send to controller
 - **priority**: regole per disambiguare pattern sovrapposti (match multipli)
 - **counters**: #bytes and #packets



OpenFlow: record nella flow table



OpenFlow: esempi

Destination-based forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	*	51.6.0.8	*	*	*	*	port6

IP datagrams destined to IP address 51.6.0.8 should be forwarded to router output port 6

Firewall:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	*	*	*	*	*	22	drop

Block (do not forward) all datagrams destined to TCP port 22 (ssh port #) 

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	128.119.1.1	*	*	*	*	*	drop

Block (do not forward) all datagrams sent by host 128.119.1.1

OpenFlow: esempi



Layer 2 destination-based forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	22:A7:23: 11:E1:02	*	*	*	*	*	*	*	*	*	port3

layer 2 frames with destination MAC address 22:A7:23:11:E1:02 should be forwarded to output port 3

OpenFlow

- **match+action**: questa astrazione unisce diversi tipi di device

Router

- *match*: longest destination IP prefix
- *action*: forward out a link

Switch

- *match*: destination MAC address
- *action*: forward or flood

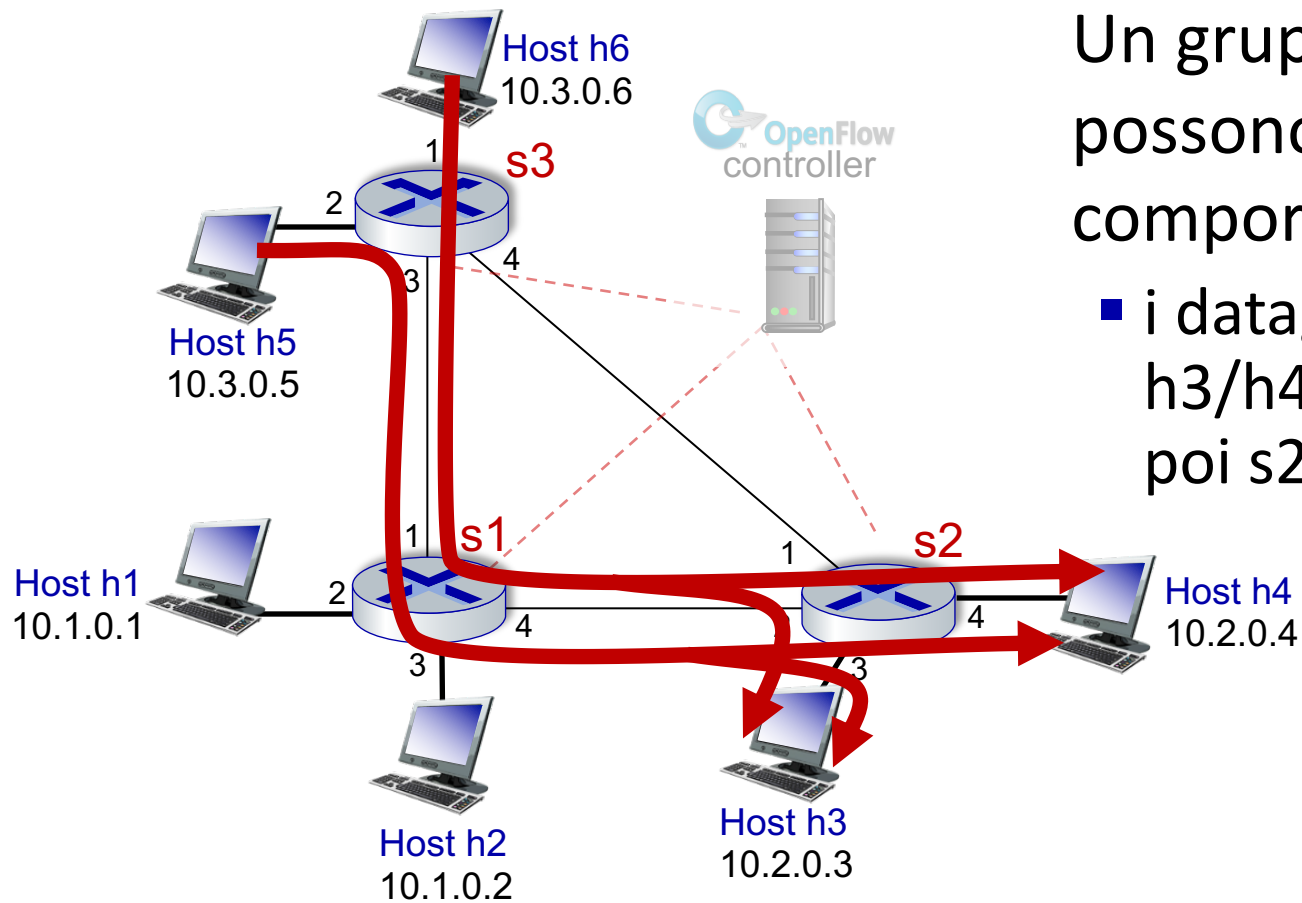
Firewall

- *match*: IP addresses and TCP/UDP port numbers
- *action*: permit or deny

NAT

- *match*: IP address and port
- *action*: rewrite address and port

Esempio OpenFlow

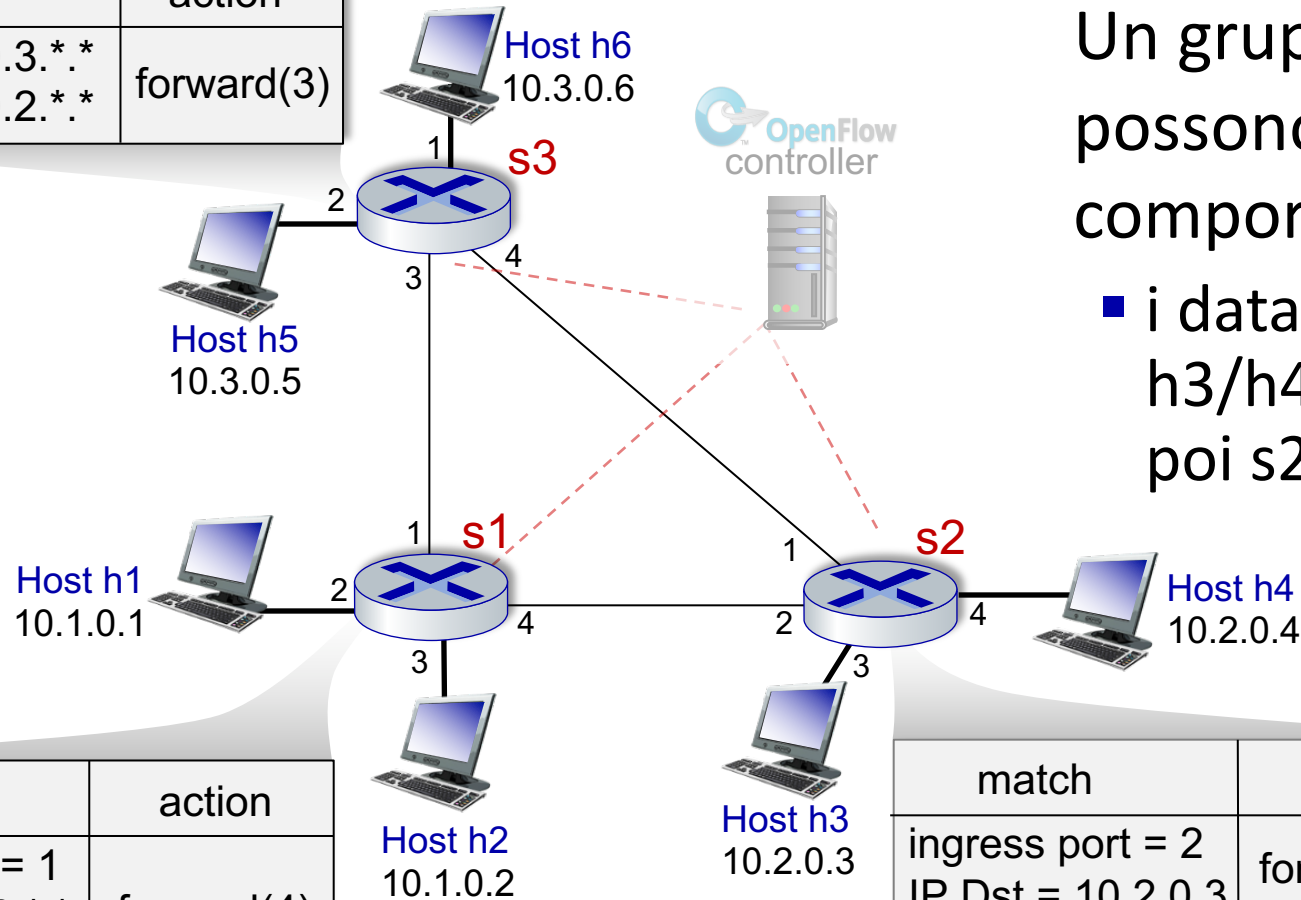


Un gruppo di flow table
possono garantire
comportamenti di rete *globali*:

- i datagrammi da h5 e h6 per h3/h4 **devono** passare da s1 e poi s2

Esempio OpenFlow

match	action
IP Src = 10.3.*.* IP Dst = 10.2.*.*	forward(3)



match	action
ingress port = 1 IP Src = 10.3.*.* IP Dst = 10.2.*.*	forward(4)

match	action
ingress port = 2 IP Dst = 10.2.0.3	forward(3)
ingress port = 2 IP Dst = 10.2.0.4	forward(4)

Un gruppo di flow table possono garantire comportamenti di rete *globali*:

- i datagrammi da h5 e h6 per h3/h4 devono passare da s1 e poi s2

Forwarding generalizzato: riepilogo

- astrazione “**match plus action**”: fare il match di bit degli header dei pacchetti in arrivo su ogni layer, ed esegui un'azione
 - match su molteplici campi (link, network, transport-layer)
 - azioni locali: drop, forward, modify, send to controller
 - permette di “programmare” comportamenti a livello di rete
- forme semplici di “network programmability”
 - processamento programmabile per ogni pacchetto
 - *origine*: active networking
 - *oggi*: programmazione generalizzata delle reti: P4 (see p4.org).

Livello di rete (data plane): sommario

- Livello di rete: panoramica
 - piano dati
 - piano di controllo
- Dentro i router
 - porte di ingresso, commutazione, porte di uscita
 - gestione del buffer, scheduling
- IP: il protocollo Internet
 - formato datagramma
 - indirizzamento
 - traduzione di indirizzi di rete
 - IPv6
- Forwarding generalizzato, SDN
 - Match+action
 - OpenFlow: match+action in azione
- Middleboxes
 - funzioni
 - evoluzione dell'architettura internet

