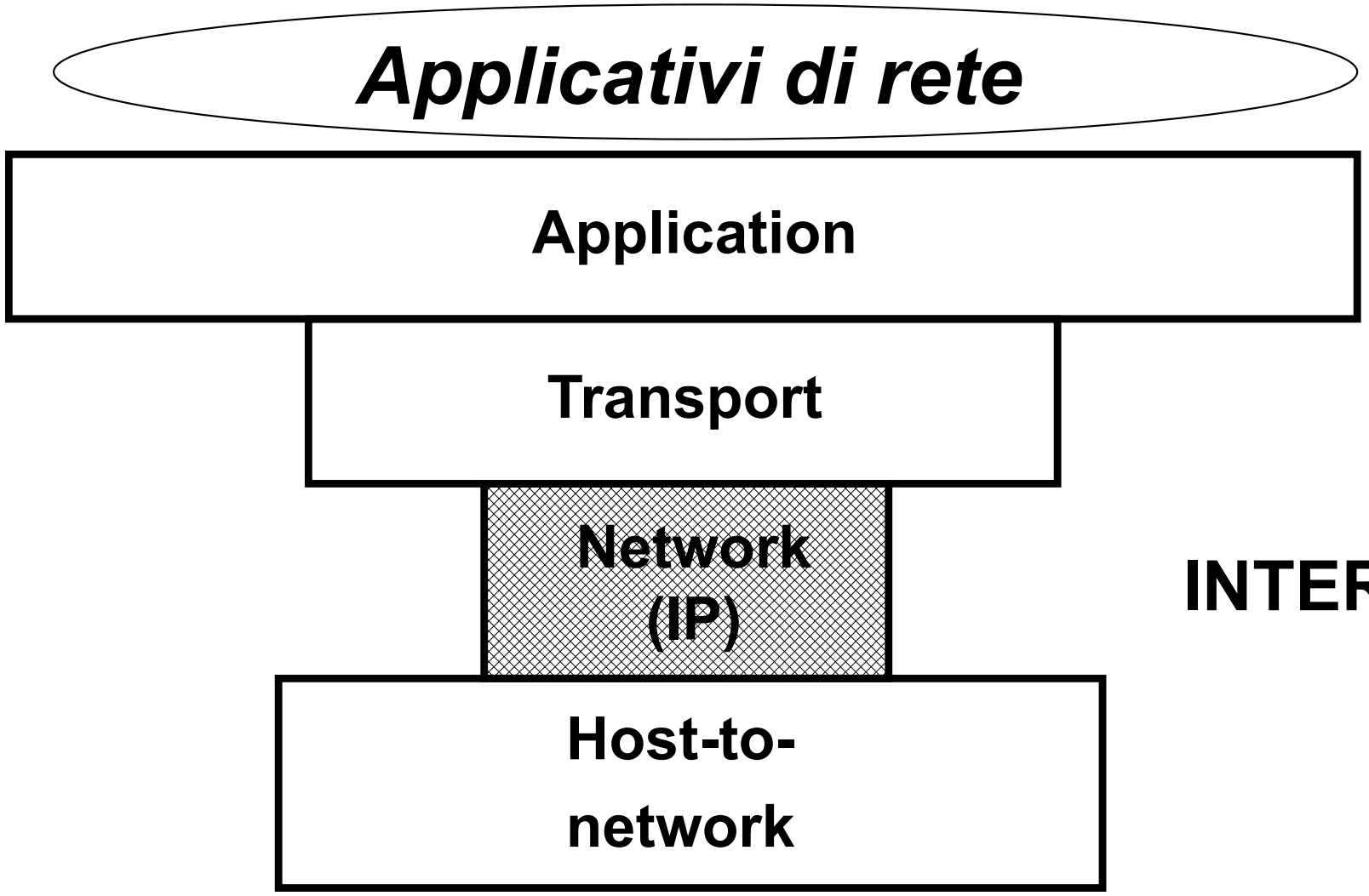
# LIVELLO IP

## (La “dorsale” di Internet)

# Importanza del livello IP

# Suite di protocolli TCP/IP



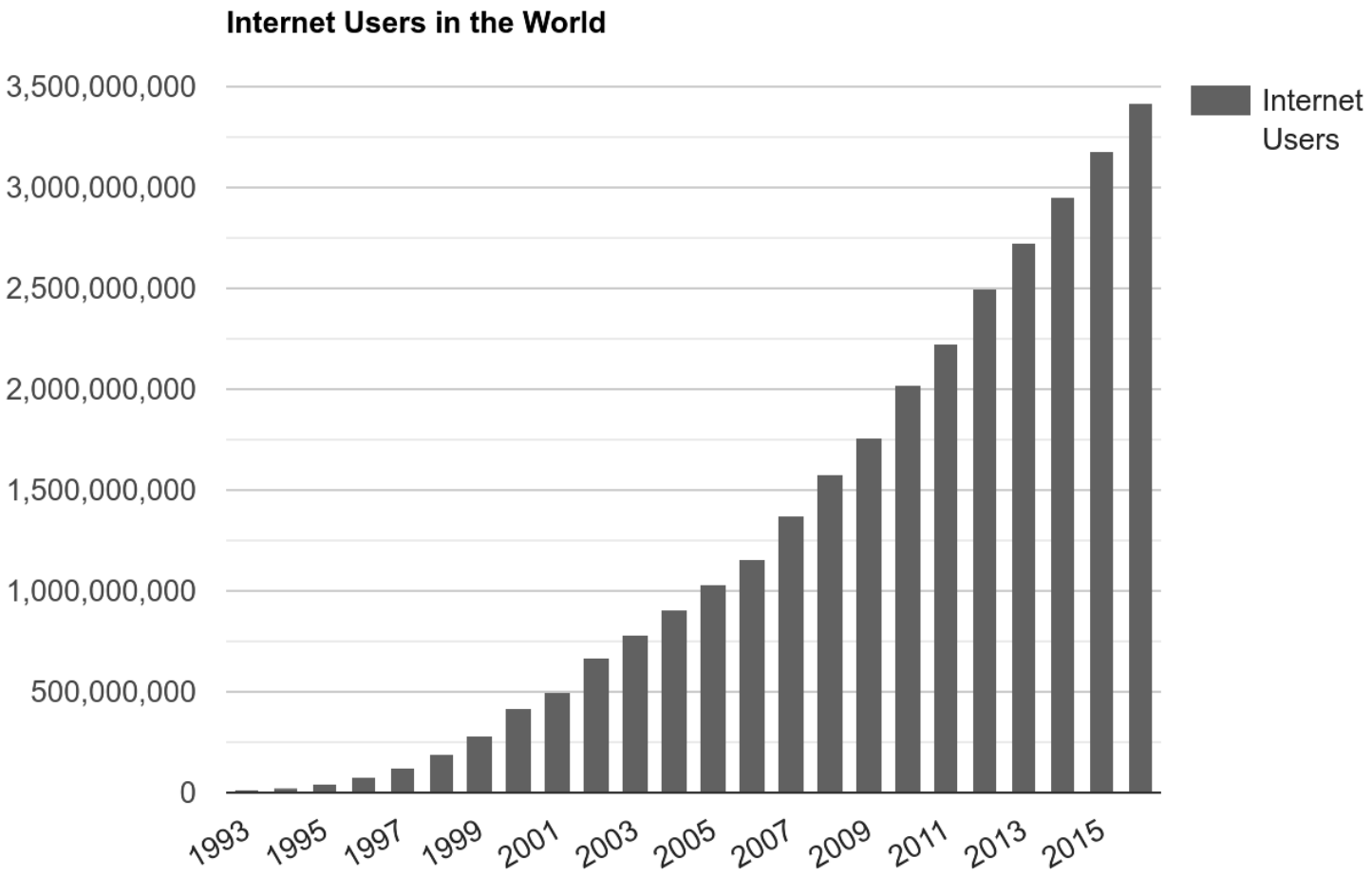
# Il successo continuo e costante di Internet

**Tutti gli host collegati ad Internet devono essere "identificati" in modo univoco**

**Numero di host collegati ad Internet**

| | |
|---|---|
| 1969 | 4 |
| 1979 | 200 |
| 1989 | 100.000 |
| Gennaio 1993 | 1.313.000 |
| Gennaio 1994 | 2.217.000 |
| Gennaio 1995 | 4.852.000 |
| Gennaio 1996 | 9.472.000 |
| Gennaio 1997 | 16.146.000 |
| Gennaio 1998 | 29.670.000 |
| Gennaio 1999 | 43.230.000 |
| Gennaio 2000 | 72.340.000 |
| Gennaio 2001 | 109.574.000 |
| Gennaio 2002 | 147.344.000 |
| Gennaio 2003 | 171.638.000 |
| Gennaio 2004 | 233.101.000 |
| Gennaio 2005 | 317.646.000 |
| Gennaio 2006 | 394.992.000 |
| Gennaio 2007 | 433.194.000 |
| Gennaio 2008 | 541.677.000 |
| Gennaio 2009 | 625.226.000 |
| Gennaio 2010 | 732.740.000 |
| Gennaio 2011 | 818.374.000 |
| Gennaio 2012 | 888.239.000 |
| Gennaio 2013 | 963.518.000 |
| Gennaio 2014 | >miliardo! |

# La crescita esponenziale del numero di host in Internet




**Numero di utenti connessi: > 5 miliardi**
**https://www.internetlivestats.com/internet-users/**
**https://ourworldindata.org/internet**

# IP: un protocollo "antico"

**Descritto nell'RFC 791**

**Pubblicato dalla IETF nel settembre 1981**

# Ma cos'è Internet?

"Internet *refers to the global information system that*

*i) is logically linked together by a globally unique address space based on the Internet Protocol (IP) or its subsequent extensions/follow-ons;*

*ii) is able to support communications using the Transmission Control Protocol/Internet Protocol (TCP/IP) suite or its subsequent extensions/follow-ons, and/or other IP-compatible protocols;*

*iii) provides, uses or makes accessible, either publicly or privately, high level services layered on the communications and related infrastructure described herein"*

**(Federal Networking Council)**

## *Ma cos'è INTERNET ?*

# Una rete globale…

**Obiettivo globale:**

- Connettere un qualsiasi numero di reti **(scalabilità) eterogenee (**distinzione netta fra H2N e Network) e **indipendenti** (alta decentralizzazione organizzativa)

- **Scelte fondamentali del progetto:**
  - Comunicazioni con paradigma ***packet switching***
  - Nodi intermedi (***router***) che inoltrano i pacchetti con "intelligenza minima" (minore complessità = minori costi e maggiori performance)
    - Logica di inoltro **stateless**
    - Intelligenza delegata agli host (ai layer trasporto e applicativo, se necessaria)

# *Ma cos'è INTERNET ?*
# Un insieme di nodi e reti dotati di indirizzi univoci

Nodi terminali

- Host

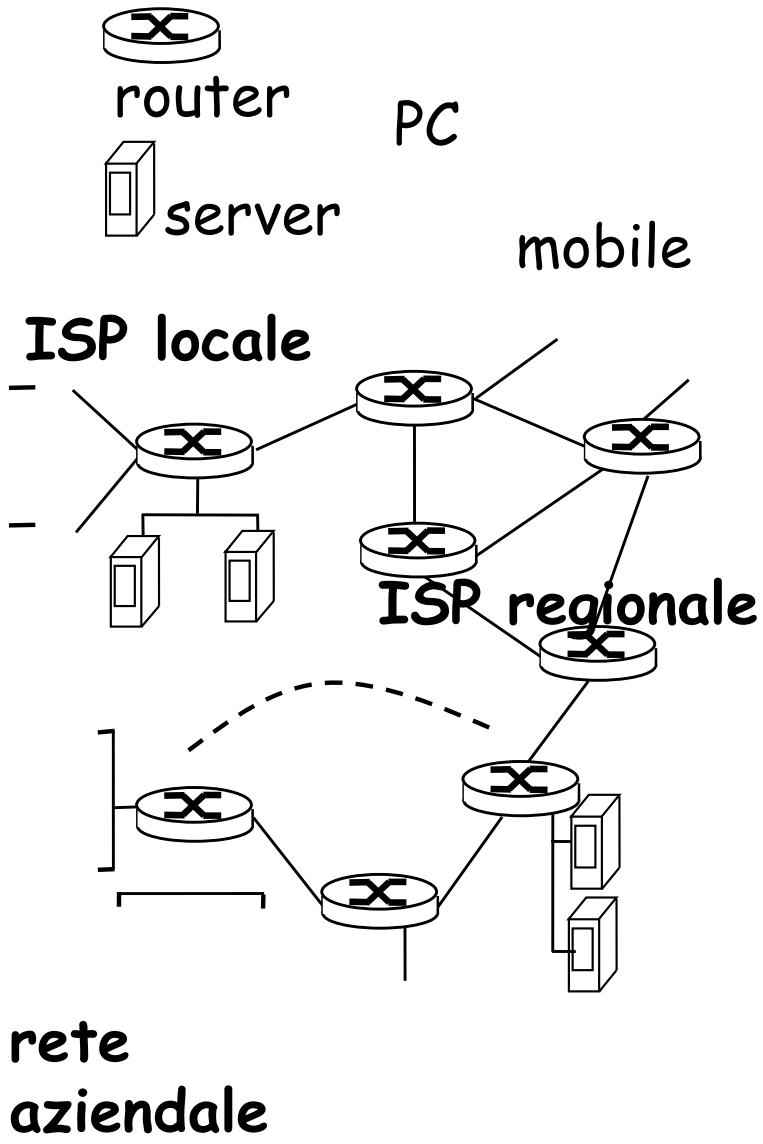
Nodi intermedi

- **Router**

***Ogni nodo ha almeno un indirizzo IP univoco***

I nodi sono "aggregati" in reti, identificabili a loro volta da **"blocchi di indirizzi"**

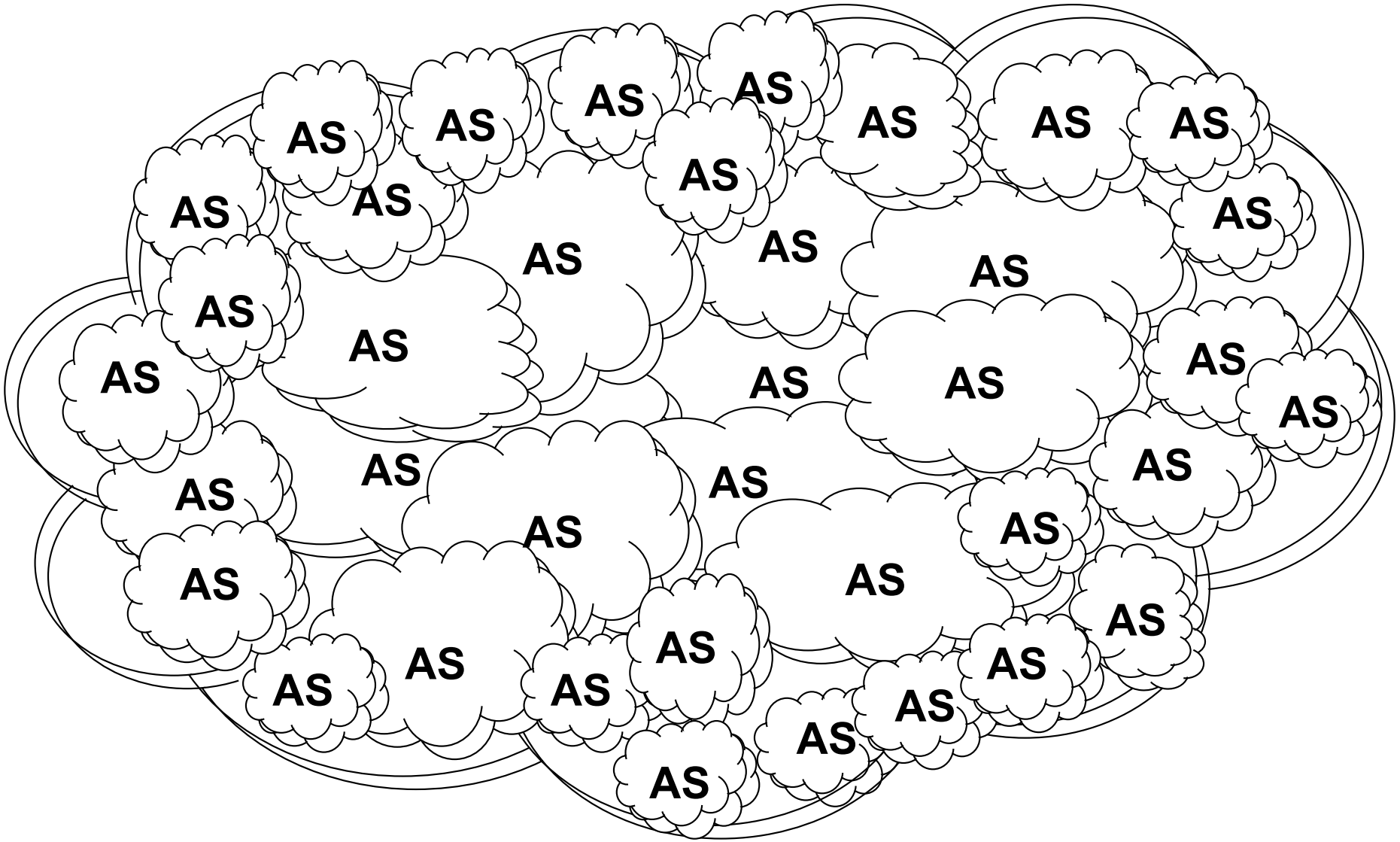
- Logica di **indirizzamento gerarchico**



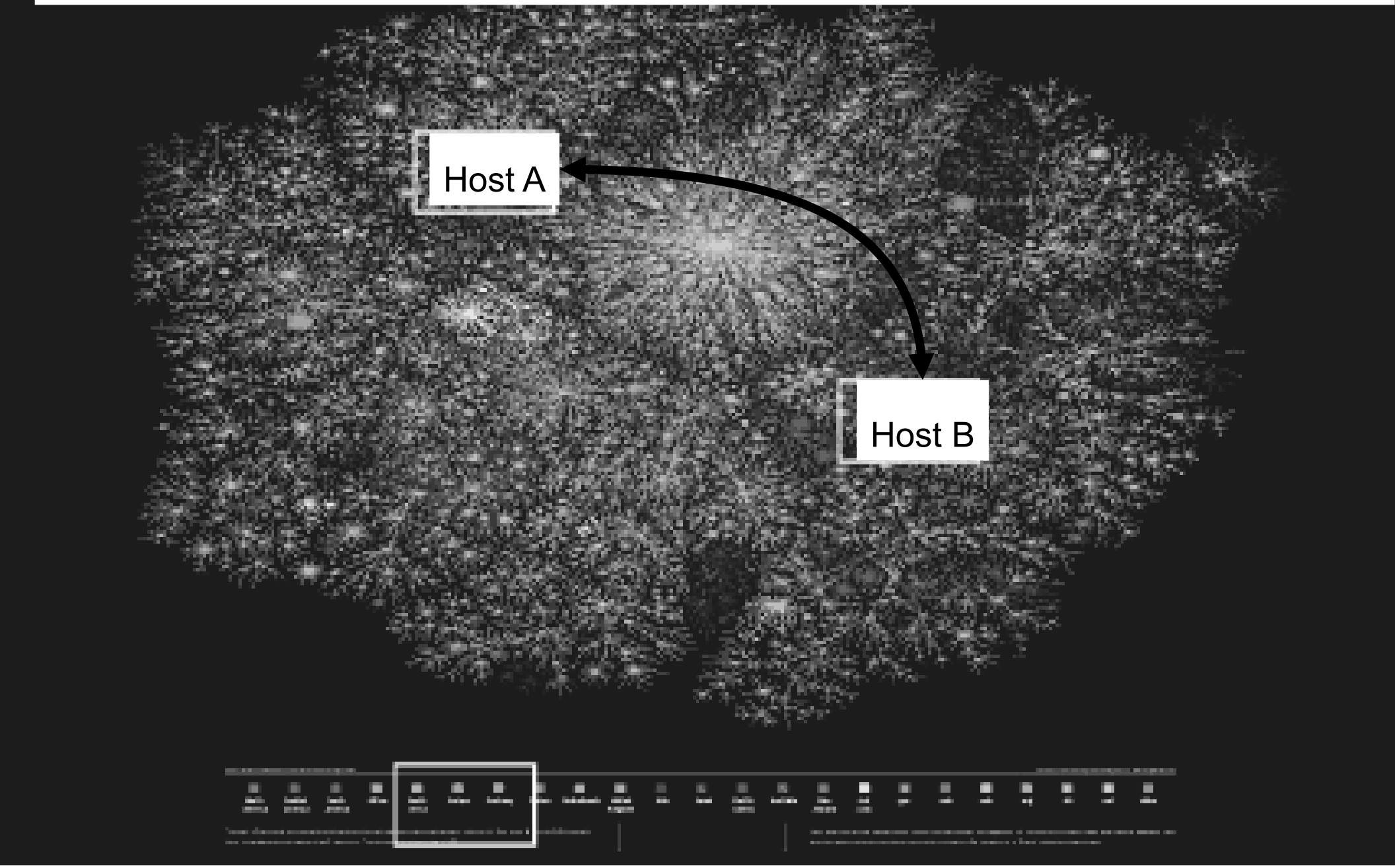
# Cos'è INTERNET ?

