



Fondamenti di Internet e Reti

**Antonio Capone, Matteo Cesana,
Ilario Filippini, Guido Maier**



1 – Introduzione e Architetture

**Antonio Capone, Matteo Cesana,
Ilario Filippini, Guido Maier**



1a – Introduzione al corso

**Scopo e finalità, testi e materiale,
esami, contatti**

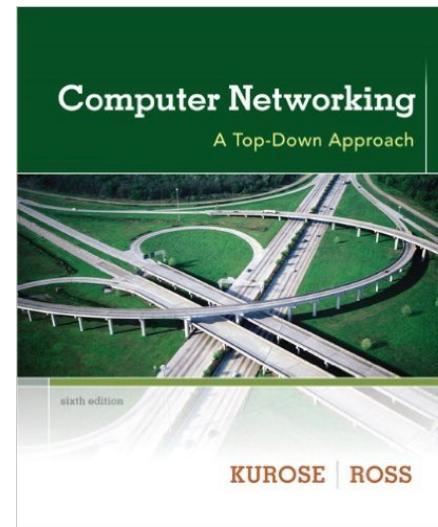
Docente

- Prof. Matteo Cesana
- Ufficio:
 - DEIB, ed. 20, 3° piano
 - stanza 335
- Tel: (02 2399) 3695
- E-mail: matteo.cesana@polimi.it
- Web page: <http://home.deib.polimi.it/cesana/>
- Orario di ricevimento:
 - Martedì 10.00-11.30
- Collaboratori per esercitazioni e laboratori:
 - Prof. Alessandro Redondi



Materiale didattico

- **Testo di riferimento:**
James F. Kurose, Keith W. Ross
Reti di Calcolatori e Internet
Un approccio top-down
Settima Edizione (2017)
Pearson
- **Disponibile anche in inglese**
James F. Kurose, Keith W. Ross
Computer Networking
A top-down approach
Seventh Edition (2017)
Pearson



Materiale didattico

- **Altro materiale**
 - *Slides* delle lezioni
 - Appunti delle esercitazioni
 - Temi d'esame
 - Materiale a supporto delle attività di laboratorio
 - Letture suggerite (*link*, articoli)
 - *Tool* per lezioni e laboratorio
 - Canale *YouTube* (video su aspetti specifici, video 2015-2016), <https://www.youtube.com/channel/UCKGJ5AI6XQBJz73Os3N1Rfg>
- **Tutto il materiale è disponibile (*linkato*) sulla pagina web del corso cui si accede dalla pagina personale del docente**



Home Page del corso

<http://www.antlab.polimi.it/teaching/fondamenti-di-internet-e-reti>

The screenshot shows the homepage of the Advanced Network Technologies LABoratory (ANTLab) website. At the top left is the logo 'ANT LAB' with a red ant icon. To the right is the text 'Advanced Network Technologies LABoratory' and a circular seal. On the left sidebar, under 'Matteo Cesana', there are links for 'Biography and Services', 'Teaching' (which is highlighted), 'Fondamenti di Internet e Reti' (under 'Internet of Things - Milano'), 'Internet of Things - Como', and 'Publications'. Under 'Main Menu', there are links for 'ANTLab', 'Staff', 'Research', and 'Projects'. The main content area features a banner for 'Anno accademico 2016-2017' with a list of links: 1. News, 2. Programma ed orario, 3. Materiale didattico, 4. Laboratorio, 5. Temi d'esame, 6. Libri, 7. Letture di approfondimento. Below this is a 'News' section with the text '24 Febbraio 2017 - Inizio del corso AA 2016/2017' and 'Il corso inizierà Lunedì 6 Marzo.' Further down are sections for 'Programma ed orario' (with a note about the official file 'fir-programma.txt') and 'Materiale didattico'.

www.antlab.polimi.it



POLITECNICO MILANO 1863

FIR:1 - Introduzione e architetture

Organizzazione del corso

- **Mix didattico:**
 - Lezione: 58 ore (circa)
 - Esercitazione: 26 ore (circa)
 - Laboratorio: 16 ore
- **Le esercitazioni sono inserite alla fine di ciascun argomento (non c'è una programmazione settimanale fissa)**
- **Il laboratorio si svolge in aula connessa senza PC, è necessario portare il proprio laptop (BYOD)**



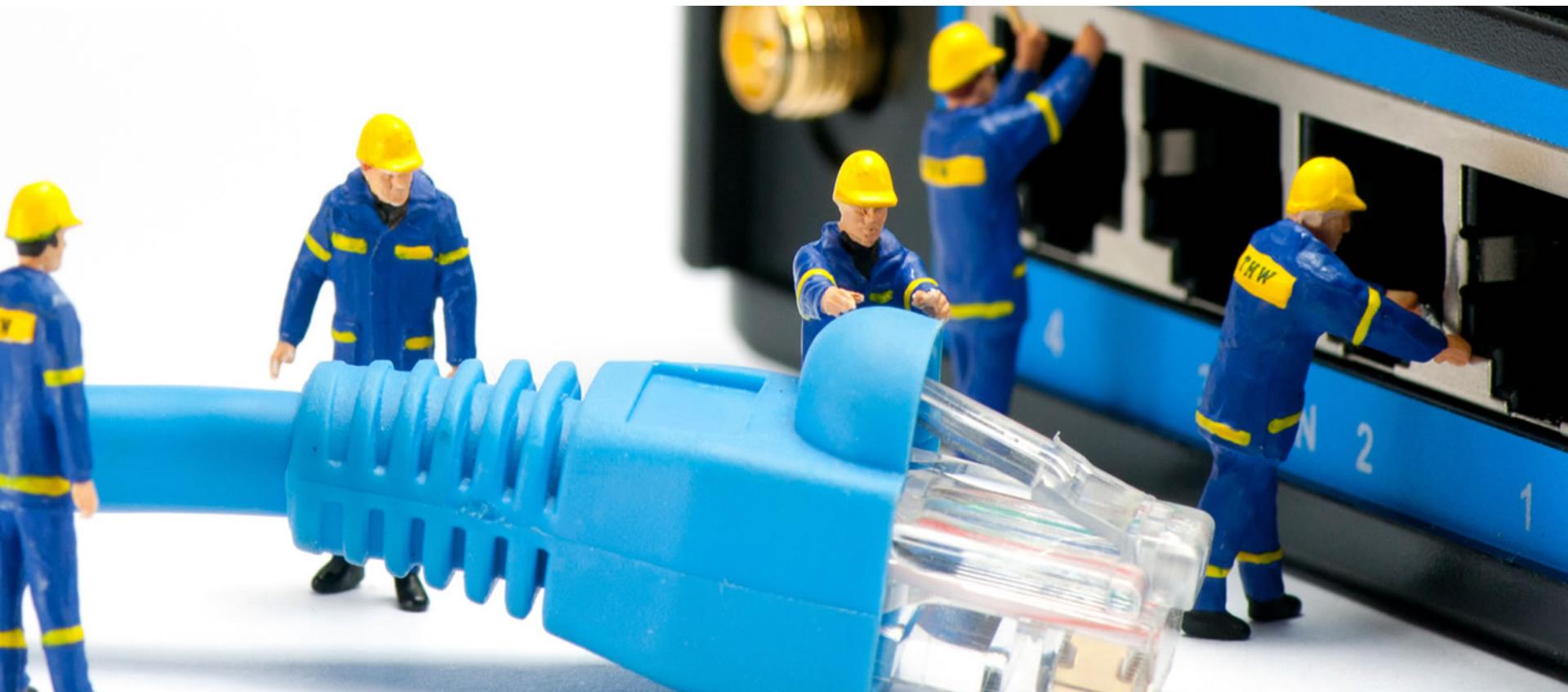
Modalità d'esame

- **Due prove in itinere**
 - Prova 1: inizio Maggio (4 Maggio)
 - Prova 2: fine corso (TBD)
- **Esame solo scritto (vedi temi d'esame su sito web)**
 - 3 esercizi simili a quelli visti a lezione/esercitazione
 - 1 esercizio di laboratorio
 - Domande (risposta aperta e/o chiusa)
 - (di solito: 6 pt per esercizio, 8 pt domande, 6 pt laboratorio, totale 32 pt)
- **Punti addizionali partecipando ai quiz in classe (kahoot.it, vedi dopo)**
 - Punti extra per chi frequenta e partecipa ai quiz alla fine di ogni parte/capitolo in classe (usando PC, tablet, o smartphone)
 - 1 pt premio a chi vince il quiz di ogni parte/capitolo
 - 3 pt x frazione di risposte esatta su totale complessivo (es. se uno studente risponde correttamente al 50% delle domande di tutti i quiz effettuati ottiene +1,5 punti sul voto d'esame)
- **Orale solo a discrezione del docente**



Scopo del corso

Iniziare a trasformarvi in esperti di Internet !



Scopo del corso

- **Fornire le conoscenze fondamentali sul funzionamento della rete tra calcolatori che oggi è alla base della quasi totalità dei servizi di comunicazione**
 - Audio e video telefonia
 - TV, video streaming, video on demand
 - Web, mobile apps, e accesso a DB in genere
 - Email
 - Messaging, chat
 - Social networks
 - Mappe e navigazione
 - Strumenti di collaborazione e condivisione
 - e molte altre ...



Scopo del corso

E in base a queste conoscenze imparare a:

Controllare l'interazione tra le applicazioni e la rete

Configurare la rete e le sue componenti

Prevedere il comportamento della rete calcolandone le prestazioni

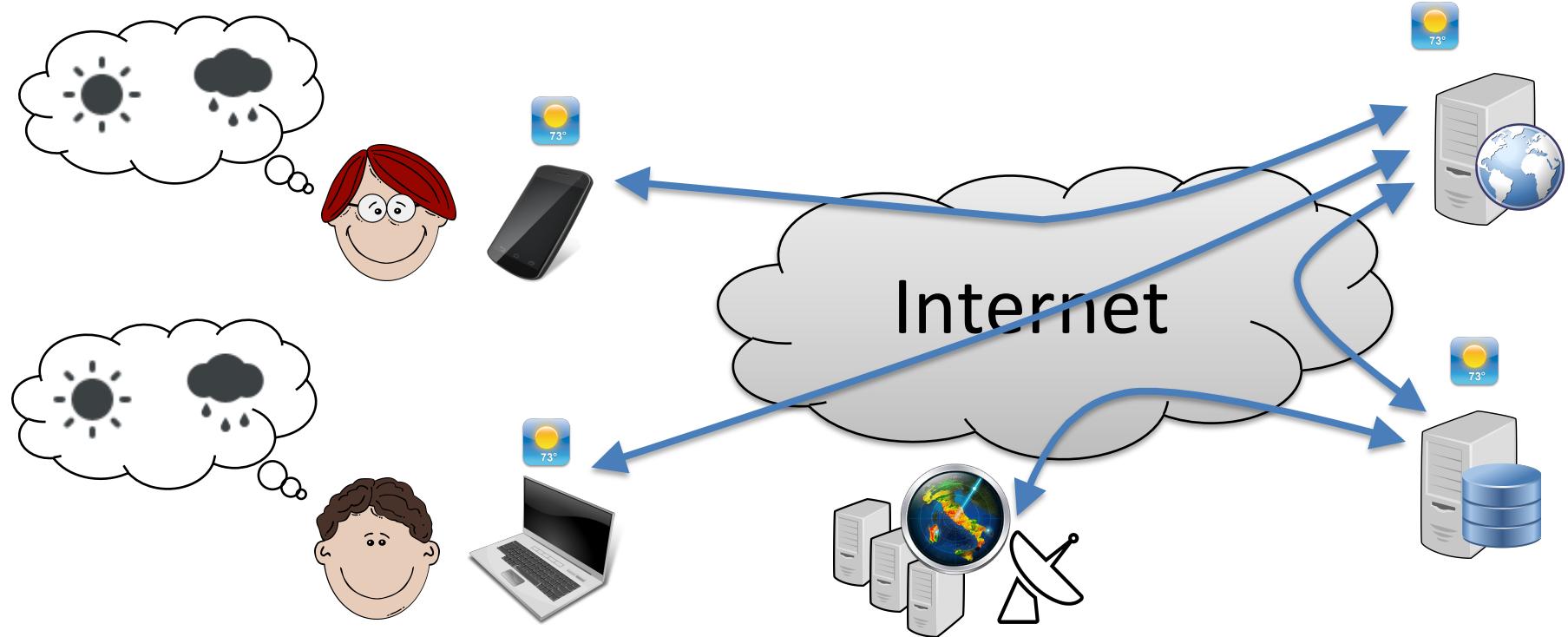
Monitorare il comportamento del sistema e risolvere i problemi

Essere pronti ai cambiamenti tecnologici della rete
(imparare ad imparare)



A cosa serve

- Tutti questi servizi del mondo della società dell'informazione sono **applicazioni distribuite**
- Le applicazioni distribuite sono programmi in esecuzione su più calcolatori che **scambiano dati tra loro usando Internet**



Vecchi mondi che non esistono più

Applicazioni isolate

- Elaborazioni isolate
- Scambio dati su rete

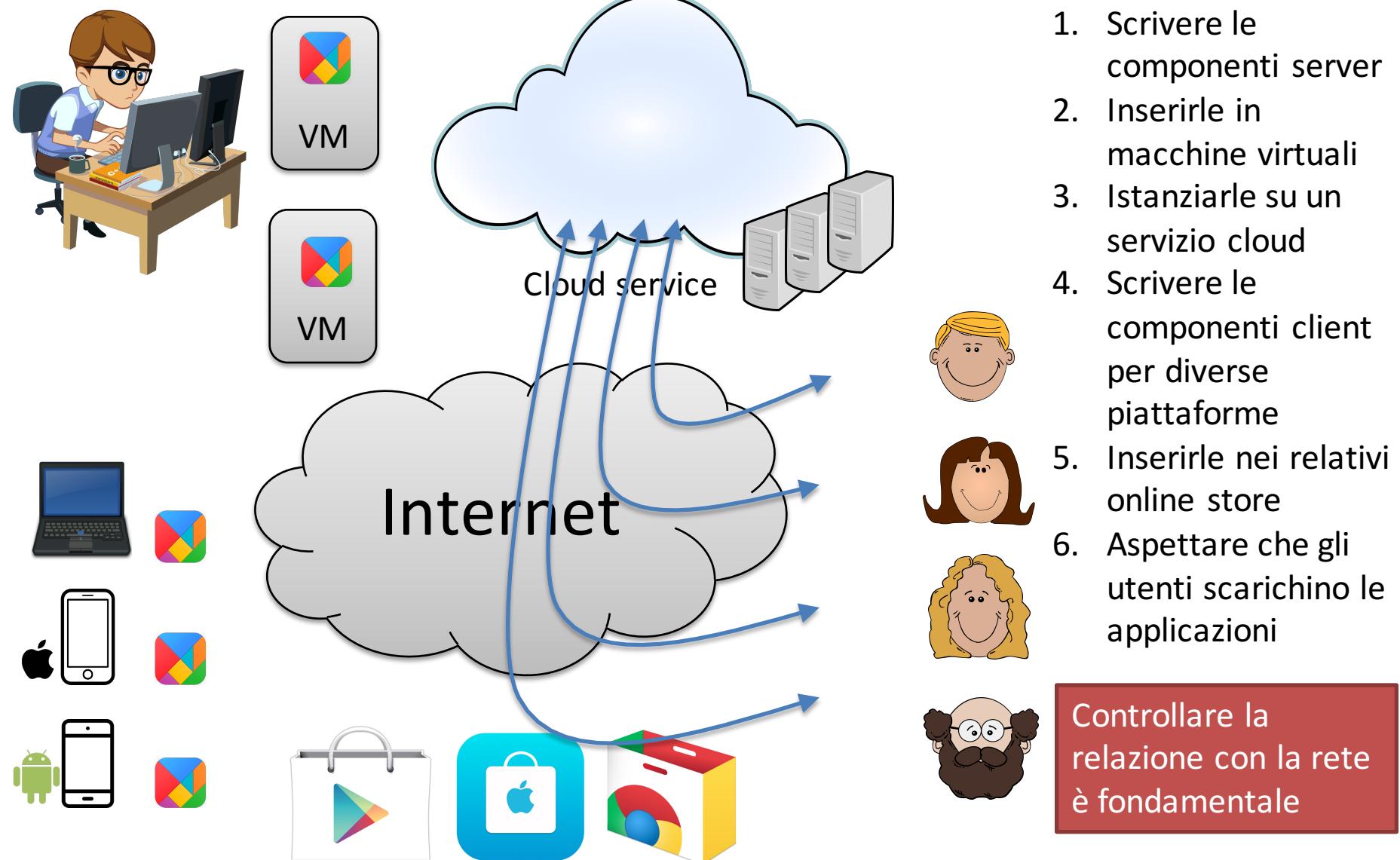


Reti di telecomunicazione

- Reti dedicate ai servizi
- Nessuna elaborazione

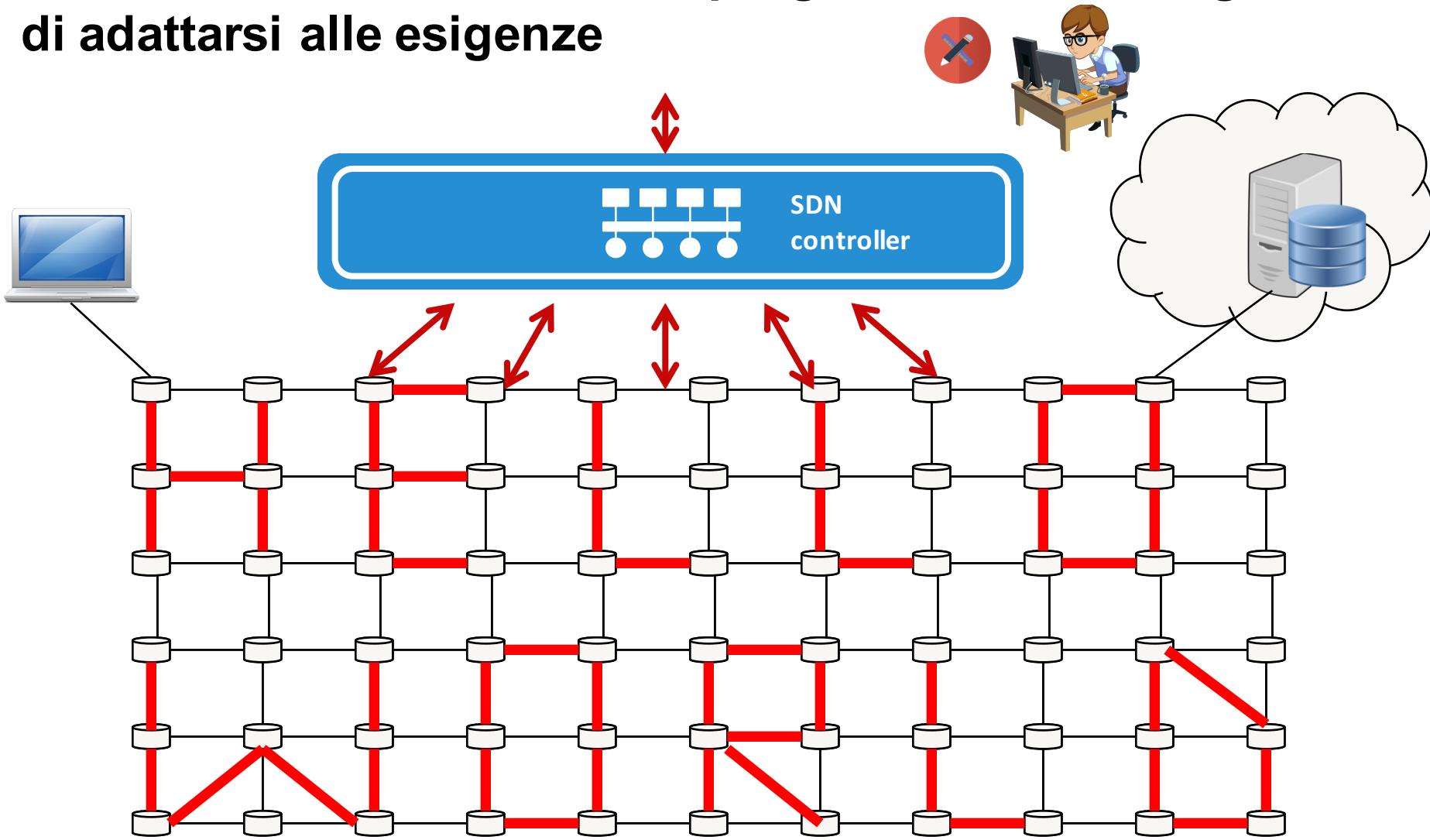


Cosa significa oggi sviluppare un'applicazione



Applicazioni di rete: Software Defined Networking

Anche la rete sta diventando programmabile ed in grado di adattarsi alle esigenze



Le conoscenze per l'ingegneria dell'informazione

Le competenze richieste a chi:



Sviluppa applicazioni e servizi

Costruisce la rete e i suoi servizi di comunicazione



Sono sempre più integrate

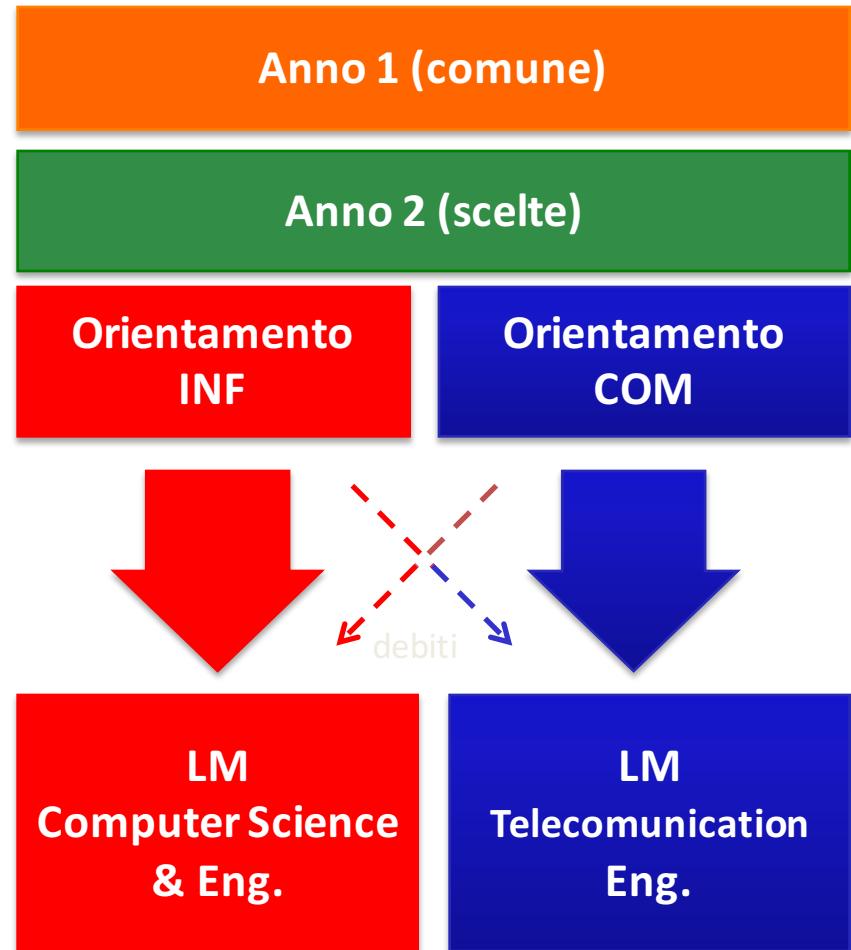


Per qualunque ingegnere che opera nel settore dell'informazione è fondamentale sia saper sviluppare applicazioni e servizi su piattaforme diverse sia progettare e gestire la rete e i servizi di comunicazione



Percorso didattico in Informatica e Comunicazioni

- Come sapete i due percorsi didattici sono stati integrati
- E' possibile rimanere puri (**INF** o **COM**) ma sono state progettate delle ibridazioni
- E una serie di corsi che consentono una solida preparazione multidisciplinare
- Questo è il primo corso di area **COM** che vedete



Percorso didattico in Informatica e Comunicazioni

1° anno

INSEGNAMENTO	CREDITI
ANALISI MATEMATICA I	10
GEOMETRIA E ALGEBRA LINEARE	8
FONDAMENTI DI INFORMATICA	10
FISICA	12
ELETTROTECNICA	9
FONDAMENTI DI INTERNET E RETI	10

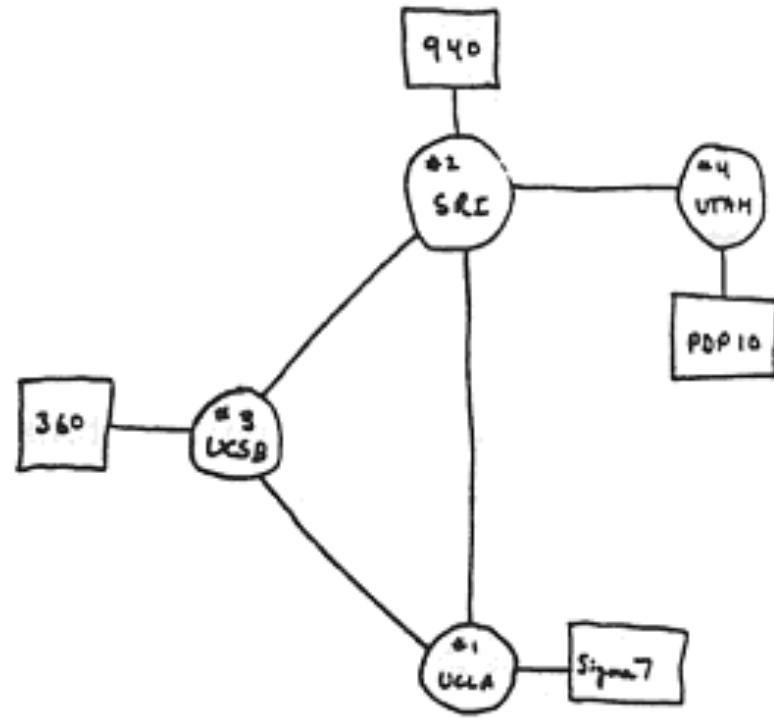
2° anno

INSEGNAMENTO	CREDITI
ANALISI MATEMATICA 2	10
ARCHITETTURA DEI CALCOLATORI E SISTEMI OPERATIVI	10
LOGICA E ALGEBRA	
Insegnamento a scelta dal gruppo TABASE	10
ELETTROMAGNETISMO E CAMPI	
STATISTICA E CALCOLO DELLE PROBABILITA'	10
TEORIA DEI FENOMENI ALEATORI E DELLA STIMA	
ALGORITMI E PRINCIPI DELL'INFORMATICA	
LAB: ALGORITMI E PRINCIPI DELL'INFORMATICA	11
SEGNALI PER LE COMUNICAZIONI	
LAB: SEGNALI PER LE COMUNICAZIONI	
FONDAMENTI DI AUTOMATICA	10