Middleboxes



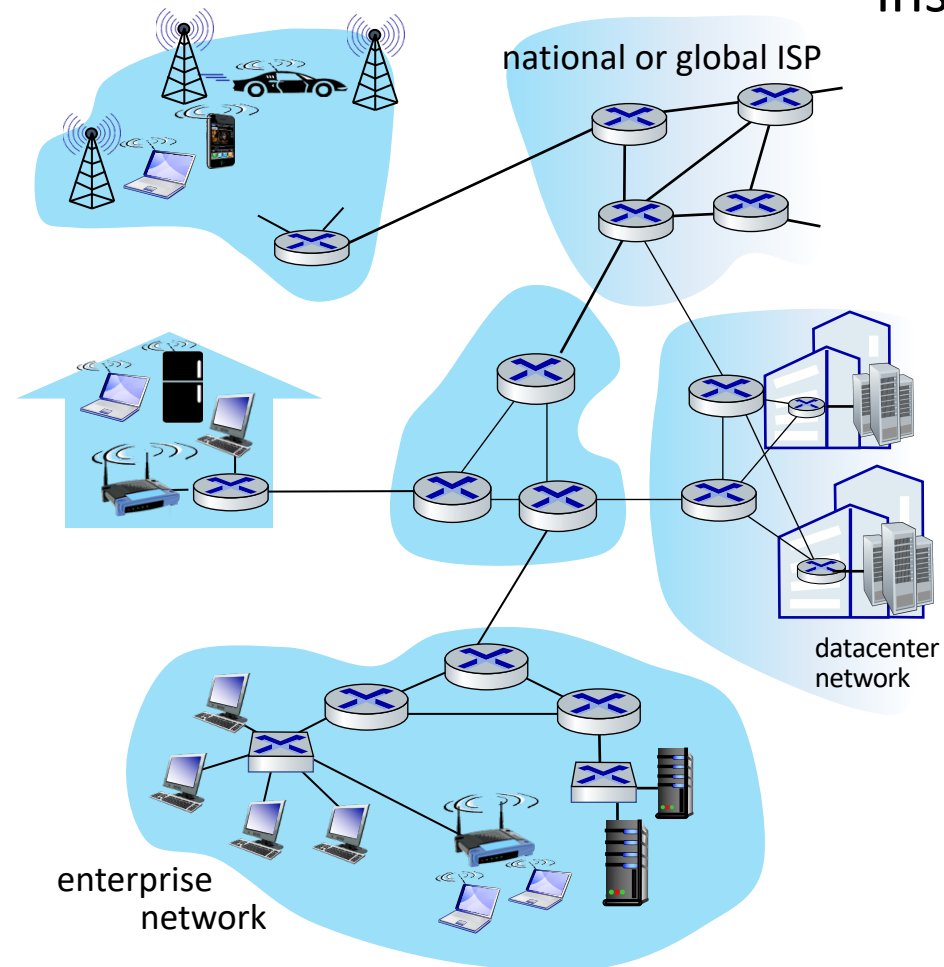
Middlebox (RFC 3234)

"qualsiasi box intermediario che esegue funzioni diverse dalle normali funzioni standard di un router IP sul percorso dati tra un host di origine e un host di destinazione"

Middleboxes everywhere!

NAT: home,
cellular,
institutional

Application-specific: service
providers,
institutional,
CDN



Firewalls, IDS: corporate,
institutional, service providers,
ISPs

Load balancers:
corporate, service
provider, data center,
mobile nets

Caches: service
provider, mobile, CDNs

Middleboxes

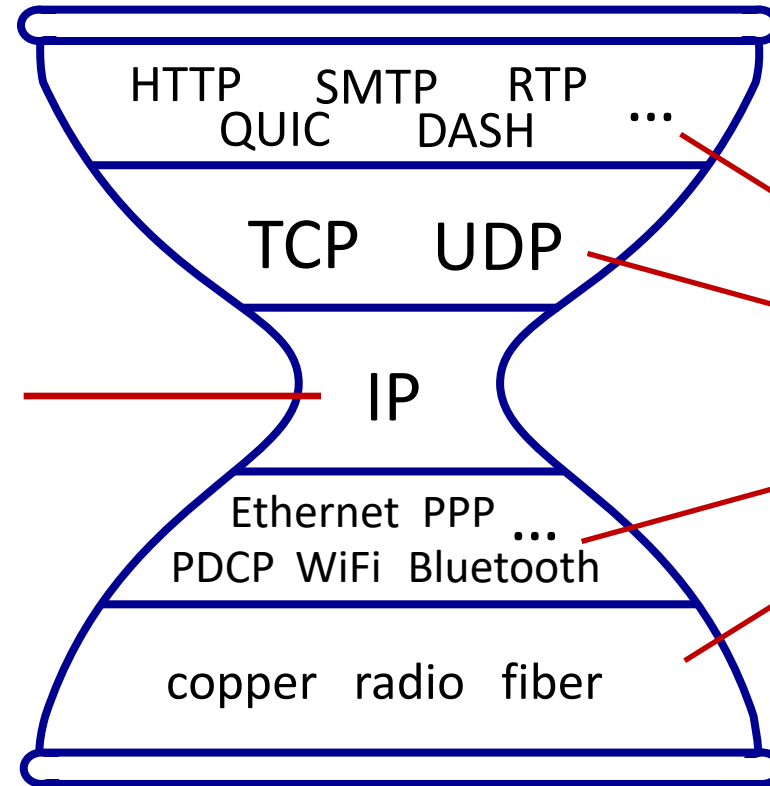


- inizialmente: soluzioni hardware proprietarie (chiuse).
- trend verso hardware "whitebox" che implementa API aperte
 - allontanarsi dalle soluzioni hardware proprietarie
 - azioni locali programmabili tramite match+action
 - muoversi verso innovazione e differenziazione lato software
- SDN: controllo "centralizzato" e gestione della configurazione spesso in cloud privato o pubblico
- virtualizzazione delle funzioni di rete (NFV): servizi programmabili su white box che include funzioni di networking, calcolo e storage