## *DAL PUNTO DI VISTA ORGANIZZATIVO:*

*Un insieme di oltre 50000* ***Autonomous Systems*** *alcuni su scala nazionale altri su scala continentale e intercontinentale*



# *Ma cos'è INTERNET* (3)?
# Un entità *trasparente* per gli host



# In sintesi: cos'è Internet?

## DAL PUNTO DI VISTA "FISICO":

*Un insieme di componenti interne (host, link, router) eterogenei, la complessità di gestire l'eterogeneità è delegata ai protocolli H2N, che devono essere "compatibili" con il protocollo IP ma possono essere progettati in modo indipendente*

## DAL PUNTO DI VISTA "FUNZIONALE":

*Rete globale organizzata in tante sotto-reti tramite un sistema di indirizzamento gerarchico, l'inoltro avviene mediate packet switching e inoltro stateless dei pacchetti, per ottenere massime prestazioni dai router*

## DAL PUNTO DI VISTA ORGANIZZATIVO:

*Un insieme di organizzazioni e aziende (Autonomous Systems - AS), che amministrano in modo esclusivo e decentralizzato "porzioni di Internet" e collaborano per mantenere la rete globale e garantire un accesso "neutrale" agli utilizzatori*

# Alcuni princìpi funzionali di progetto

- **Survivability**
  - Se tra due host esiste un qualsiasi percorso, la comunicazione deve poter avvenire
- **Forma a clessidra su più livelli**
  - IP effettua minime assunzioni sui mezzi di trasporto sottostanti e deve funzionare per tutti i tipi di applicazioni di rete
- **Mancanza di "stato"**
  - La "intelligenza" è mantenuta ai bordi della rete (host) e non all'interno (router). Si facilitano la survivability del sistema e le prestazioni di trasmissione
- **Net neutrality**
  - Ogni pacchetto con qualsiasi mittente e destinazione è trattato nello stesso modo
  - Negli ultimi dieci anni tentativi/azioni di indebolimento e ripristino
- **Decentralizzazione organizzativa**
  - Ogni rete è potenzialmente posseduta e gestita da un ente diverso (AS)

# Funzioni del livello 3 network (e scaletta degli argomenti da affrontare)

1. **Si garantisce l'indirizzamento univoco degli host**
   - Tutti gli host collegati a Internet devono essere identificati ed in modo esclusivo → ***Indirizzo IP***
2. **Si definisce l'unità di trasferimento dati**
   - Definisce l'unità informativa utilizzata da Internet per trasferire dati → ***Datagram IP***
3. **Si chiarisce l'architettura di Internet**
   - Definisce i componenti fondamentali di una rete distribuita su scala geografica → ***Autonomous Systems***, ***Router***
4. **Si illustrano le diverse funzioni di routing**
   - Gli algoritmi di routing determinano il percorso nell'ambito di una rete geografica attraverso il quale si consegnano i datagram
   - Caratteristica ***best-effort***: la consegna dei datagram è *non affidabile*

# Un po’ di storia di Internet (extra)

# Internet: *storia e leggenda*

- **La leggenda**

  **Un progetto finanziato dal Ministero della Difesa USA con lo scopo di realizzare una rete in grado di comunicare anche in seguito ad attacchi nucleari**

- **La realtà**

  - **Finanziata dal Ministero della Difesa USA**
  - **Motivazione: successi spaziali dell'URSS e timore del sorpasso tecnologico in clima di guerra fredda**
  - **Obiettivo: consentire l'accesso alle poche risorse di calcolo potenti (e costose) da vari centri di ricerca e Università USA**

# Anni ’60: *la teoria e i primi esperimenti*

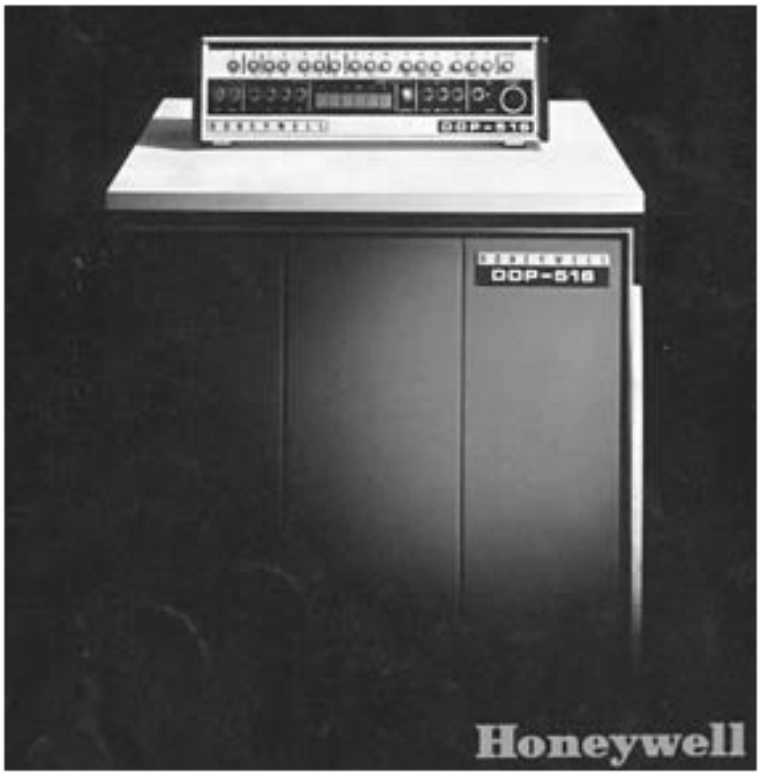
- **1961**: Leonard Kleinrock del MIT pubblica “*Information flow in large communication nets*” sulla teoria del *packet switching*
- **1962**: J.C.R. Licklider e Wesley Clark del MIT pubblicano “*On-line man computer communication*” che rappresenta il primo articolo sul concetto di Internet
- **1962-1964**: J.C.R. Licklider è il primo direttore dell’ufficio IPT dell’Arpa e scrive diversi articoli sul concetto di “galactic network”
- **1964**: Leonard Kleinrock descrive il funzionamento di una rete basata sul ***packet switching*** nel libro “Communication net”, ma non incontra il favore dei principali esperti dell’epoca, che considerano una tale rete irrealizzabile

# Anni ’60: *la teoria e i primi esperimenti*

- **1965**: Larry Roberts e Thomas Marrill effettuano il primo collegamento dati fra Massachussets e Santa Monica in California:
  - Prima volta che due computer si scambiano informazioni
  - Primo utilizzo dei “packets”
  - Risultati sorprendenti: il collegamento a commutazione di circuito era inaffidabile, mentre le teorie di Kleinrock sul “packet switching" funzionavano
- **1966:** Roberts e Marrill pubblicano i risultati in “*Toward a cooperative network of time-shared computers*”, dove viene utilizzato per la prima volta il termine ***protocollo***
- **1966:** Robert Taylor diventa il terzo direttore dell’ufficio IPT dell’Arpa e assume Larry Roberts per portare avanti il progetto Arpanet. **Charlie Hertzfeld, direttore dell’agenzia Arpa, stanzia 1 milione di dollari per il progetto *ARPAnet***

# Internet 1967-1972: *gli albori*

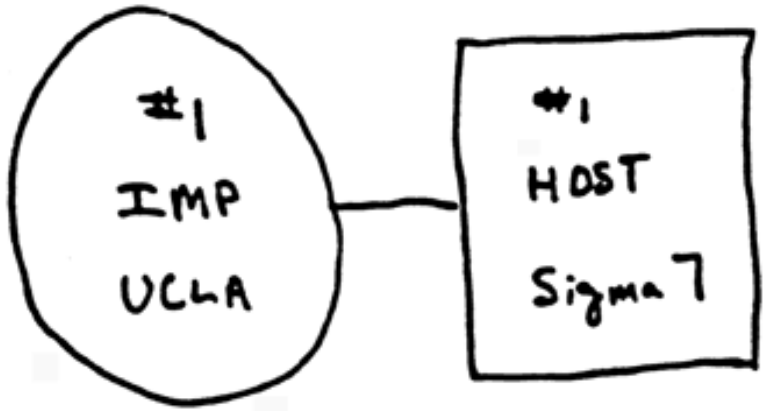
- **1967**: Wesley Clark suggerisce di utilizzare una sottorete di minicomputer, tutti uguali, dedicata esclusivamente alla ricezione e trasmissione dei dati. Suggerisce di chiamare questi computer IMP *(Interface Message Processors)*. Questa idea consente di evitare i problemi hw/sw causati dalla diversità e incompatibilità dei computer dell'epoca
- **1967**: Larry Roberts presenta il primo disegno di Arpanet e rilascia la *Request For Proposals*" (RFP) per la realizzazione degli IMP della rete Arpanet che viene inviata a 140 società
- **1968**: La società BBN vince la gara per la realizzazione degli IMP (→Honeywell con 32 Kbyte di memoria)




Honeywell DDP-516:
- 32 Kbyte di memoria
- Processore a 1,1 MhZ

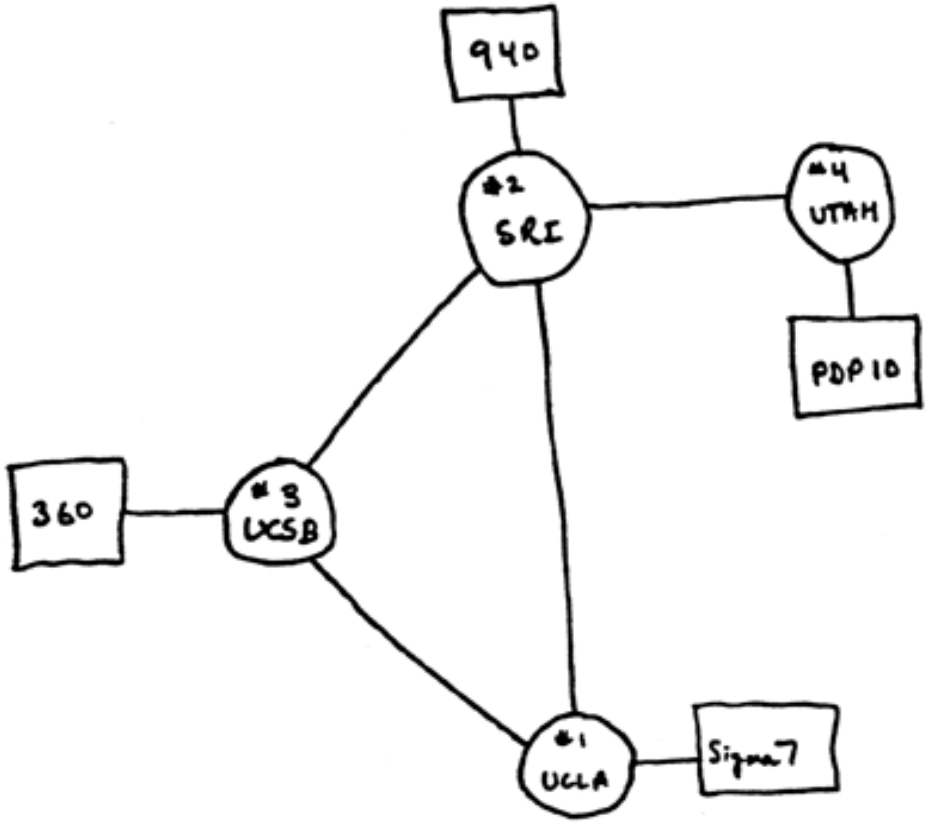
# Internet 1967-1972: *gli albori*

- 1969: Bob Kahn scrive "*host to imp specification 1822" che* descrive le interfacce tra gli host della rete Arpanet e gli IMP. Gli IMP devono essere collegati ai computer attraverso questa interfaccia che deve essere implementata e configurata per ogni computer collegato
- 1969 (apr.): Steve Crocker scrive il Request For Comment (RFC) #1 che tratta l'host-to-host protocol
- 1969 (sett.): Viene installato il primo nodo della rete Arpanet, presso UCLA che si collega a un computer Sigma 7



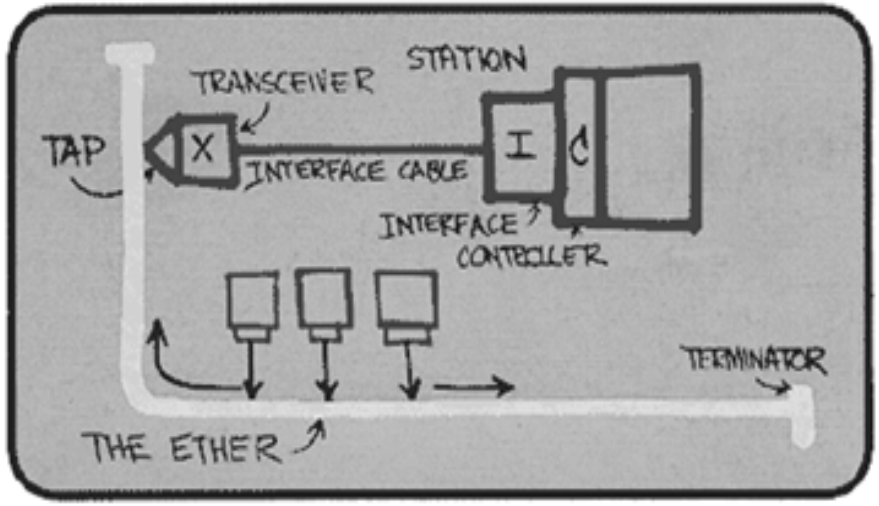
# I primi nodi di Arpanet (1969)

- **1969 (ott.):** Nasce il secondo nodo della rete Arpanet presso lo Stanford Research Institute (SRI) di Doug Engelbart dove si riesce a collegare il computer SDS 940 all'IMP. Il primo messaggio della rete Arpanet passa questo giorno
- **1969 (nov.):** Viene installato il terzo nodo della rete Arpanet presso l'Università di Santa Barbara (UCSB). La rete assume una "topologia ridondante"
- **1969 (dic.):** Viene installato il quarto nodo della rete Arpanet presso l'Università dello Utah



# Internet 1972-1980: *la ricerca*

- 1972:
  - ARPAnet viene presentata pubblicamente
  - NCP (Network Control Protocol): *primo protocollo host-to-host*
  - primo programma di posta elettronica e utilizzo di @ (at)
  - ARPAnet ha 15 nodi (ovvero collega 15 host)
- 1973: Nella tesi di PhD, Metcalfe propone il protocollo e architettura per reti locali *Ethernet*
- 1974: Cerf e Kahn definiscono i princìpi che tuttora regolano l'architettura di Internet (*autonomia, minimalismo, best effort service model, controllo distribuito, router senza stato*)
- fine anni '70: architetture di rete proprietarie: DECnet (Digital), SNA (IBM), XNA
- 1979: ARPAnet collega 200 nodi



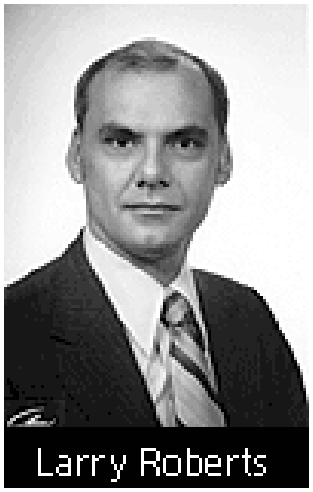
# “Slogan” per il progetto ARPAnet

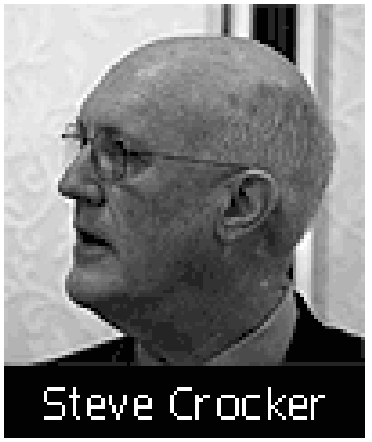
- “Perfection is achieved not when there is no longer anything to add, but when there is no longer anything to take away” [*Antoine de Saint-Exupery*]
- “The simplest explanation is the best” [*Occam’s razor*]
- “Be liberal in what you accept, and conservative in what you send” [*Jon Postel*]
- “In allocating resources, strive to avoid a disaster rather than to achieve an optimum” [*Butler Lampson*]

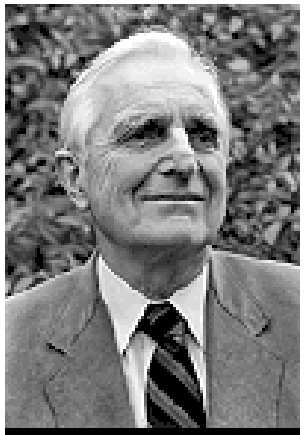
# Alcuni “padri” di Internet


J.C.R. Licklider

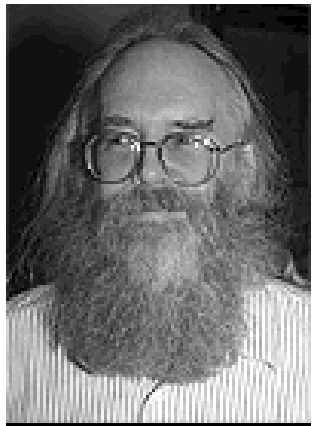

L. Kleinrock

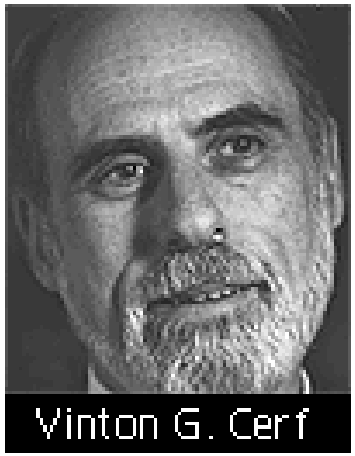

Larry Roberts


Steve Crocker


Doug Englebart

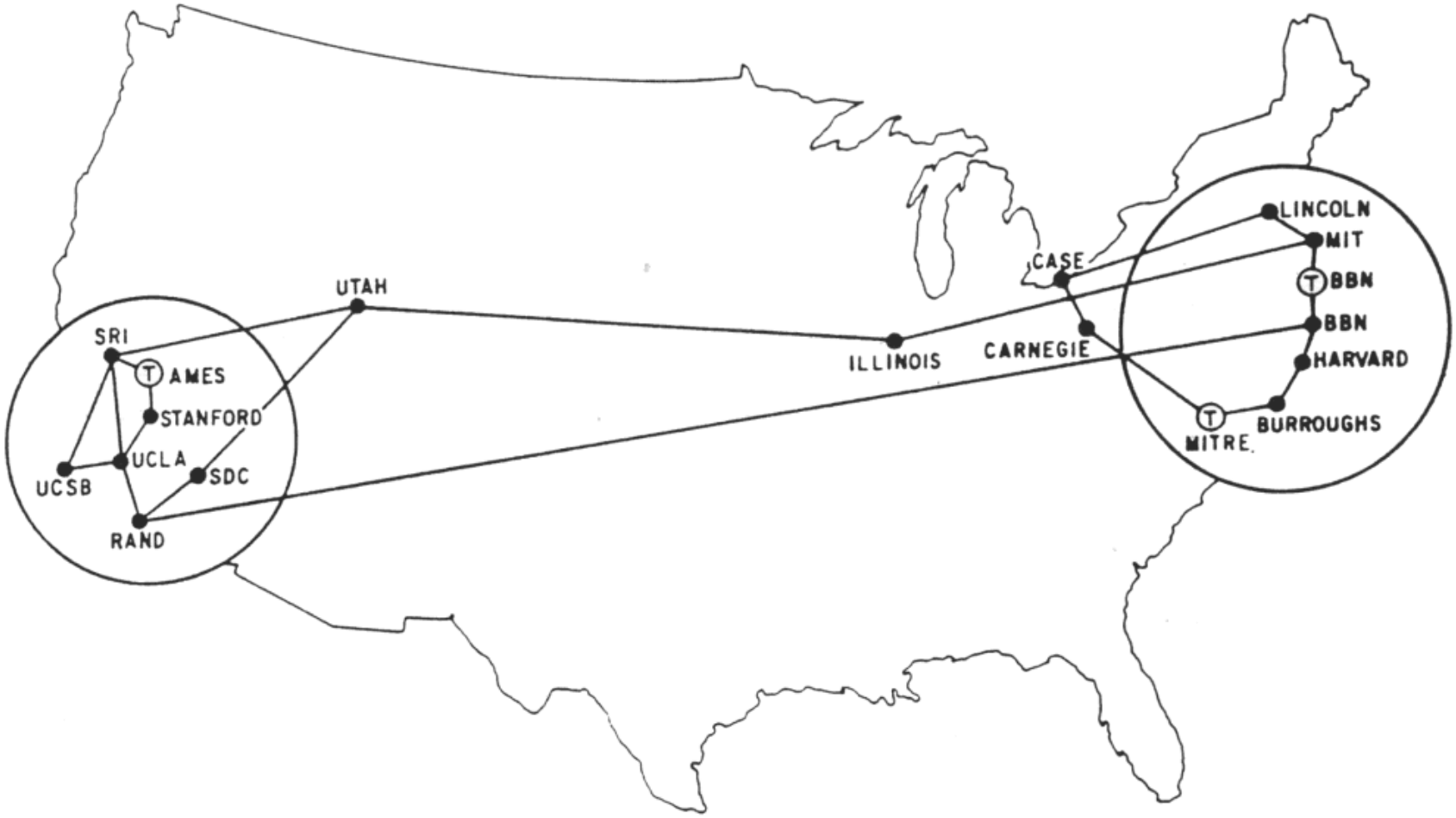

Robert E. Kahn


Jon Postel


Vinton G. Cerf

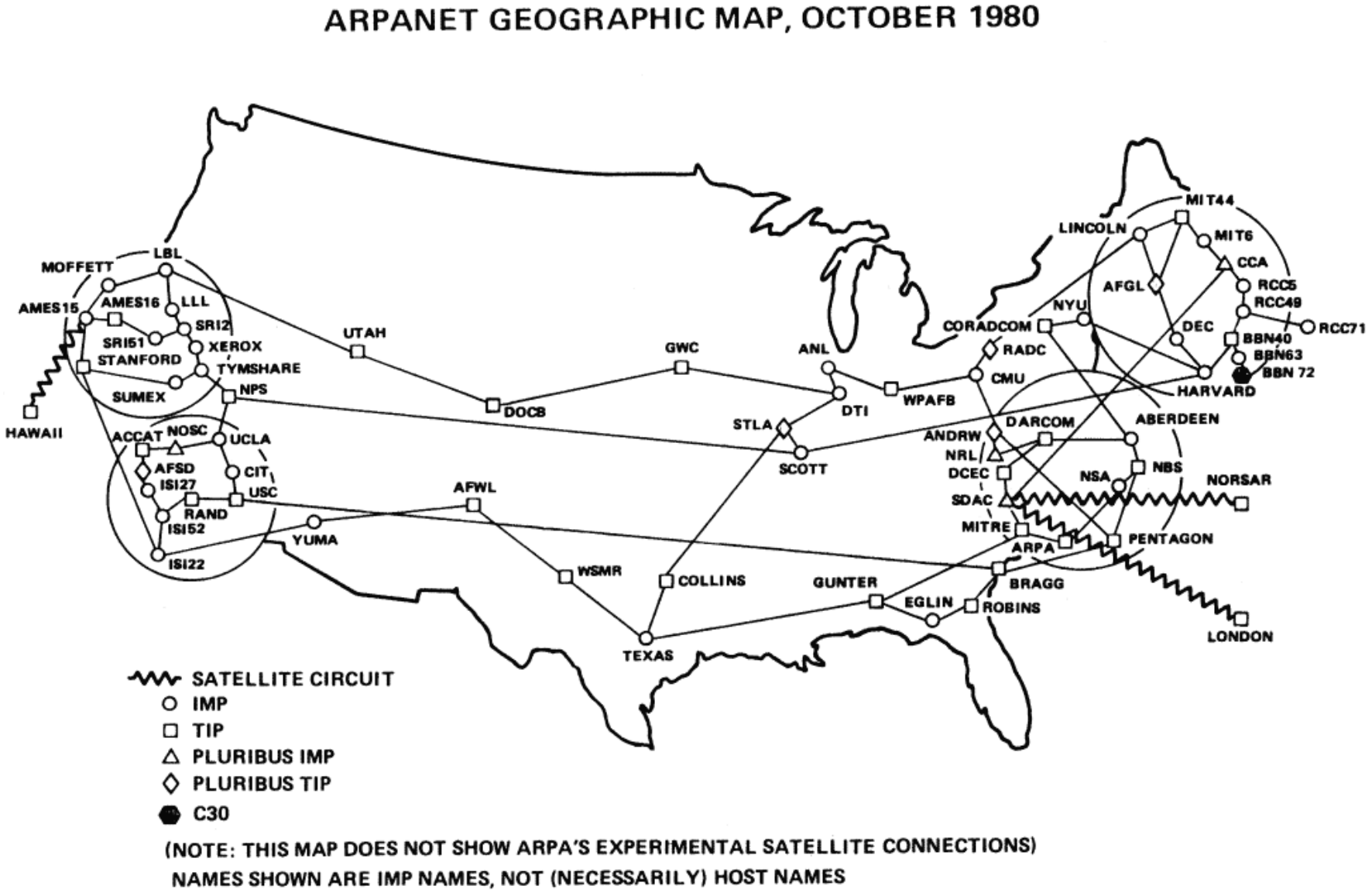

Bob Metcalfe

# Situazione Arpanet (1971)




MAP 4 September 1971

# Situazione Arpanet (1980)



# Internet 1980-1990: *la maturità*

- **1983**: stack TCP/IP
- **1982**: protocollo SMTP per e-mail
- **1983**: sistema DNS distribuito per la traduzione da *hostname* a *indirizzo IP*
- **1985**: protocollo FTP
- **1988**: controllo di congestione del TCP
- **1989**: nuove reti nazionali: Csnet, BITnet, NSFnet, Minitel
- **1990**: 100.000 nodi sono connessi a reti confederate