- Al 3° anno sono possibili dei recuperi per chi cambia idea
- Ci sono corsi che possono essere scelti dell'altra area
 - SOFTWARE DEFINED NETWORKING (5cr)
 - SEGNALI PER LE COMUNICAZIONI (10 cr)



Brevissima storia di Internet



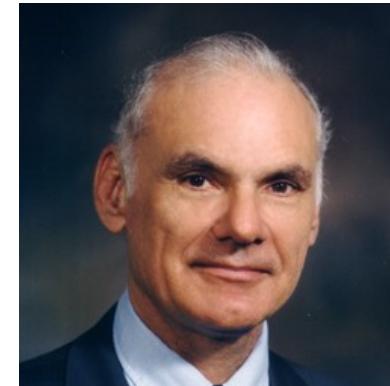
THE ARPA NETWORK

DEC 1969



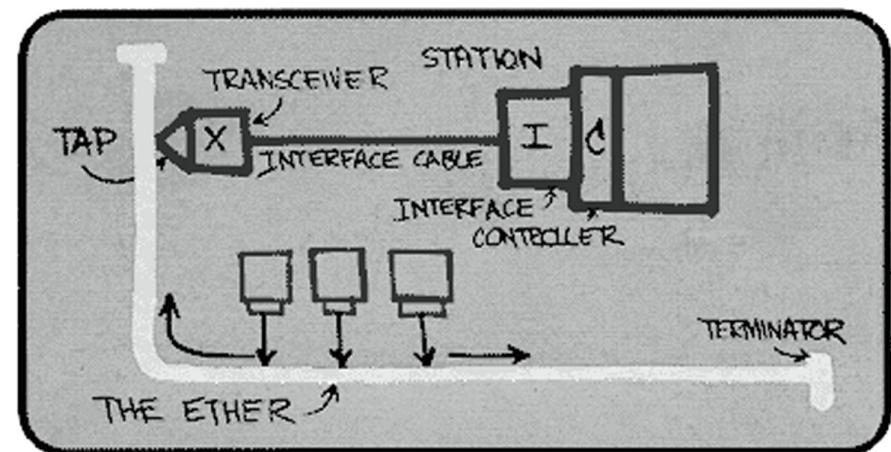
Storia di Internet: anni '60

- **1961:** Kleinrock – dimostra l'efficacia della commutazione di pacchetto grazie alla teoria delle code
- **1967:** Lawrence Roberts progetta ARPAnet (Advanced Research Projects Agency)
- **1969:** primo nodo di IMP (Interface Message Processor) di ARPAnet a UCLA



Storia di Internet: anni '70

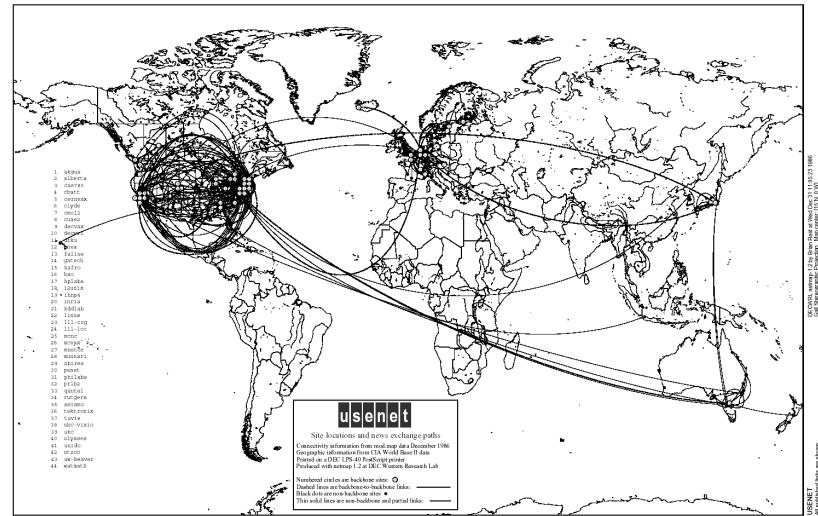
- **1972:**
 - Nasce NCP (Network Control Protocol) il primo protocollo di Internet
 - Primo programma per la posta elettronica
 - ARPAnet ha 15 nodi
- **1970:**
 - ALOHAnet rete radio a pacchetti al Univ. of Hawaii
- **1974:**
 - Cerf and Kahn – definiscono i principi dell'internetworking (rete di reti)
- **1976:**
 - Nasce Ethernet nei laboratori di Xerox
- **1979:**
 - ARPAnet ha 200 nodi



Storia di Internet: anni '80

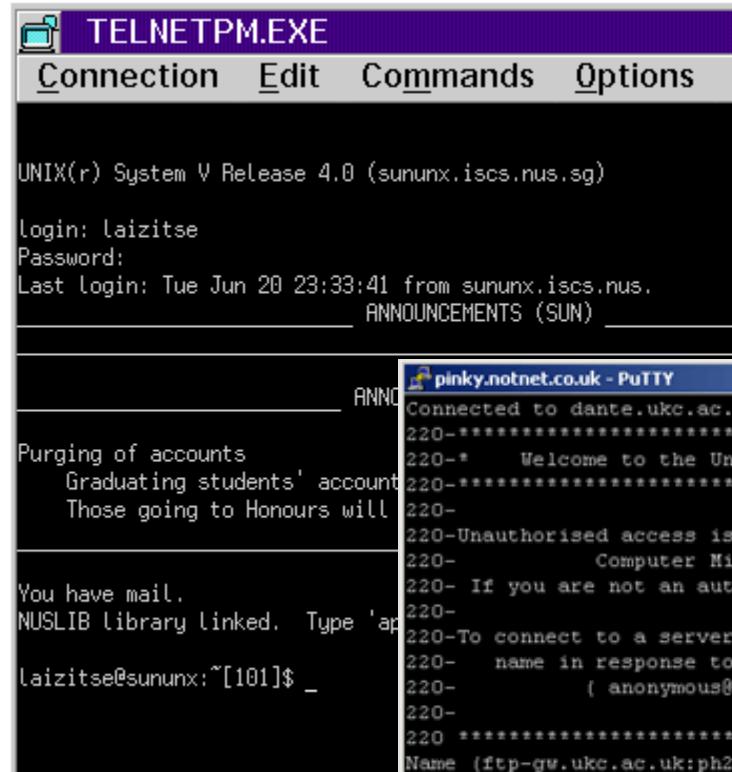
- **1982:** definizione del protocollo SMTP per la posta elettronica
- **1983:** rilascio di TCP/IP che sostituisce NCP
- **1983:** definizione del DNS per la traduzione degli indirizzi IP
- **1985:** definizione del protocollo FTP
- **1988:** controllo della congestione TCP

- **Nuove reti nazionali: Csnets, BITnet, NSFnet, Minitel**
- **100.000 host collegati**

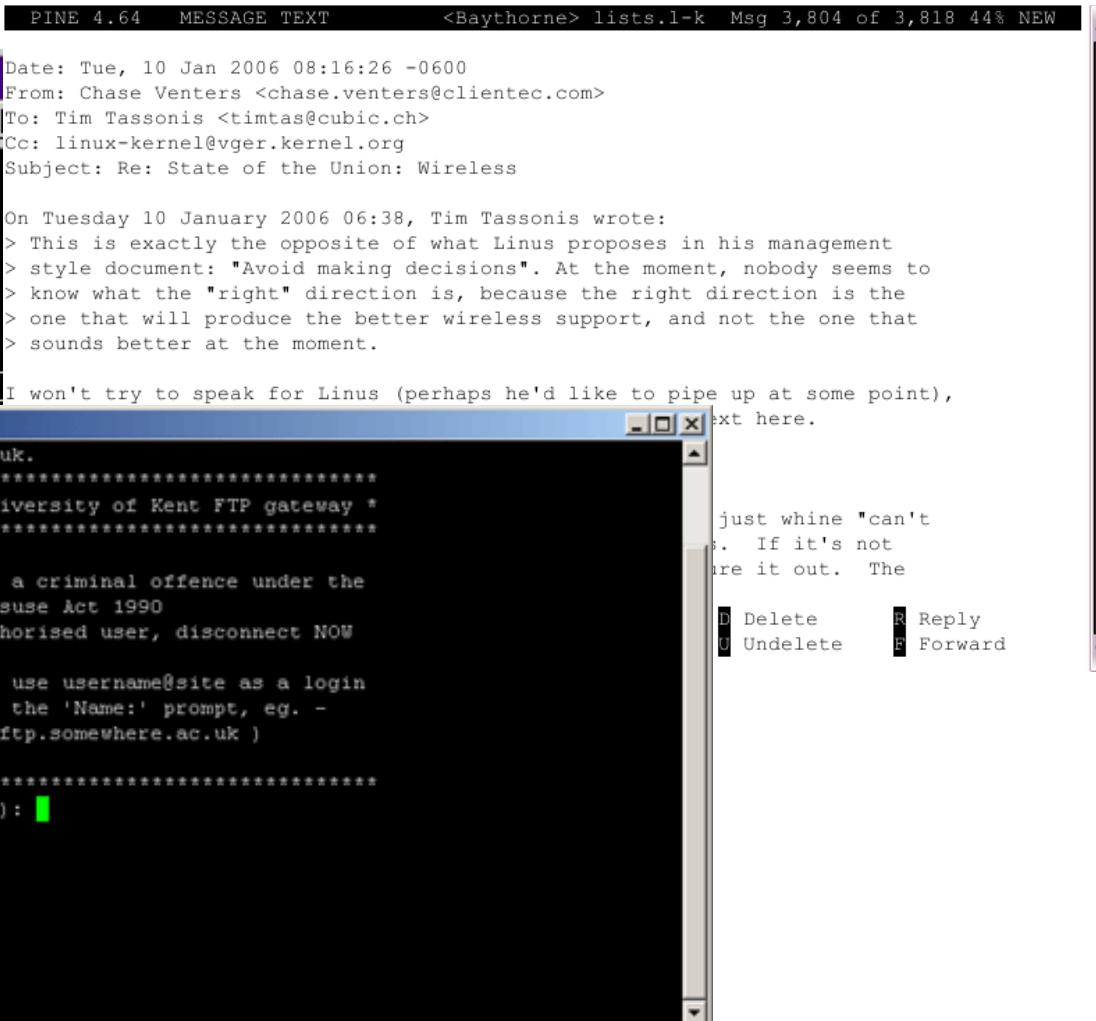


Storia di Internet: Le prime applicazioni

Telnet



Email

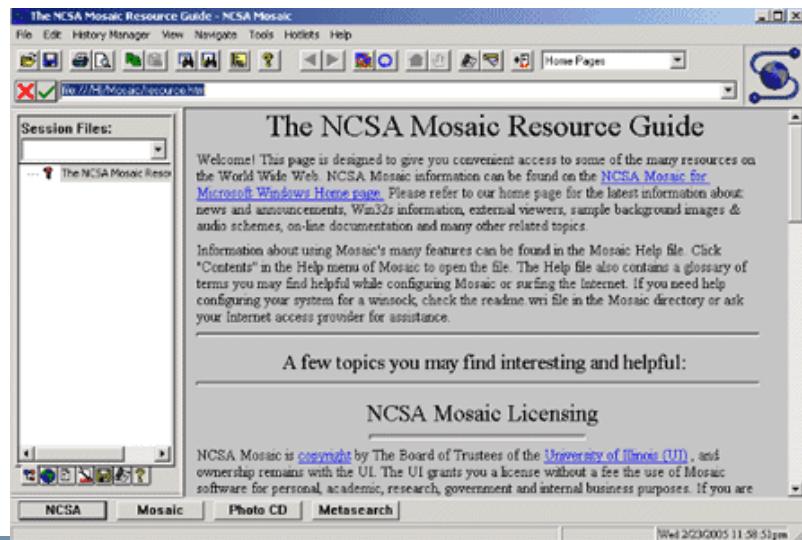
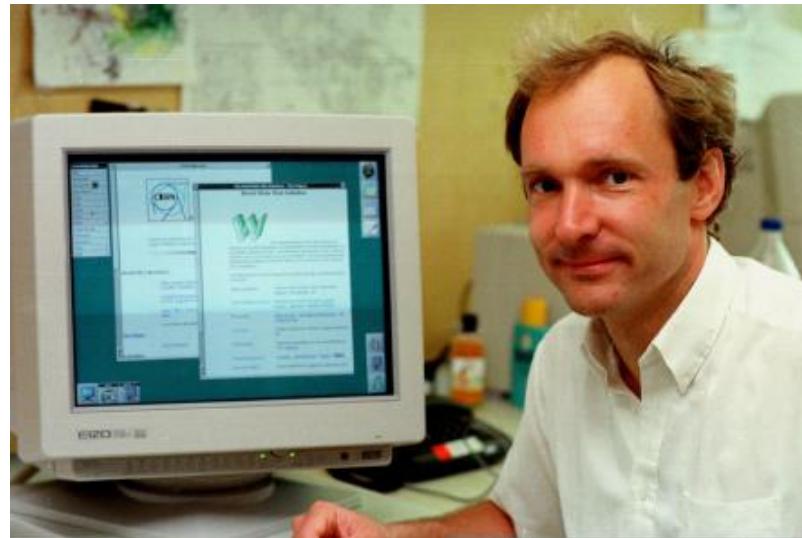


FTP



Storia di Internet: anni '90

- **1990:** ARPAnet viene dismessa
- **1991:** NSF lascia decadere le restrizioni sull'uso commerciale di NSFnet
- **Primi anni '90:** Tim Berners-Lee inventa il web al Cern di Ginevra
- **1994:** Mosaic, poi Netscape
- **Fine '90 :** commercializzazione del Web



Storia di Internet: anni '00

2000 – 2009:

- Arrivano le “killer applications”: messaggistica istantanea, condivisione di file P2P, IP Telephony, social networks
- La sicurezza di rete diventa un problema
- Centinaia di milioni di host, un miliardo di utenti
- Velocità nelle dorsali dell'ordine dei Gbps



Shawn
Fanning



Diffie-Hellman-Merkle



Zuckerberg



Storia di Internet: anni '10

2010 – oggi:

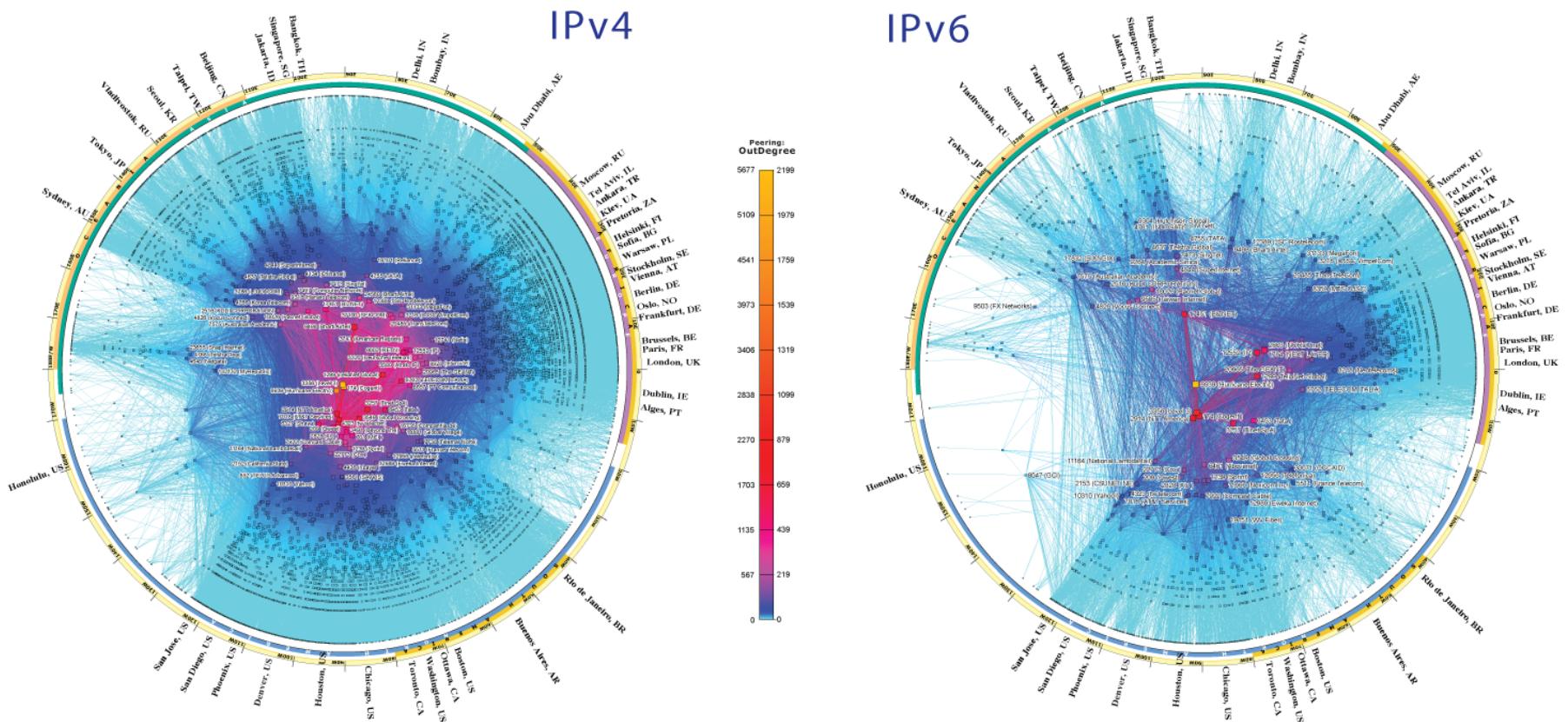
- Esplosione della *Mobile Internet*
- Arrivano gli smartphone
- La telefonia si trasferisce definitivamente su Internet
- I contenuti video diventano il traffico predominante sulla rete



Mappa di Internet oggi

CAIDA's IPv4 & IPv6 AS Core AS-level INTERNET GRAPH

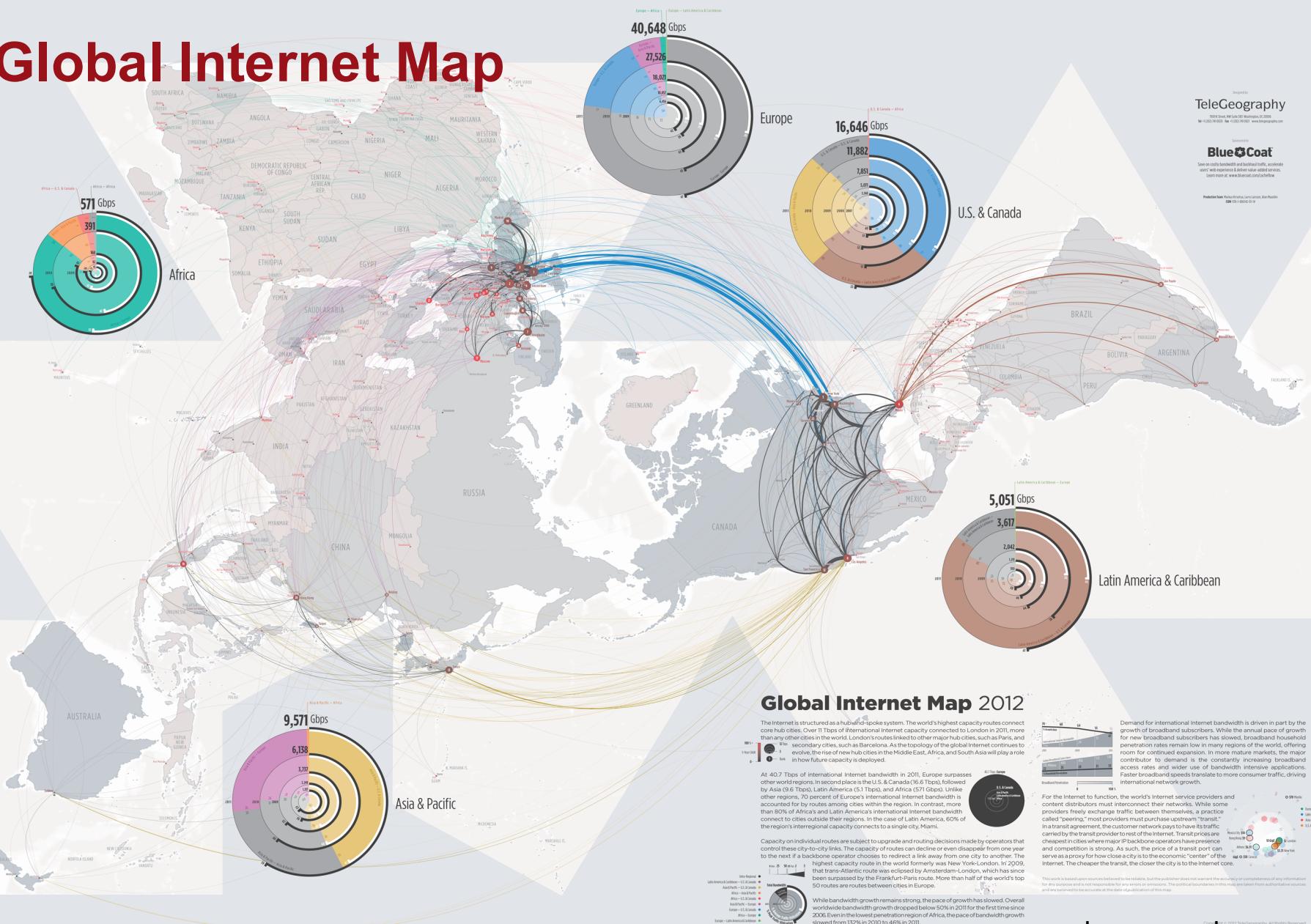
Archipelago January 2015



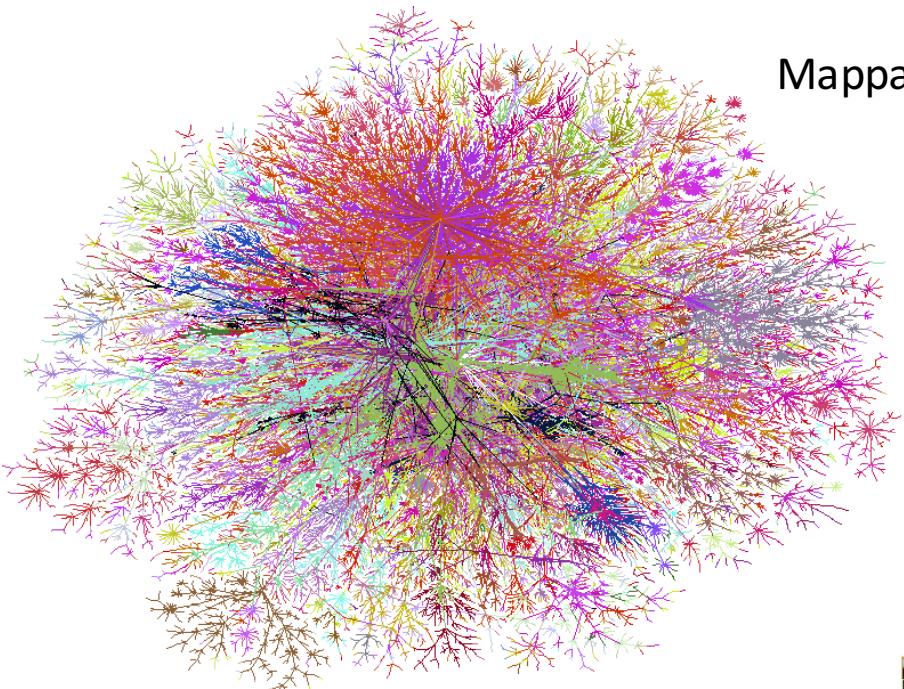
Copyright © 2015 UC Regents. All rights reserved.