La clessidra IP

Girovita sottile:

- *one* network layer protocol: IP
- *must* be implemented by every (billions) of Internet-connected devices

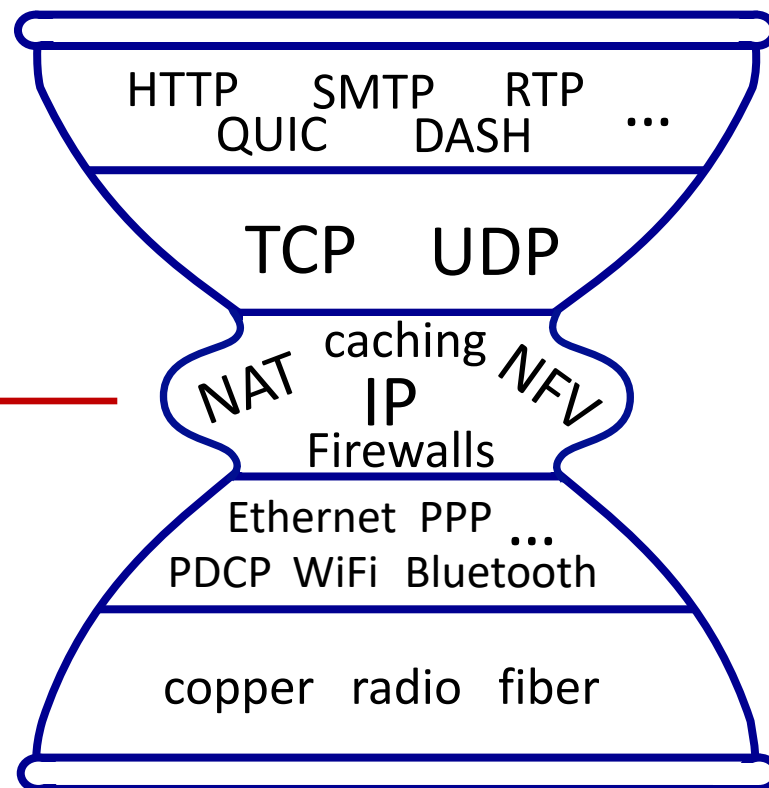


many protocols
in physical, link,
transport, and
application
layers

La clessidra IP dopo 40 anni

Maniglie dell'amore?