# Internet 1990-2000: *l'esplosione*

- Inizi '90: si dismette ARPAnet
- 1991: NSF rimuove le restrizioni sull'uso commerciale di NSFnet (dismessa poi nel 1995)
- Inizi '90: nascita del WWW
  - Ipertesti [Bush 1945, Nelson 1960's]
  - HTML, http: Berners-Lee
  - 1994: *Mosaic*, poi *Netscape,* ed *Explorer, poi Firefox*
  - seconda metà anni '90: commercializzazione del WWW

Inizi anni 2000:

- circa 100 milioni di host connessi a Internet
- più di 300 milioni di utenti
- le dorsali di Internet (*backbone*) hanno capacità di trasferimento di 1 Gbps

**Oggi**

- più di 5 miliardi di host connessi ad Internet
- host sempre più eterogenei

# Indirizzi IP (IPv4)

# Indirizzi IP (IPv4)

- Un indirizzo IP ha dimensione 32 bit, ed è solitamente rappresentato tramite i valori di ciascuno dei 4 byte che lo compongono in notazione decimale

- Esempio: 155.185.121.7

# Possibili scelte progettuali

- **Lunghezza indirizzi**
  - lunghezza fissa ⟸ **IP**
  - lunghezza variabile
    - Vantaggi a livello di flessibilità, ma maggiori costi nella gestione dei pacchetti e del routing

- **Spazio di indirizzamento**
  - Gerarchico (strutturato) ⟸ **IP**
  - Flat

# Indirizzi IP

- Per fornire un servizio di comunicazione universale (ogni nodo della rete può comunicare con ciascun altro nodo) occorre un metodo che permetta di identificare univocamente ogni nodo
  - A ogni nodo è assegnato un unico indirizzo Internet (indirizzo IP) formato da 32 bit $\rightarrow 2^{32} \cong 4{,}3$ miliardi di indirizzi diversi
- L'indirizzo IP (32 bit) è suddiviso in 4 campi:
  - Ciascun campo è formato da un byte (8 bit)
  - E' separato da un punto (notazione decimale puntata o ***dotted notation***)
  - Esempio: **130.192.5.189**

# Componenti dell'indirizzo IP

Ogni indirizzo IP è solitamente strutturato in una coppia:

<netid, hostid>

dove ***netid*** (o prefisso di rete) identifica la rete

e ***hostid*** identifica un host di quella rete

**Questa notazione consente di indicare sinteticamente intervalli contigui di indirizzi**

**(anche detti "blocchi" o "range" di indirizzi definiti da "prefissi di rete")**

# Componenti dell'indirizzo IP

Ad esempio: **128.211.121.7**

L'indirizzo può essere composto da due parti

**NetId: 128.211 HostId: 121.7**

Cosa determina le parte di NetId e di HostId?

Dipende:

- Assegnazione di **classi predefinite**
- Ripartizione manuale tramite notazioni **classless**

Ricordare l'obiettivo: ridurre il numero di regole di routing da memorizzare nelle tabelle di routing

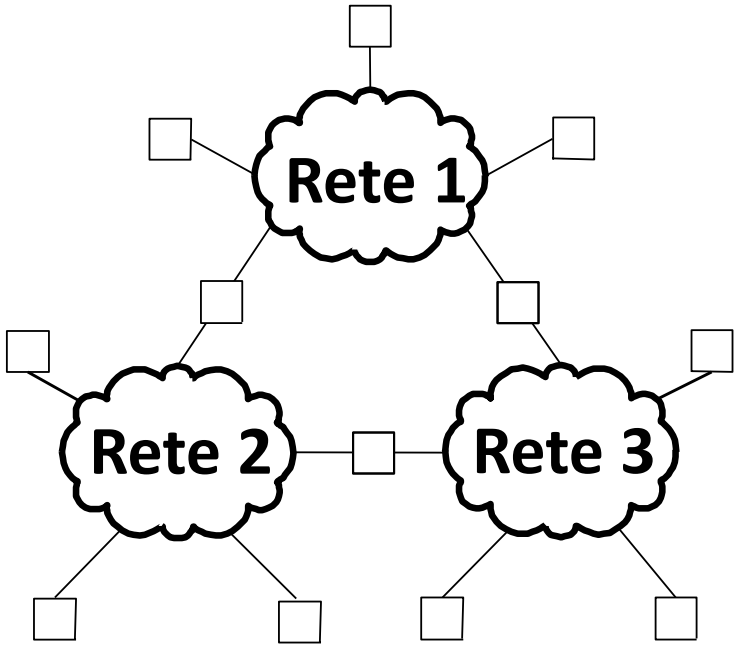
# Assegnazione indirizzi IP nelle reti [1]

- Lo **spazio degli indirizzi** di IP viene *solitamente* gestito in «blocchi» (o «intervalli») di indirizzi

**NetId Rete 1: 128.211**

**NetId Rete 2: 128.212**

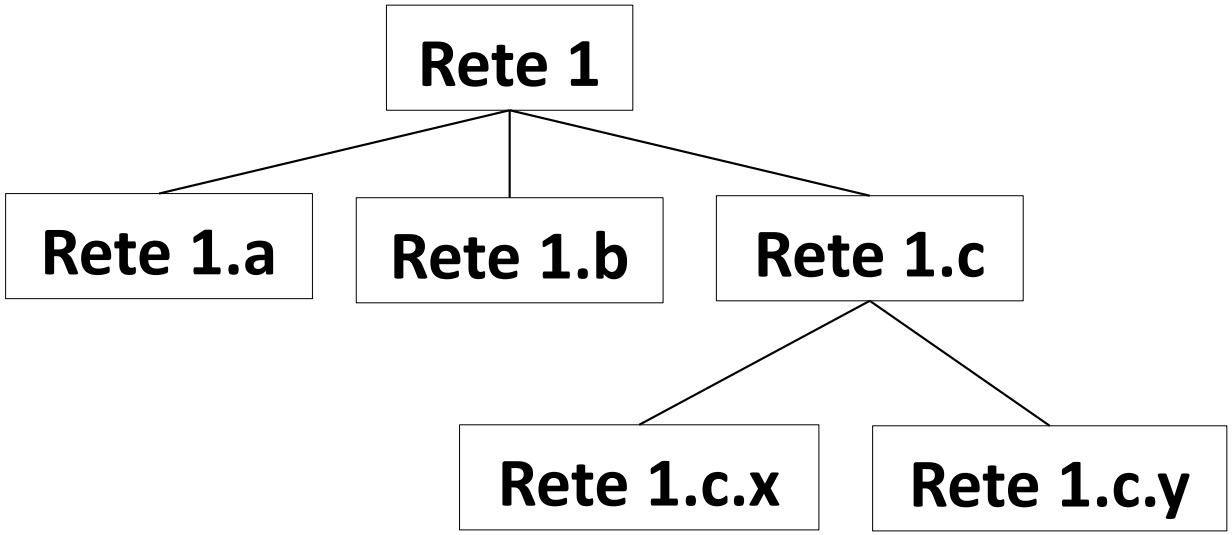
**NetId Rete 3: 128.213**

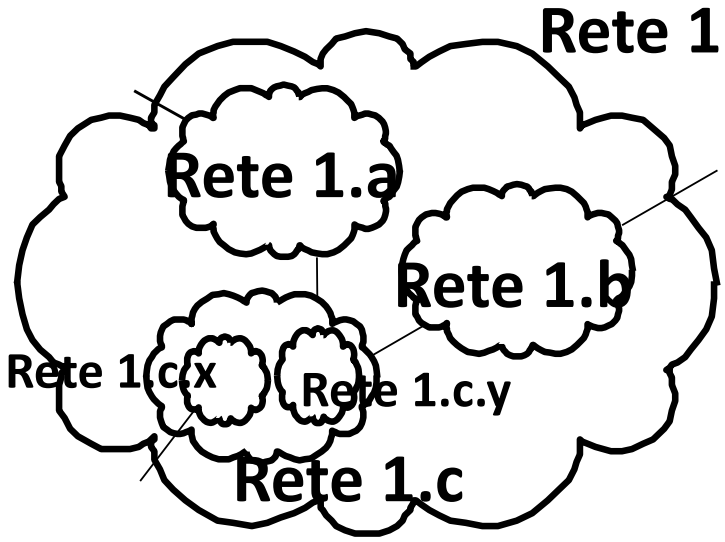



- Ogni rete **gestisce in modo esclusivo** tutti gli indirizzi IP sottintesi dal NetId assegnato
  - Univocità di indirizzi a livello di rete

# Assegnazione indirizzi IP nelle reti [2]

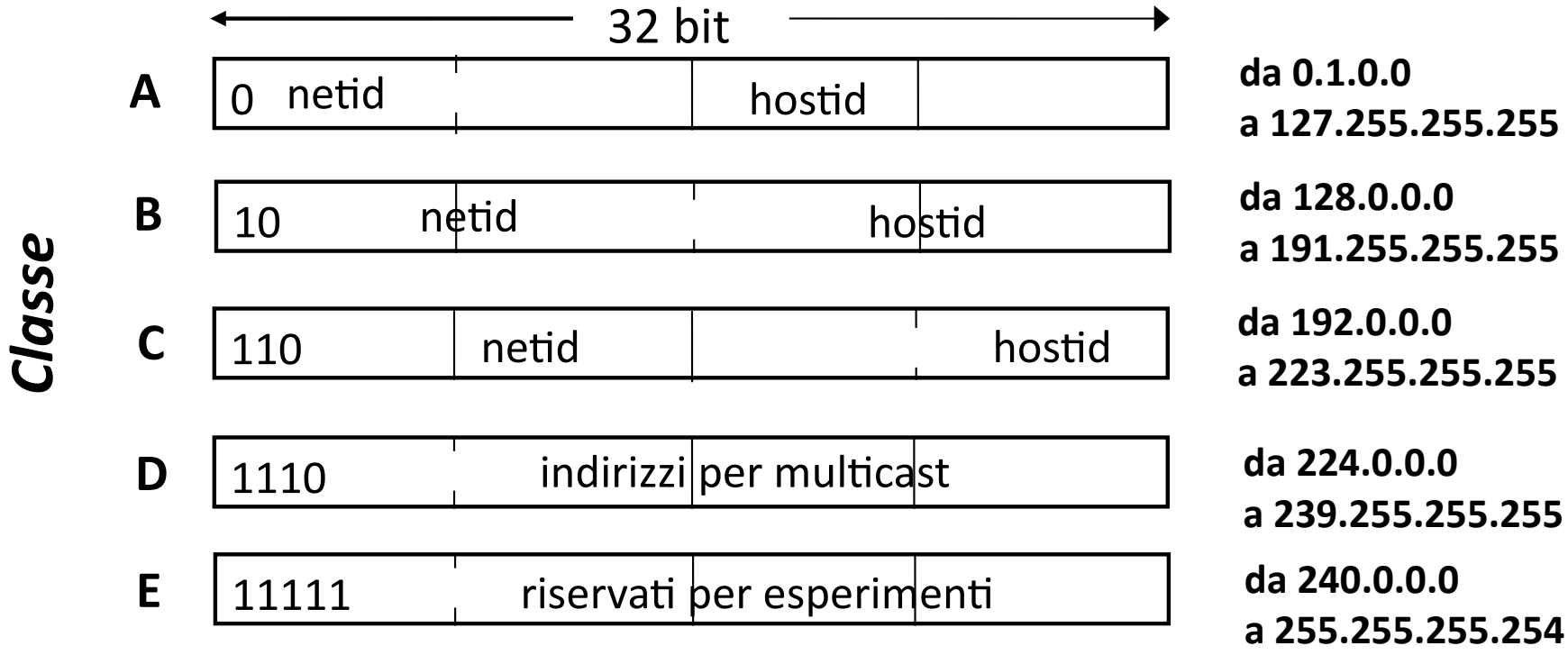
- L'approccio è **ricorsivo** (consistente con la definizione di rete)
  - Un blocco di indirizzi può essere assegnato a una rete locale in cui tutti gli indirizzi possono comunicare a livello H2N
  - Oppure essere assegnato a una rete «logica» che contiene più reti («logiche» o locali)
    - In questo caso si parla di **subnetting** (approfondimenti dopo)



# Classi di indirizzi IP

- 3 classi utilizzabili per l'indirizzamento di host (*classe A*, *classe B*, *classe C*), più 1 classe per multicast address (*classe D*), più 1 classe riservata (*classe E*)
  - La quantità di bit destinati al prefisso di rete dipende dalla classe cui l'indirizzo appartiene
  - La classe è codificata dai bit più significativi dell'indirizzo



# Dimensioni delle classi di indirizzi

- **Classe A** (7 bit per netid, 24 bit per hostid)
  - 128 (**2** ) possibili network ID
  - Oltre 16 milioni di host ID per ciascun network ID
- **Classe B** (14 bit per netid, 16 bit per hostid)
  - 16K =16384 (**2** ) possibili network ID
  - 64K = 65536 (**2** ) host ID
- **Classe C** (21 bit per netid, 8 bit per hostid)
  - Oltre 2 milioni (**2** ) di possibili network ID
  - 256 (**2^** ) host ID

# Indirizzi *classless (CIDR)*

- Oggi sono necessarie architetture più flessibili
- Gli indirizzi non sono considerati in classi fisse, ma l'intero spazio di indirizzamento può essere suddiviso in blocchi di dimensioni differenti
- Si usa la notazione **CIDR (Classless Inter-Domain Routing)** dove ciascun insieme di bit del netid è indicato dal suffisso ***n*** nella notazione

**a.b.c.d/n**

# Indirizzi *classless*

- La ripartizione degli indirizzi in classi è molto rigida e poco graduale perché basata su interi byte:
  - si passa da reti con 250 host (Classe C) a reti con 65000 host (Classe B) a reti con milioni di nodi (classe A)

- Per motivi gestionali e di efficienza del routing interno, può convenire definire degli "insiemi logici" di indirizzi più flessibili rispetto alla suddivisione rigida in 1, 2, 3 byte per il *netid*
- Più flessibili significa passare *da una suddivisione in byte ad una suddivisione in bit* per la coppia <*netid*, *hostid*>

# Indirizzi Classless (CIDR)

- Con la notazione CIDR si elimina il concetto di indirizzamento a classi fisse A-B-C-D: l'indirizzo IP non ha più un confine fisso tra netid e hostid
- Si utilizza la notazione slash per indicare il numero di bit usati per netid **a.b.c.d/*x***. Es., 197.8.3.0/24
- Gli indirizzi CIDR richiedono l'utilizzo di strutture dati e algoritmi opportuni da utilizzare per consultare in modo efficace le **tabelle di routing**
  - **Le tabelle di routing devono conservare anche l'informazione relativa alla netmask, e non solo la rete di destinazione**
- Si utilizza un approccio di tipo **longest prefix** per gestire possibili conflitti fra diverse regole di routing

**Approfondiremo in seguito**

# Assegnamento indirizzi IP

- **Gli indirizzi IP sono indirizzi *logici* (non fisici)**
- Ciascun host deve essere identificato da un indirizzo IP, che può essere assegnato:
  - permanentemente ad un host
  - oppure dinamicamente al momento del boot di un host

- **Come fa un host a conoscere il proprio indirizzo IP**?
  - **Configurazione manuale**: l'indirizzo IP è configurato in un file dall'amministratore del sistema
  - **Dynamic Host Configuration Protocol (DHCP)**: allocazione dinamica effettuata da un server speciale

# Indirizzi IP speciali

- **Network address**: **hostid** con tutti i bit uguali a 0 (es., 128.211.0.0 indica la rete di classe B avente netid 128.211) → denota il **netid** (prefisso) assegnato ad una rete
- **Directed broadcast address**: **hostid con tutti i bit uguali a 1** (es., 128.211.255.255 indica il broadcast per la rete di classe B avente netid 128.211) → permette il broadcast a tutti gli host di una certa rete
- **Limited broadcast address**: **tutti i bit uguali a 1** (ossia 255.255.255.255) → permette il broadcast sulla rete fisica locale
- **Nessun indirizzo IP**: **tutti i bit uguali a 0** (ossia 0.0.0.0) → usato per il boot dell'host o per configurazioni «particolari»
- **Loopback address (localhost)**: la classe A con netid pari a 127 (es., 127.0.0.1) → è un indirizzo software virtuale senza corrispettivo hardware e senza connessioni di rete: è usato per il testing di applicazioni di rete (ad es., consente di comunicare con un server sulla stessa macchina: http://127.0.0.1)

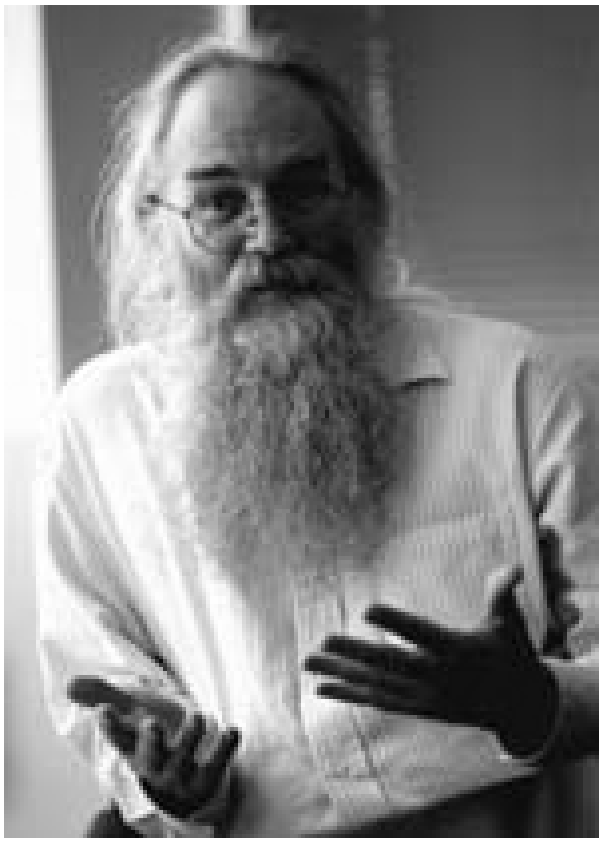
# Gestione indirizzi IP Pubblici (IPv4)

# Gestione indirizzi e domini (dal 1986)

- Il governo USA creò la **Internet Assigned Numbers Authority** (**IANA**), **in pratica il gruppo di Jon Postel**, per gestire le assegnazioni di gruppi di indirizzi
- "IANA è nata con Jon Postel, **era Jon Postel"** (in qualche modo "muore" con lui)

# Jon Postel [1943-1998]

- Autore degli RFC 791-793 (Internet Protocol standard)
- Autore di oltre 200 RFC
- Verificatore degli standard
- **Definito lo "Zar dei numeri"**
  - Curatore delle *well known port* dei protocolli a livello trasporto
  - Editor degli RFC
  - Direttore di IANA
- RFC2468 ("I remember IANA") scritto in sua memoria da Vint Cerf: «I doubt that anyone could possibly duplicate his record, but it stands as a measure of one man's astonishing contribution to a community he knew and loved»



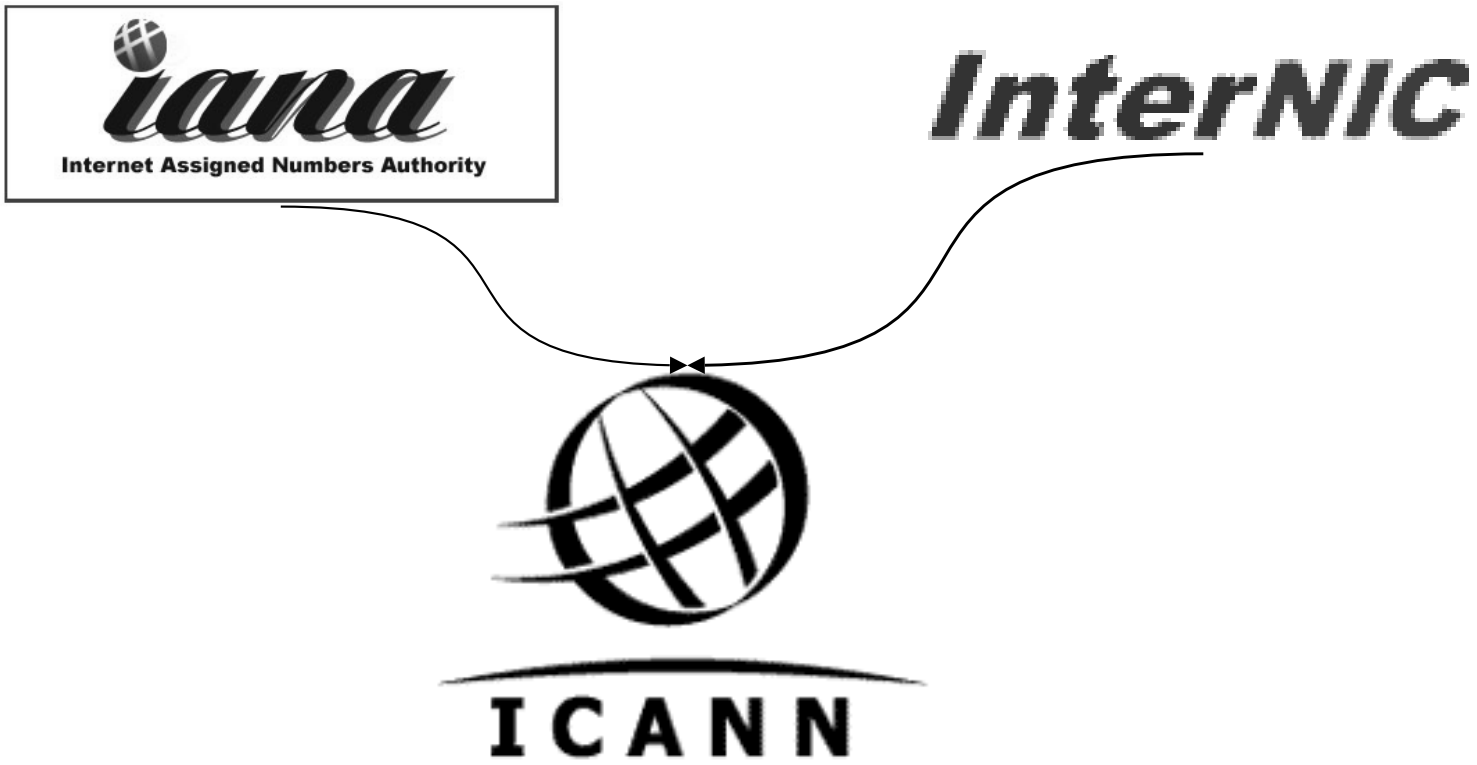
# Gestione di Internet

- Struttura estremamente decentralizzata in cui esiste soltanto un coordinamento molto lasco a livello di:
  - definizione ed accettazione degli standard
  - distribuzione della documentazione
  - assegnamento degli indirizzi e dei nomi