Global Internet Map



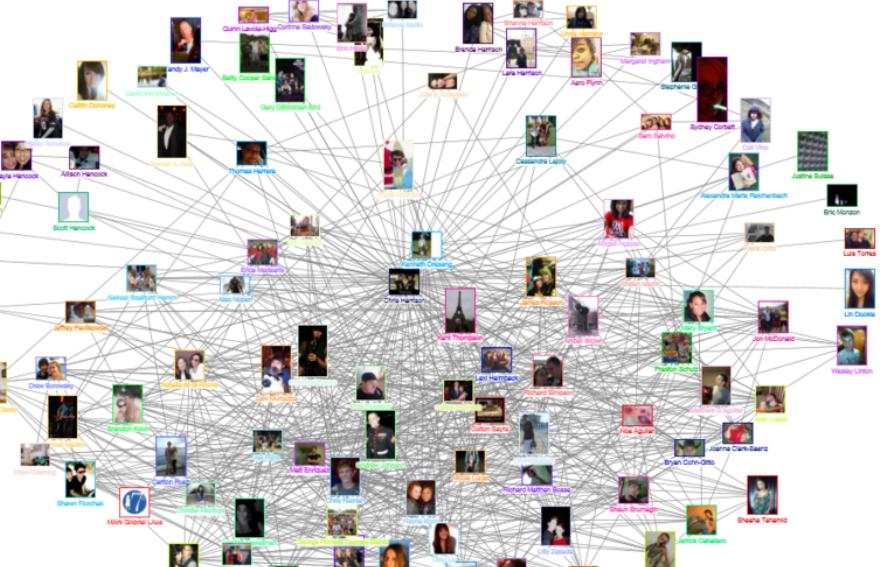
Il mondo è piccolo



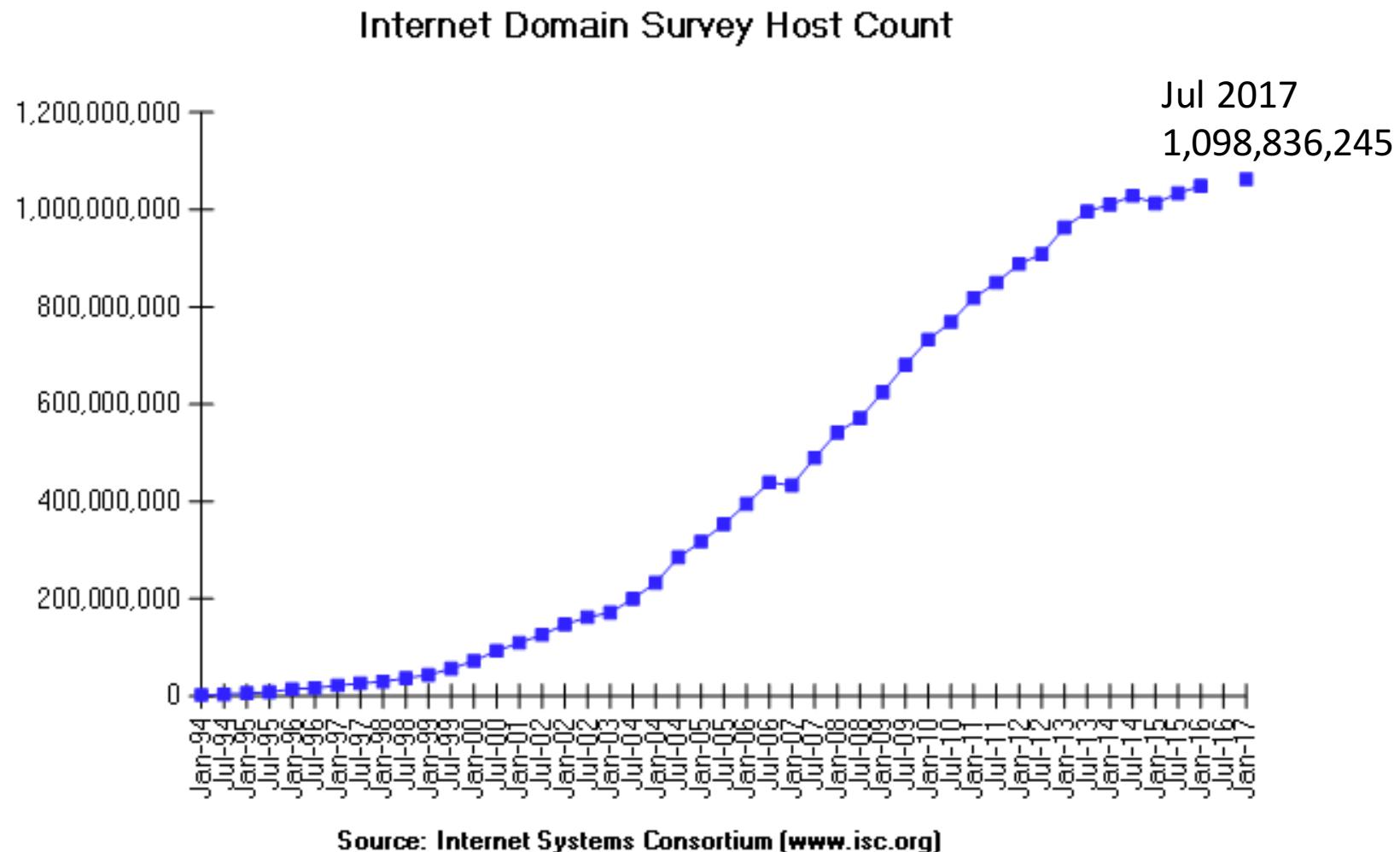
Mappa sottoreti IP (2001)



Esempio di topologia di social network



La crescita di internet



isc.org

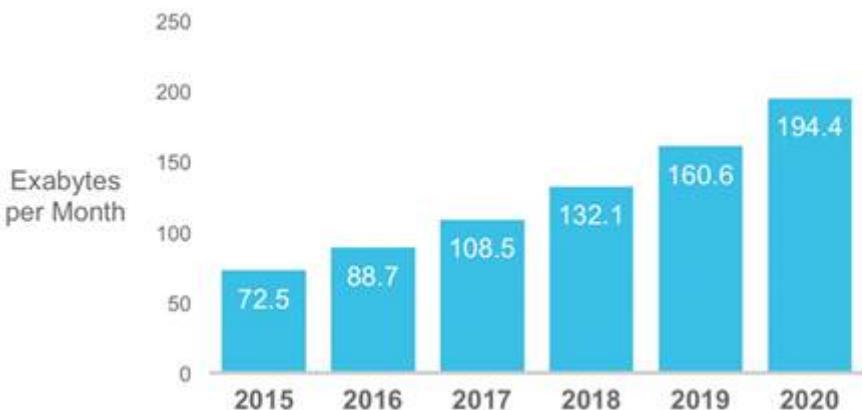


POLITECNICO MILANO 1863

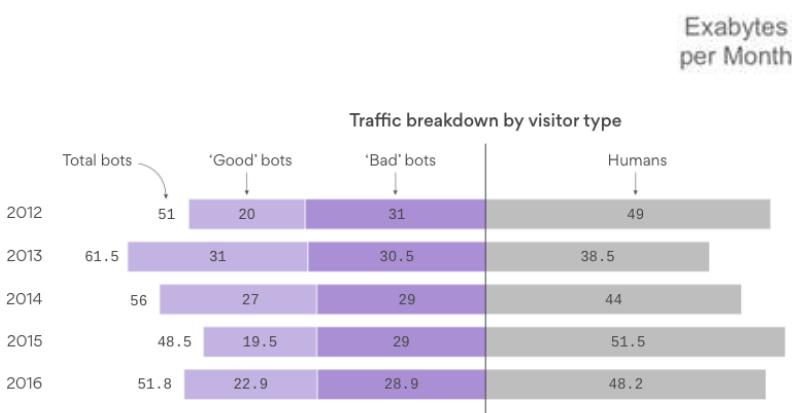
FIR:1 - Introduzione e architetture

31

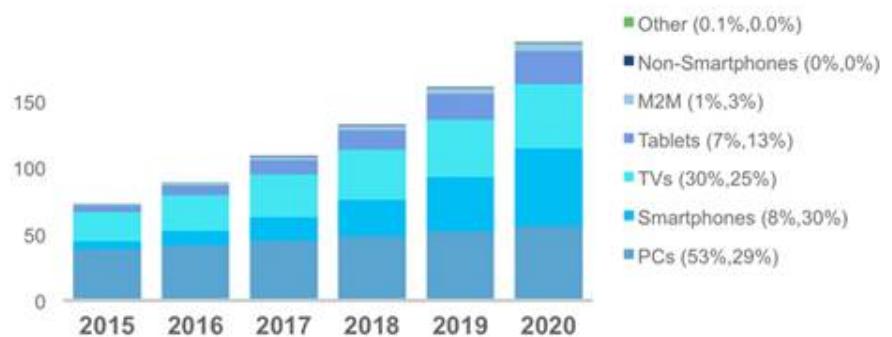
Il traffico di internet



(exa= 10^{18})



Data: [Imperva Incapsula Bot Traffic Report](#). Based on 16.7+ billion visits to 100,000 randomly-selected domains on the Incapsula network; Chart: Lazaro Gamio / Axios



CISCO VNI
(2016)



Internet è nel ~~CLOUD~~ MARE



99% del traffico internazionale passa attraverso cavi sottomarini

telegeography.com



Internet NON è un posto sicuro



ATTACK ORIGINS

#	Country
117	Germany
62	United States
46	China
15	Russia
15	Netherlands
14	Taiwan
10	Italy
9	Turkey
6	South Korea
5	Costa Rica

ATTACK TYPES

#	Port	Service Type
107	5060	sip
37	23	telnet
36	53413	netis-router
20	50856	xsan-filesystem
14	443	https
14	50864	xsan-filesystem
13	138	netbios-dgm
7	445	microsoft-ds
6	80	http
5	18245	unknown

ATTACK TARGETS

#	Country
274	United States
10	Thailand
10	Russia
8	Germany
7	Spain
6	Iceland
4	Norway
3	United Arab Emirates
3	Singapore
3	Hong Kong

LIVE ATTACKS

Timestamp	Attacker
18:59:43.015	Aapt Limited
18:59:42.612	Metro Denver Title Llc
18:59:42.245	Carinet Inc.
18:59:41.832	Icelab Networks
18:59:41.590	China Unicom Hebei Province Network
18:59:41.266	Hurricane Electric Inc.
18:59:41.170	Ovh Static Ip
18:59:41.083	Telecom Italia S.P.A.
18:59:41.082	Telecom Italia S.P.A.
18:59:41.082	Telecom Italia S.P.A.

Attacker IP	Attacker Geo	Target Geo	Attack Type	Port
59.100.18.84	Brisbane, AU	Lynnwood, US	telnet	23
208.42.231.233	Englewood, US	Englewood, US	db-lsp-disc	17500
71.6.165.200	San Diego, US	San Francisco, US	gprs-sig	3386
72.1.109.27	Englewood, US	Englewood, US	netbios-dgm	138
121.20.24.137	Shijiazhuang, CN	Lynnwood, US	xsan-filesystem	50856
65.19.130.35	West Hollywood, US	West Hollywood, US	unknown	21027
5.196.197.186	Roubaix, FR	San Francisco, US	unknown	14481
95.247.245.139	Cesena, IT	Reykjavik, IS	netis-router	53413
95.247.245.139	Cesena, IT	Reykjavik, IS	netis-router	53413
95.247.245.139	Cesena, IT	Reykjavik, IS	netic-router	53413



Programma del corso

1. INTRODUZIONE E ARCHITETTURE

- Principi generali, architettura e componenti, meccanismi di base

2. SISTEMI DI COMUNICAZIONE

- Come misuro le prestazioni di una rete: il concetto di *throughput*, i ritardi nelle reti di telecomunicazioni (tempo di trasmissione, ritardo di trasferimento, tempo di processing, tempo di accodamento).

3. MODELLI FUNZIONALI

- Come è gestita la comunicazione in rete: il concetto di protocollo di comunicazione, modelli architetturali a strati, commutazione di pacchetto e commutazione di circuito



Programma del corso

4. PROTOCOLLI APPLICATIVI

- architetture delle applicazioni di rete: approccio *client-server* ed approccio *peer-to-peer*;
- esempi di protocolli applicativi *client-server*: HTTP, FTP, SMTP;
- architetture *peer-to-peer*: la rete Gnutella, BitTorrent

5. IL LIVELLO DI TRASPORTO

- caratterizzazione del servizio di comunicazione tra processi applicativi; trasporto non affidabile: il protocollo UDP (formato dei segmenti);
- trasporto affidabile: il protocollo TCP (formato dei segmenti, apertura della connessione, controllo di flusso, controllo di congestione e controllo d'errore).



Programma del corso

6. IL LIVELLO DI NETWORKING:

- *I'Internet Protocol (IP)*: servizi offerti da IP, formato dei pacchetti IPv4
- Gestione di indirizzi IP: formati e notazioni degli indirizzi IPv4, le classi e gli indirizzi speciali, pianificazione di uno spazio di indirizzamento IPv4, tecniche di *subnetting* e *supernetting*, assegnamento automatico di indirizzi IP: il Dynamic Host Control Protocol (DHCP)
- Corrispondenza tra indirizzi IP ed indirizzi simbolici (*Il Domain Name System*);
- *I'Internet Control Message Protocol (ICMP)*



Programma del corso

7. INOLTRO ED INSTRADAMENTO IN INTERNET:

- inoltro diretto ed indiretto
- uso delle tabelle di *routing*;
- instradamento su cammini minimi, la costruzione dell’albero dei cammini minimi,
- Instradamento *link state* ed instradamento *distance vector*
- Esempi di protocolli: RIP, OSPF, BGP,



Programma del corso

8. RETI LOCALI E LIVELLO DI LINEA:

- problema dell'accesso multiplo,
- indirizzamento nelle reti locali,
- l'*Address Resolution Protocol* (ARP),
- interconnessione di reti locali con bridge/switch,
- lo standard Ethernet/802.3: principi e funzionamento,
- lo standard IEEE 802.11 (WiFi): principi e funzionamento.

9. INTRANET

- Indirizzamento privato e traduzioni di indirizzi IP (NAT, NAPT)
- Interconnessione di reti private (*IP Tunneling*)
- IPv6 (cenni)



Programma del corso

LABORATORIO:

- Attività di base svolta a lezione ed esercitazione:
 - Sniffer di rete (Wireshark)
 - Ping (PingPlotter), Traceroute, Dig, Strumenti del browser (*Chrome*)
 - Protocolli applicativi (con utilizzo di *server* e *client email e web*)
- Attività di laboratorio
 - Lab 1: Python e *scripting* per analisi di rete
 - Lab 2: Programmazione socket in Python, parte I
 - Lab 3: Programmazione socket in Python, parte II
 - Lab 4: Configurazione e simulazione di rete (Packet Tracer), parte I
 - Lab 5: Configurazione e simulazione di rete (Packet Tracer), parte II
 - Lab 6: Attività sperimentali su reti wireless



Programma CISCO Networking Academy

- Il Politecnico offre agli studenti la possibilità di seguire online e gratuitamente i corsi di formazione per la certificazione CISCO CCNA
- Il programma CISCO NA è uno strumento aggiuntivo e facoltativo offerto per integrare la preparazione e acquisire competenze anche di tipo implementativo che non vengono normalmente fornite nei corsi
- Il programma CCNA si articola in 4 moduli/semestri
 - CCNA 1, CCNA 2, CCNA 3, CCNA 4



Come si segue il programma CCNA ?

- Il programma CCNA si segue mediante un piattaforma di E-learning
- Ogni studente studia autonomamente usando la piattaforma
- I docenti dei corsi di reti e il responsabile del programma CISCO (Prof. Antonio Capone) sono a disposizione per chiarimenti durante l'orario di ricevimento
- Seguire il programma CCNA richiede dell'impegno aggiuntivo rispetto al normale impegno per i corsi universitari
- A fronte di questo impegno si riceve una preparazione complementare a quella universitaria molto apprezzata nel mercato del lavoro



Piattaforma di E-Learning

- **Esempi della piattaforma:**
 - Portale d'ingresso: NetSpace

The screenshot shows the Cisco Networking Academy NetSpace portal. At the top, there's a navigation bar with links for 'NetSpace Home', 'About Us', 'Program', 'Offerings', 'Communities', and user icons for email and profile. The main content area features a banner for 'Scheduled Site Maintenance'. Below it, there are three main sections: 'Access the Courses You Are Taking' (with a photo of a student at a computer), 'Security in the Cloud Webinar' (with a thumbnail of a presentation slide), and 'Self-Enroll Made Easy' (with a 'Learn' button graphic). A sidebar on the right is titled 'My Tasks'.

Welcome, Antonio Capone, Sign Out | Help |

Cisco Networking Academy Mind Wide Open

NetSpace Home About Us Program Offerings Communities

Scheduled Site Maintenance Show

Teach Manage Learn

Access the Courses You Are Taking

When you are enrolled in courses you will be able to access them from this tab.

Actions View Course History

Security in the Cloud Webinar

Learn about the security risks and threats in cloud, and other topics such as Identity and Access Management & data security

Register Here

Self-Enroll Made Easy

Explore the revamped Course Catalog & Self-Paced Courses, under Offerings menu, choose "Enroll Now" and start learning.

More Details

My Tasks

Your classroom related tasks will display here.



Piattaforma di E-Learning

- Esempi della piattaforma:
 - Esempio (1)

NOTA: il materiale è in inglese

The screenshot shows the 'Routing and Switching Essentials' course structure. On the left, a vertical navigation bar lists chapters from Chapter 0 to Chapter 6. To the right, a detailed view of Chapter 1 is shown, with sections 1.1 and 1.1.1 expanded. Each section includes a thumbnail image and a brief description. At the bottom, there is a footer with various navigation icons.

Routing and Switching Essentials

Chapter 0
Course Introduction

Chapter 1
Introduction to Switched Networks

Chapter 2
Basic Switching Concepts and Configuration

Chapter 3
VLANs

Chapter 4
Routing Concepts

Chapter 5
Inter-VLAN Routing

Chapter 6
Static Routing

Section 1.0
Introduction

Section 1.1
LAN Design

Section 1.2
The Switched Environment

Section 1.3
Summary

Topic 1.1.1
Converged Networks

Topic 1.1.2
Switched Networks

Page 1.1.1.1
Growing Complexity of Networks

Page 1.1.1.2
Elements of a Converged Network

Page 1.1.1.3
Cisco Borderless Networks

Recent Pages **Bookmarks** **Course Index** **Search** **Select Background** **Help** **Return to Class**



Come si fanno gli esami CCNA ?

- Gli esami si fanno sempre mediante un piattaforma di E-learning
- Gli esami consistono in quiz e attività interattive
- Ogni semestre prevede un esame finale
- Il giorno dell'esame lo studente ha tempo dalle 9 alle 17 per completare i quiz
- Normalmente alla fine di ogni semestre sono fissate 4-5 date per l'esame
- Ogni “assessment” ha un tempo massimo dopo l’attivazione
- Copiare o barare agli esami non conviene, conta solo quello che si impara



Come ci si iscrive al CCNA?

- ▶ Per iscrizioni e maggiori informazioni consultare la pagina del POLIMI:
<http://home.deib.polimi.it/capone>
 - > teaching
 - > CISCO NA
- ▶ ISCRIZIONI:
 - ▶ 2 semestre (mar-apr)
 - ▶ Dopo l'iscrizione (siate pazienti viene fatta a mano) riceverete un email con login e password
 - ▶ Usateli per effettuare il primo accesso al portale
 - ▶ Dopo il primo accesso cambiate login (preferibilmente in nome.cognome) e password e inserite tutte le informazioni personali

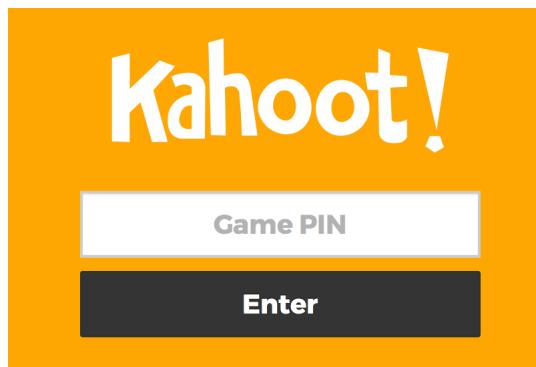


Quanto ne sapete di Internet

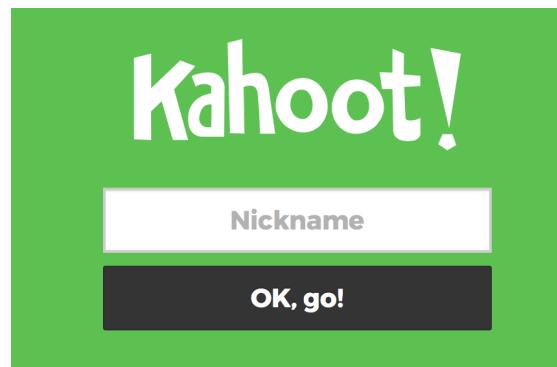
- **Test d'ingresso:**
 - Quiz gara
 - Usate il vostro smartphone/tablet/PC collegato a Internet
 - Collegativi con il browser a:

kahoot.it

Inserite il PIN che vi
darò tra un attimo



Inserite come nickname
Matricola.Cognome





1b – Concetti base

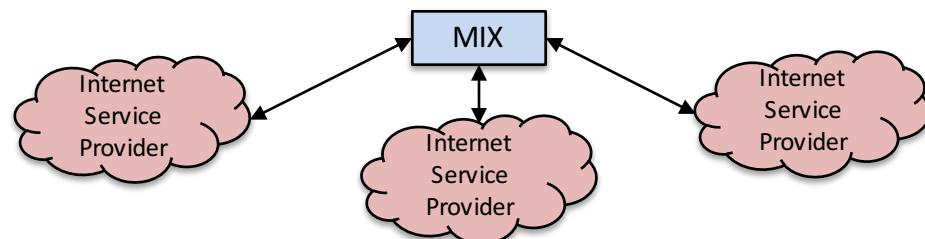
**Cos'è Internet, Architettura e
componenti, Meccanismi base**

Cos'è Internet?

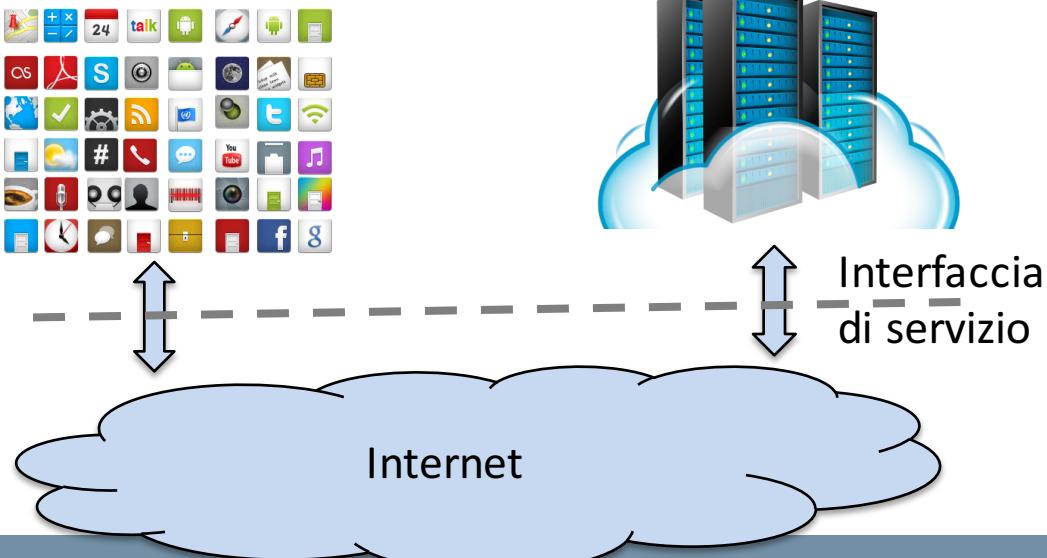
1) Una **infrastruttura fisica** fatta di componenti



2) Un'architettura di rete



3) Un **servizio di comunicazione** usato dalle applicazioni e i **protocolli di comunicazione** tra le componenti del sistema



Cos'è Internet?

Componenti fisiche

- Milioni di computer connessi alla rete chiamati **host = terminali**
- Canali di comunicazione di diversi tipi (fibra, cavo, radio, satellite, ...) **link = collegamenti**
- Nodi di rete chiamati **router = nodi**
- Altri nodi di rete locali (*switch, access point, modem, ...*)



router



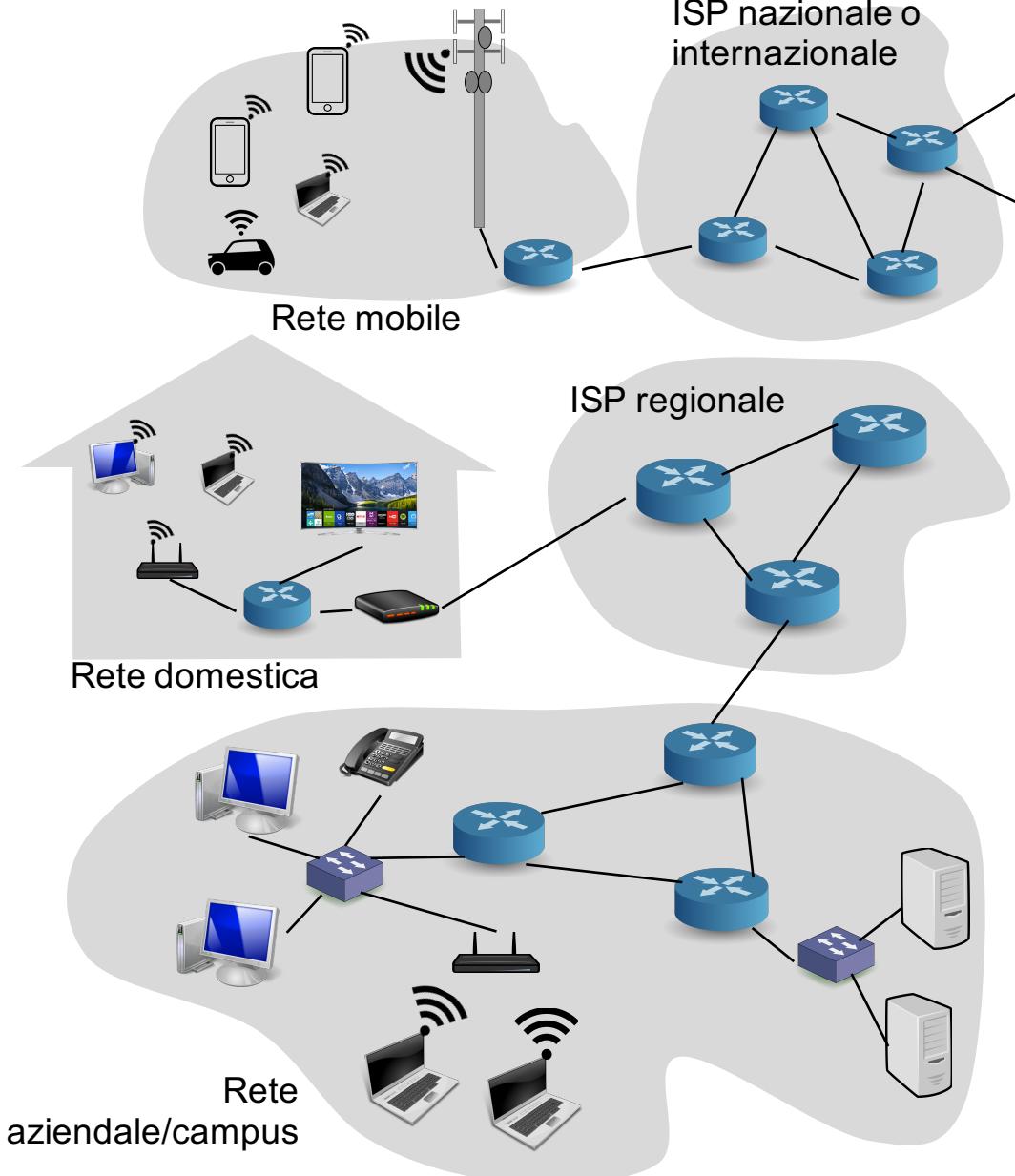
desktop



server

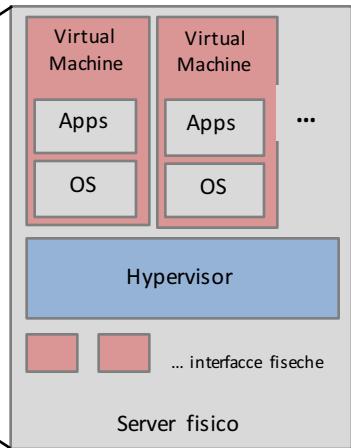


laptop



Componenti fisiche: host (terminali)