- middleboxes, operating inside the network



Principi architetturali di Internet

RFC 1958

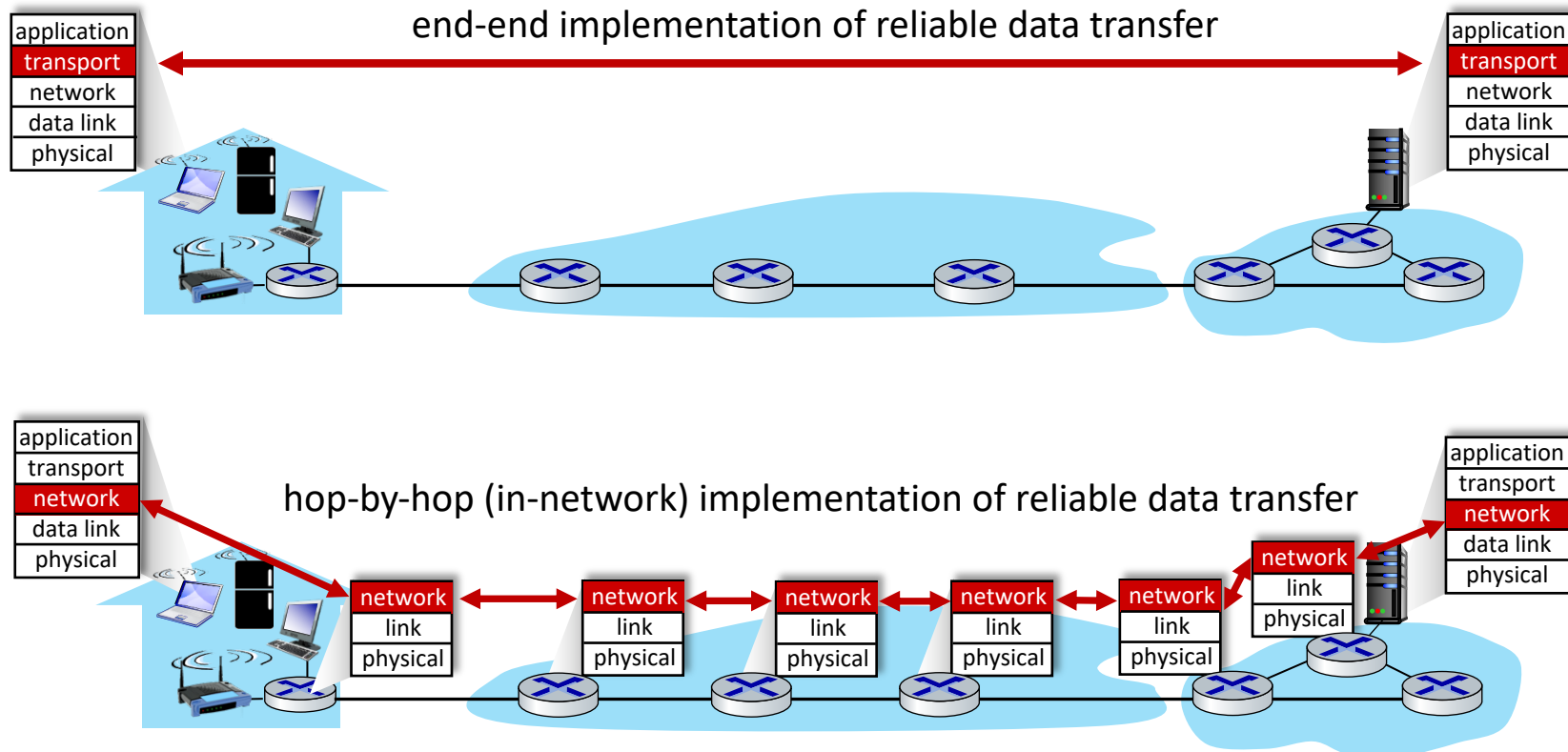
“Many members of the Internet community would argue that there is no architecture, but only a tradition, which was not written down for the first 25 years (or at least not by the IAB). However, in very general terms, the community believes that **the goal is connectivity, the tool is the Internet Protocol, and the intelligence is end to end rather than hidden in the network.**”

Tre principi fondamentali:

- Connettività semplice
- Protocollo IP: girovita sottile
- Complessità e intelligenze sulla periferia della rete

Il principio end-to-end

- Molte funzionalità (ad es. reliable data transfer, congestion control) possono essere implementate **lungo** la rete, o **sul bordo** della rete



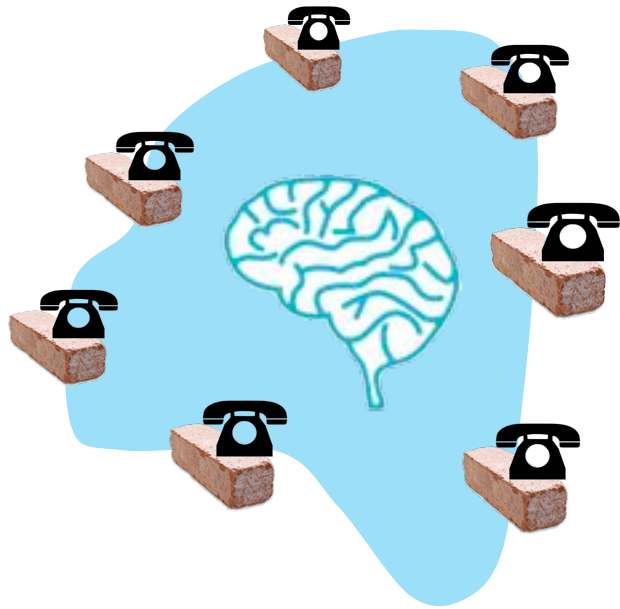
Il principio end-to-end

- Molte funzionalità (ad es. reliable data transfer, congestion control) possono essere implementate **lungo** la rete, o **sul bordo** della rete