- La giurisdizione sugli IP number era della **IANA** (**Internet Assigned Number Authority**)
- La distribuzione era effettuata da **INTERNIC** (**Internet Network Information Center**)
- A livello locale, gli indirizzi si ottenevano da un provider che aveva a disposizione degli insiemi su delega di INTERNIC
- Per la ricerca scientifica italiana (Università e centri di ricerca) l'organo di riferimento era ed è il **GARR**

# Dal 1998

- Il Governo statunitense riconosce l'autorità della **Internet Corporation for Assigned Names and Numbers** (**ICANN**) internazionale:

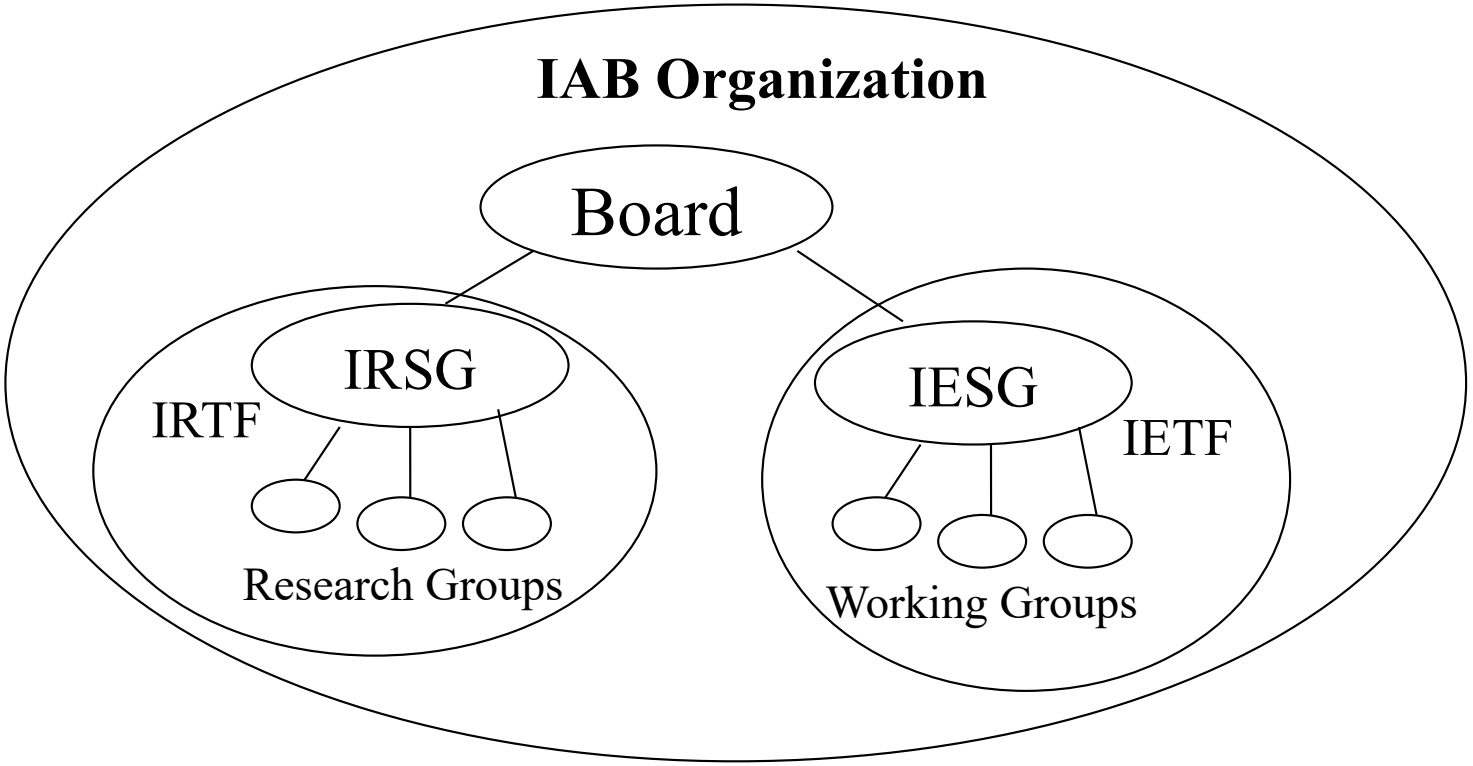


# IANA → ICANN

- **IANA è adesso sotto il controllo formale di ICANN**
- **ICANN incorpora tutte le responsabilità di IANA anche se delega a IANA alcune funzioni di gestione:**
  - IANA alloca lo spazio di indirizzi IP in collaborazione con i cinque Regional Internet Registry (RIR): **AfriNIC** (Africa), **APNIC** (Asia/Pacific), **ARIN** (North America), **LACNIC** (Latin America), **RIPE NCC** (Europe, Middle East, Central Asia)
  - IANA gestisce il Servizio di registrazione per gli identificativi dei numeri di porta dei protocolli (il significato si vedrà a livello 4 “trasporto”)
  - Responsabile della gestione della DNS root zone

# Standard e distribuzione documentazione

IAB = Internet Architecture Board
IRTF = Internet Research Task Force
**IETF** = Internet Engineering Task Force – valuta RFC per Internet standards
IRSG = Internet Research Steering Group
IESG = Internet Engineering Steering Group



# Ricordare la tempistica degli RFC

- [RFC 0001] “Host software”, April **1969**
- [RFC 1000] “RFC reference guide”, Aug. **1987**
- [RFC 2000] “Internet Official Protocol standards”, Feb. **1997**
- [RFC 3000] “Internet Official Protocol standards”, Nov. **2001**
- [RFC 4001] “Textual Conventions for Internet Network Addresses”, Feb. **2005**
- [RFC 5001] “DNS Name Server Identifier option”, Aug. **2007**

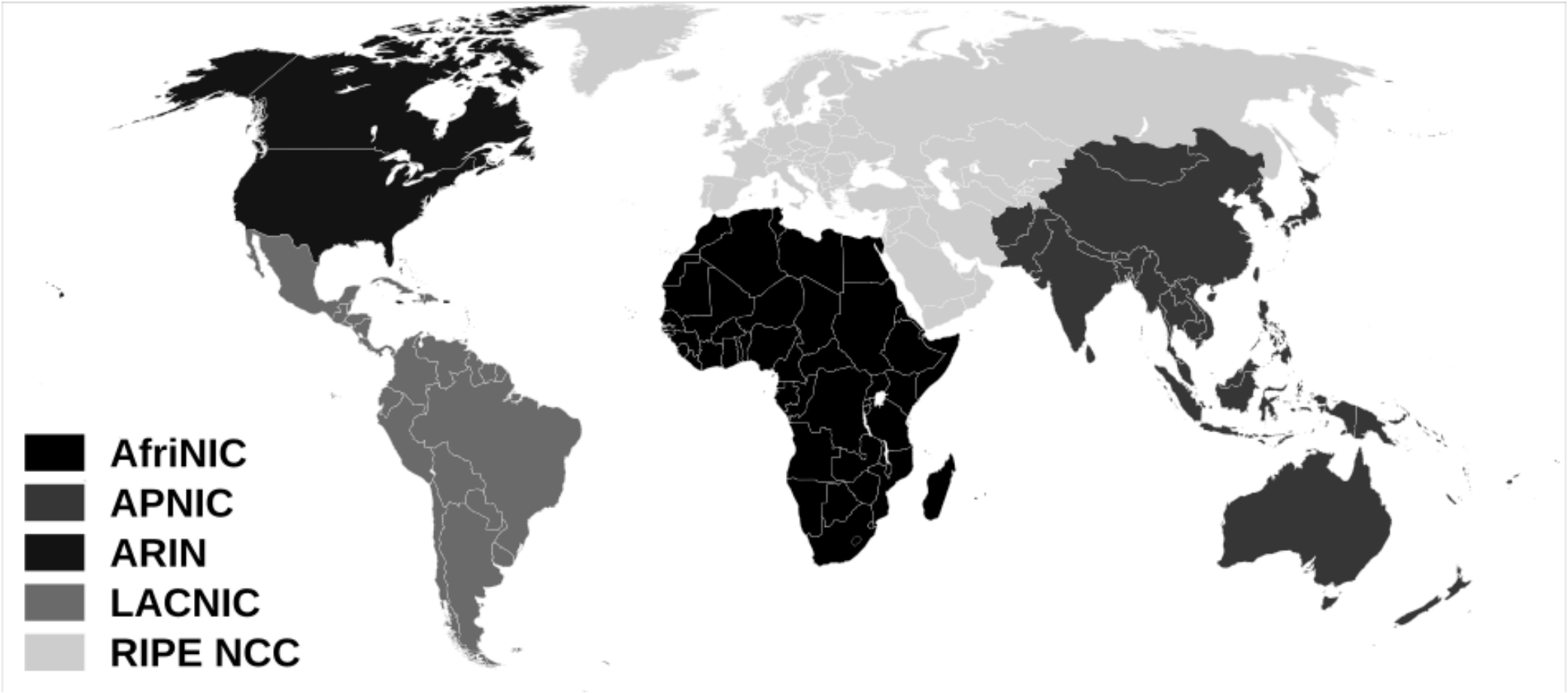
***A ottobre 2022, siamo intorno a RFC 9291***

*https://www.ietf.org/rfc/rfc-index-latest.txt*

# Assegnamento indirizzi IP

- Un **network ID**, corrispondente a un insieme di indirizzi IP, è assegnato a (poche) organizzazioni e tipicamente agli ***Internet Service Provider*** da **IANA/ICANN**
- Un'organizzazione richiede un network ID a qualche ISP
- Gli **host ID** sono assegnati localmente a ciascun host dall'amministratore di rete della organizzazione

# Regional Internet Registry




**https://www.iana.org/numbers**

# Chi possiede indirizzi di Classe A

- IANA
- General Electric
- Level 3 Communications
- Army Information Systems Center
- IBM
- DoD
- AT&T Bell Laboratories
- Xerox Corporation
- Hewlett-Packard Company
- Digital Equipment Corporation
- Apple Computer Inc.
- Ford Motor Company
- …

- Japan Inet
- Bell-Northern Research
- Prudential Securities
- Army Information Systems Center
- Deparment Social Security (UK)
- APNIC
- DoD Network Information Center
- US Postal Service
- UK Ministry of Defence
- AfriNIC
- ARIN
- …

# Subnetting e Supernetting (IPv4)

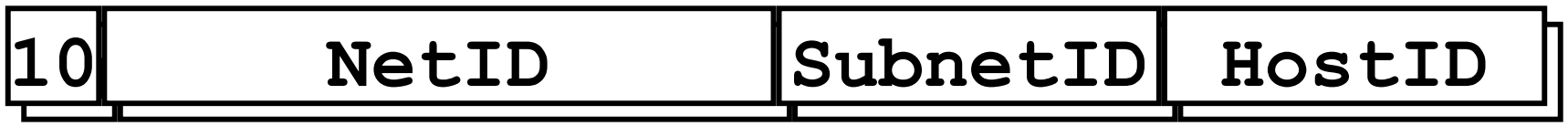
# Subnetting e Supernetting

- Due opportunità:
  - sottoclassi di indirizzi IP (***subnet***), soprattutto per organizzazioni con indirizzi di classe B
  - sopraclassi di indirizzi IP (***supernet***), per organizzazioni grandi con più indirizzi di classe C ovvero per ISP

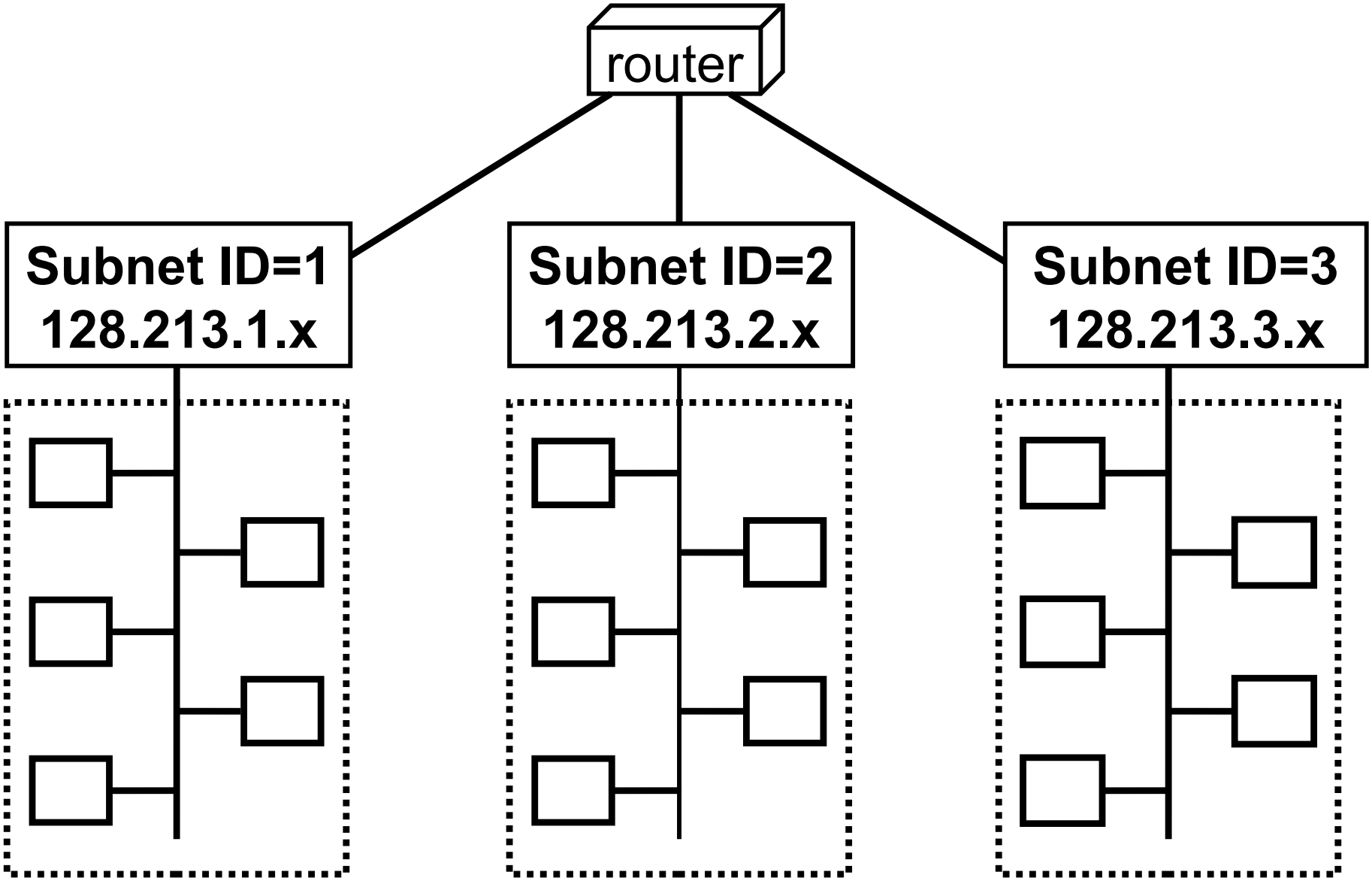
- Due vantaggi:
  - Si crea maggiore flessibilità nella ripartizione degli indirizzi all'interno di un'organizzazione (es., Università con indirizzi di Classe B)
  - Si facilitano le operazioni di routing dei pacchetti identificando insiemi di indirizzi di host contigui

# Subnetting

- Un'organizzazione può suddividere il suo spazio di host address in gruppi detti **subnet**
- Il **subnet ID** è tipicamente utilizzato per raggruppare host basati sulla topologia fisica della rete
- Per esempio, per un indirizzo di classe B, si può avere:

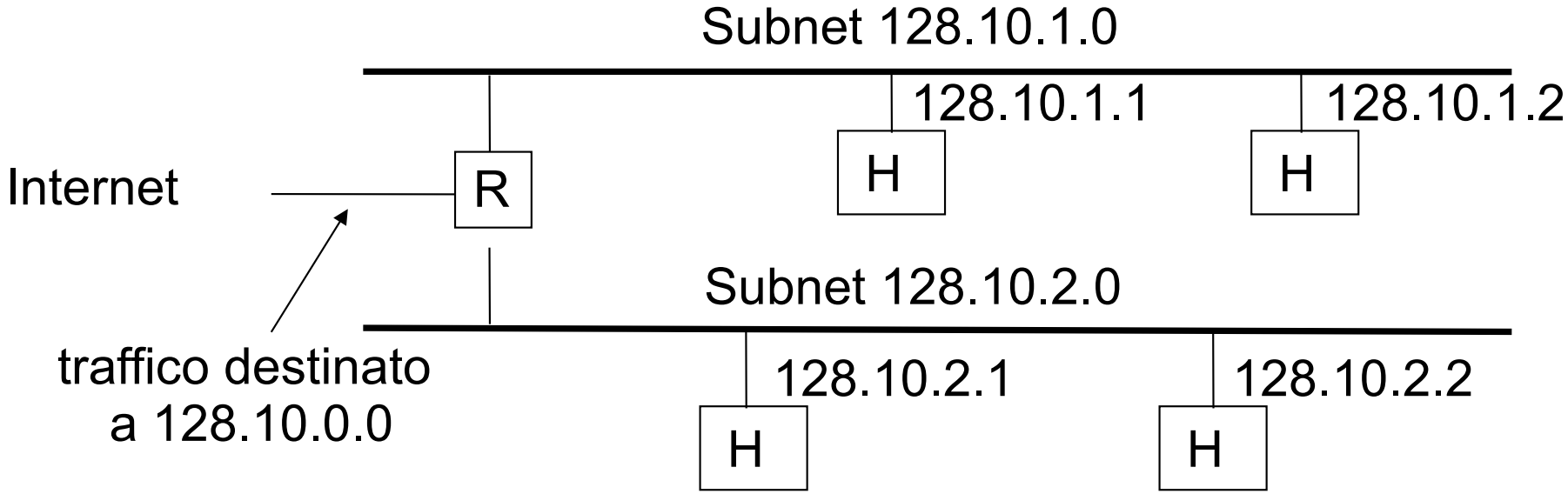


# Subnetting (*cont.*)



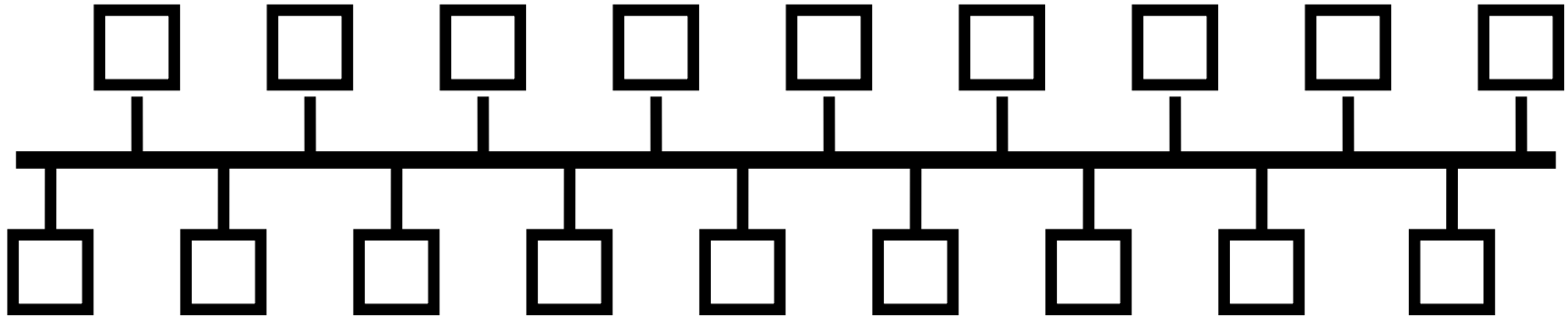
# Subnet addressing

- **Schema di indirizzamento IP originale:**
  - **ad ogni rete fisica è assegnato un unico "indirizzo di sottorete"**
  - **ogni host appartenente a questa rete <u>ha come netid</u> l'indirizzo di sottorete**



# **Subnetting addressing** (*cont.*)

- Il subnetting consente la massima flessibilità
- E' possibile anche avere uno stesso segmento di rete fisico suddiviso in multiple subnet logiche, corrispondenti per esempio a diversi gruppi di una organizzazione
- Es.,



# Uso della network mask

- **Per definire i bit (non i byte!) dedicati al *netid* si usa una *network mask* di 4 byte**. Es.