- Tutti gli **host** per la rete sono sistemi in grado di **inviare e ricevere informazioni** per le loro applicazioni finali
- Ma in realtà hanno caratteristiche molto diverse



Server fisici e virtuali per data center di servizi cloud



Dispositivi personali



Oggetti intelligenti



Componenti fisiche: link (collegamenti)

- I collegamenti possono essere di natura fisica molto diversa (fibra ottica, cavi coassiali, doppini, radio, ecc.)
- Differiscono anche per tecnologia di trasmissione dell'informazione
- E ovviamente per la velocità di trasmissione (rate) misurato in bit al secondo (b/s, Kb/s, Mb/s, Gb/s, Tb/s)

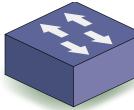


Componenti fisiche: nodi di rete

- I nodi di rete di internet sono i **router** che operano su unità di informazione (sequenze di bit) finite dette pacchetti
- Esistono altri nodi di rete che a livello locale svolgono altre funzioni di collegamento
- Vedremo che il “livello” a cui opera un nodo di rete è un aspetto importante della tecnologia



router



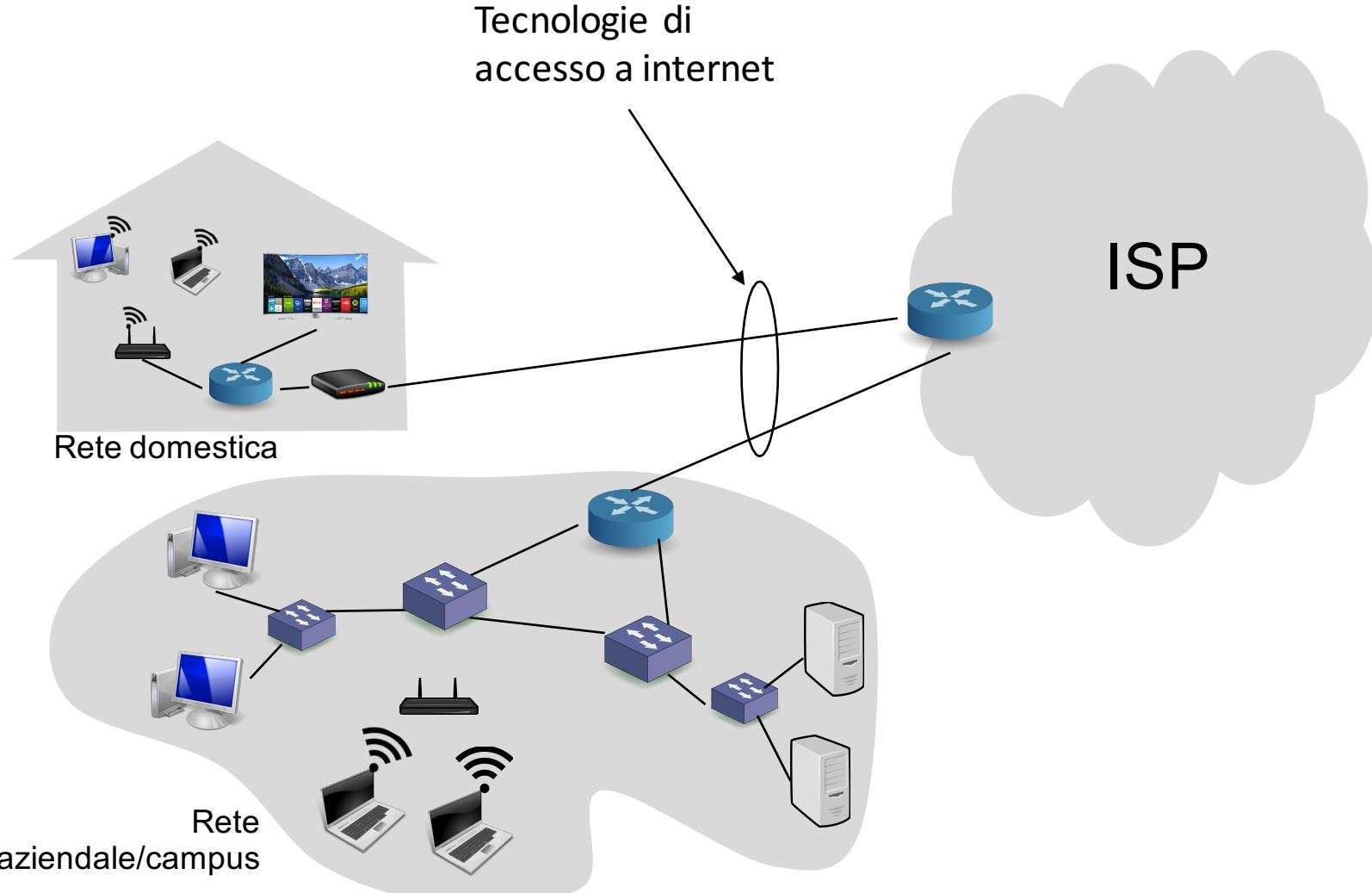
switch



access point

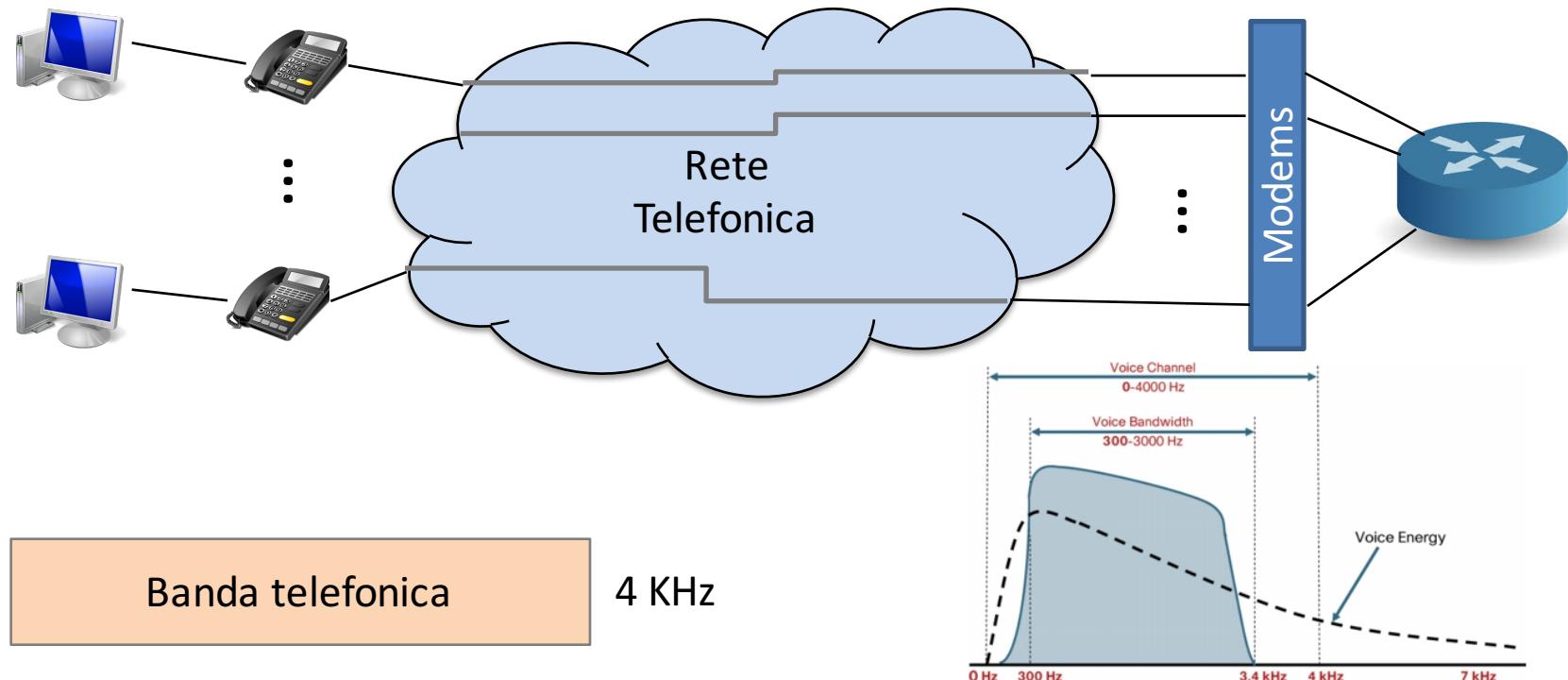


Architettura fisica: Accesso a Internet



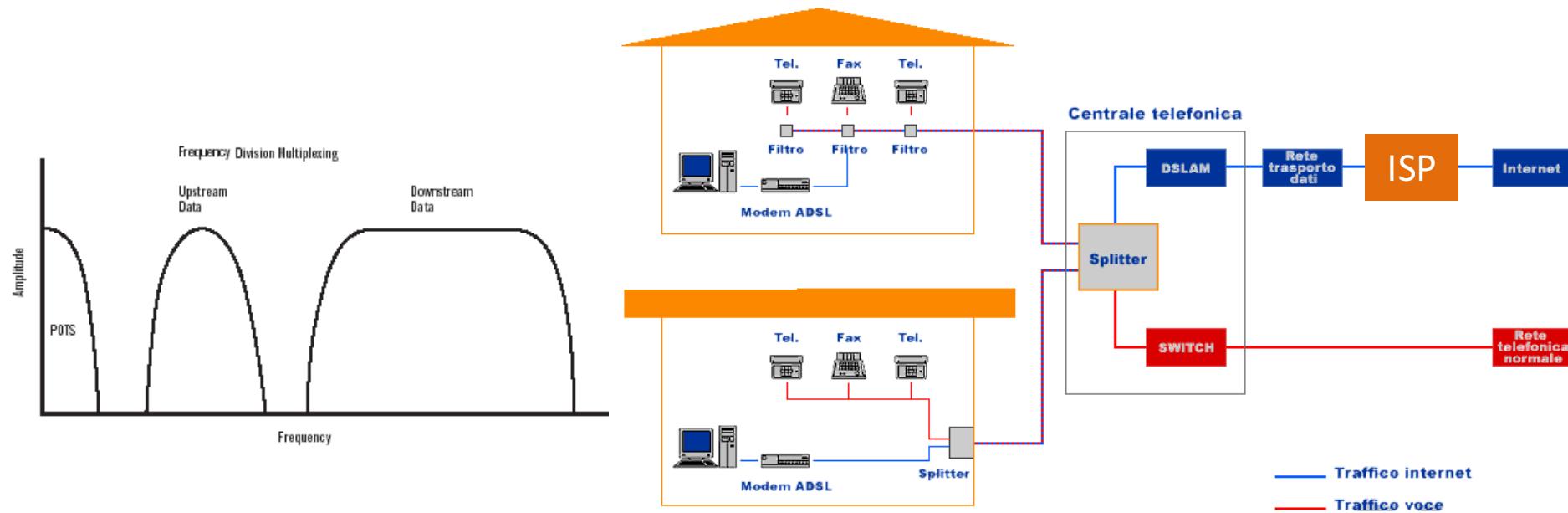
Accesso a Internet: Dialup

- **Dialup via modem**
 - Fino a 56Kbps
 - Accesso diretto al router del ISP mediante circuito telefonico
 - Trasmissione del segnale in banda fonica



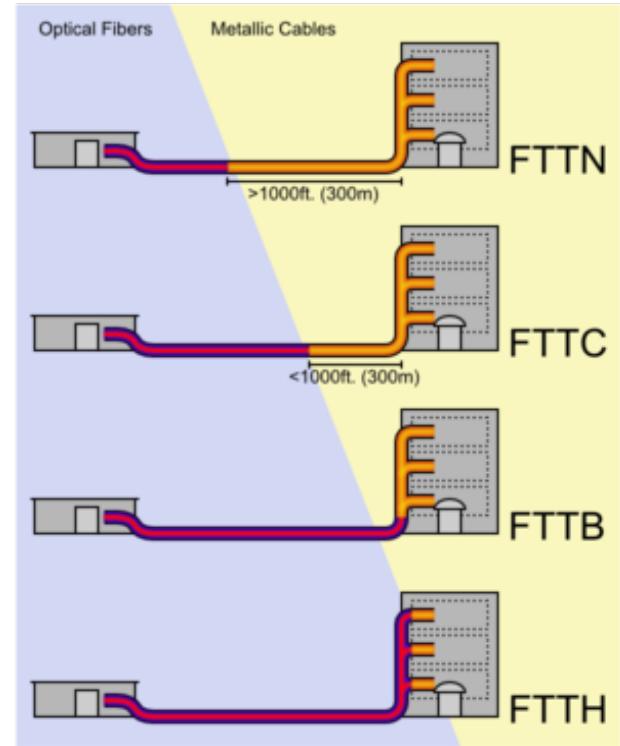
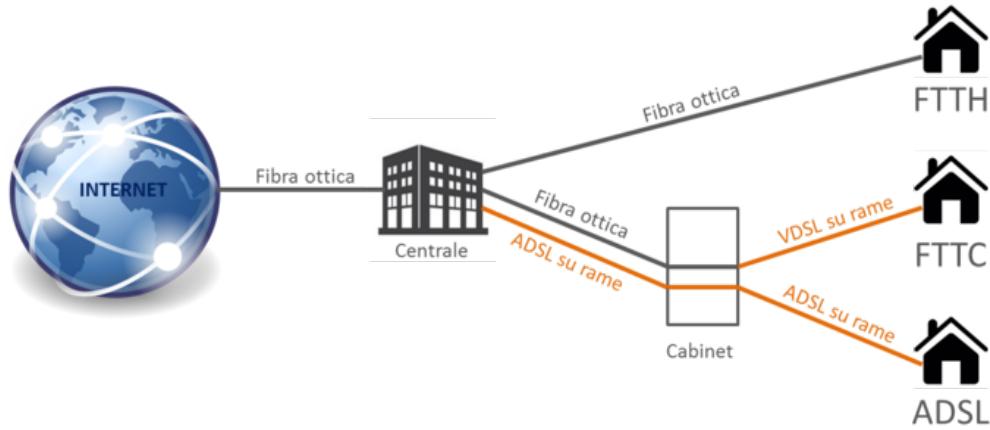
Accesso a Internet: ADSL

- **ADSL: asymmetric digital subscriber line**
 - Fino a 1 Mbps upstream, Fino a 20 Mbps downstream
 - Condivisione del doppino con la rete telefonica fino alla centrale (divisione di frequenza)
 - Accesso al router del provider mediante rete dati ad alta velocità



Accesso a Internet: Fibra

- Rete d'accesso di nuova generazione ad alta velocità
 - Sostituzione parziale o totale del doppino telefonico con fibra ottica



FTTH - Fiber To The Home

FTTB - Fiber To The Basement

FTTC - Fiber To The Curb

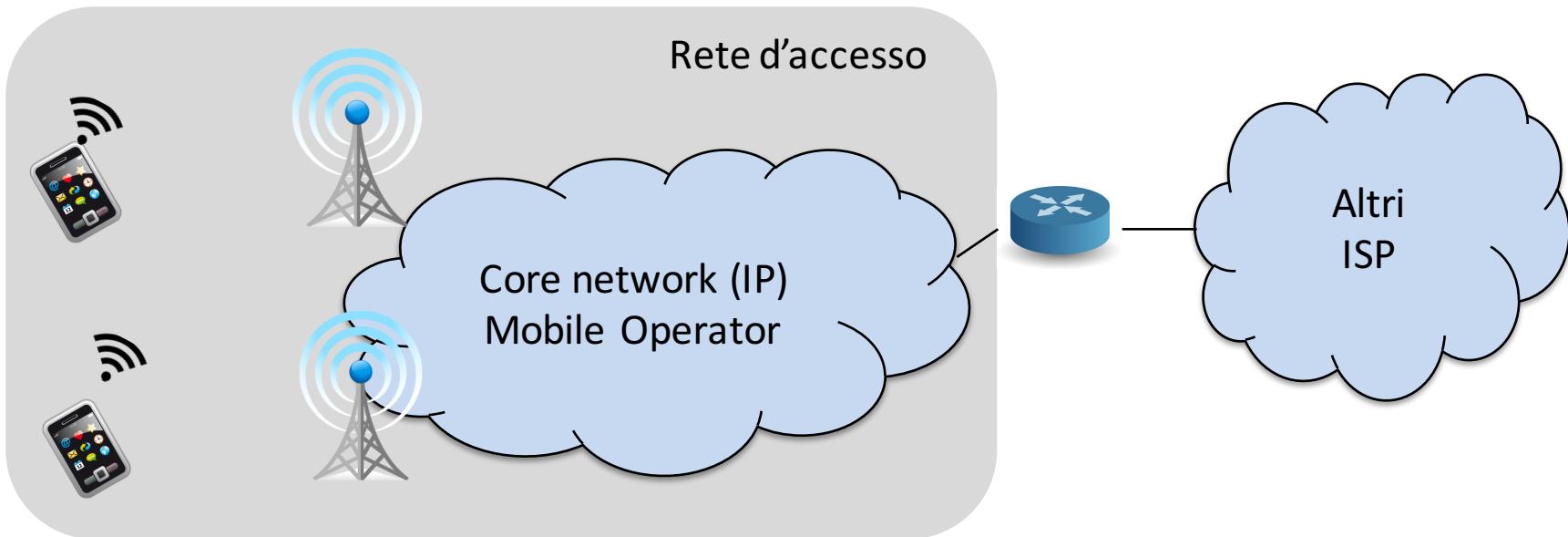
FTTN - Fiber To The Neighborhood



Accesso a Internet: Rete cellulare

- **Reti cellulari**

- GPRS/EDGE ~ 200 kbps
- HSPA ~ 14,5(down)/5,7(up) Mbps
- LTE ~ 300(down)/85(up) Mbps



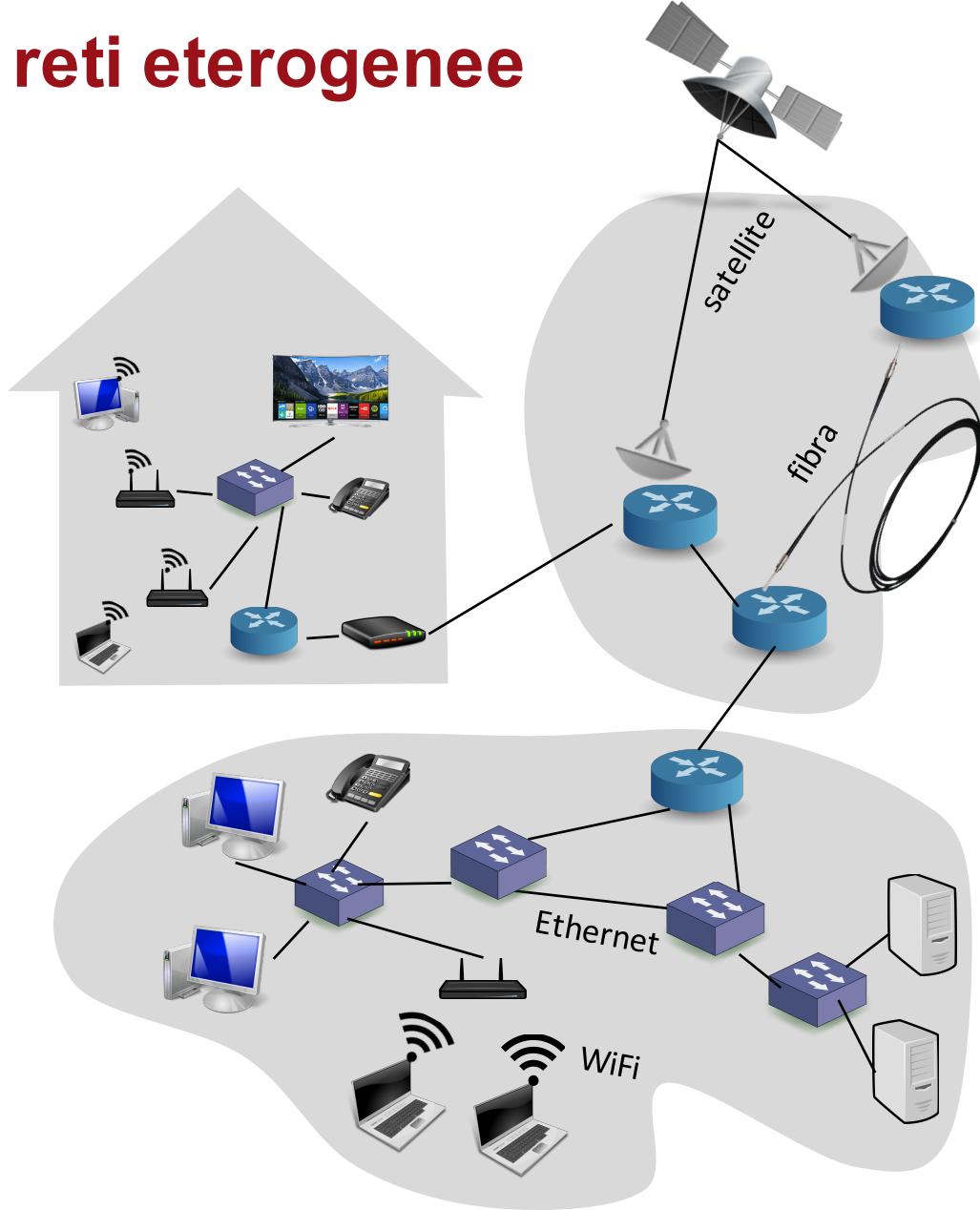
Reti di reti

- Nella descrizione di cosa sia Internet da punto di vista fisico e di servizio abbiamo trascurato una aspetto architettonale fondamentale: **Internet in realtà è un puzzle di tante reti interconnesse**
- **Questo ha due risvolti importanti:**
 - 1) La tecnologia di Internet (IP - Internet Protocol) può essere usata per interconnettere sotto-reti di tipo eterogeno
 - 2) L'intera rete Internet mondiale è composta da tante reti gestite da operatori indipendenti (ISP – Internet Service Provider) che si accordano per collegarle insieme



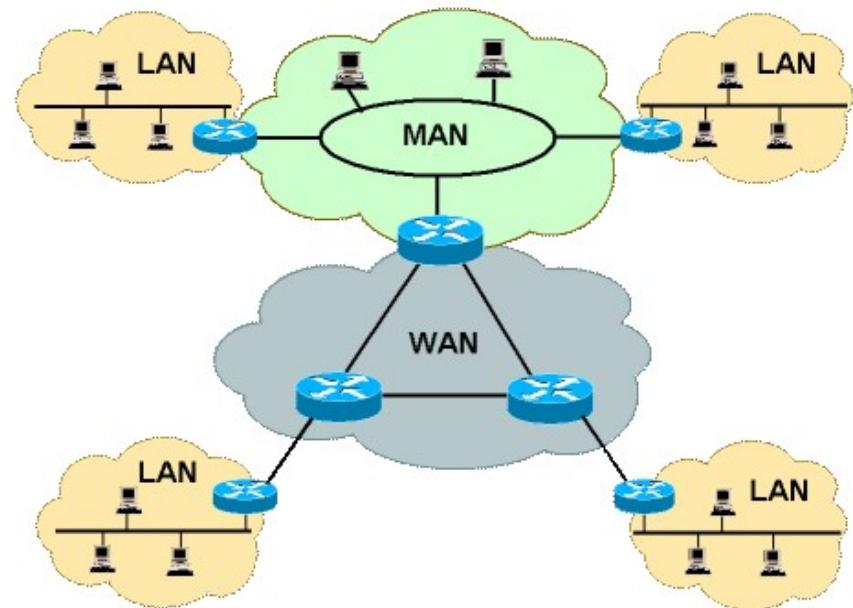
Reti di reti: Insieme di reti eterogenee

- Le diverse porzioni di rete sono composte da **tecnicologie diverse**
- I router possono essere interconnessi da **link di vario tipo**
- Ma anche da “**sotto-reti**” che gestiscono internamente propri nodi e link
- Esempio: reti locali Ethernet e/o WiFi

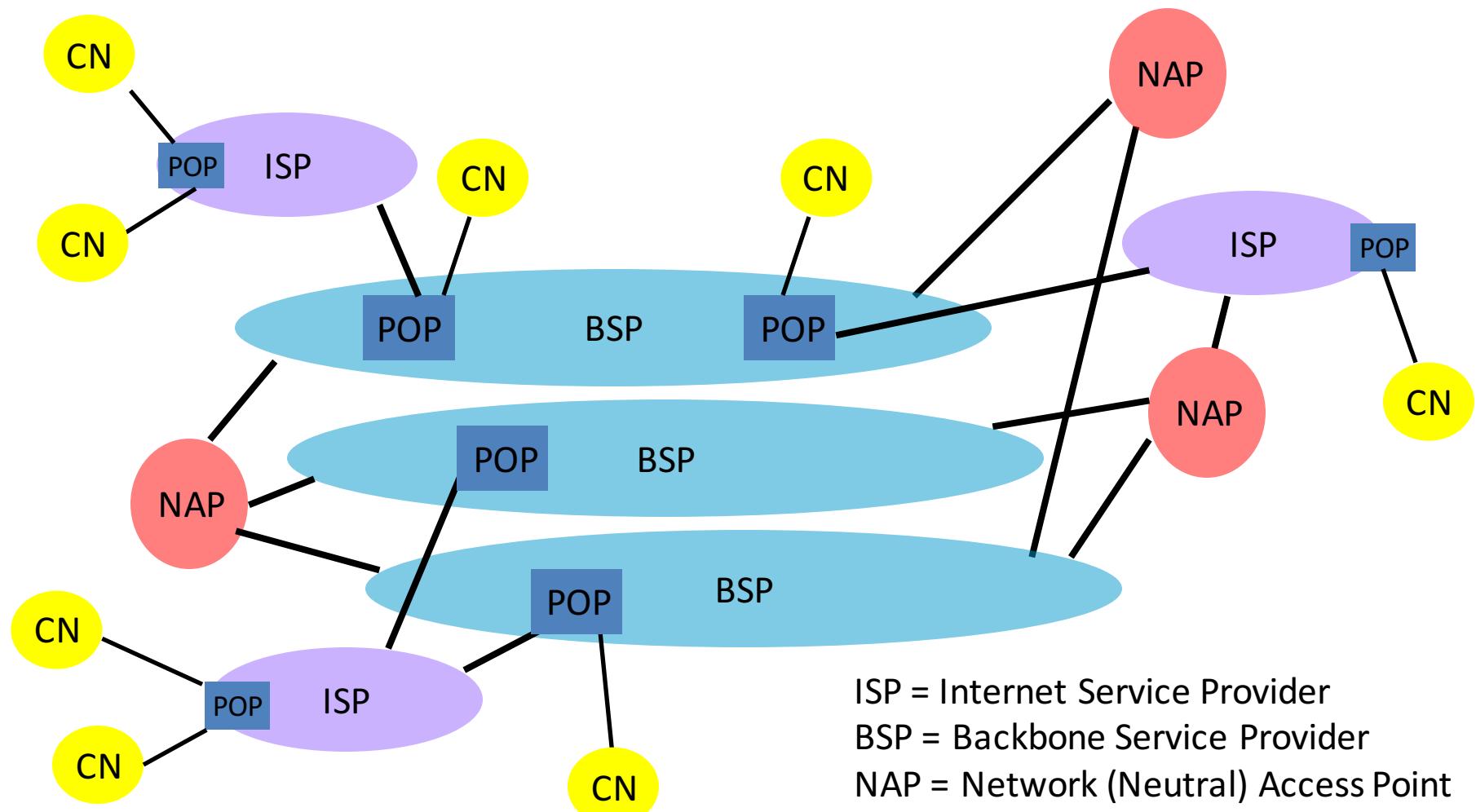


Rete di reti: insieme di reti eterogenee

- Così come Ethernet e WiFi, si possono usare tipi di reti diversi non solo in ambito locale
- **Tipi di rete in base all'estensione geografica:**
 - **LAN**: Local Area Network
 - Impiegate in aree limitate (tipicamente edifici, campus)
 - **MAN**: Metropolitan Area Network
 - Coprono estensioni fino ad alcune decine di km
 - **WAN**: Wide Area Network
 - Hanno copertura ampia a piacere



Rete di reti: architettura di interconnessione



ISP = Internet Service Provider

BSP = Backbone Service Provider

NAP = Network (Neutral) Access Point

POP = Point of Presence

CN = Customer Network



Reti di reti: Internet Exchange Map



TeleGeography Internet Exchange Map

The Internet Exchange Map is a free resource from [TeleGeography](#). Data contained in this map was compiled by TeleGeography and is updated on a regular basis.