“The function in question can **completely and correctly be implemented only with the knowledge and help of the application standing at the end points of the communication system**. Therefore, providing that questioned function as a feature of the communication system itself is not possible. (Sometimes an incomplete version of the function provided by the communication system may be useful as a performance enhancement.)

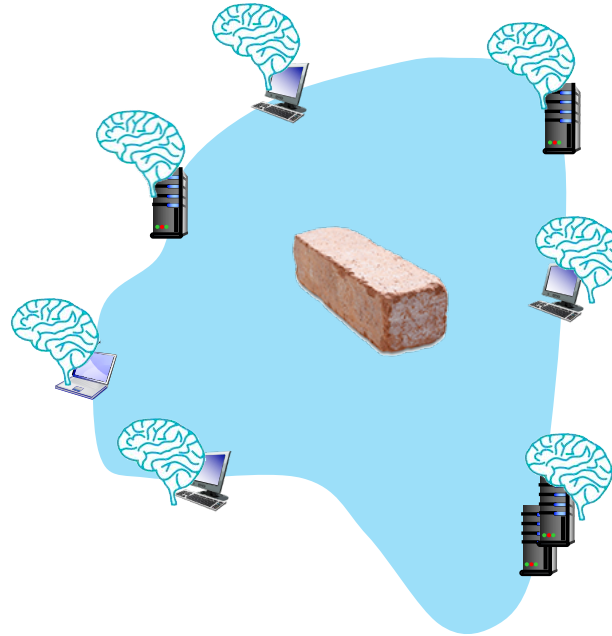
We call this line of reasoning against low-level function implementation the “end-to-end argument.”

Dov'è l'intelligenza?



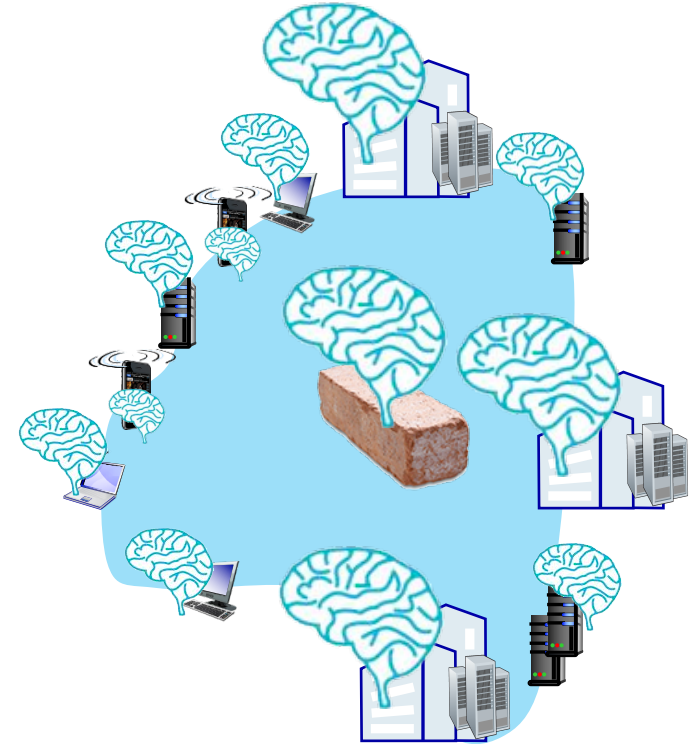
20th century phone net:

- intelligence/computing at network switches



Internet (pre-2005)

- intelligence, computing at edge



Internet (post-2005)

- programmable network devices
- intelligence, computing, massive application-level infrastructure at edge