  **`Net mask: 11111111.11111111.11111111.11000000`**

- La ***network mask*** permette di individuare due dati mediante un AND logico con l'indirizzo IP:
  - quale parte di un indirizzo IP è riservata per il *netid* (la parte di 1)
  - quale parte è disponibile per gli *hostid* (la parte di 0)

# Esempi di subnet mask

- Implementazione delle subnet usando le maschere:
  - **subnet mask** formata da 32 bit per ciascuna rete che usa il subnet addressing
  - nella mask, i bit settati ad 1 corrispondono alla parte di rete, quelli settati a 0 alla parte host
- Esempio di rete di classe B con cinque reti fisiche suddivise su tre livelli:
  - maschera = **11111111 11111111 11100000 00000000**
- Esempio di rete in cui tutto il terzo byte dell'indirizzo IP è usato per la subnet:
  - maschera = **11111111 11111111 11111111 00000000**

# Subnet mask: esempio di uso

- Indirizzo IP: **156.154.81.56**
- Network mask: **255.255.255.240**
- A quale sottorete appartiene?

```
Indirizzo IP: 10011100.10011010.01010001.00111000
 Subnet mask: 11111111.11111111.11111111.11110000
(AND) ----------------------------------------
      Subnet: 10011100.10011010.01010001.00110000
```

- Qual è il range di host della sottorete?
  - Ci sono 2 -2 host nella subnet, dove n è il numero degli ultimi 0 della subnet mask. Nell'esempio: 2 -2=14, ovvero da 156.154.81.49 a 156.154.81.62
- Qual è il *broadcast address* della sottorete?
  - 10011100.10011010.01010001.00111111 → 156.154.81.63

# Esempio di subnetting

Contesto

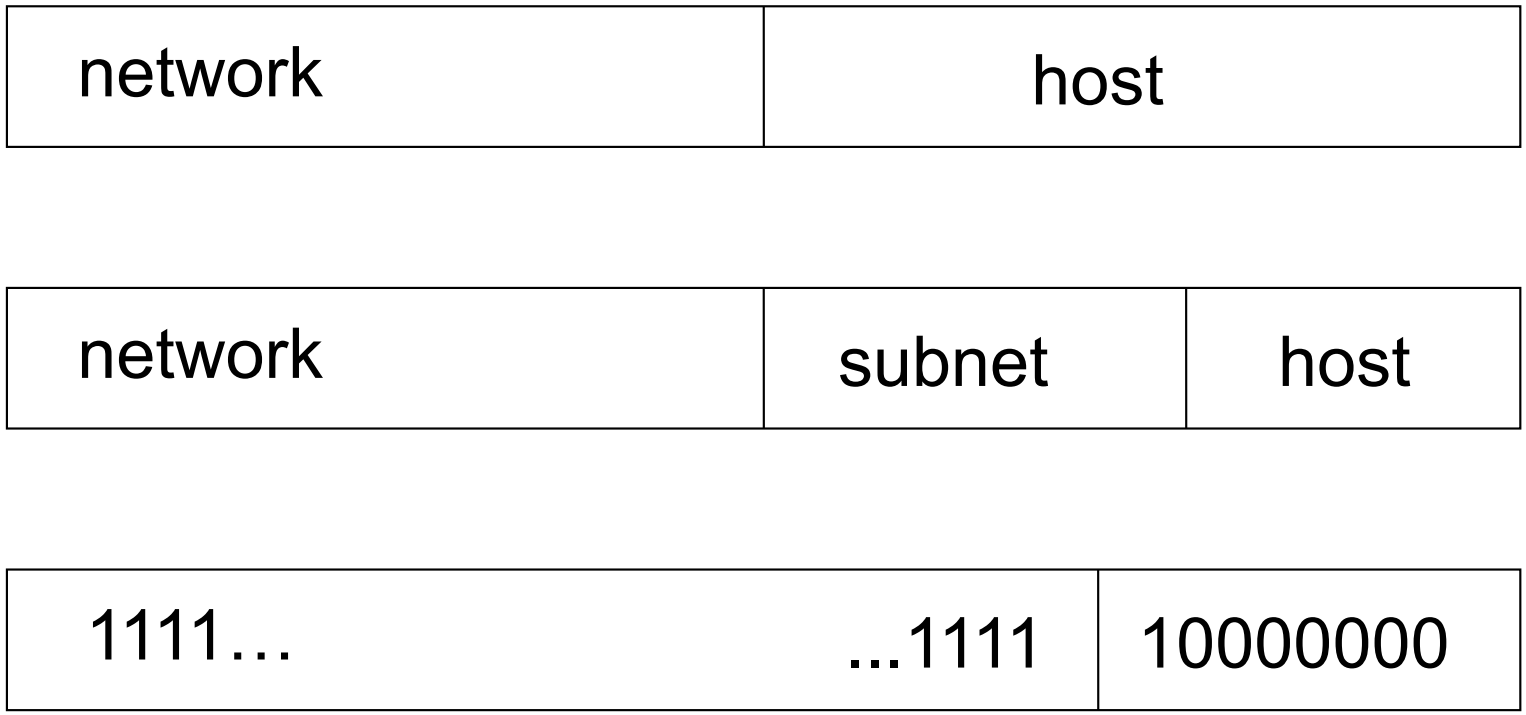
- Una università con un indirizzo di classe B: **150.100**
- Si assuma che ciascun dipartimento abbia meno di 100 host

- Quanti bit servono per identificare gli host di una sottorete?

**7**

- Qual è la network mask?
  - **11111111 11111111 11111111 10000000**
  - **255.255.255.128**

# Esempio di subnetting (*cont.*)



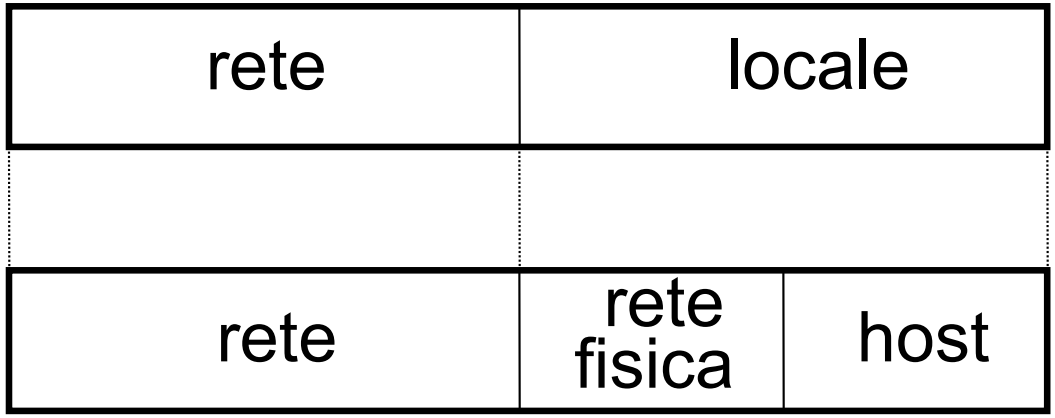
# Come usare le subnet mask per routing

- **Le subnet servono soprattutto per facilitare il routing dei pacchetti all'interno della rete amministrata**
- Si assuma, nel caso dell'università precedente, che arrivi un pacchetto con indirizzo destinazione: **150.100.12.176**
- Si effettua un AND tra l'indirizzo e la subnet mask
  - (150.100.12.176) *AND* (255.255.255.128)
  - Risultato: **150.100.12.128** che corrisponde alla sottorete di destinazione i cui host si trovano nel range

    150.100.12.**129** - 150.100.12.**254**

# Subnet addressing

Il **subnet addressing** modifica l'interpretazione degli indirizzi IP: l'indirizzo IP è composto da una porzione di **rete** e una **locale**



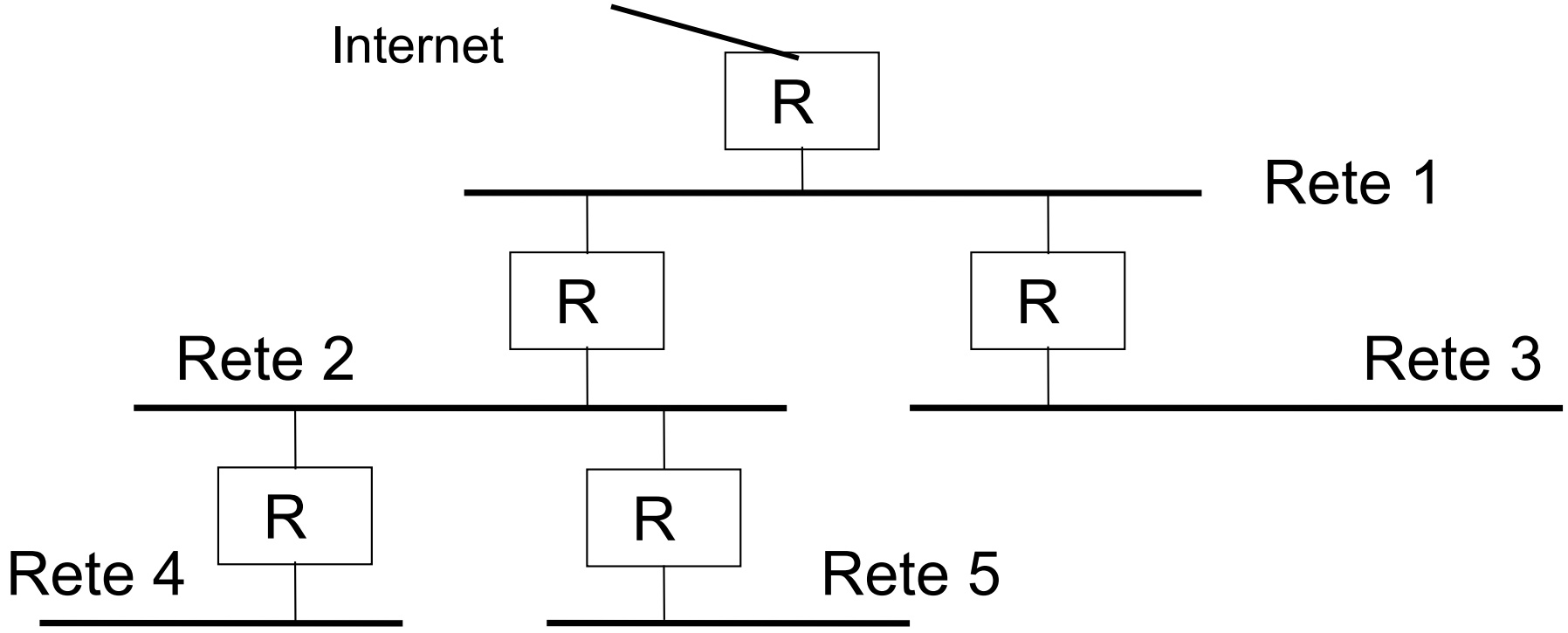

Risultato: *indirizzamento gerarchico* → **routing gerarchico**

Es. routing gerarchico: i router esterni usano i primi due byte dell'indirizzo IP per il routing, mentre il router della rete locale usa il terzo byte dell'indirizzo IP

# Esempio di subnet gerarchico

Esempio di rete con cinque reti fisiche suddivise in tre livelli:

- rete di classe B (16 bit per parte locale)
- 5 reti fisiche: occorrono 3 bit (essendo 5 < 2 = 8) per identificarle
- ad ognuna delle 5 reti fisiche è possibile collegare: 2 = 8192 host



# Supernet

- **PROBLEMA** → Esaurimento dello spazio di indirizzamento all'interno di una stessa classe di indirizzi (B o C)
- **SOLUZIONE**→ Un'organizzazione può richiedere più indirizzi della stessa classe per la sua rete. Es.
  - un blocco di indirizzi di classe C contigui viene assegnato a un'organizzazione
  - un blocco di indirizzi di classe B contigui viene assegnato a un Internet Service Provider
- **Come gestirli?**

# **Supernet** (*cont.*)

- **Gestione mediante supernet addressing**:
  - approccio opposto al subnet addressing
  - in pratica, si utilizzano meno bit di un intero byte per identificare il netid

- Formalmente, nei router si utilizza il meccanismo di **Classless Inter-Domain Routing** (**CIDR**) in cui:

  **(network address, count)**

  - **network address** è il più piccolo indirizzo (in bit) nel blocco di indirizzi di classe B o C assegnati
  - **count** è il numero di blocchi di indirizzi di classe B o C contigui

# Indirizzi privati (non routable)

# Indirizzi Pubblici e Privati (non routable)

- Gli indirizzi IP assegnabili (e quindi raggiungibili) sulla rete Internet sono detti **Pubblici**
  - L'utilizzo di un indirizzo IP pubblico deve essere **autorizzato** per evitare indirizzi duplicati (ne discutiamo fra poco – vedere «assegnazione indirizzi IP Pubblici»)
- Esistono però classi di indirizzi **IP privati** (anche detti **non routable**) che:
  - Possono essere utilizzati senza autorizzazione all'interno di reti private
  - Non possono essere utilizzati su Internet

# Indirizzi IP Non Routable

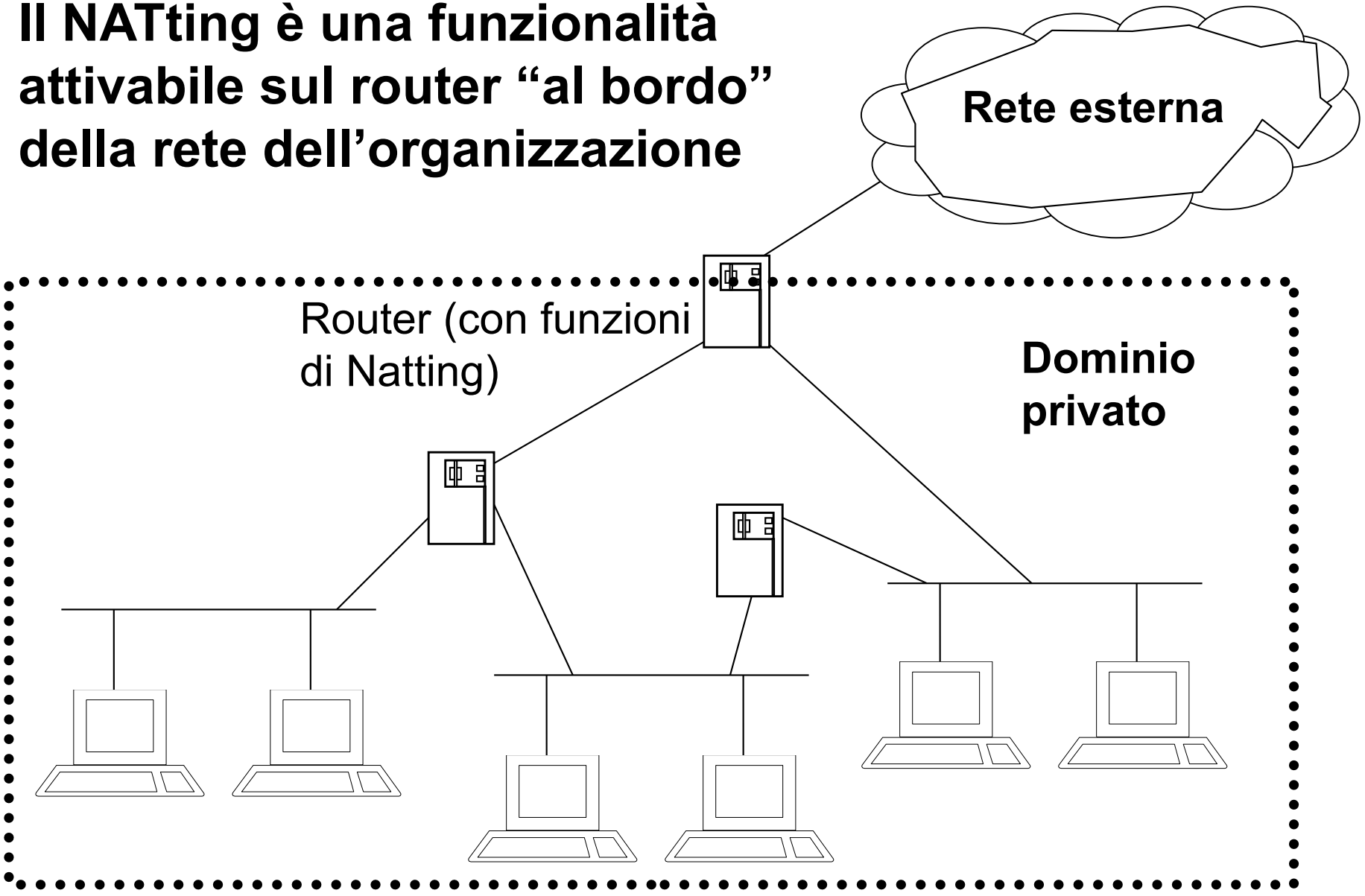
| Intervallo di indirizzi | |
|---|---|
| 10.0.0.0/8 | (10.0.0.0 - 10.255.255.255) |
| Default: Indirizzi di Classe A (1 Rete da 2^24 indirizzi) | |
| 172.16.0.0/12 | (172.16.0.0 - 172.31.255.255) |
| Default: Indirizzi di Classe B (2^4 Reti da 2^16 indirizzi) | |
| 192.168.0.0/16 | (192.168.0.0 - 192.168.255.255) |
| Default: Indirizzi di Classe C (2^8 Reti da 2^8 indirizzi) | |

# Reti private e semi-private

- Per alcune (poche) organizzazioni è importante avere **reti private in senso stretto**:
  - nessun nessun pacchetto esce da una rete privata e nessun pacchetto entra in una rete privata
  - indirizzi univoci solo all'interno della rete privata
- Per molte altre organizzazioni è importante avere **reti semi-private** con tre categorie di host:
  - nessun accesso da/a host fuori "dall'organizzazione" (molti host)
  - accesso parziale (host che possono raggiungere l'esterno ma non sono raggiungibili dall'esterno)
  - accesso completo (pochi host, es. server Web)

# *NATting* per reti semi-private

- **Il NATting è una funzionalità attivabile sul router "al bordo" della rete dell'organizzazione**



# Indirizzi IP privati per Intranet

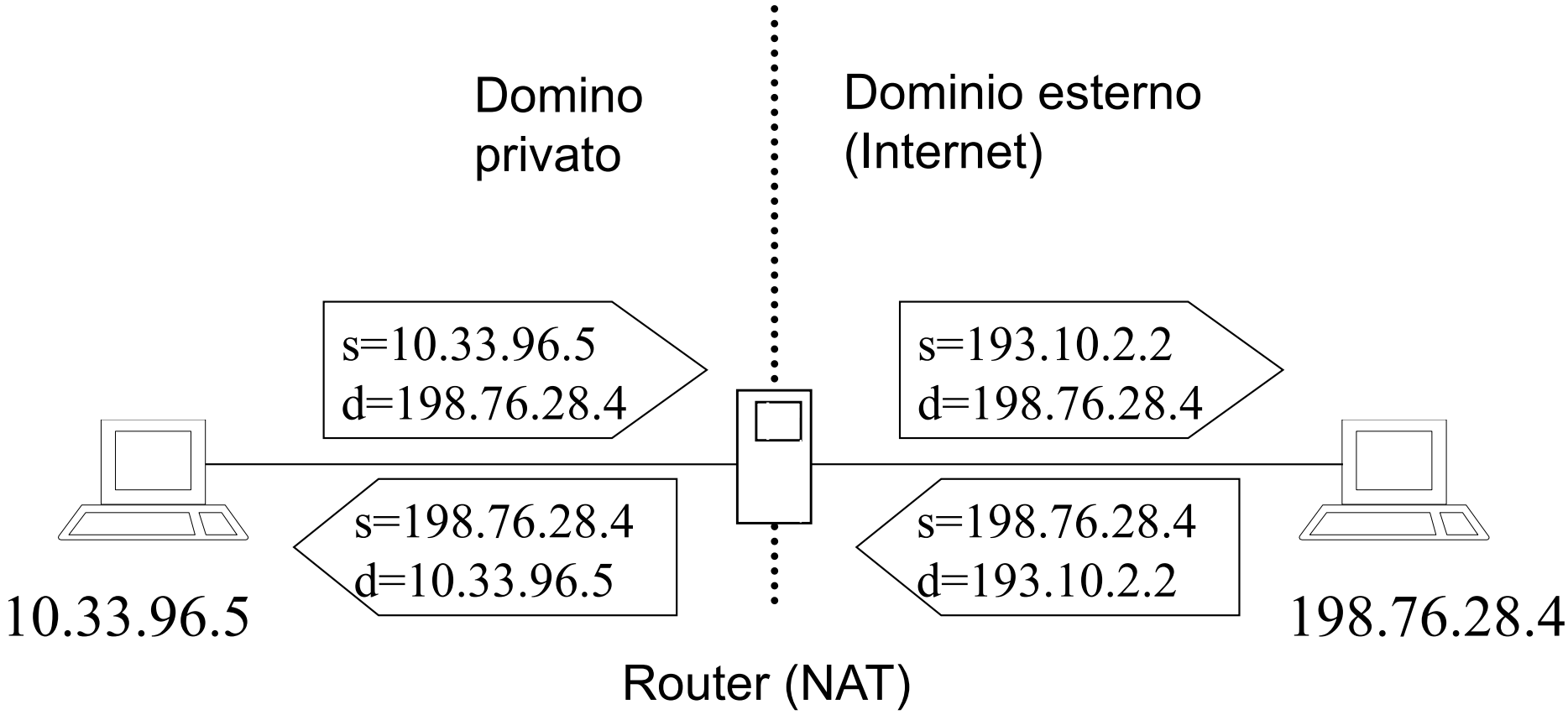
- In questo modo, un'organizzazione tipicamente ha la possibilità di progettare una rete che:
  - include host visibili da Internet (*host pubblici*)
  - altri host che non sono visibili (*host privati*)
- Gli *host privati* possono scambiare pacchetti:
  - solo con altri host privati all'interno della stessa rete senza intermediari
  - con host pubblici mediante:
    - **application gateway (*proxy*) sugli host pubblici**
    - **Network Address Translation (NAT)**

# NAT router

- Il *NAT router* (un router con funzionalità di NATting) si interpone tra la rete locale di una organizzazione e Internet con i seguenti compiti:
  - Mappa gli indirizzi IP tra due domini (interno-esterno)

    indirizzi locali ←→ indirizzi IP globali
  - Garantisce la trasparenza del routing tra gli *end system*
  - "Moltiplica" le possibilità di interconnessioni di host di una organizzazione (nel caso in cui l'organizzazione abbia a disposizione un numero di indirizzi IP inferiore al numero di host)
  - Aumenta la sicurezza evitando di rendere visibili all'esterno alcuni computer di una organizzazione

# Traduzione indirizzi



# *Natting*: contro

Svantaggi

- Distrugge la semantica della comunicazione *end-to-end* in quanto gli host interni non possono essere raggiunti dall'esterno
- Il router NAT modifica i pacchetti al volo:
  - qualche volta questo richiede modifiche a livello di informazioni application e non solo header del datagramma IP (es., indirizzo IP nel protocollo FTP)
  - È necessario usare dei gateway NAT box livello application

# *Natting*: pro

Vantaggi

- Distrugge la semantica della comunicazione *end-to-end* in quanto gli host interni non possono essere raggiunti dall'esterno

  → Ottima cosa per la SICUREZZA

- Soluzione economica, relativamente facile e veloce
- Consente massima flessibilità nella gestione interna degli indirizzi senza richiedere alcun permesso al proprio ISP …