To learn more about [TeleGeography](#) or this map, please visit www.telegeography.com.



Sponsored by Datamena

Feedback

Search

Internet Exchanges

- 6NGIX (Seoul, Korea, Rep.)
- AAIX (Klagenfurt, Austria)
- ADN-IX (Valence, France)
- AIXP (Arusha, Tanzania)
- AIXP (Port-au-Prince, Haiti)
- ALB-IX (Tirane, Albania)
- AMPATH (Miami, United States)
- AMS-IX (Amsterdam, Netherlands)
- AMS-IX Bay Area (San Francisco, United States)
- AMS-IX Caribbean (Willemstad, Netherlands Antilles)

All content © 2015 PriMetrica, Inc.

telegeography.com



POLITECNICO MILANO 1863

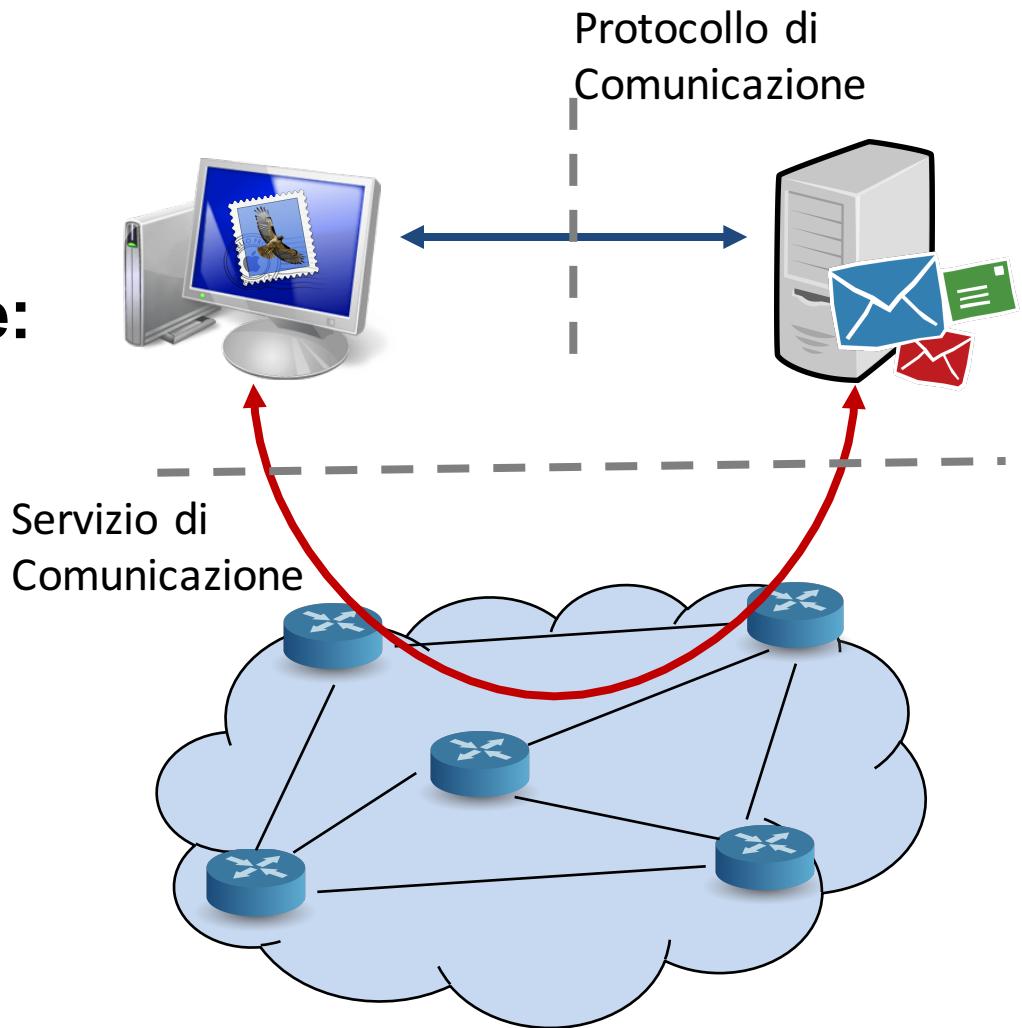
FIR:1 - Introduzione e architetture

63

Cos'è Internet?

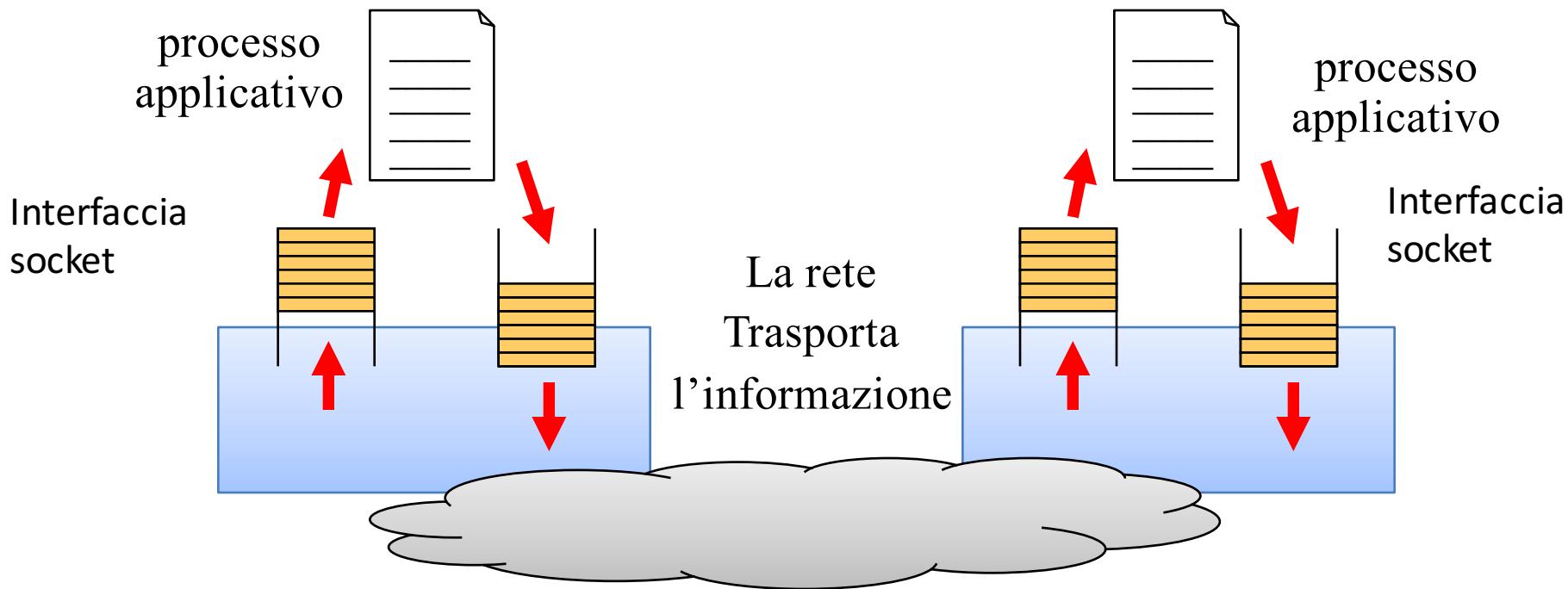
Il servizio e i protocolli di comunicazione

- **Infrastruttura di comunicazione consente le applicazioni distribuite:**
 - Web, email, games, e-commerce, file sharing
- **Protocolli di comunicazione per inviare e ricevere messaggi**



Servizio di Comunicazione

- La rete fornisce un servizio di comunicazione alle applicazioni per il trasporto delle informazioni tra i processi remoti
- Il servizio di trasporto offerto dalla rete alle applicazioni può essere di vari tipi

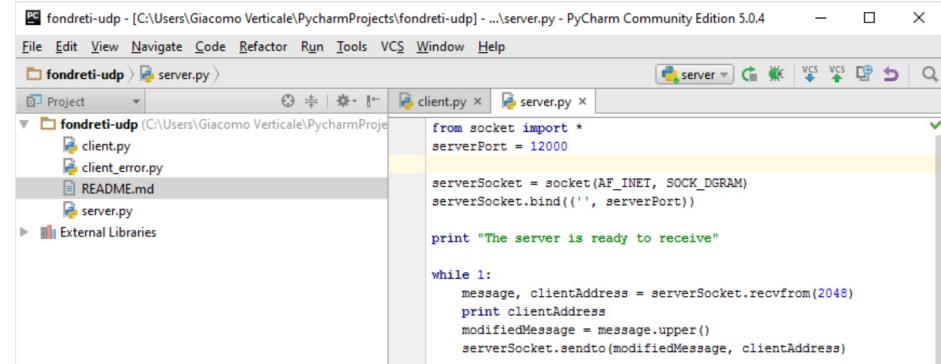


Servizio di Comunicazione

- Possono essere trasportati brevi messaggi in modo non affidabile (esempi: DNS, segnalazione, ecc.)
- Possono essere trasportate sequenze anche lunghe di byte in modo affidabile (web, email, file transfer, ecc.)

Laboratorio: Socket programming

Vedrete in laboratorio come usare Python per accedere all'interfaccia software sia lato client che lato server per trasporto affidabile e non



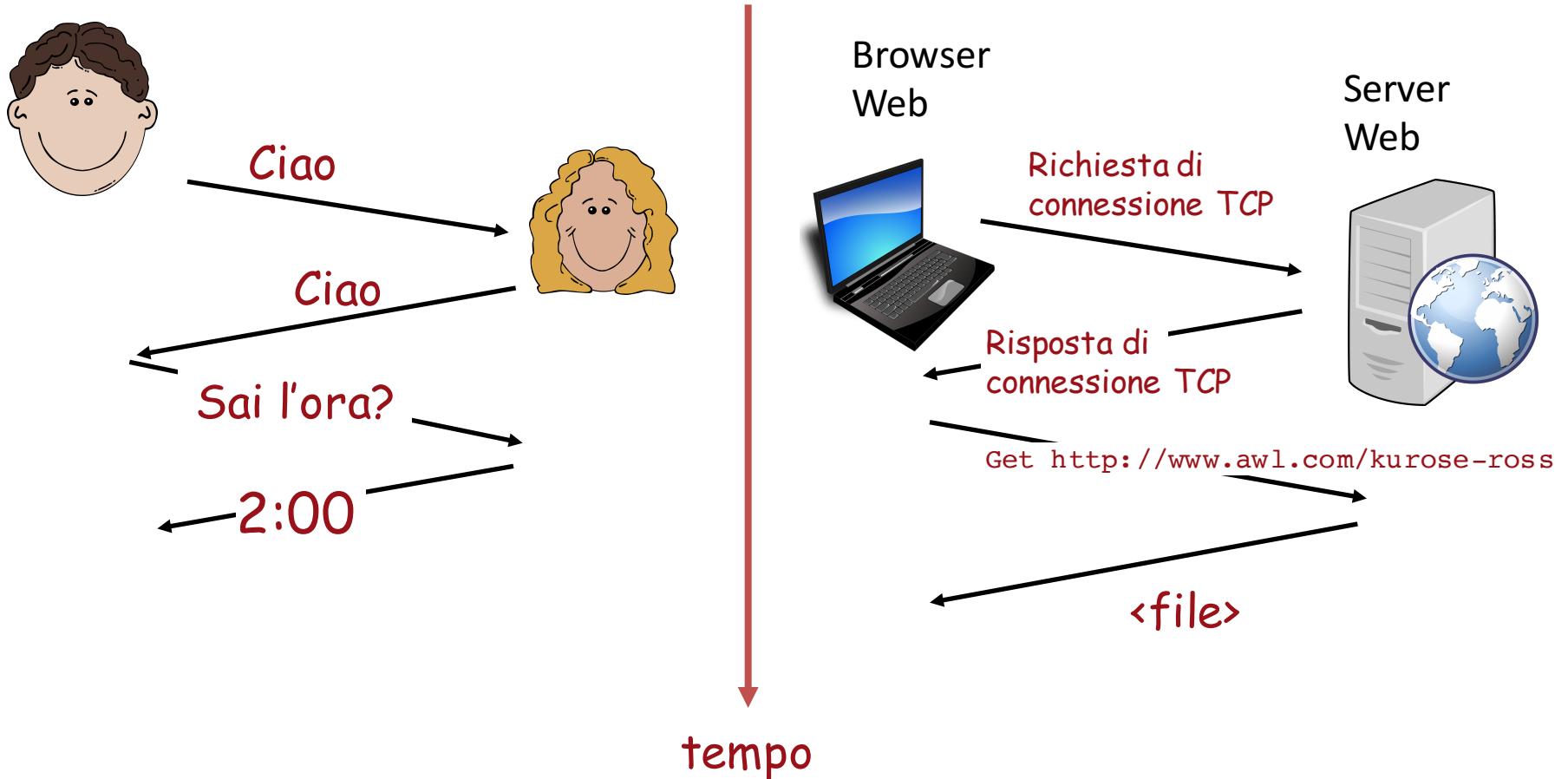
The screenshot shows the PyCharm Community Edition 5.0.4 interface. The project navigation bar indicates the current file is 'server.py'. The left sidebar shows a project structure with files: 'client.py', 'client_error.py', 'README.md', and 'server.py'. The right pane displays the code for 'server.py':

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(('', serverPort))
print "The server is ready to receive"
while 1:
    message, clientAddress = serverSocket.recvfrom(2048)
    print clientAddress
    modifiedMessage = message.upper()
    serverSocket.sendto(modifiedMessage, clientAddress)
```



Protocolli di comunicazione

Protocollo umano e protocollo di rete



Protocolli di comunicazione: Esempio posta elettronica

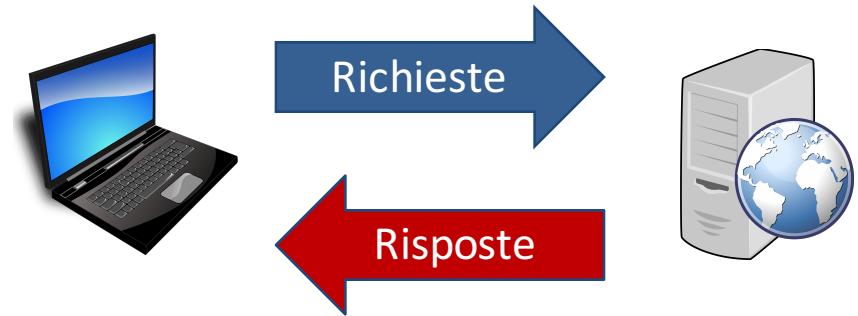
```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```



Protocolli di comunicazione: modelli

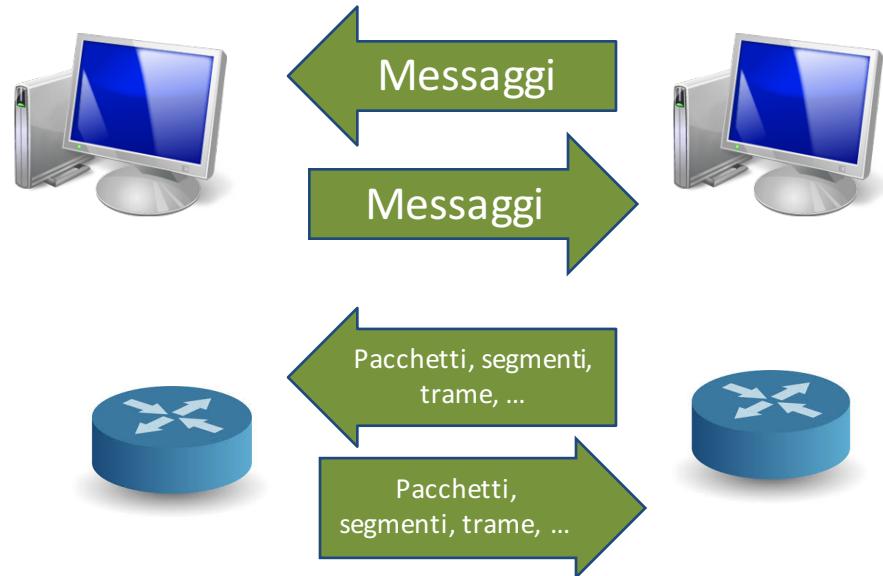
- **Modello client/server**

- client chiedono il servizio, i server lo forniscono
- I client fanno domande, i server rispondono



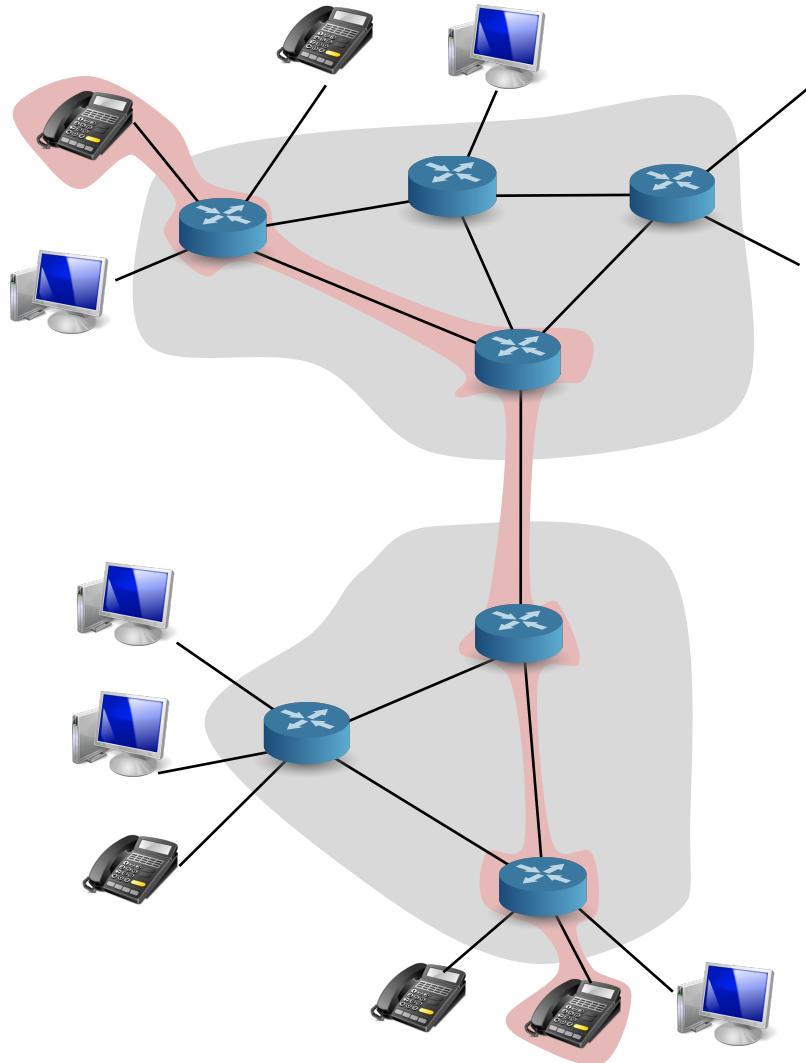
- **Modello peer-to-peer:**

- Tutti i terminali collaborano senza distinzione di ruoli (o quasi)



Come funziona Internet?

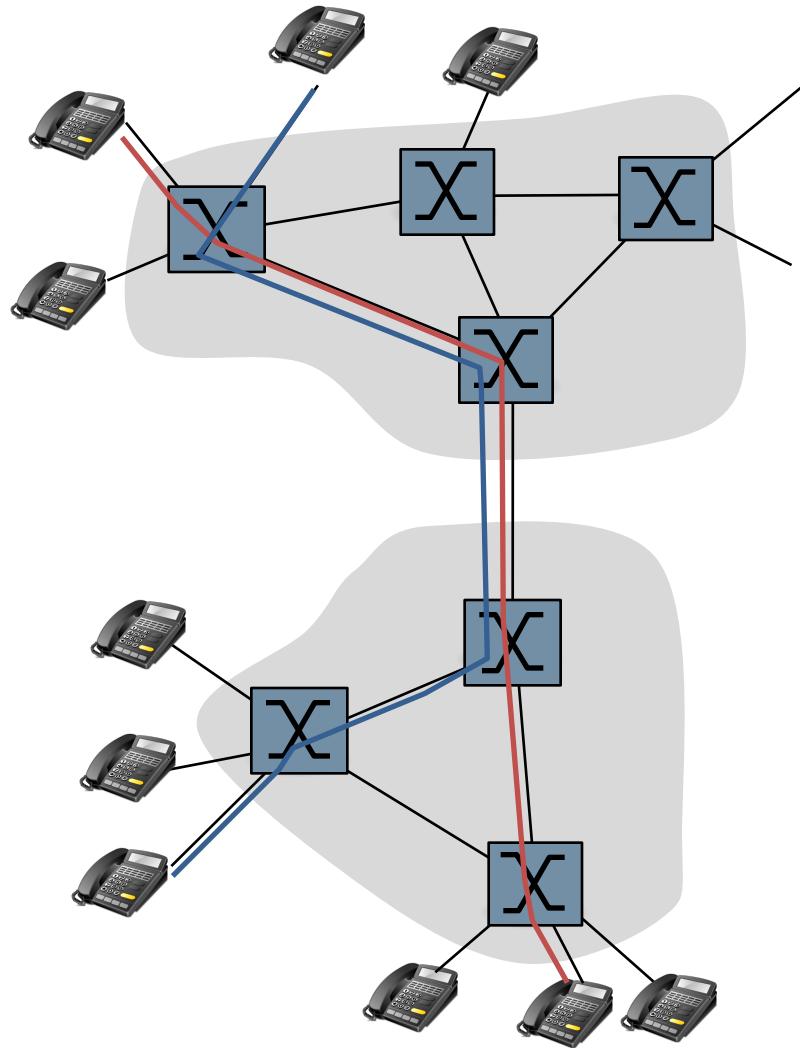
- E' la domanda che ci occuperà l'intero corso
- Ma partiamo da meccanismo di base
- **Come può essere trasferita l'informazione in rete?**
 - Comutazione di circuito: circuito dedicato per chiamata
 - **Comutazione di pacchetto:** dati inviati in rete con messaggi



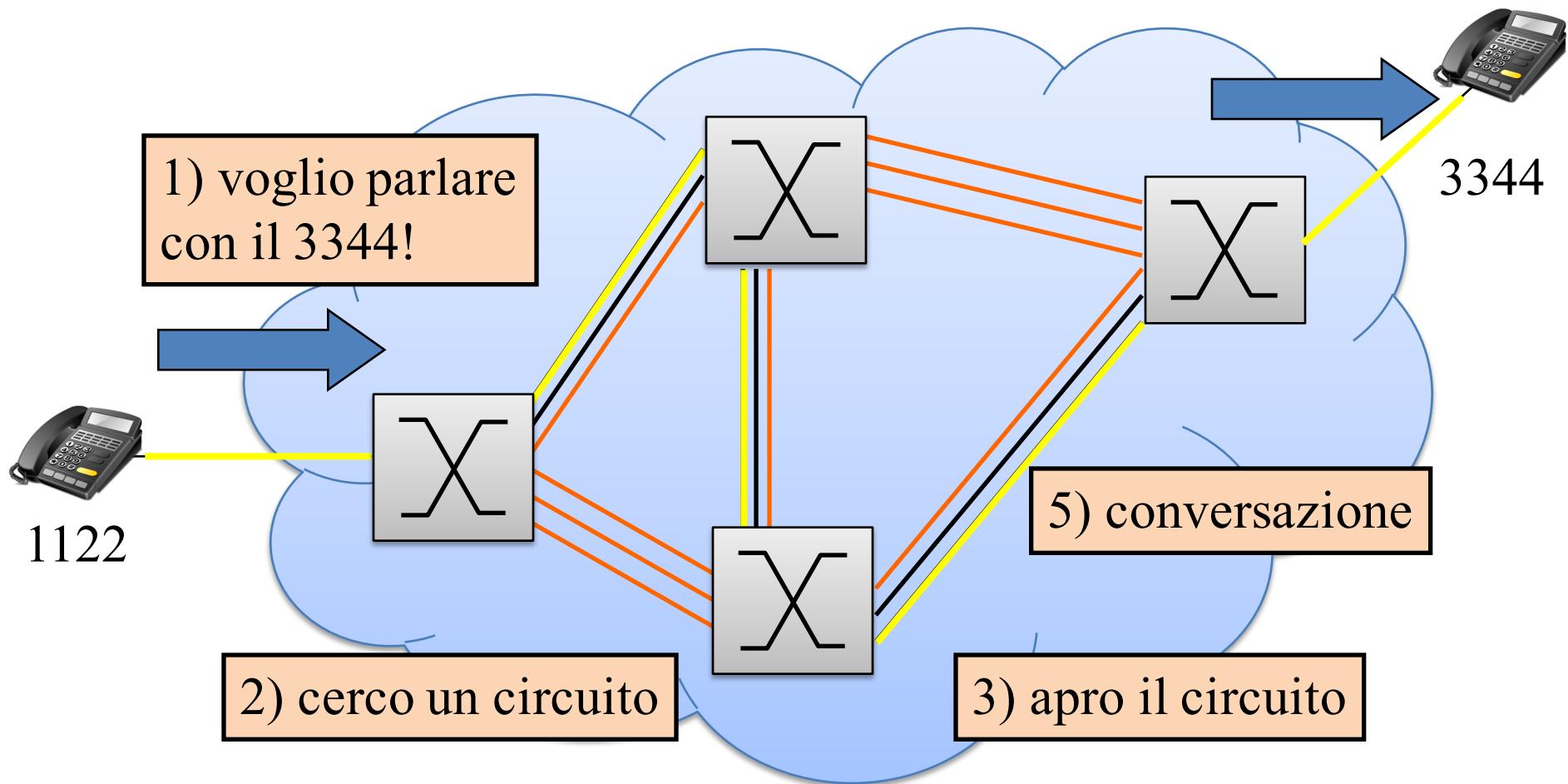
Commutazione di circuito

Le risorse per la comunicazione sono riservate per la chiamata

- Esempio rete telefonica



Commutazione di circuito



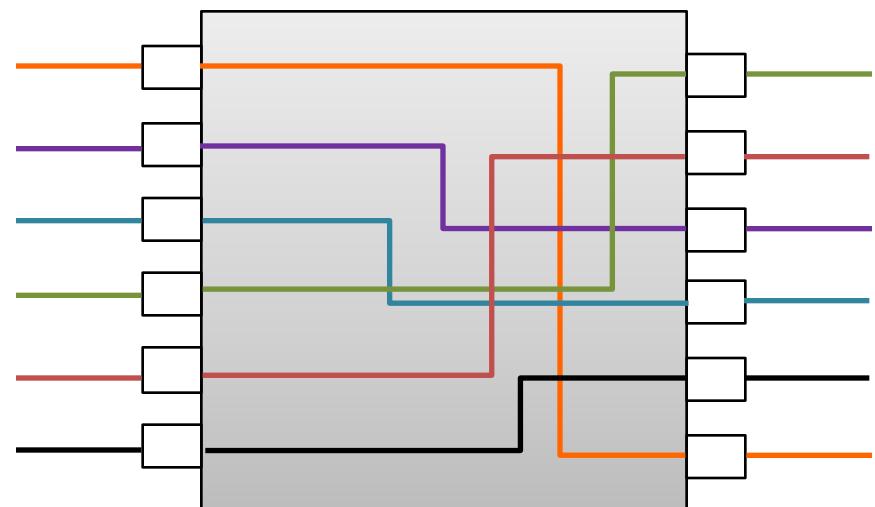
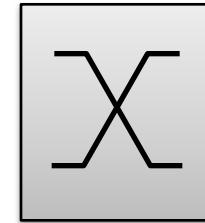
Commutazione di circuito

- **Risorse di rete (collegamenti) suddivise in “pezzi”**
 - ciascun “pezzo” (= circuito) viene allocato ai vari collegamenti
 - le risorse rimangono *inattive* se non utilizzate (*non c'è condivisione*)
- suddivisione della banda in “pezzi”
- divisione di frequenza
 - divisione di tempo



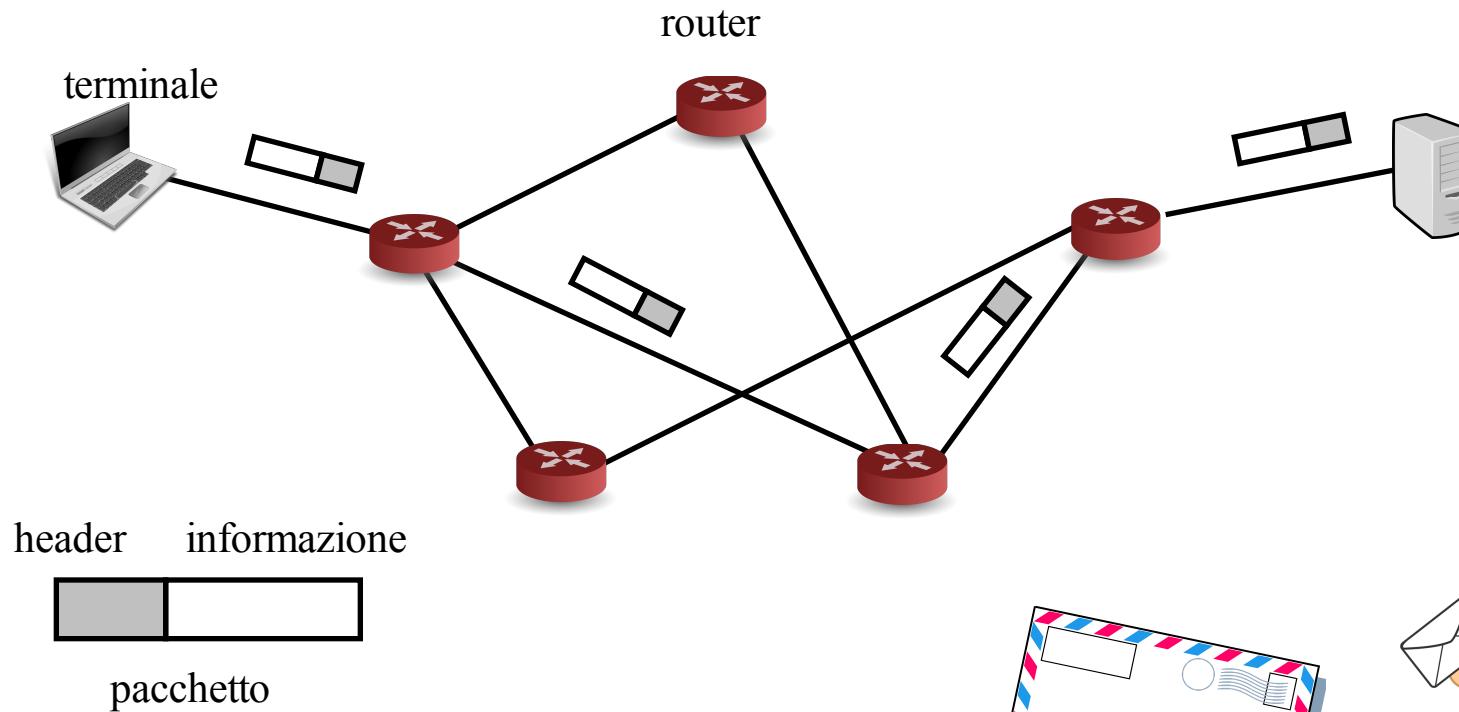
Commutazione di circuito

- **Modello di nodo
(comutatore a
circuito)**
 - La capacità dei canali in ingresso è pari alla capacità (in bit al secondo) di quelli in uscita
 - Non serve memorizzare temporaneamente l'informazione

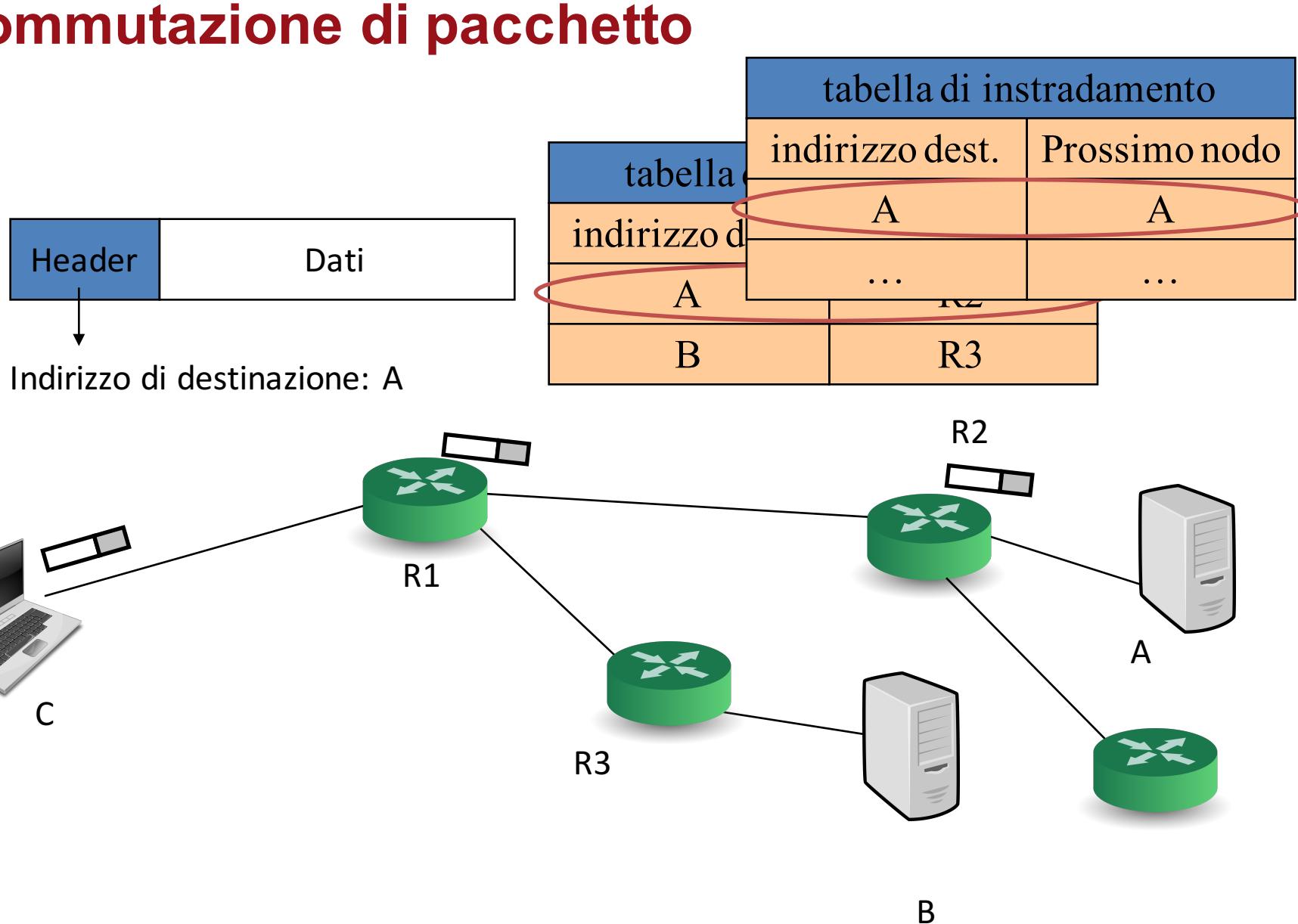


Commutazione di pacchetto

- **Informazione suddivisa in pezzi**
- **Collegamenti non suddivisi**



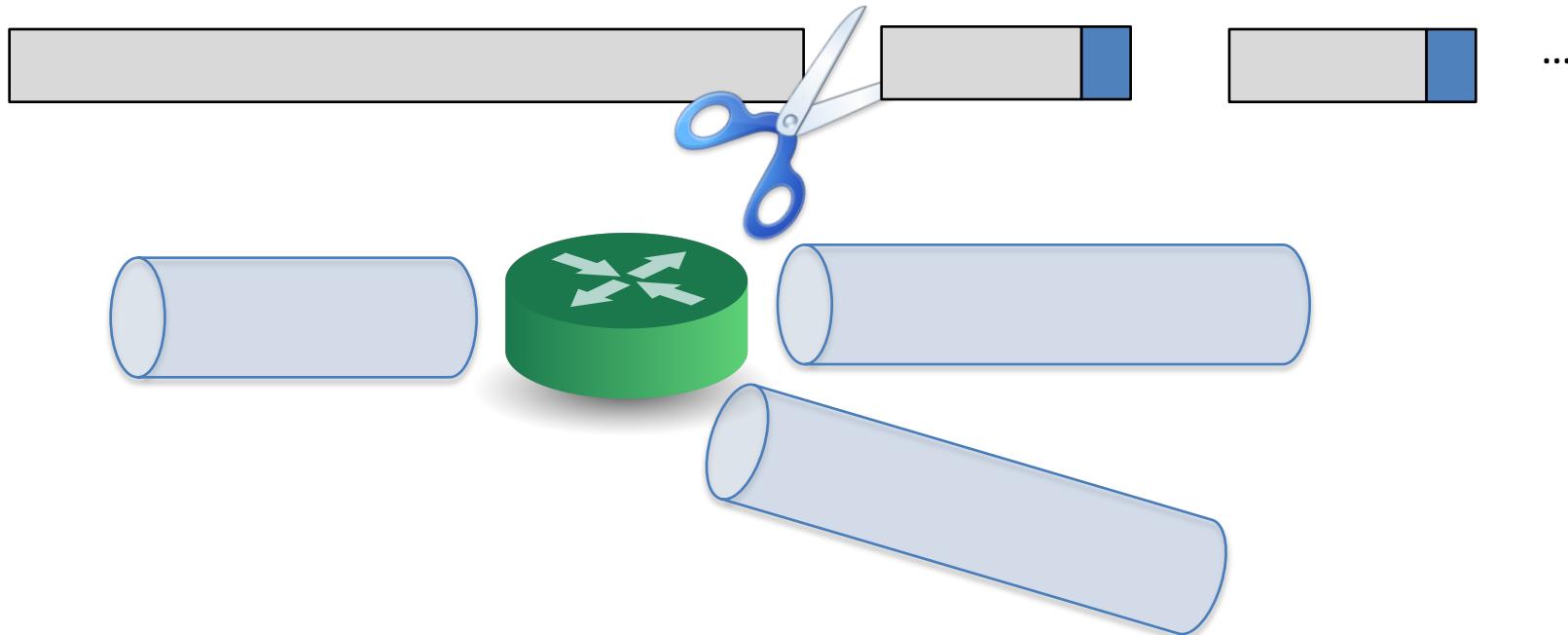
Commutazione di pacchetto



Commutazione di pacchetto

Il flusso di dati viene suddiviso in pacchetti

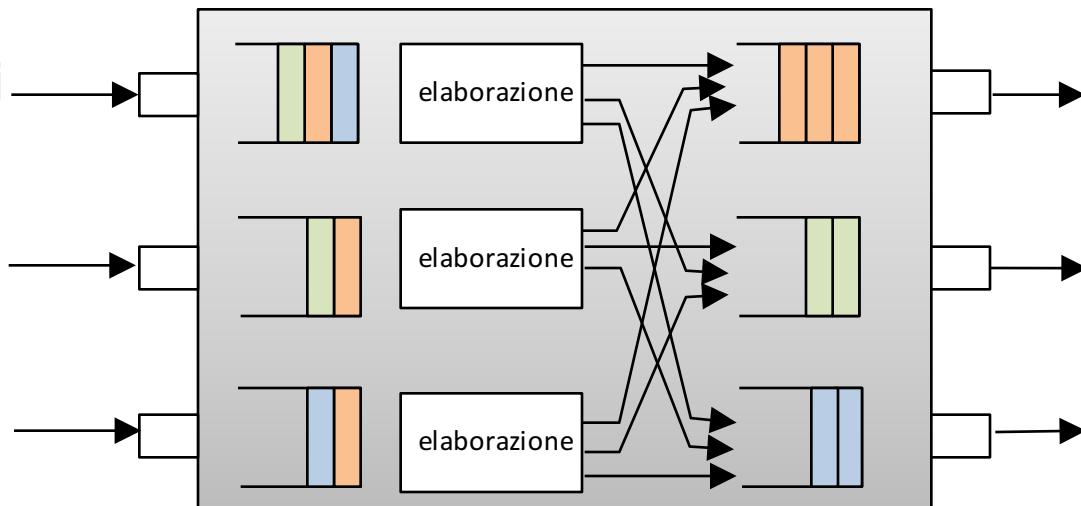
- I pacchetti di tutti gli utenti *condividono* le risorse di rete
- Ciascun pacchetto utilizza completamente il canale
- Le risorse vengono usate a seconda delle necessità



Commutazione di pacchetto

- **Modello di nodo
(packet switch/router)**

- L'arrivo dei pacchetti è **asincrono**
- La capacità dei collegamenti **arbitraria**
- Possono esserci **conflitti temporali** per la trasmissione
- Serve **memorizzare temporaneamente (coda)**
 - **All'ingresso** per analizzare indirizzo destinazione
 - **All'uscita** per gestire conflitti

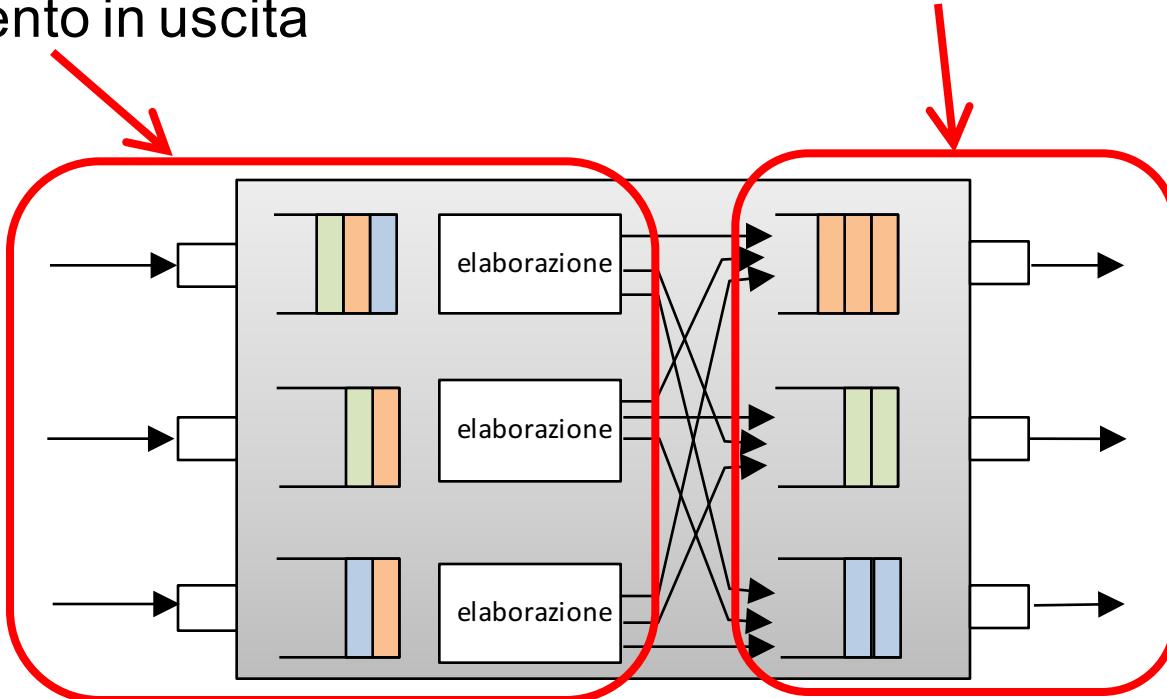


Commutazione di pacchetto

Contesa per le risorse:

store and forward: il commutatore deve ricevere l'intero pacchetto prima di poter cominciare a trasmettere sul collegamento in uscita

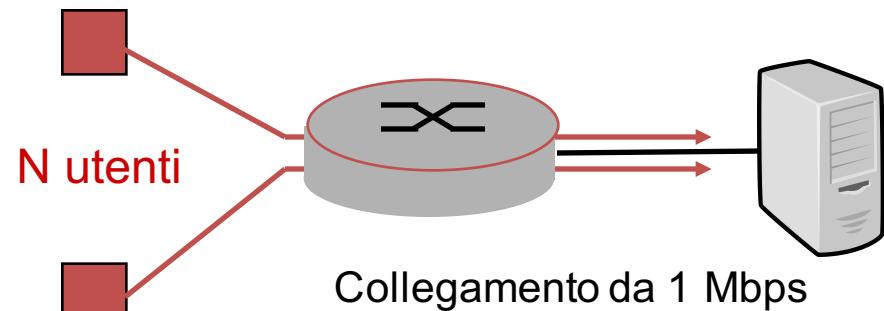
Multiplazione statistica: accodamento dei pacchetti, attesa per l'utilizzo del collegamento



Confronto tra pacchetto e circuito

Esempio:

- 1 collegamento da 1 Mbps
- Ciascun utente:
 - Genera 100 kbps quando è “attivo”
 - È attivo per il 10% del tempo
- **commutazione di circuito:**
 - 10 utenti ($1 \text{ Mbps} / 100 \text{ kbps} = 10$)



- **commutazione di pacchetto:**
 - con 35 utenti, la probabilità di averne > 10 attivi è inferiore a 0,0004
(risultato di teoria della prob.)

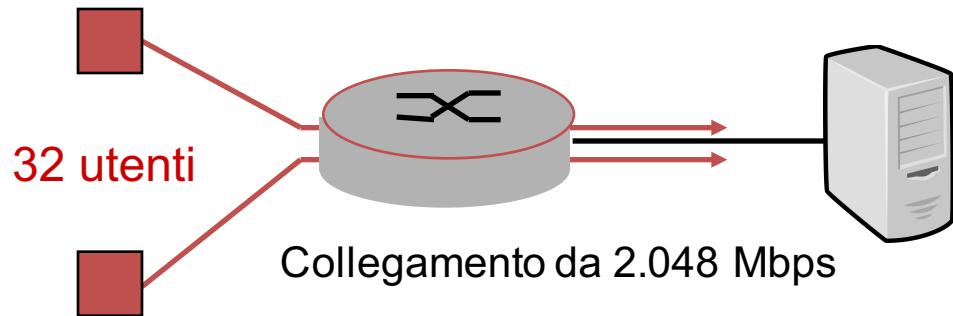
La commutazione di pacchetto consente a più utenti di usare la rete!



Confronto tra pacchetto e circuito

Esempio:

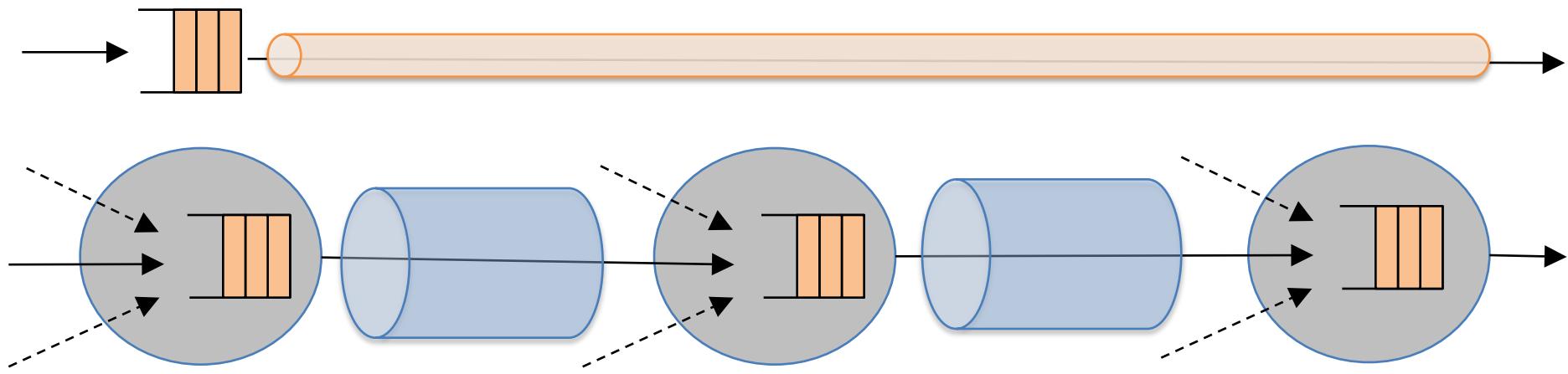
- 1 collegamento da 2.048 Mbps
- 32 utenti, ciascun utente:
 - Chiede pagine web di 50KB ogni 62.5s in media
- **commutazione di circuito:**
 - 1 canale 64 kbps per utente
 - Ritardo di trasferimento pagina web: 6.25s
 $(400 \text{ kbit} / 64 \text{ kbps} = 6,25\text{s})$
- **commutazione di pacchetto:**
 - Ritardo di trasferimento medio pagina web: 0.22s
(risultato di teoria delle code)



La commutazione di pacchetto consente di scaricare le informazioni più velocemente!