# RFC per NAT

- **RFC 1631**

  **The IP Network Address Translator (NAT)**

  K. Egevang, P. Francis

  May 1994

- **RFC 2663**

  **IP Network Address Translator (NAT) Terminology and Considerations**

  P. Srisuresh, M. Holdrege

  August 1999

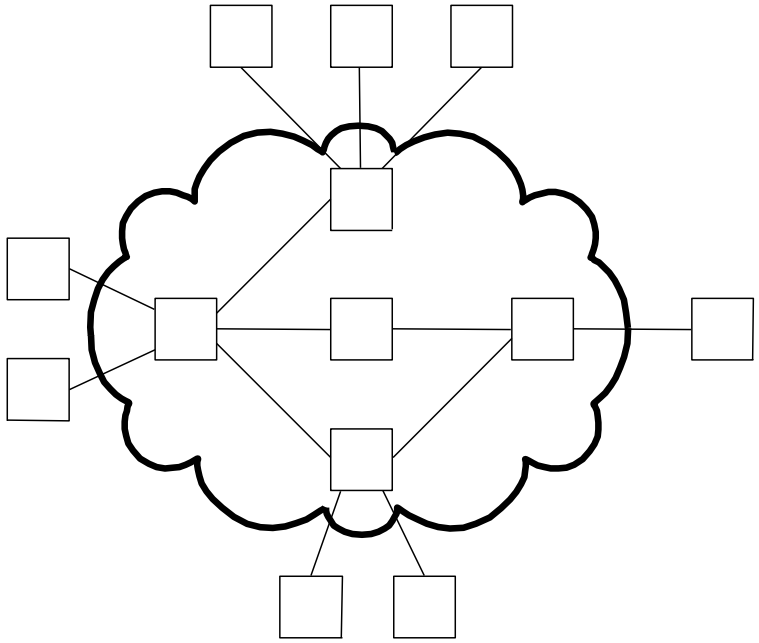
# NOTA

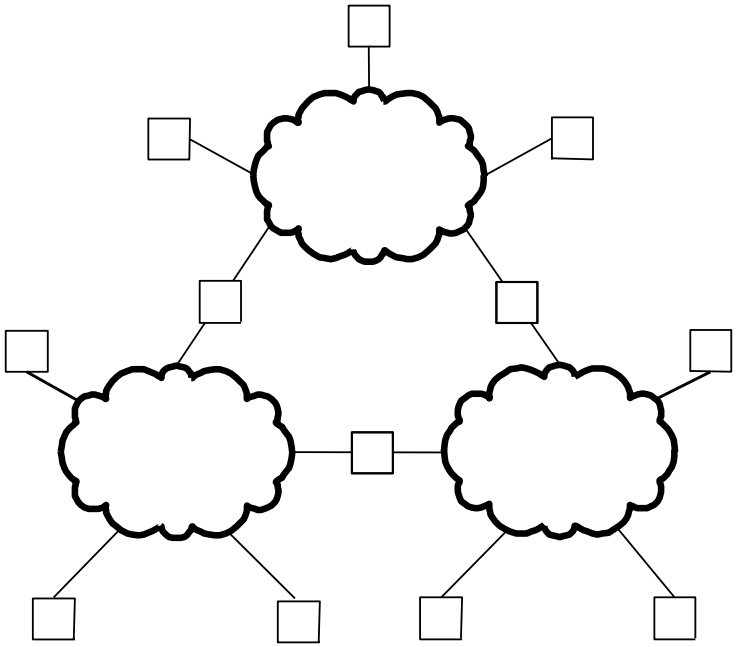
- Il NAT è solitamente associato alla trasformazione fra indirizzi pubblici e privati
- Può essere considerato un meccanismo generalizzato per mettere in comunicazione reti IP che utilizzano **spazi di indirizzamento IP separati**
  - Ad esempio, sono molto diffusi meccanismi NAT fra più reti private

# Concetti di instradamento dei pacchetti IP

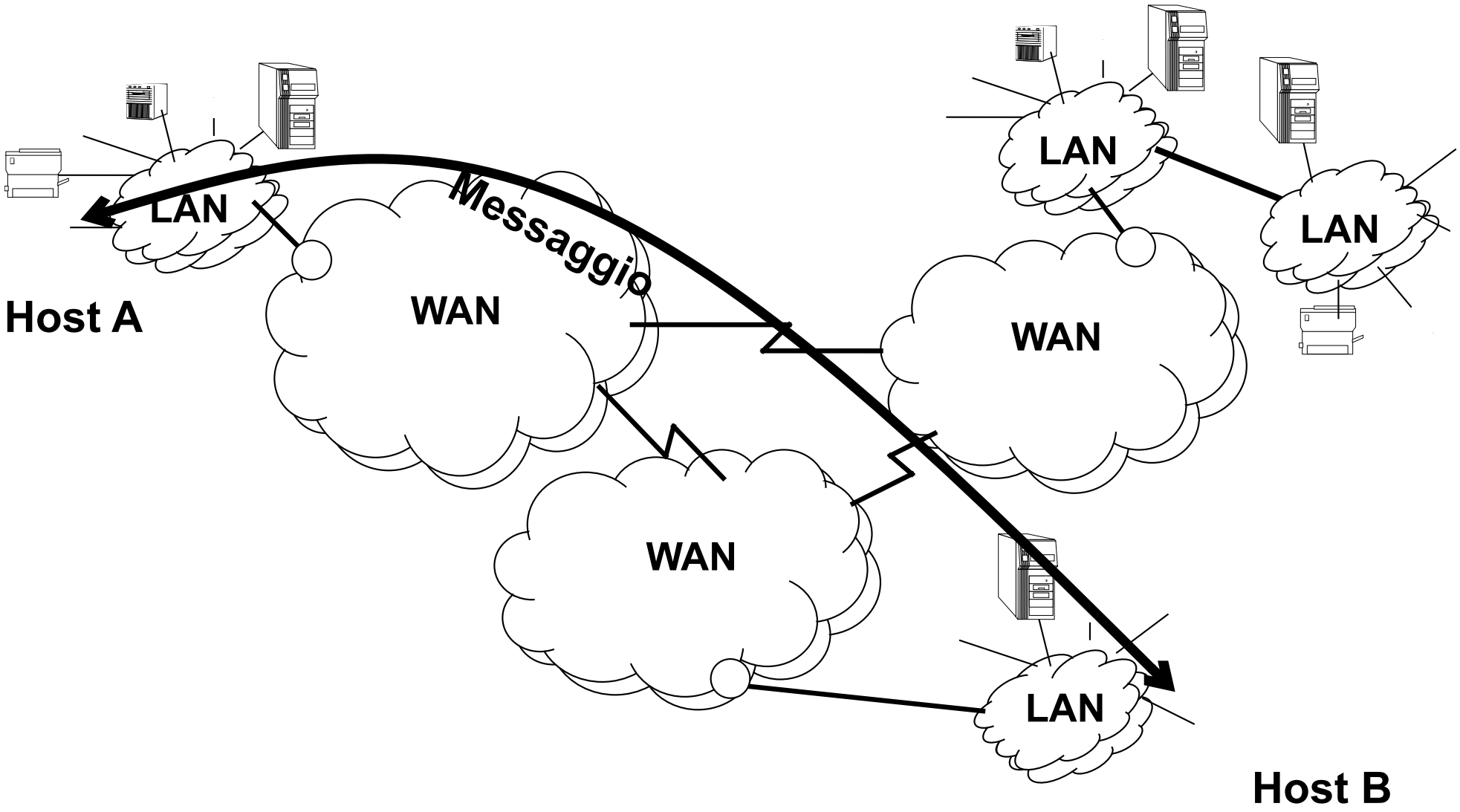
# Modellazione ideale reti

- Una rete può essere definita ricorsivamente
  - **Due o più nodi connessi tramite collegamenti**
  - **Due o più reti connesse tramite nodi**





# Comunicazione logica tra due host

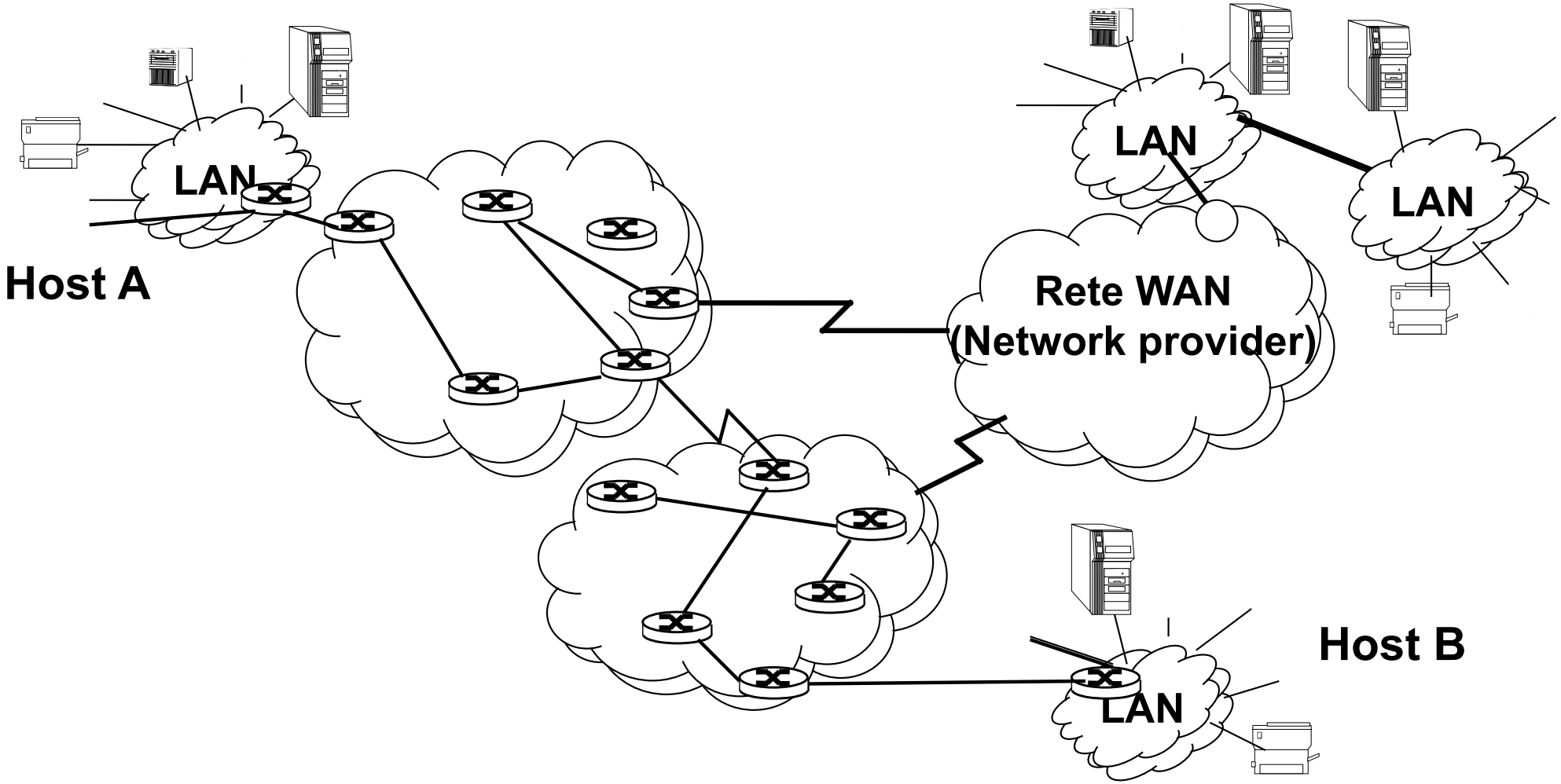



## Logicamente comunicano i due host terminali

# Comunicazione reale

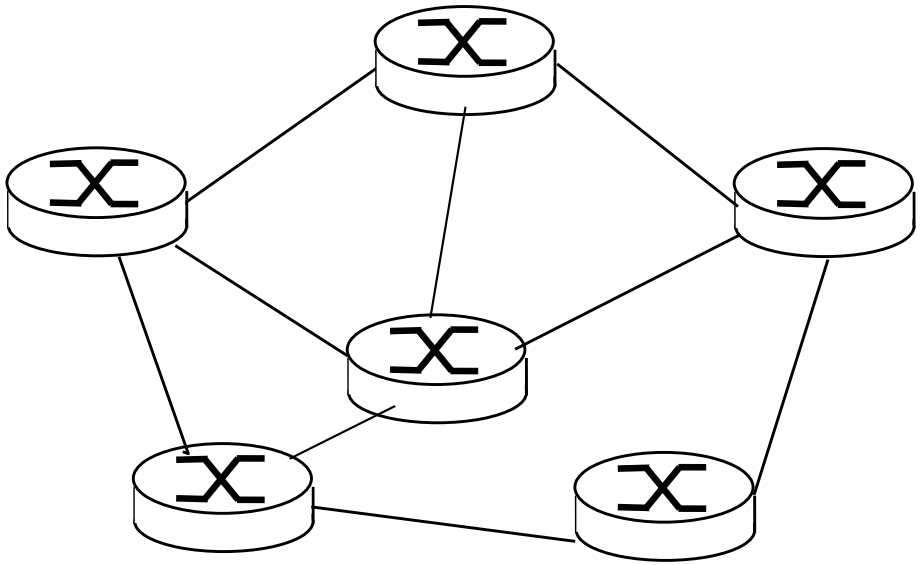
**In realtà, le informazioni attraversano tutti i nodi e i collegamenti**

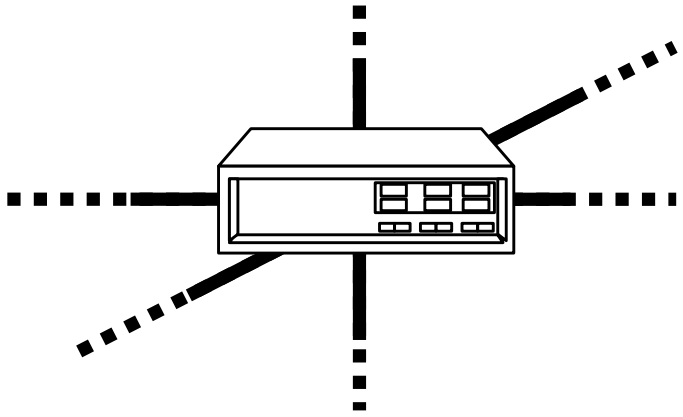
- **Problemi di *instradamento* e di *condivisione delle risorse***



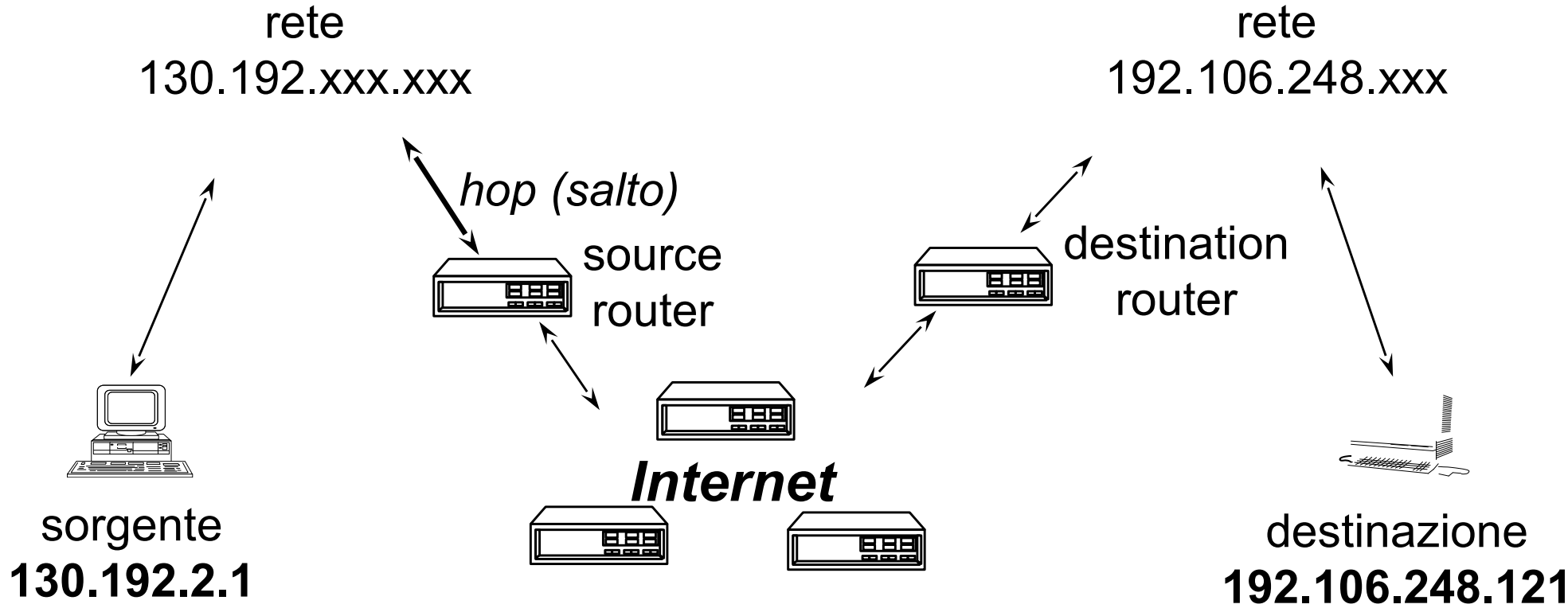
# Router

**Il router deve risolvere un problema molto ben definito: Instradare i pacchetti nella rete da un qualsiasi host ad un qualsiasi altro host, sulla base dell'indirizzo IP destinazione incluso nel pacchetto stesso**





# Routing IP




- **I router si passano i pacchetti hop-by-hop: nessuno decide il percorso complessivo, ma solo il router successivo**
- A volte il routing non ha successo perché i **router sovraccarichi** scartano pacchetti (***congestione***, ***limite fisico***) o vi possono errori di routing (***errore logico, ad esempio cicli nella rete***)

# Inoltro hop-by-hop dei pacchetti IP

- Un host che invia un pacchetto all'**esterno della propria rete locale** deve decidere **tramite router inviarlo**: questo router viene detto ***first hop router*** o ***source router***

- Ogni router deve decidere a sua volta il router (***next-hop router***) a cui inoltrare il pacchetto

- Infine, il pacchetto dovrebbe arrivare all'host destinazione
  - Se il pacchetto percorre troppi router, potrebbe essere scartato (si vedrà il TTL nell'header IP)
  - L'ultimo router prima dell'arrivo del pacchetto a destinazione è anche detto ***destination router***

# Problema del routing

- Consegna i pacchetti da un host sorgente a uno destinazione potenzialmente attraversando molteplici router intermedi
  - in modo best effort, privo di connessione, e quindi non garantito

- Quando un problema è complesso si suddivide in sottoproblemi più semplici:
  - **Sottoproblema 1**: ad ogni pacchetto in ingresso, determinare il link di uscita in modo che il pacchetto si avvicini alla destinazione (**IP forwarding**)
  - **Sottoproblema 2**: mantenere **informazioni aggiornate** per risolvere il sottoproblema 1 (**protocollo di routing**)

# IP Forwarding

- **IP forwarding (inoltro)**: meccanismo con cui un router trasferisce i datagram da un'interfaccia d'ingresso a quella in uscita
- Effettuato da ogni router
- Il **next-hop router** appartiene a una rete alla quale il **router è collegato a livello H2N**

Per inoltrare i pacchetti:

- l'indirizzo di destinazione viene estratto dall'header del datagram *(prossime slide)*
- l'indirizzo di destinazione è usato come indice nella **tabella di routing** *(prossime lezioni)*

# Caratteristiche dell'IP forwarding

- **Indipendenza dal mittente**: il next-hop routing, tipicamente, non dipende dal mittente del pacchetto o dal cammino che il pacchetto ha attraversato fino a quel momento
  - **Il router estrae dal pacchetto soltanto l'indirizzo del destinatario**

- La tabella di routing deve contenere un next-hop router per ciascuna destinazione

- **Il next-hop router appartiene a una rete alla quale il router è collegato direttamente**

# Tabella di routing [1]

- **Ogni host e ogni router** hanno una **tabella di routing** in cui ciascuna riga fornisce il ***next-hop*** per **ogni possibile destinazione**
  - Il percorso dei pacchetti viene selezionato **hop-by-hop**

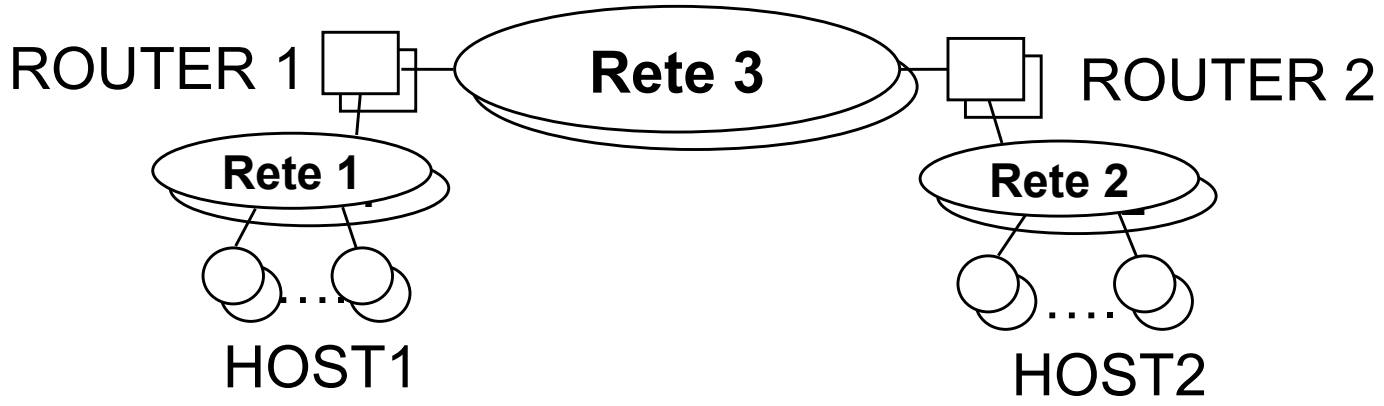



**Tabella di routing host Rete 1**

| Destinazione | Metodo |
|---|---|
| Rete 1 | H2N |
| Rete 2 | IP tramite Router 1 |
| Rete 3 | **IP tramite Router 1** |

**Tabella di routing host Rete 2**

| Destinazione | Metodo |
|---|---|
| Rete 1 | IP tramite Router 2 |
| Rete 2 | H2N |
| Rete 3 | IP tramite Router 2 |

# Tabella di routing [2]

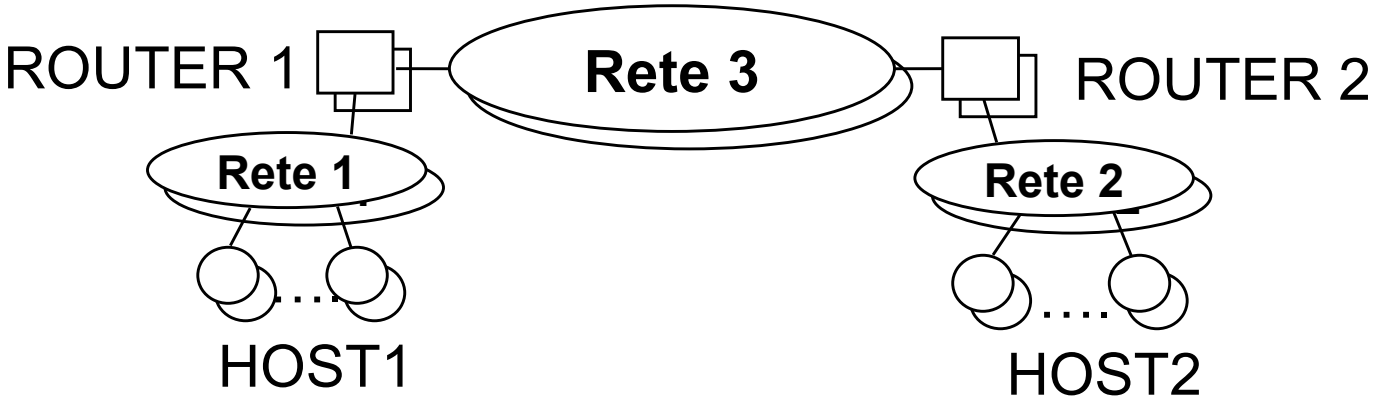



**Tabella di routing host Rete 3**

| Destinazione | Metodo |
|---|---|
| Rete 1 | IP tramite Router 1 |
| Rete 2 | IP tramite Router 2 |
| Rete 3 | **H2N** |