Confronto tra pacchetto e circuito



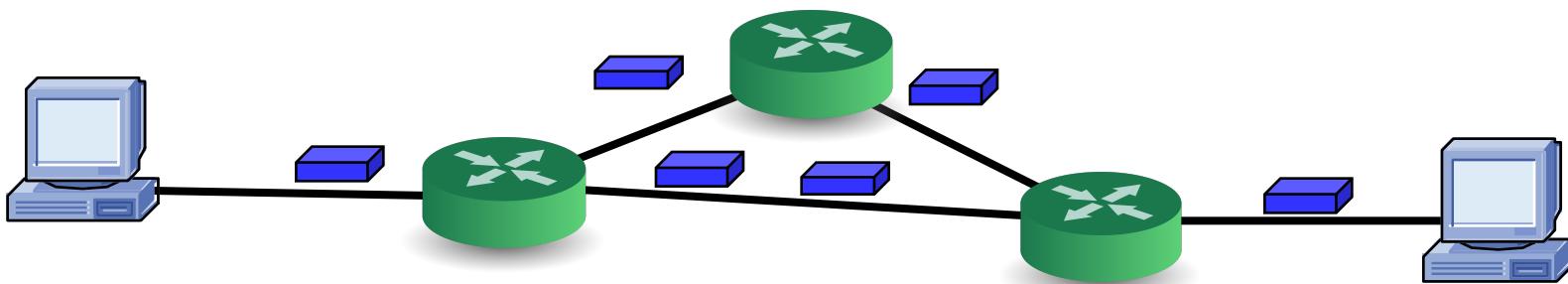
La commutazione di pacchetto è la scelta di Internet

- **Il problema delle coda: ritardo e perdita di pacchetti**
 - Sono necessari protocolli per il trasferimento affidabile dei dati e per il controllo della congestione



Commutazione di pacchetto: datagram e circuito virtuale

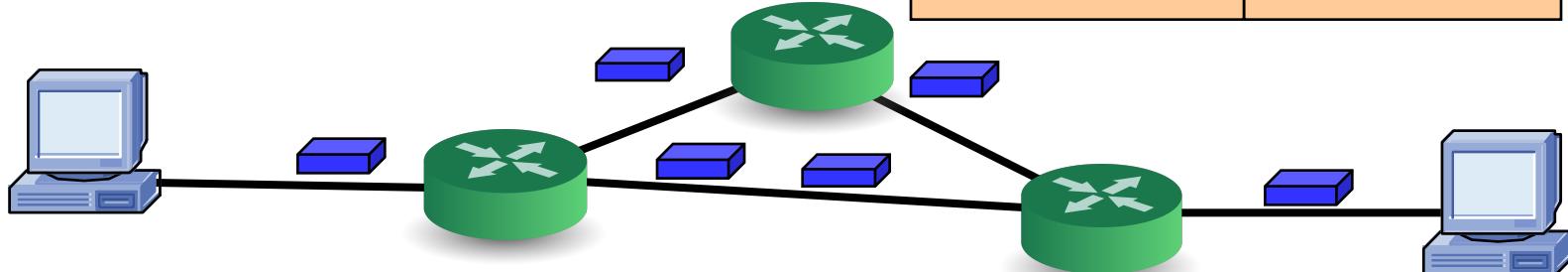
- **Esistono due tipi di commutazione di pacchetto:**
 - Datagram
 - Circuito virtuale



Commutazione di pacchetto: Datagram

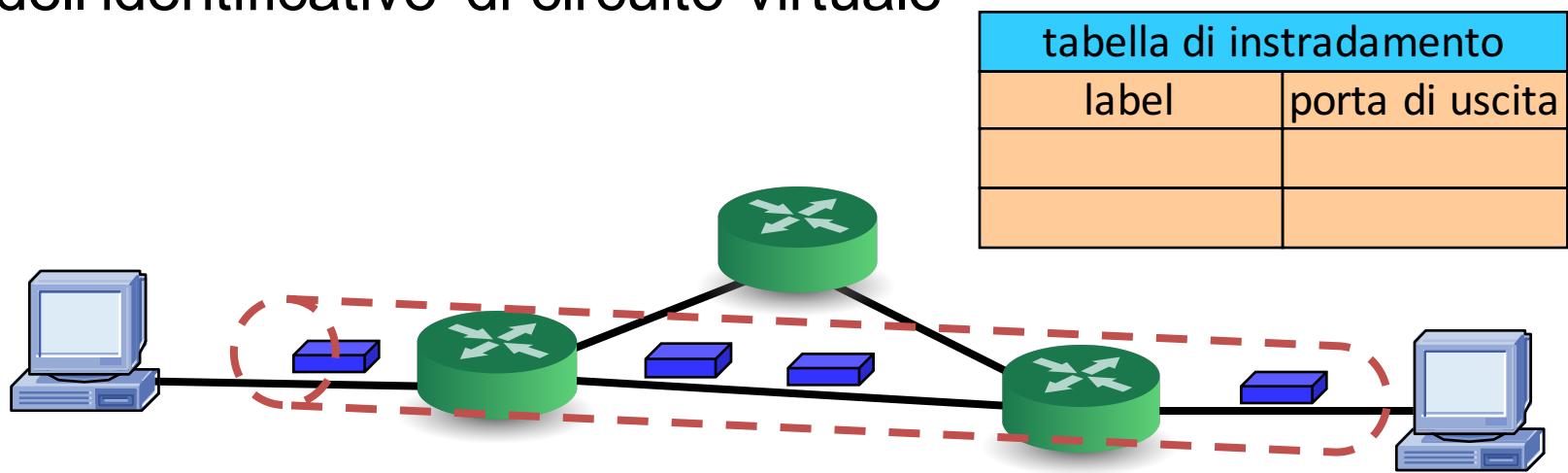
- Nella commutazione datagram la scelta della porta d'uscita viene fatta sulla base del solo indirizzo di destinazione
- I pacchetti dello stesso flusso sono inoltrati indipendentemente

tabella di instradamento	
destinazione	porta di uscita



Commutazione di pacchetto: Circuito virtuale

- I nodi identificano i pacchetti di un flusso informativo sulla base di un identificativo di circuito virtuale (CVI o label)
- Il circuito virtuale viene instaurato in una fase di setup prima della fase dati
- Dopo la fase di setup i pacchetti seguono tutti lo stesso percorso in rete perché sono instradati sulla base dell'identificativo di circuito virtuale



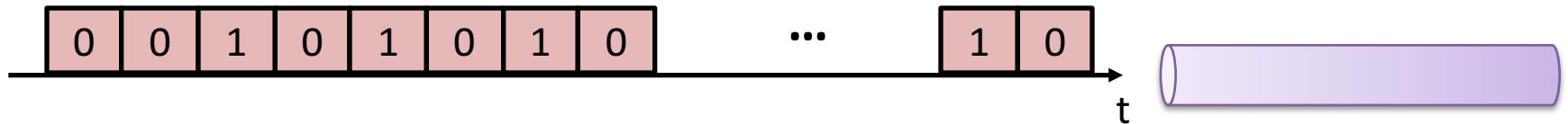


1c – Ritardi e Throughput

Tempo di trasmissione, Propagazione,
Altri ritardi, Throughput

Velocità di trasmissione

- E' la **velocità (rate) R** con la quale l'informazione digitale viene trasmessa su una linea
- E' misurata in **bit/s (bps)**
(dopo scopriremo da cosa è determinata)



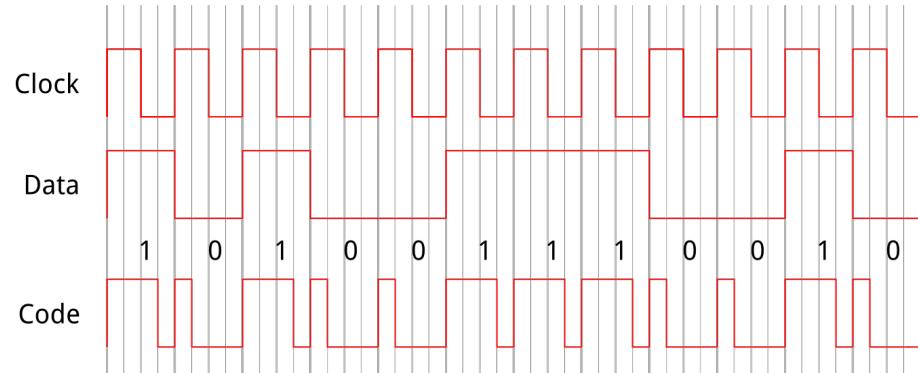
- **Unità di misura:**

	$1 \text{ B} = 8 \text{ bit}$
$1 \text{ kbps (kb/s)} = 10^3 \text{ bps}$	$1 \text{ kB} = 10^3 \text{ B}$
$1 \text{ Mbps (Mb/s)} = 10^6 \text{ bps}$	$1 \text{ MB} = 10^6 \text{ B}$
$1 \text{ Gbps (Gb/s)} = 10^9 \text{ bps}$	$1 \text{ GB} = 10^9 \text{ B}$



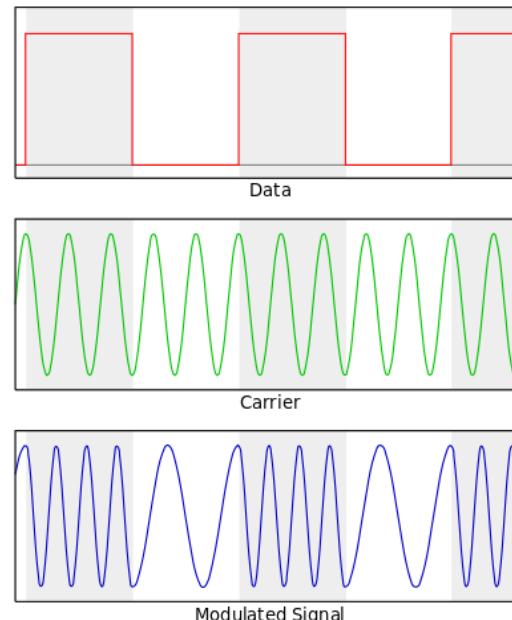
Velocità di trasmissione: esempi

- **Segnale nel tempo di Ethernet con codifica Manchester:**



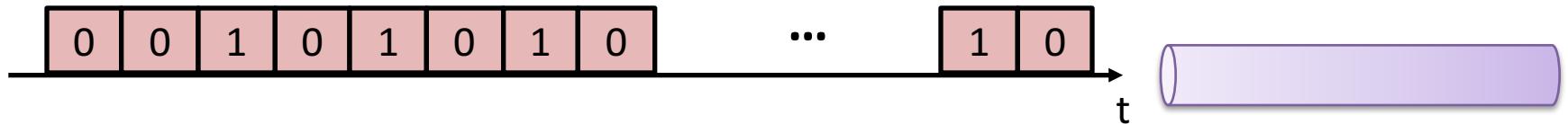
- **Segnale modulato in frequenza del GSM:**

NOTA: Il motivo per il quale la durata del singolo bit non può essere piccolissima (e quindi la velocità grandissima) deriva a limiti dei canali trasmissivi che saranno trattati nell'ultima parte del corso.



Velocità di trasmissione

- E' la **velocità (rate) R** con la quale l'informazione digitale viene trasmessa su una linea
- E' misurata in **bit/s (bps)**
(dopo scopriremo da cosa è determinata)



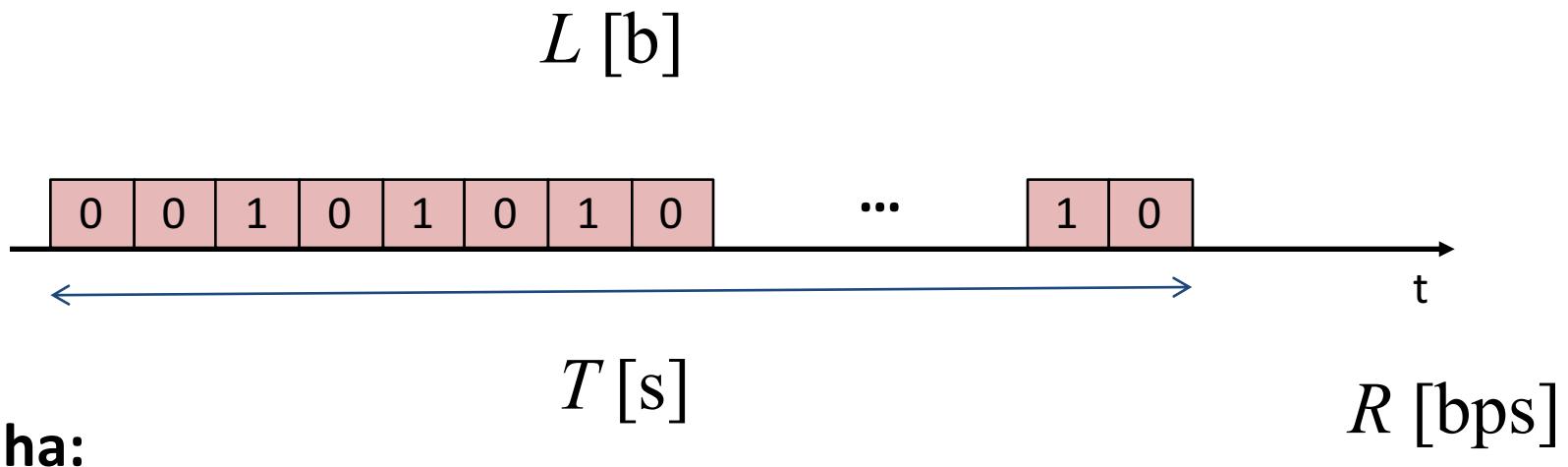
- **Unità di misura:**

$1 \text{ B} = 8 \text{ bit}$	
$1 \text{ kbps (kb/s)} = 10^3 \text{ bps}$	$1 \text{ kB} = 10^3 \text{ B}$
$1 \text{ Mbps (Mb/s)} = 10^6 \text{ bps}$	$1 \text{ MB} = 10^6 \text{ B}$
$1 \text{ Gbps (Gb/s)} = 10^9 \text{ bps}$	$1 \text{ GB} = 10^9 \text{ B}$



Tempo di trasmissione

- Il tempo T per trasmettere L bits dipende dalla velocità di trasmissione R



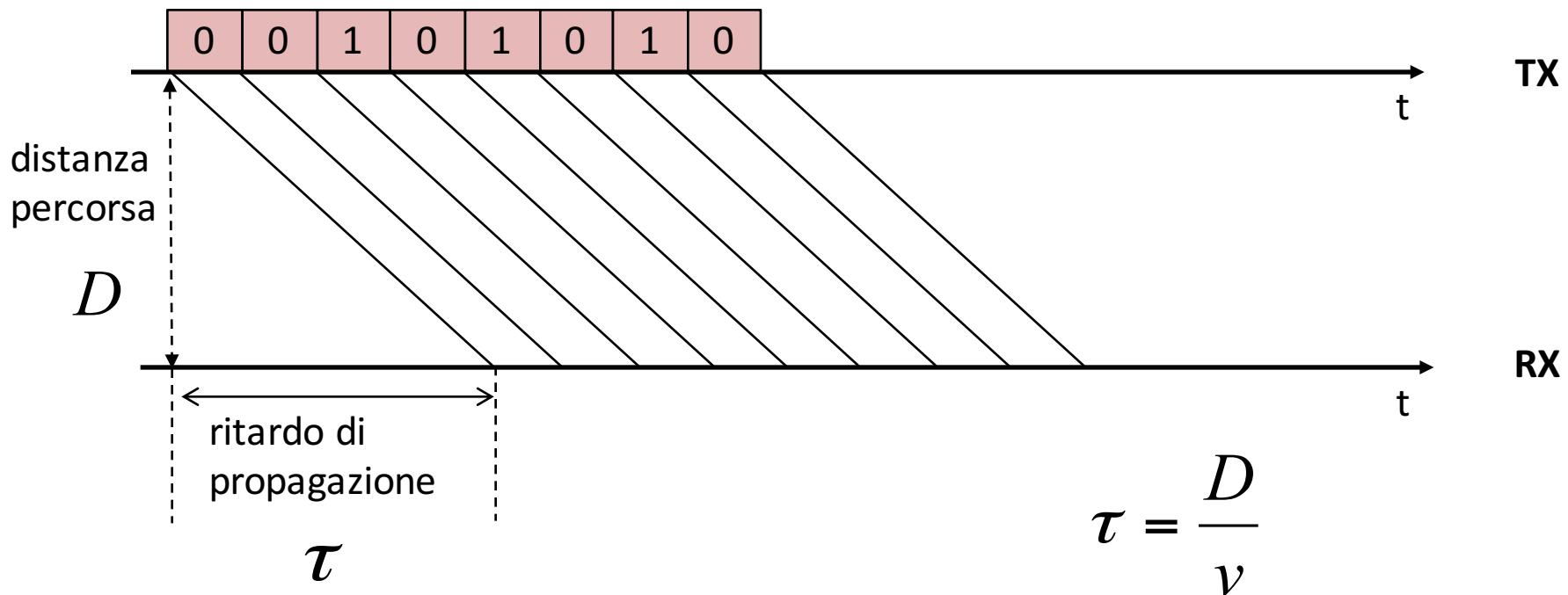
Si ha:

$$T = \frac{L}{R}$$

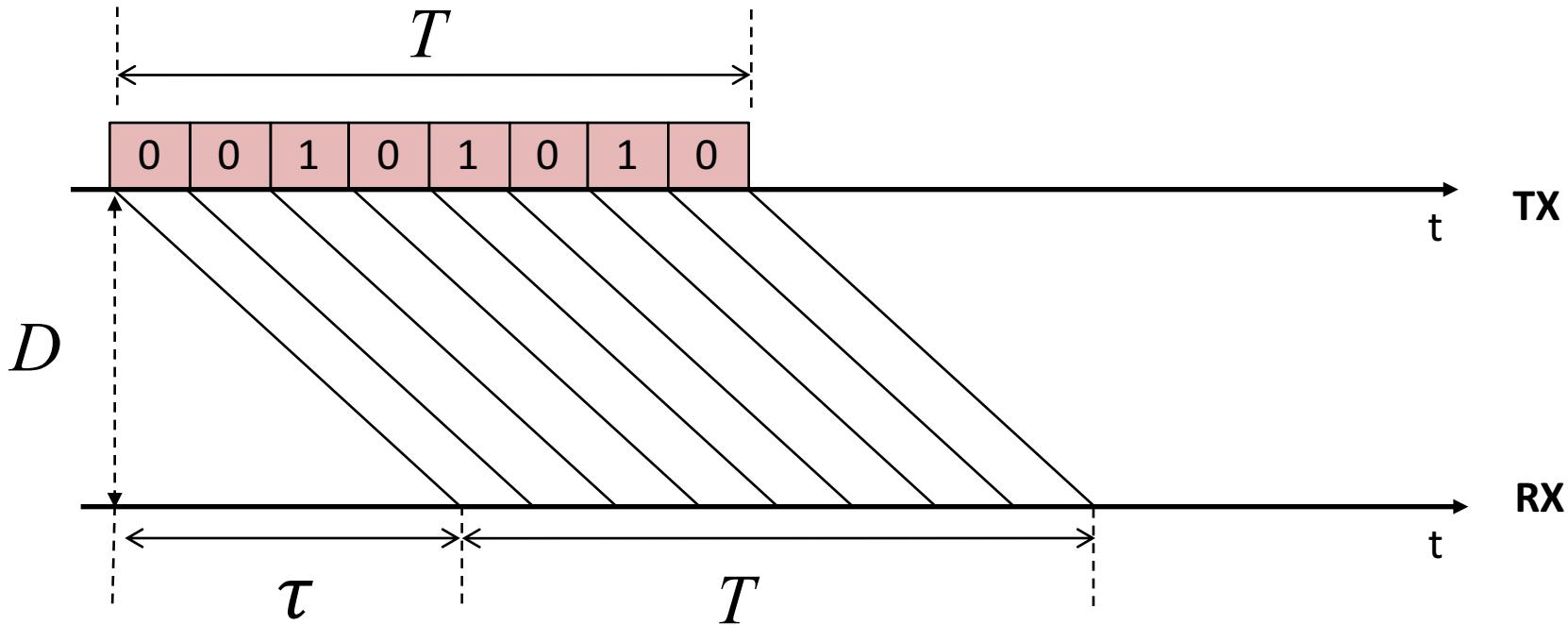


Ritardo di propagazione

- Il tempo τ affinché un impulso trasmesso dal trasmettitore TX raggiunga il ricevitore RX dipende dalla distanza D (in m) e dalla velocità di propagazione v (in m/s, prossima alla velocità della luce)



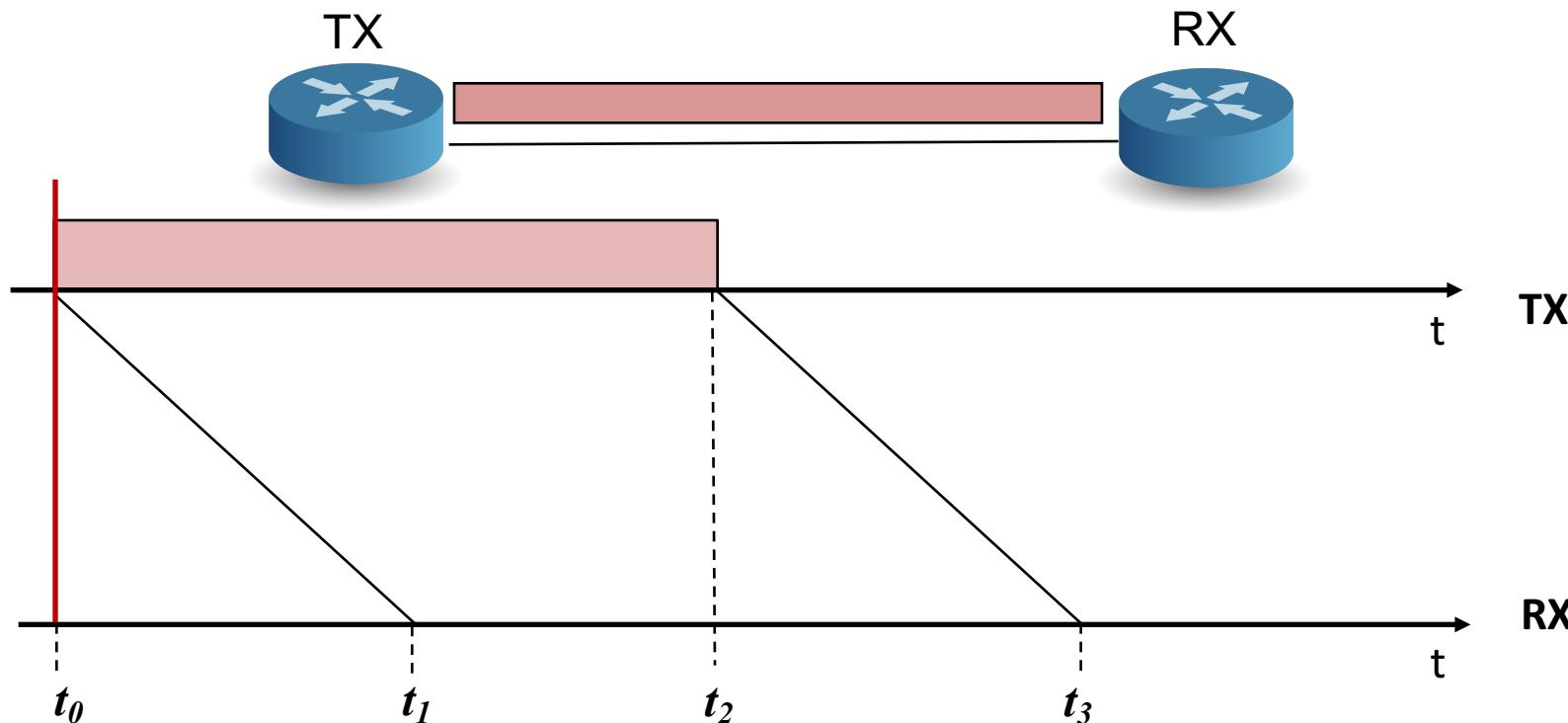
Tempi di attraversamento del canale



Ritardo fra la trasmissione del
primo bit e la ricezione dell'ultimo $T_{tot} = T + \tau$



Tempi di attraversamento del canale



t_0 =inizio trasmissione

t_1 =arrivo primo bit

t_2 =fine trasmissione

t_3 =arrivo ultimo bit

Tempo di trasmissione:

$$T = t_2 - t_0 = t_3 - t_1 = L/R \quad L=\text{lunghezza del pacchetto [bit]}$$

$$R=\text{velocità di trasm. [bit/s]}$$

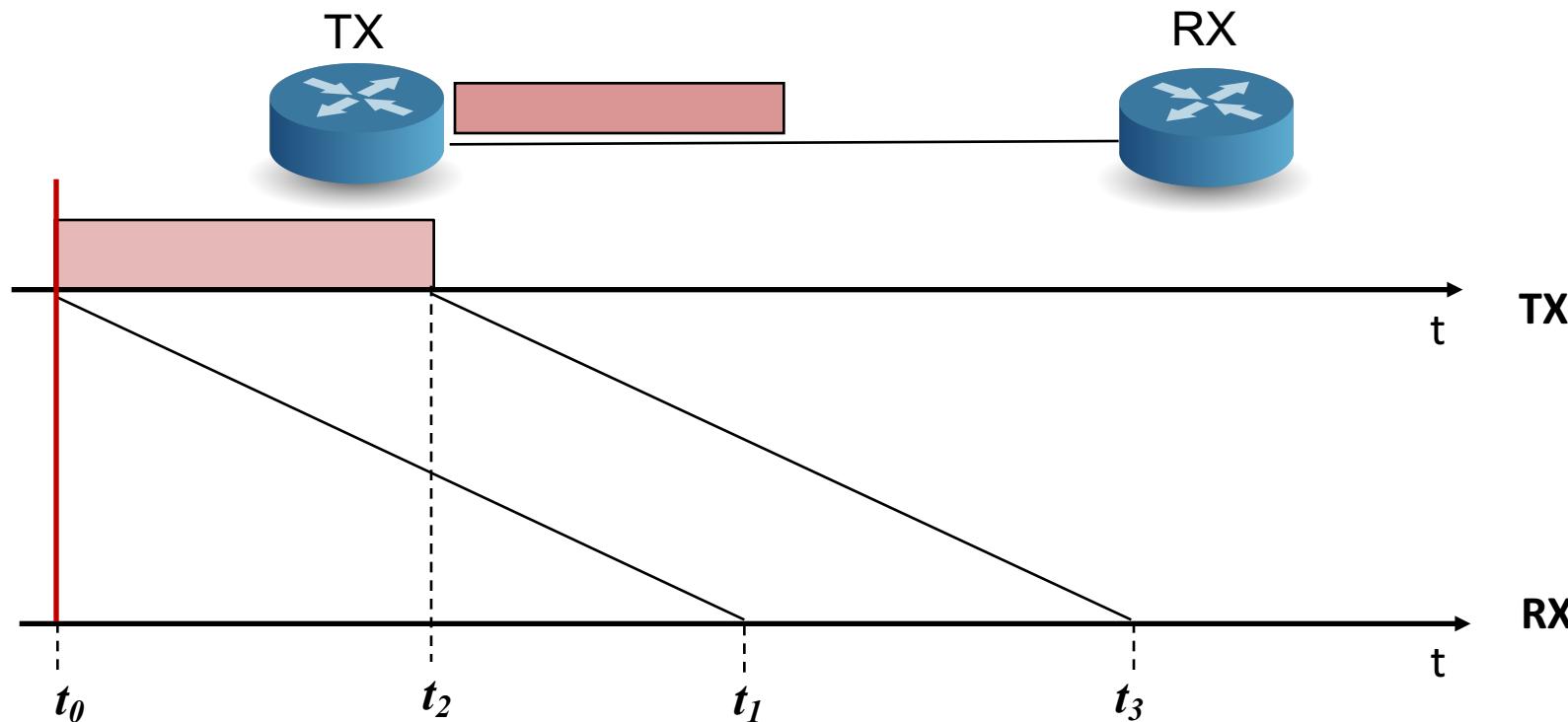
Ritardo di propagazione:

$$\tau = t_1 - t_0 = t_3 - t_2 = D/v \quad D=\text{lunghezza del coll. [m]}$$

$$v=\text{velocità di prop. [m/s]}$$



Tempi di attraversamento del canale



t_0 =inizio trasmissione

t_1 =arrivo primo bit

t_2 =fine trasmissione

t_3 =arrivo ultimo bit

Tempo di trasmissione:

$T = t_2 - t_0 = t_3 - t_1 = L/R$ L =lunghezza del pacchetto [bit]

R =velocità di trasm. [bit/s]