**Tabella di routing host Router 1**

| Destinazione | Metodo |
|---|---|
| Rete 1 | IP tramite Router 1 |
| Rete 2 | IP tramite Router 2 |
| Rete 3 | **H2N** |

**Tabella di routing host Router 2**

| Destinazione | Metodo |
|---|---|
| Rete 1 | IP tramite Router 1 |
| Rete 2 | H2N |
| Rete 3 | **H2N** |

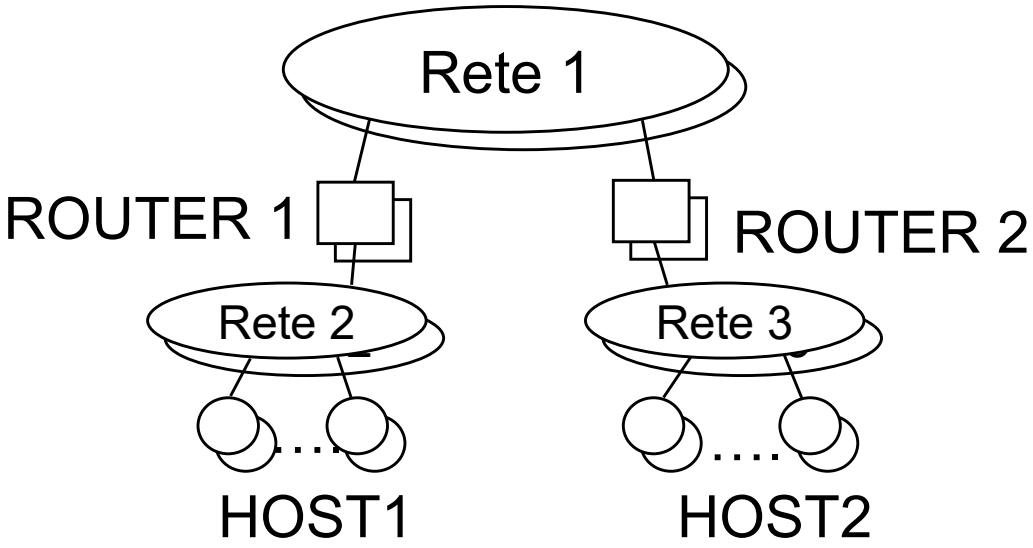
# Funzionamento del router

**Ogni router l'indirizzo IP di destinazione dall'header IP e consulta la tabella di routing per determinare:**

1. Se l'indirizzo appartiene a una **rete nota a cui il router è connesso a livello H2N**, viene usato il protocollo H2N per inviare il pacchetto a D
   - Se Ethernet, risoluzione indirizzo HW con ARP e costruzione frame verso D)
2. Se l'indirizzo appartiene a una **rete nota a cui il router non è connesso a livello H2N**, nella tabella di routing è presente il **next-hop router** a cui inviare il pacchetto
   - Comunico a **livello H2N con il next-router**, e a livello IP con D
3. Se l'indirizzo non appartiene ad alcuna rete nota, ma esiste un **router di default (default gateway)**, si invia il pacchetto al router in modo analogo a come descritto nel punto 2
4. Altrimenti, non invio il pacchetto (tipico errore **network unreachable**)

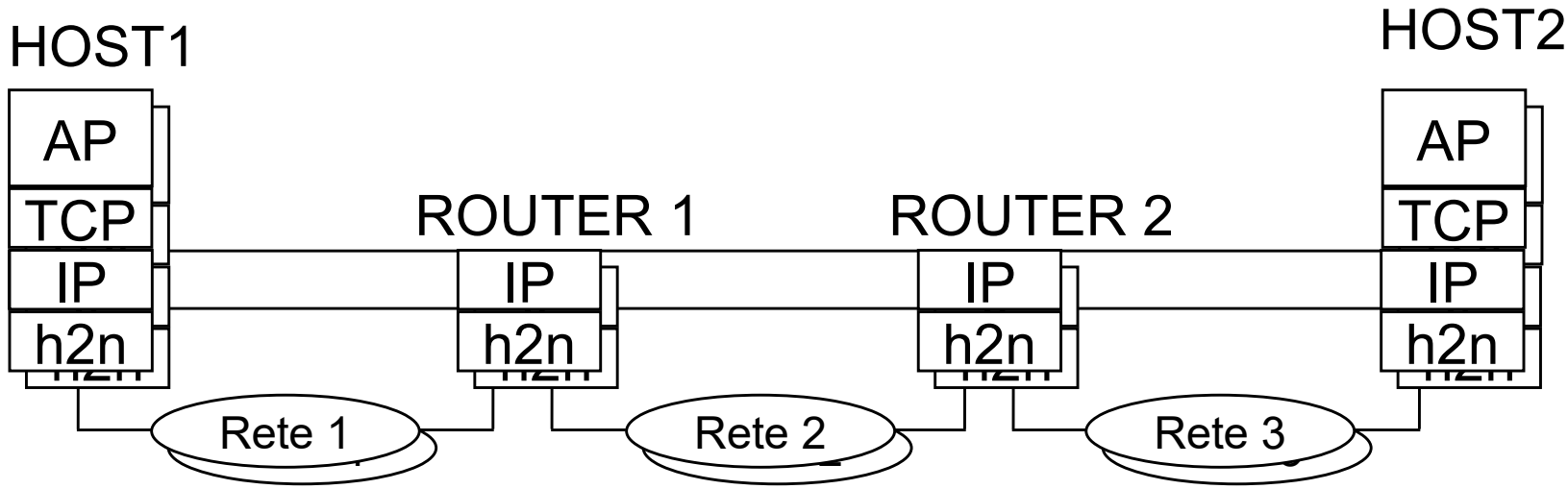
**Nota:** nel caso nella tabella di routing esistano più reti a cui la destinazione può far riferimento, si usa la regola del "prefisso più lungo" (longest prefix) – discusso in laboratorio

# Distinguere i due casi fondamentali




**Host mittente e destinatario sono nella stessa sottorete**

**Host mittente e destinatario sono in sottoreti differenti**



# Dimensioni tabella di routing

- Le dimensioni (crescenti) delle tabelle di routing potrebbero essere un limite allo sviluppo di Internet

- Per questo si sfruttano **tecniche di aggregazione** per fare in modo che ogni riga possa "catturare" molte destinazioni
  - Essenziale **progettare le reti IP assegnando opportunamente gli indirizzi IP** (e.g., indirizzi adiacenti per una stessa rete locale)
  - Essenziale utilizzare **indirizzamento gerarchico** (e non piatto)

- Studieremo questi aspetti nelle esercitazioni

# Protocolli di routing

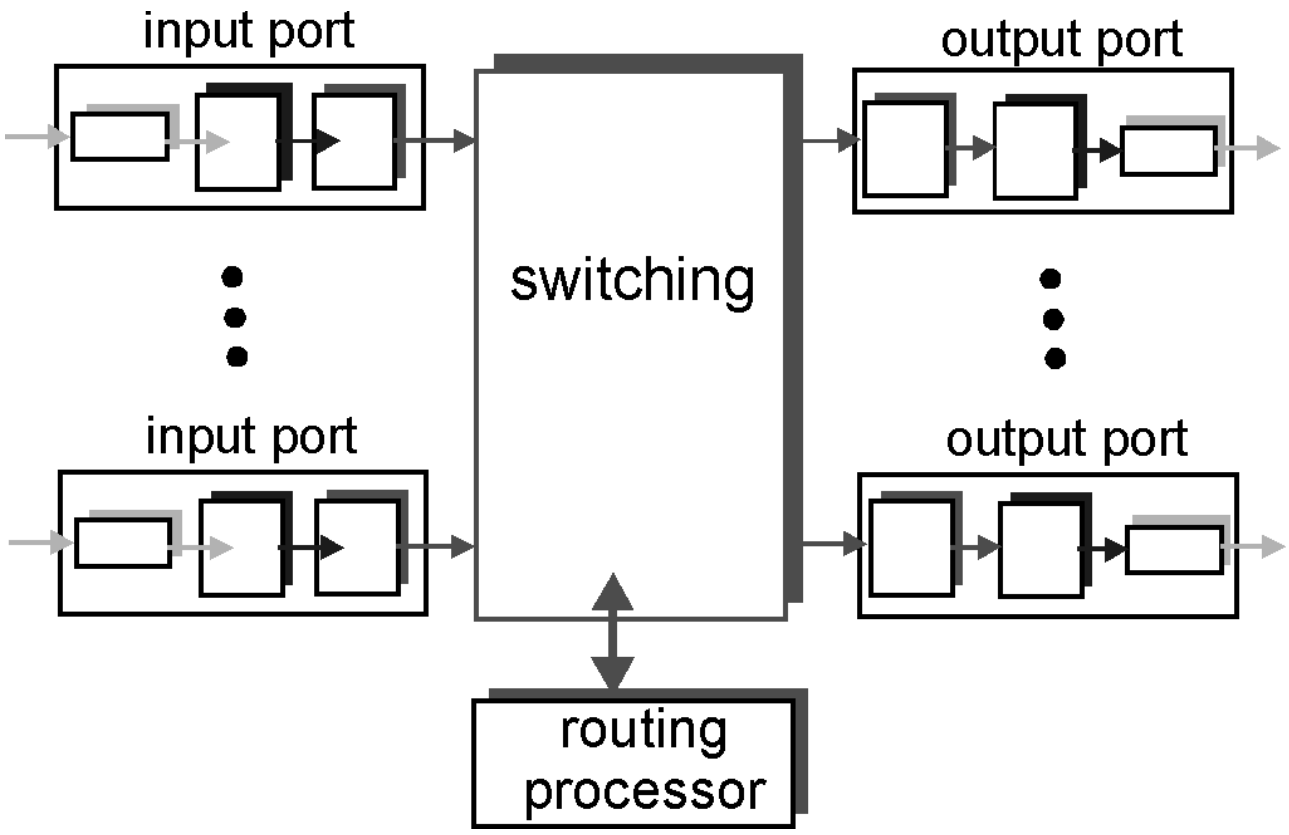
- In contesti semplici, la tabella di routing può essere definita in maniera **statica** da un amministratore di rete o da un protocollo di configurazione (e.g., DHCP)
  - Ad esempio, tabella di un host o router di reti locali molto semplici
- I **protocolli di routing** (e.g., **RIP**, **OSPF**, **BGP**) servono invece a costruire dinamicamente le tabelle di routing presenti sui router
  - Capacità di popolare in modo «ottimale» la tabella in base dalla topologia della rete e delle sue caratteristiche
  - Capacità di **adattamento** a fronte di **guasti** o, potenzialmente, **congestioni**
  - In reti mediamente complesse, l'intervento manuale umano per definire e aggiornare configurazioni statiche non è accettabile
- **Approfondiremo in seguito**

# Cenni di architetture di router

# Cenni di architetture di router [1]

4 componenti fondamentali nell'architettura di un router:

- porta di ingresso
- commutatore
- processore di routing
- porta di uscita




Funzionalità di:

- livello 1 (fisico)
- livello 2
- livello 3 (IP)

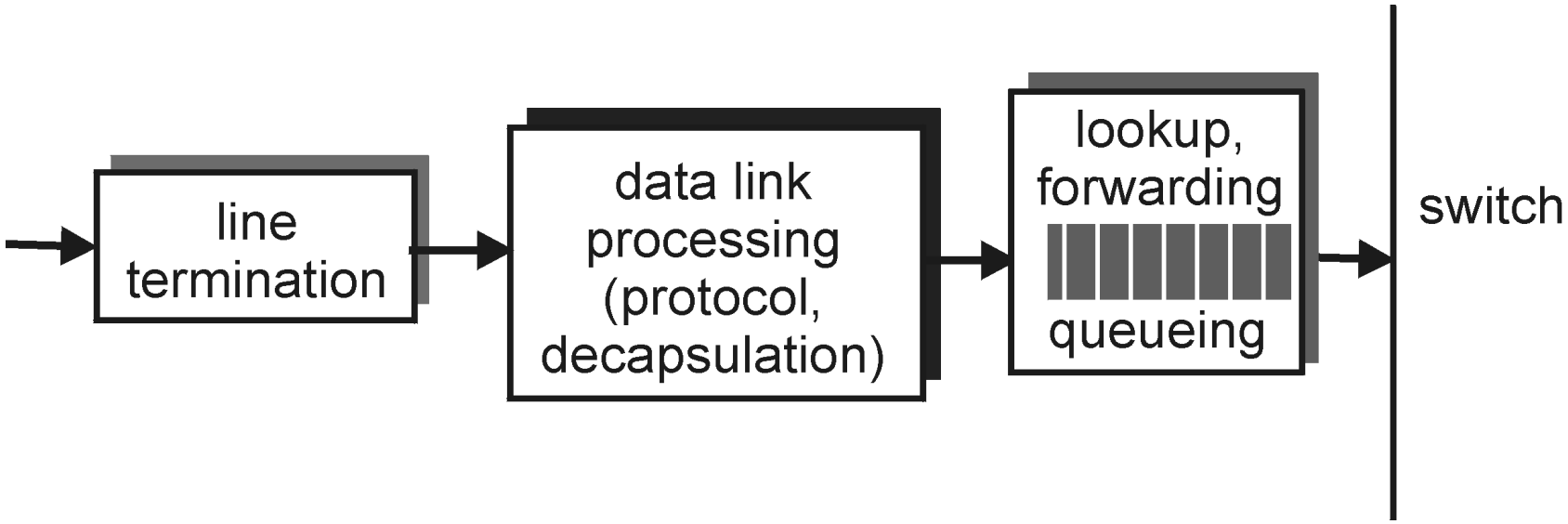
# Cenni di architetture di router [2]

**Porta di ingresso**:

- funzioni del livello 1
- funzioni del livello 2

} associate a un singolo link di ingresso

- funzioni del livello 3 → funzioni di ricerca e forwarding della porta di uscita; ottimizzazione della ricerca nella tabella di routing

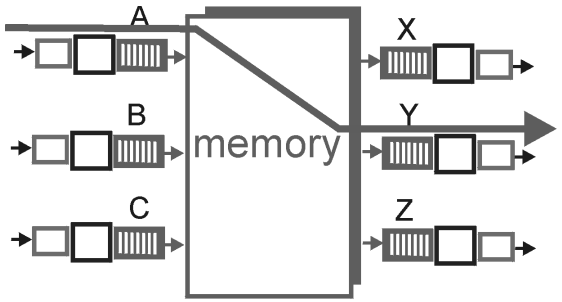


# Cenni di architetture di router [3]

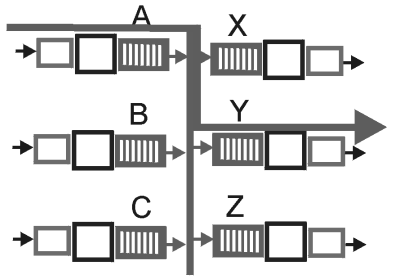
## Componenti di switching

**FUNZIONE**: spostamento del pacchetto dalla porta di ingresso a quella di uscita “opportuna”

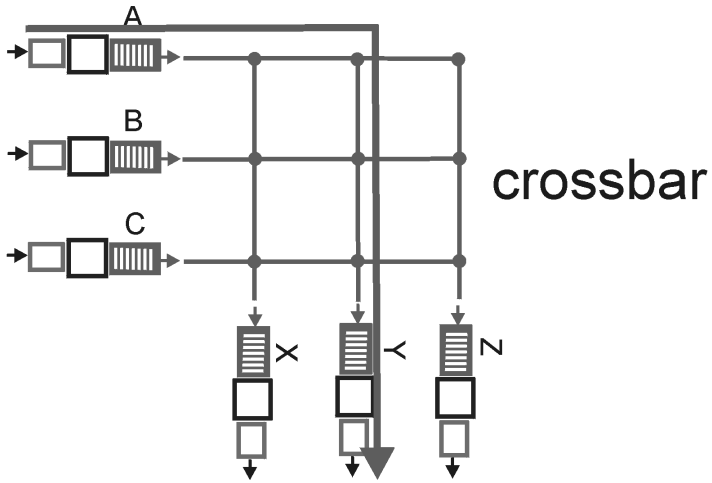
**TECNICHE**: Commutazione basata su switch, bus o rete di interconnessione crossbar




switch




bus




crossbar

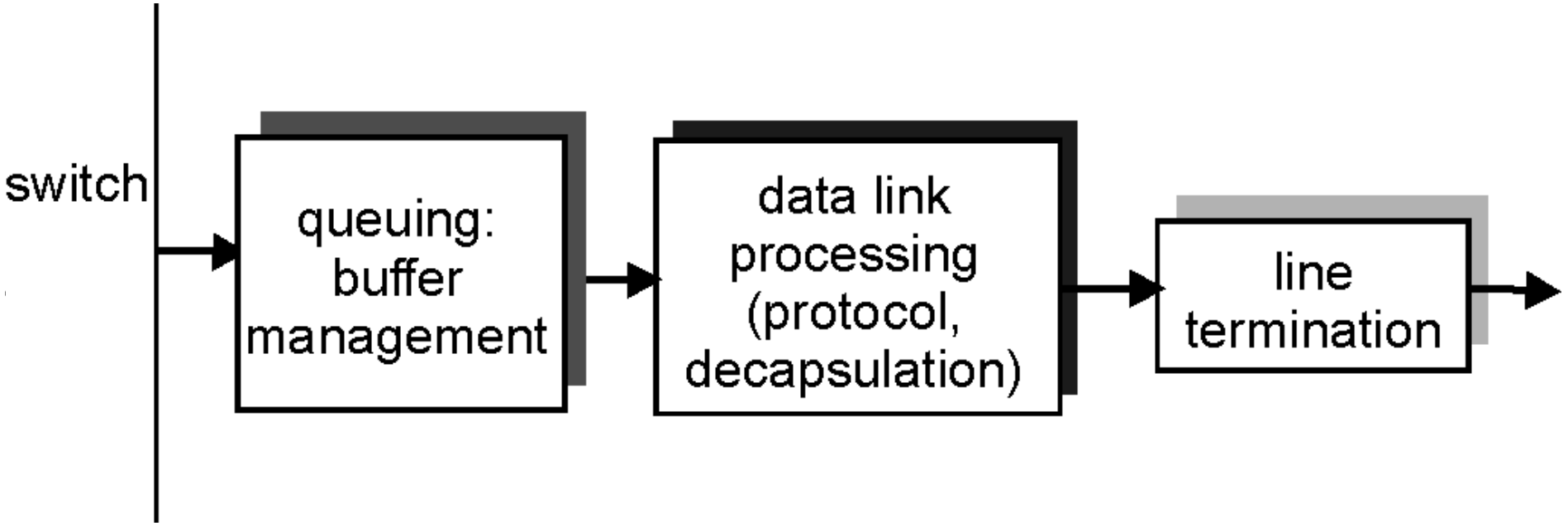
# Cenni di architetture di router [4]

**Porta di uscita**:

- funzioni del livello 1
- funzioni del livello 2

} associate a un singolo link di uscita

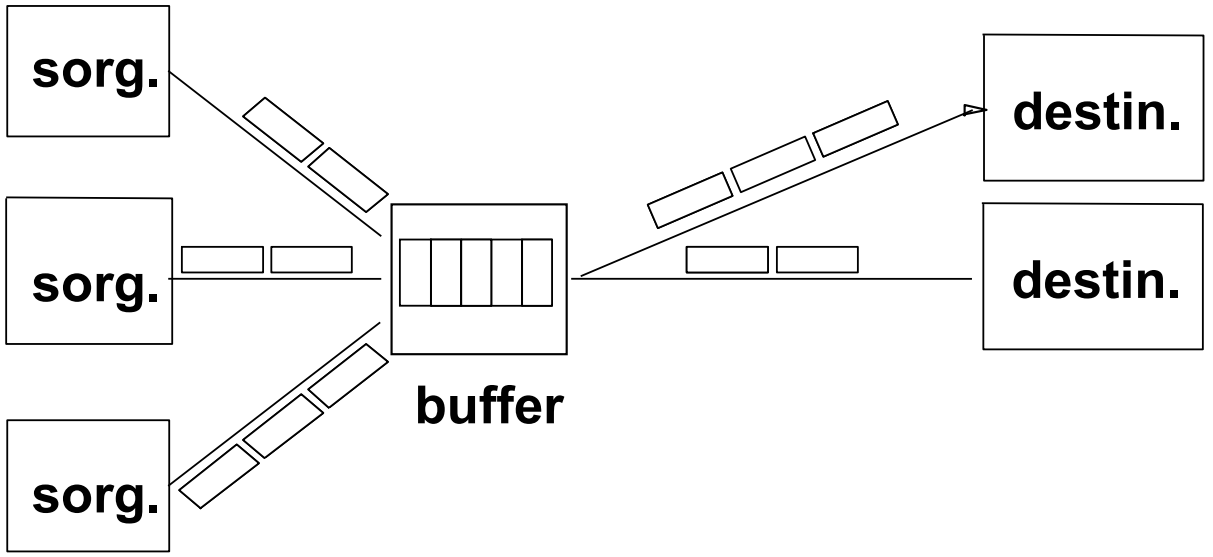
- funzioni del livello 3 → funzioni di gestione della coda e del buffer di uscita (la velocità con cui il commutatore consegna i pacchetti deve essere superiore alla capacità del link di uscita)



# Gestione del conflitto

- Si bufferizzano i pacchetti in conflitto per lo stesso link
- Il buffer determina in pratica una coda di pacchetti che può essere processata in ordine **FIFO** (**First-In-First-Out**), ma non necessariamente (es., in base alla priorità)

➔ **Congestione = riempimento del buffer**



# Trasmissioni e conflitti nel packet switching

**Comunicazione *store and forward***:

(i pacchetti si muovono di un hop alla volta)

1. trasmessi su un link, arrivano ad un router
2. aspettano (presso il router), il loro turno per poter essere trasmessi sul successivo

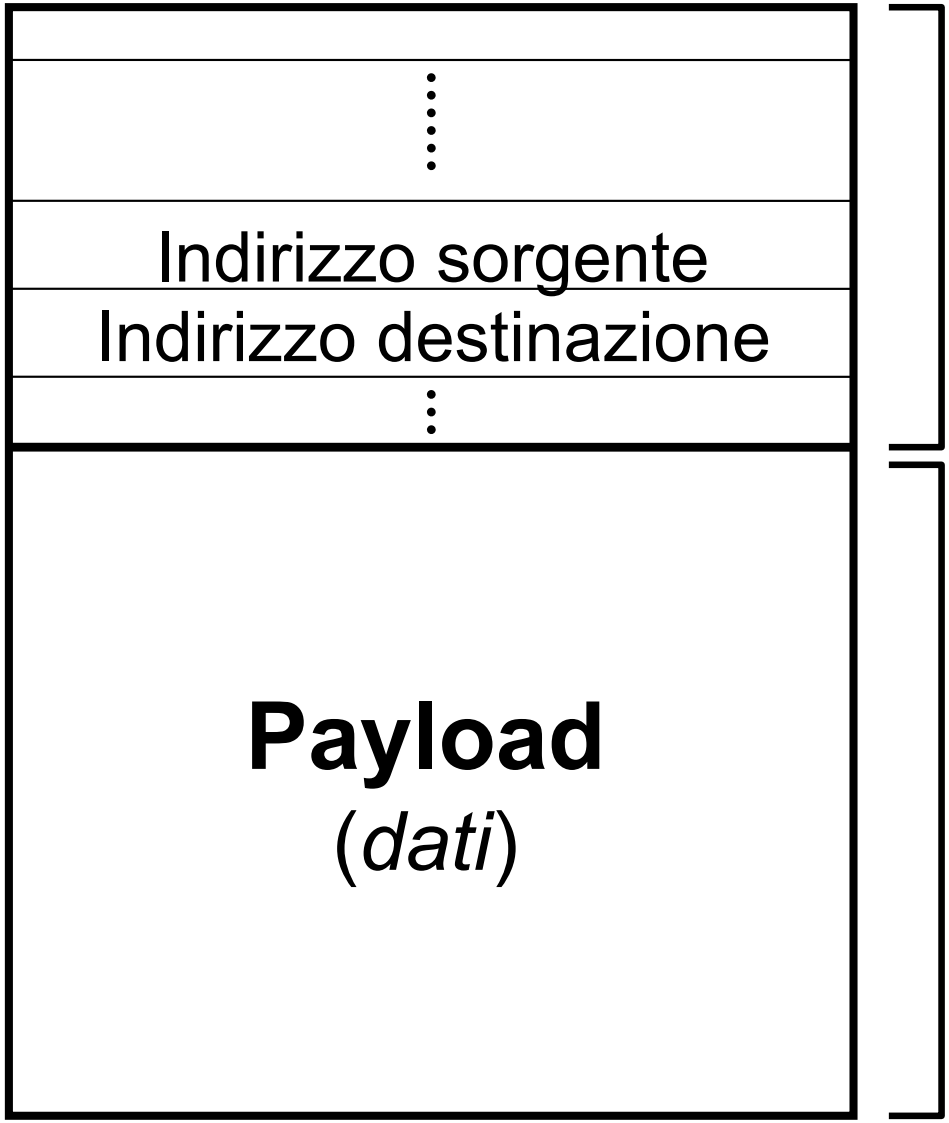
## Conflitto di risorse

- La domanda aggregata di risorse può eccedere la quantità disponibile
- Non essendoci prenotazione, si possono creare **congestioni** (***impreviste***):
  - i pacchetti rimangono accodati (se c'è spazio) in attesa di poter utilizzare il link
  - Se la coda è piena, il pacchetto viene perduto (senza avvisi!)
- Possibilità di utilizzare un link differente a seconda dello stato della rete

# Formato del datagramma IP (IPv4)

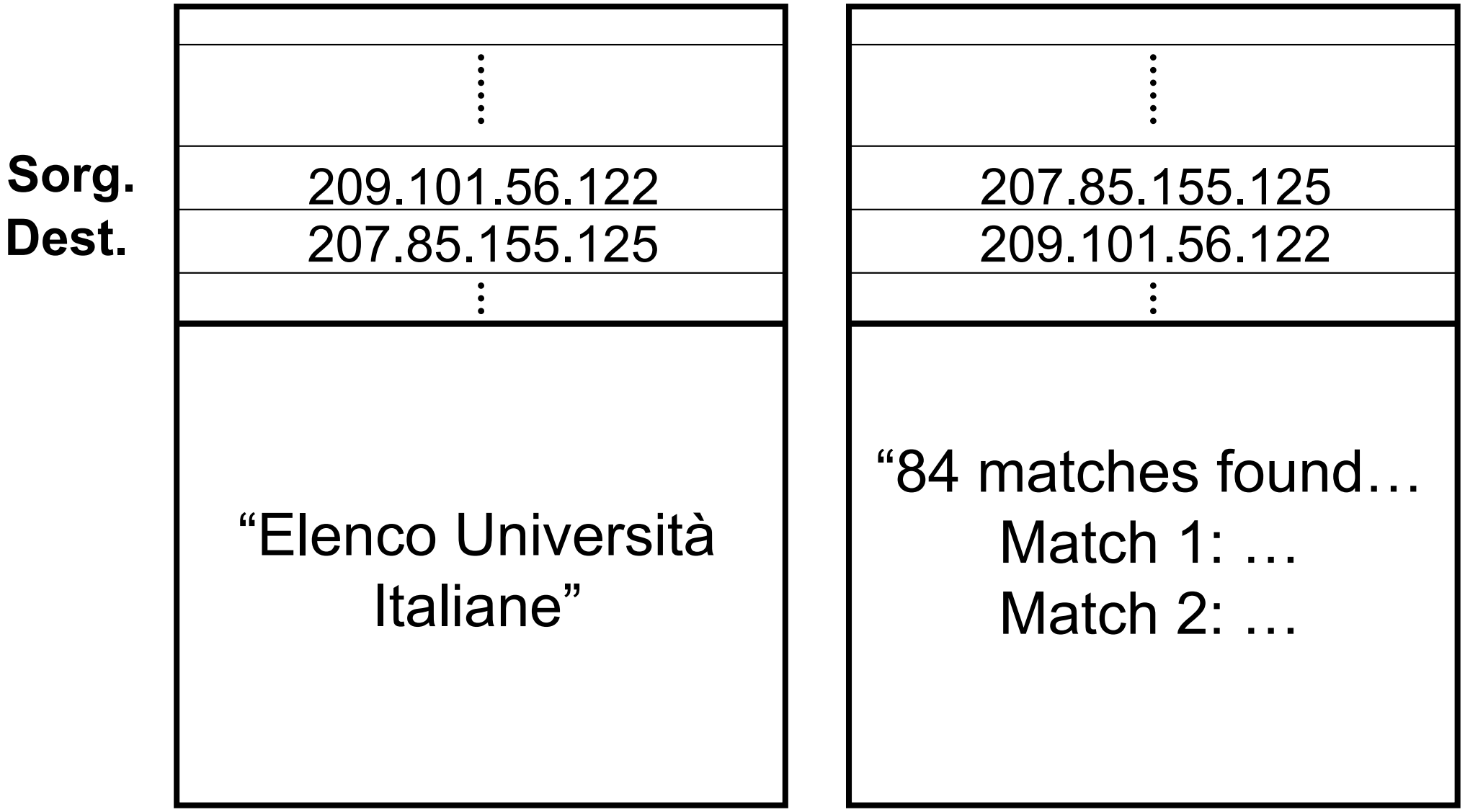
# Unità di trasferimento dati: *datagram*

## Layout dell'Internet datagram (IP datagram)

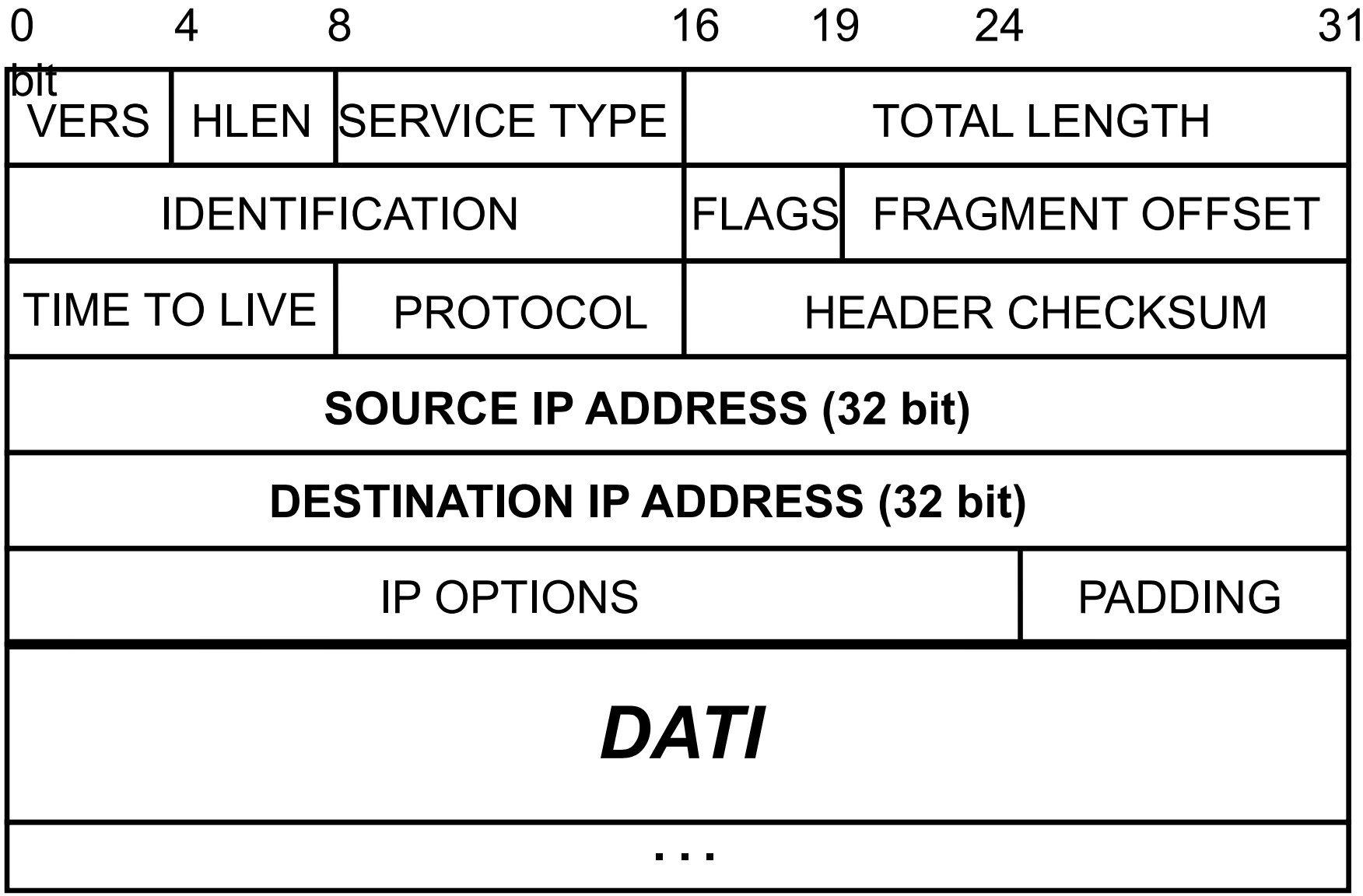



**Tutto il traffico Internet consiste di pacchetti. Ciascun pacchetto è lungo fino a 64 Kbyte**

# Esempi di datagram



# Formato del datagram IP

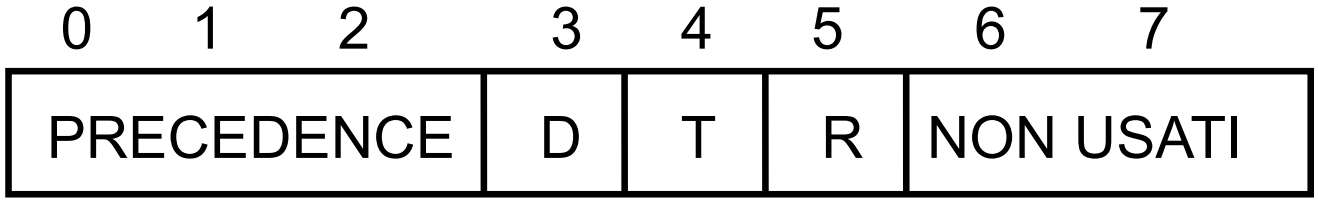


# Analisi header del datagram IP (1)

- **VERS**: versione del protocollo IP usata per creare il datagram (4 bit)
- **HLEN**: lunghezza dell'header del datagram (in parole di 32 bit); in generale uguale a 5 (20 byte)
- **TOTAL LENGTH**: lunghezza del datagram IP (in byte); max dimensione 2 = 65536 byte (64 Kbyte)
- **TYPE OF SERVICE (TOS)**: campo il cui scopo è stato modificato negli anni
  - Impiego originale: includere informazioni per la gestione differenziata dei pacchetti in base a requisiti applicativi (e.g., bassa latenza, alto throughput)
  - Attuale: uso misto per funzionalità legate a concetti di *classe di traffico* e *segnalazione esplicita di congestione*
    - protocolli definiti nelle RFC 2474 e 3168 del 1998 e 2001, implementazioni reali successive e sperimentazioni in alcuni casi ancora in corso

# Type of Service: impiego originale

- **TYPE OF SERVICE (TOS)**: campo utilizzato per scopi differenti negli anni
  - Impiego originale: includere informazioni per la gestione differenziata dei pacchetti in base a requisiti applicativi (e.g., bassa latenza, alto throughput)




**PRECEDENCE**: specifica l'importanza del datagram

**D** (delay): basso ritardo

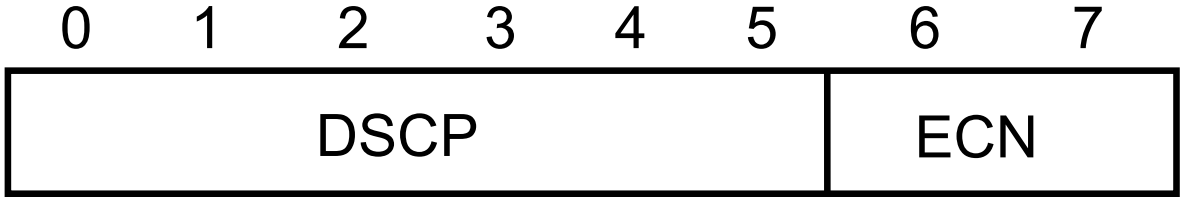
**T** (throughput): alto throughput

**R** (reliability): alta affidabilità

} tipo di trasporto desiderato

# Type of Service: impiego attuale

- **TYPE OF SERVICE (TOS)**: campo utilizzato per scopi differenti negli anni
  - Attuale: uso misto per funzionalità legate a concetti di *classe di traffico* e *segnalazione esplicita di congestione*




**DSCP**: Code Points for Differentiated Services

Codice che identifica classi di servizio

- Simile al TOS precedente, ma cambia l'interpretazione dei valori e non ci sono «bit» specifici legati a un particolare requisito
- Un router può ignorarlo: possibilmente utilizzato fra i router di una stessa organizzazione, solitamente ignorato da router di altre organizzazioni

**ECN**: Explicit Congestion Notification

- Meccanismo opzionale per permettere a un router di segnalare congestione prima di iniziare a «droppare» pacchetti (discussione su congestione in TCP)

# Analisi header del datagram IP (3)

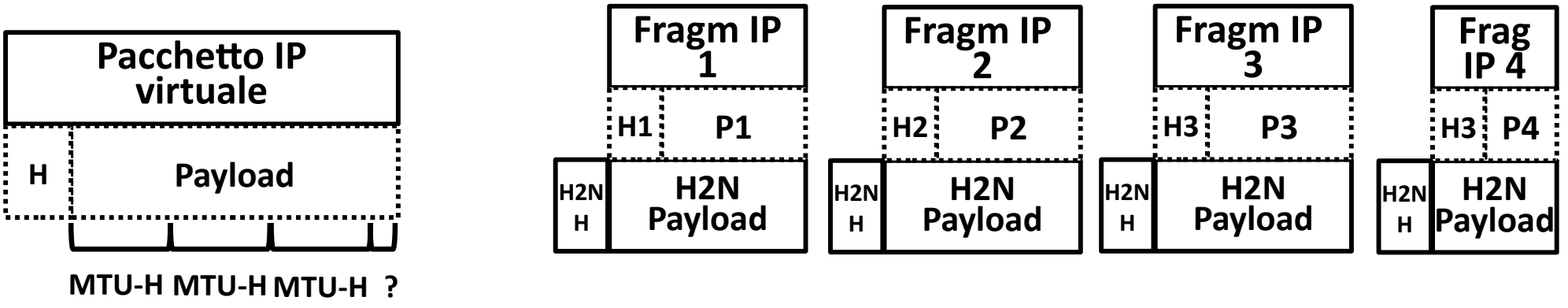
- I successivi tre campi dell'header del datagram (denotati in figura come *identification*, *flags*, *fragment offset*) servono per gestire, quando si rende necessaria, a livello H2N, la ***frammentazione*** e la ***ricostruzione*** del datagram
  - **IDENTIFICATION**: intero che identifica il datagram
  - **FLAGS**: controllo della frammentazione
  - **FRAGMENT OFFSET**: la posizione del frammento nel datagram originale
- Nei sistemi operativi moderni vengono anche supportati protocolli di Path MTU discovery per permettere agli host della rete di apprendere le dimensioni più opportune di frammentazione

# Analisi header del datagram IP (4)

- **TIME TO LIVE**: non è un vero valore temporale! Indica per quanto tempo il datagram può circolare in Internet. E' decrementato da ciascun router che gestisce il datagram: quando diviene uguale a 0, è eliminato dal router corrispondente
- **PROTOCOL**: indica quale protocollo applicativo può utilizzare i dati contenuti nel datagram
- **HEADER CHECKSUM**: serve per controllare l'integrità dei dati trasportati nell'header
- **SOURCE IP ADDRESS**: indirizzo IP (32 bit) del mittente del datagram
- **DESTINATION IP ADDRESS**: indirizzo IP (32 bit) del destinatario del datagram
- **IP OPTIONS**: campo opzionale di lunghezza variabile; serve per il testing ed il debugging della rete
- **PADDING**: campo opzionale che serve per fare in modo che l'header abbia lunghezza multipla di 32 bit (*byte stuffing*); è presente soltanto se il campo IP OPTIONS denota una lunghezza variabile

# Frammentazione per trasporto di «pacchetti IP grandi»

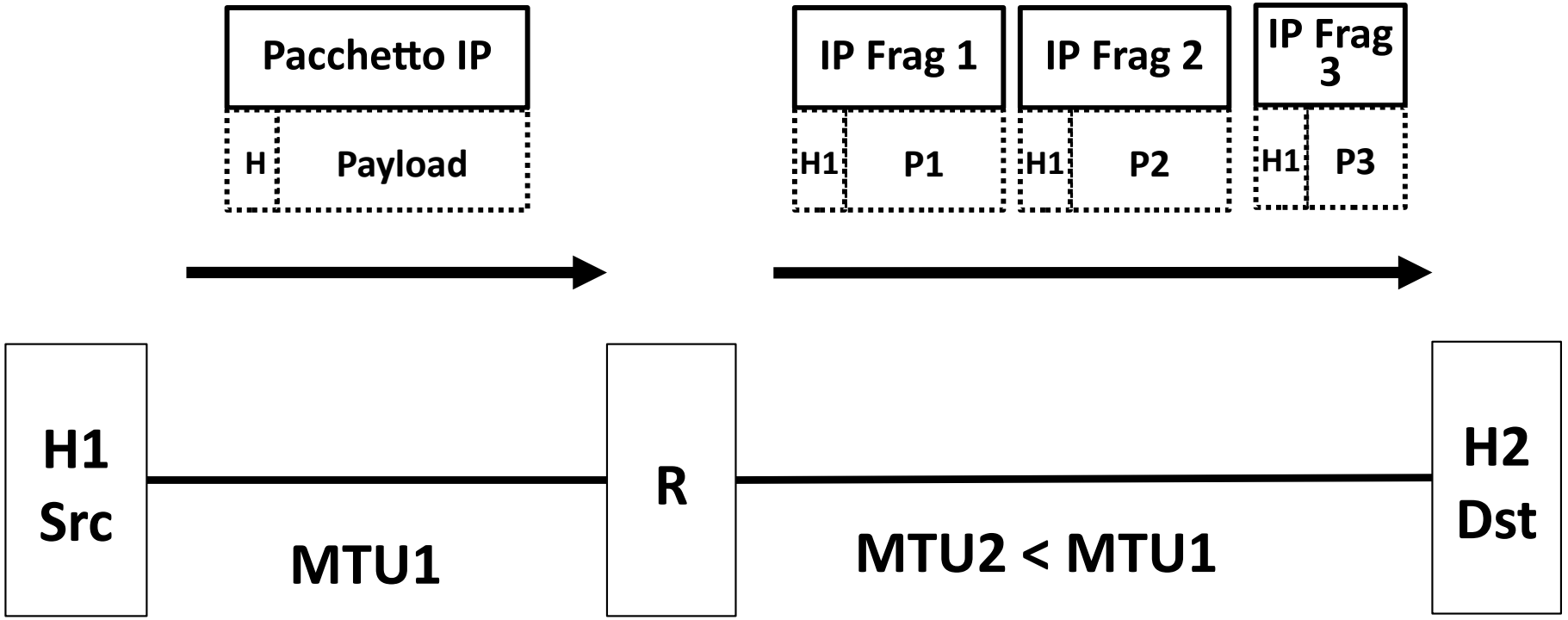
- Dimensione massima pacchetto IP: 64KB
- Dimensione MTU del livello H2N sottostante?
  - Ricordare tipica MTU di 1500 Byte di Ethernet
- Il pacchetto IP è da considerarsi un pacchetto «virtuale»
  - Il mittente frammenta il pacchetto IP «virtuale» di grandi dimensioni in tanti pacchetti IP inviabili a livello H2N
  - Il destinatario si occupa di ricostruire il pacchetto originale




**Esempio: MTU = 1500B; IP Header = 5 (20B); IP Tot Length = 5000B.**
**Quanti frammenti verranno realizzati e di quali dimensioni?**

# Frammentazione per MTU piccolo
# Operato da router

- Nota: i pacchetti potrebbero essere anche essere già frammentati. Possiamo frammentare più volte?
- L'host destinatario si occupa della ricostruzione
  - Non il router successivo, cosa che avviene in caso di tecniche di frammentazione a livello H2N (se supportato, ad esempio Ethernet no!), se il livello H2N non supporta MTU minimo stabilito dalla rete IP (e.g., 576B per IPv4, 1280B per IPv6).



# Frammentazione per invio in rete operata end-to-end (Path MTU discovery)

- Frammentazione da parte dei router ormai poco supportato su Internet
  - Completamente non supportato da IPv6
  - Perché? Ricordare il principio di rete con nodi intermedi semplici
- I router segnalano l'impossibilità a inoltrare il pacchetto a causa di MTU piccolo
  - Forzabile dal mittente settando il flag do not fragment (vedere esercitazioni ICMP): prende il nome di Path MTU discovery (Path MTU: "the minimum link MTU of all the links in a pa…urce node and a destination node" – RFC2460)