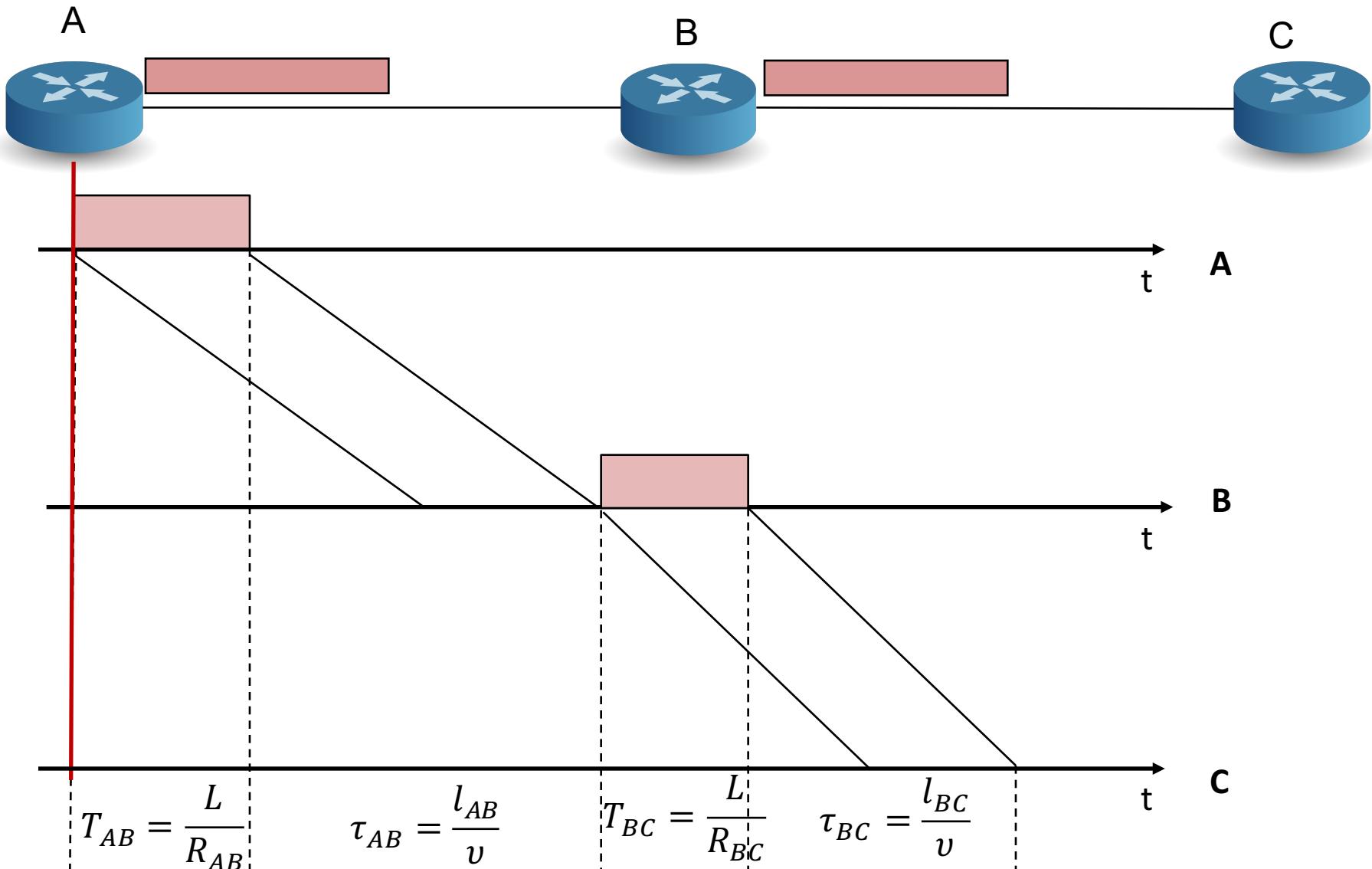
Ritardo di propagazione:

$\tau = t_1 - t_0 = t_3 - t_2 = D/v$ D =lunghezza del coll. [m]

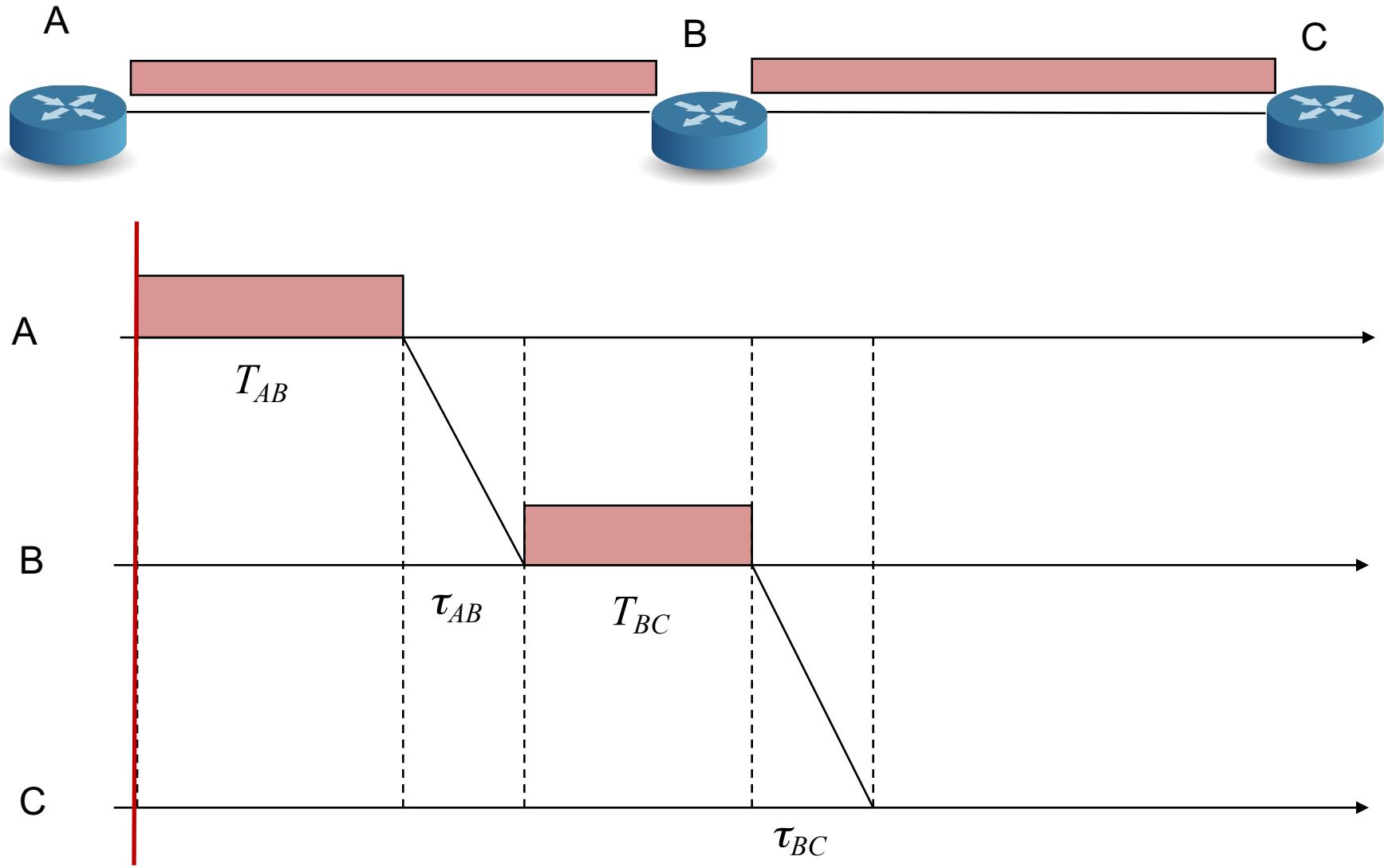
v =velocità di prop. [m/s]



Store and forward

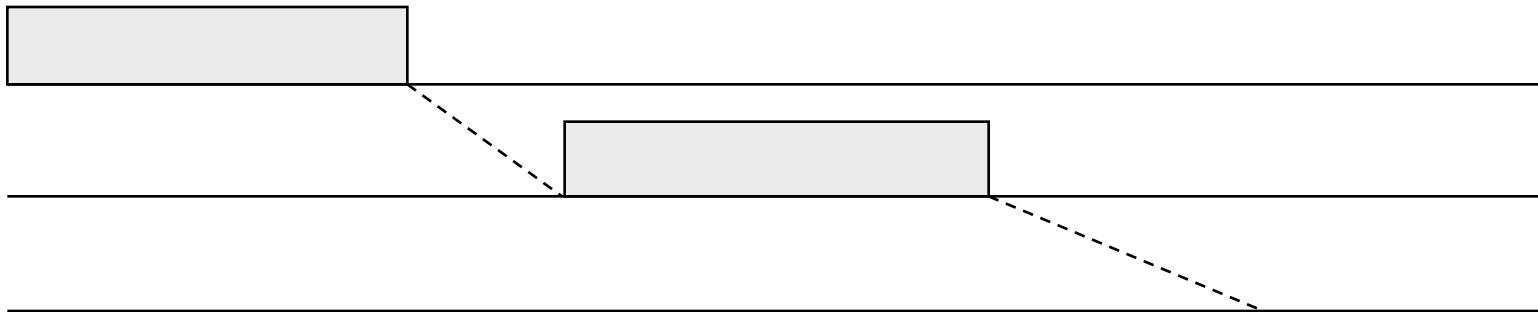


Store and forward

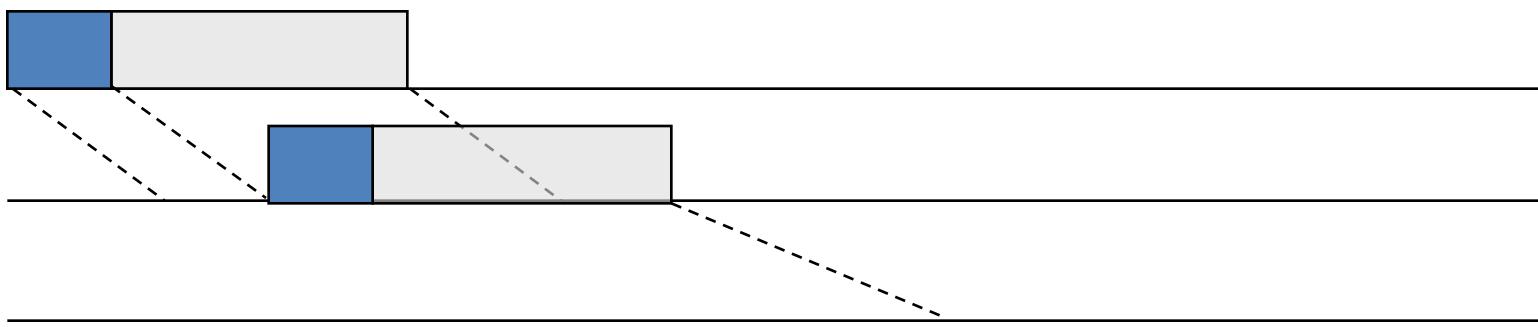


Commutazione a Pacchetto Cut-Through

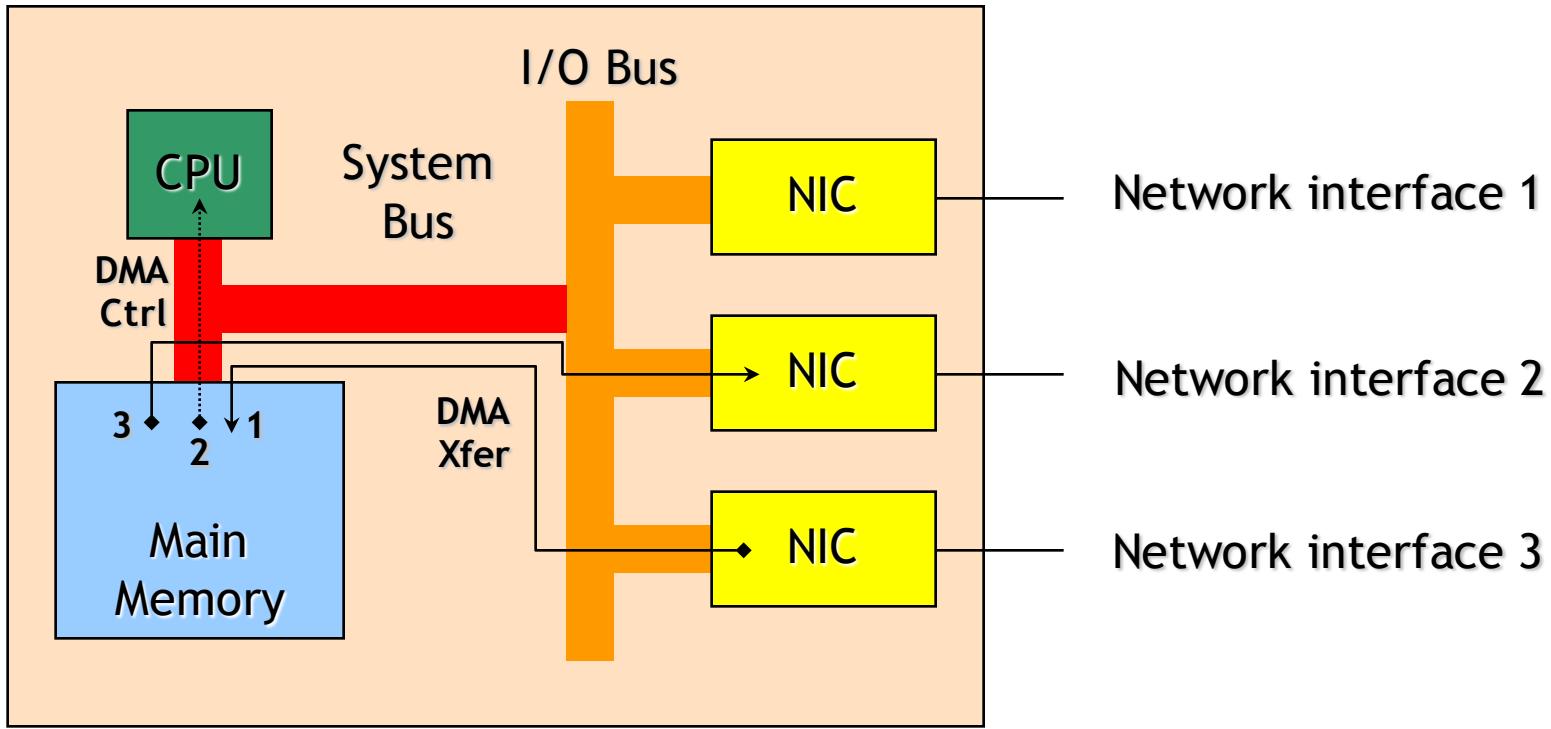
- ***Store-and-forward:*** il pacchetto deve essere completamente ricevuto prima di essere ritrasmesso



- ***Cut-Through:*** il pacchetto viene ritrasmesso alla completa ricezione dell'header



Architettura semplificata di un nodo



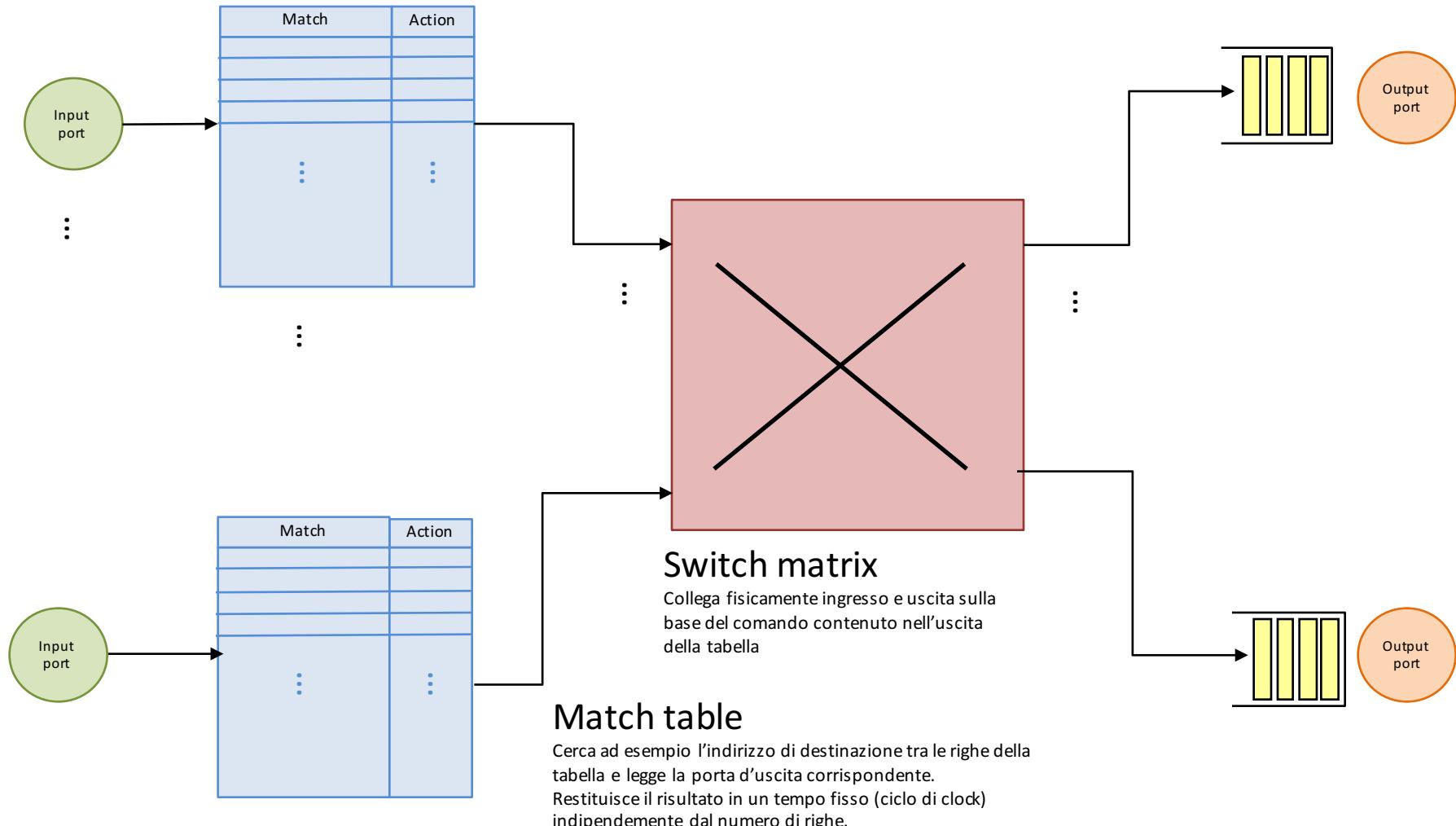
1. Packet input
2. Header processing
Routing table lookup
DMA transaction
3. Packet output

NIC = Network Interface Controller
DMA = Direct Memory Access

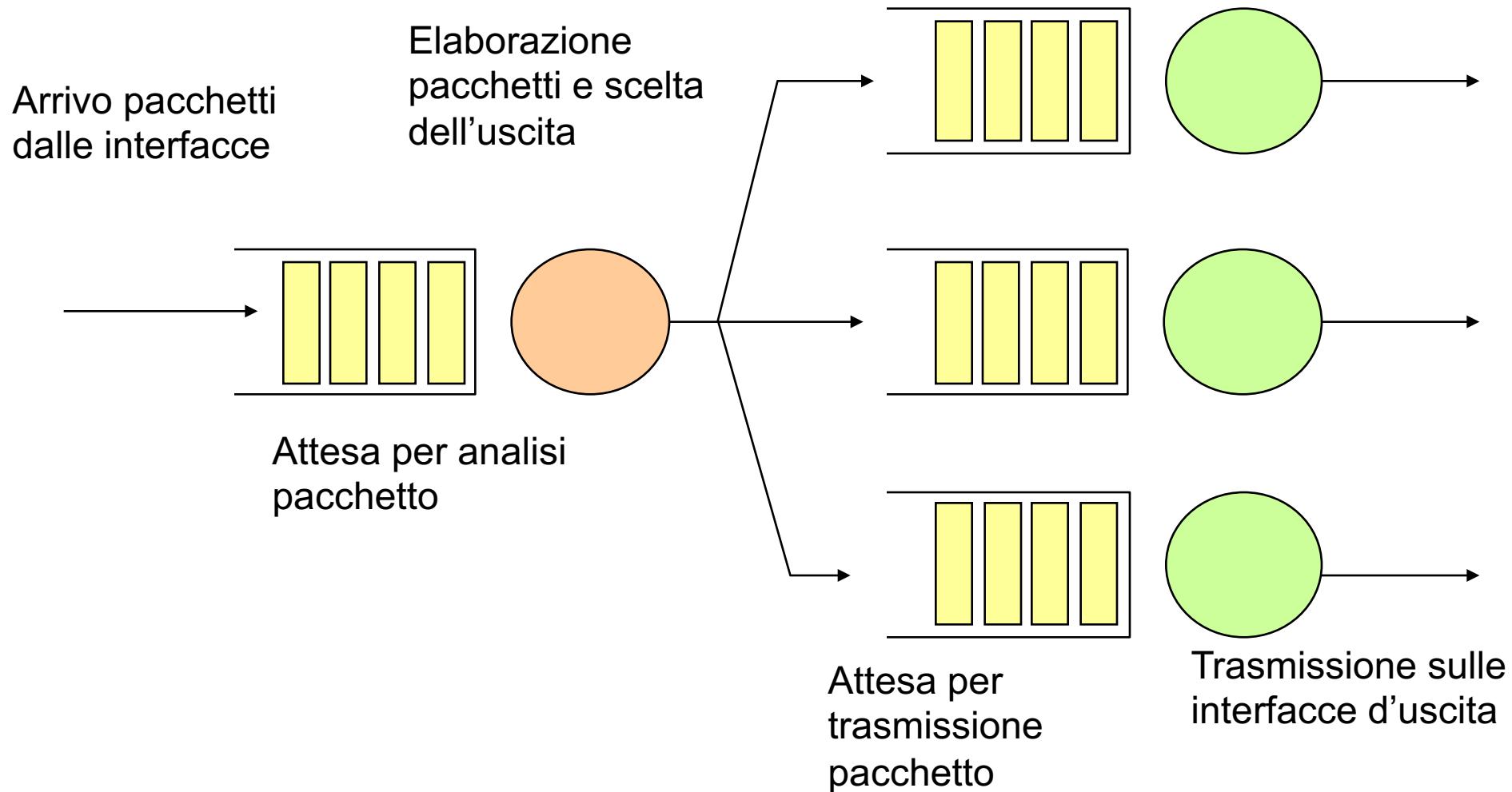


Architettura semplificata di un nodo

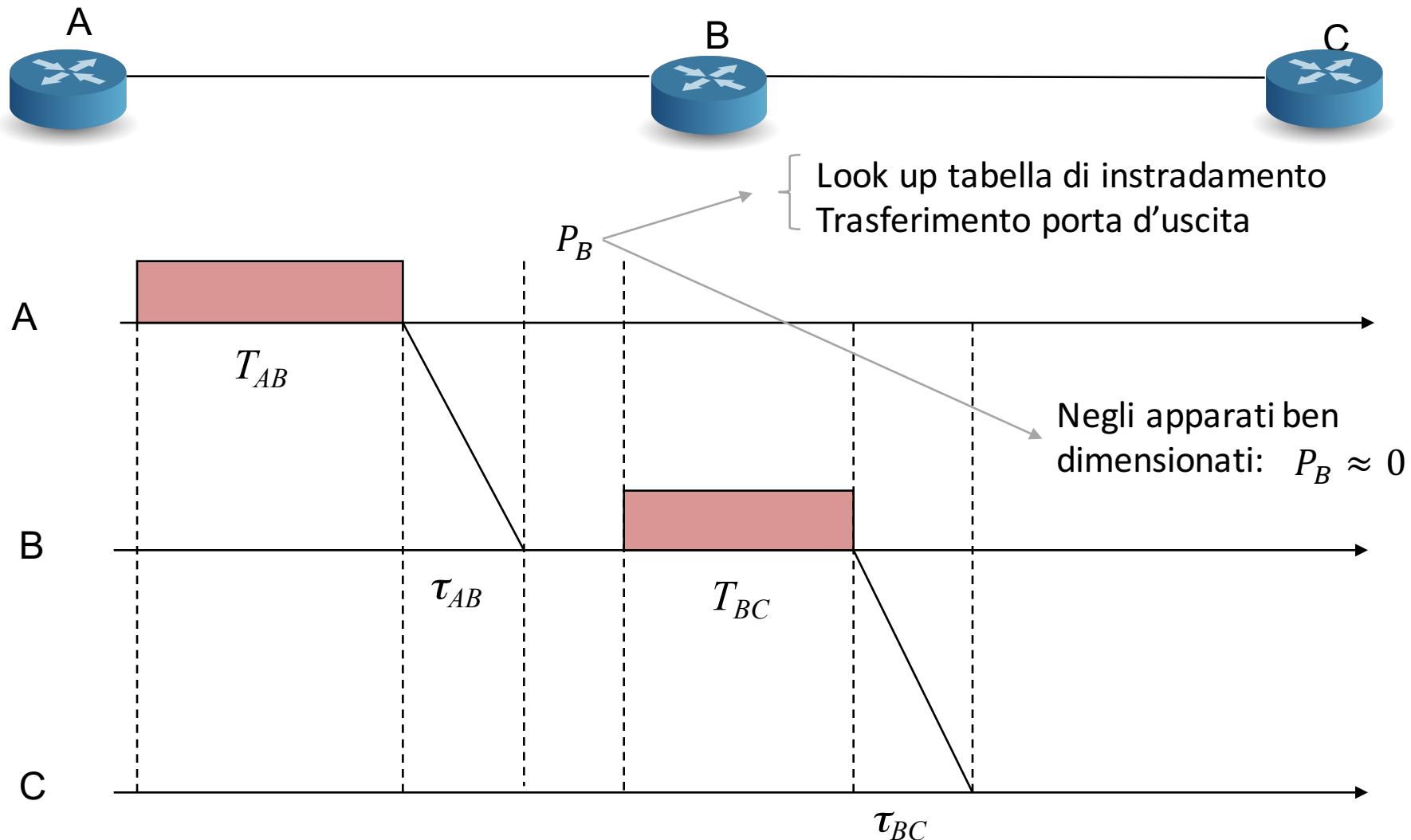
Packet switch su architettura hardware dedicata (hardware-switch)



Modello di un nodo



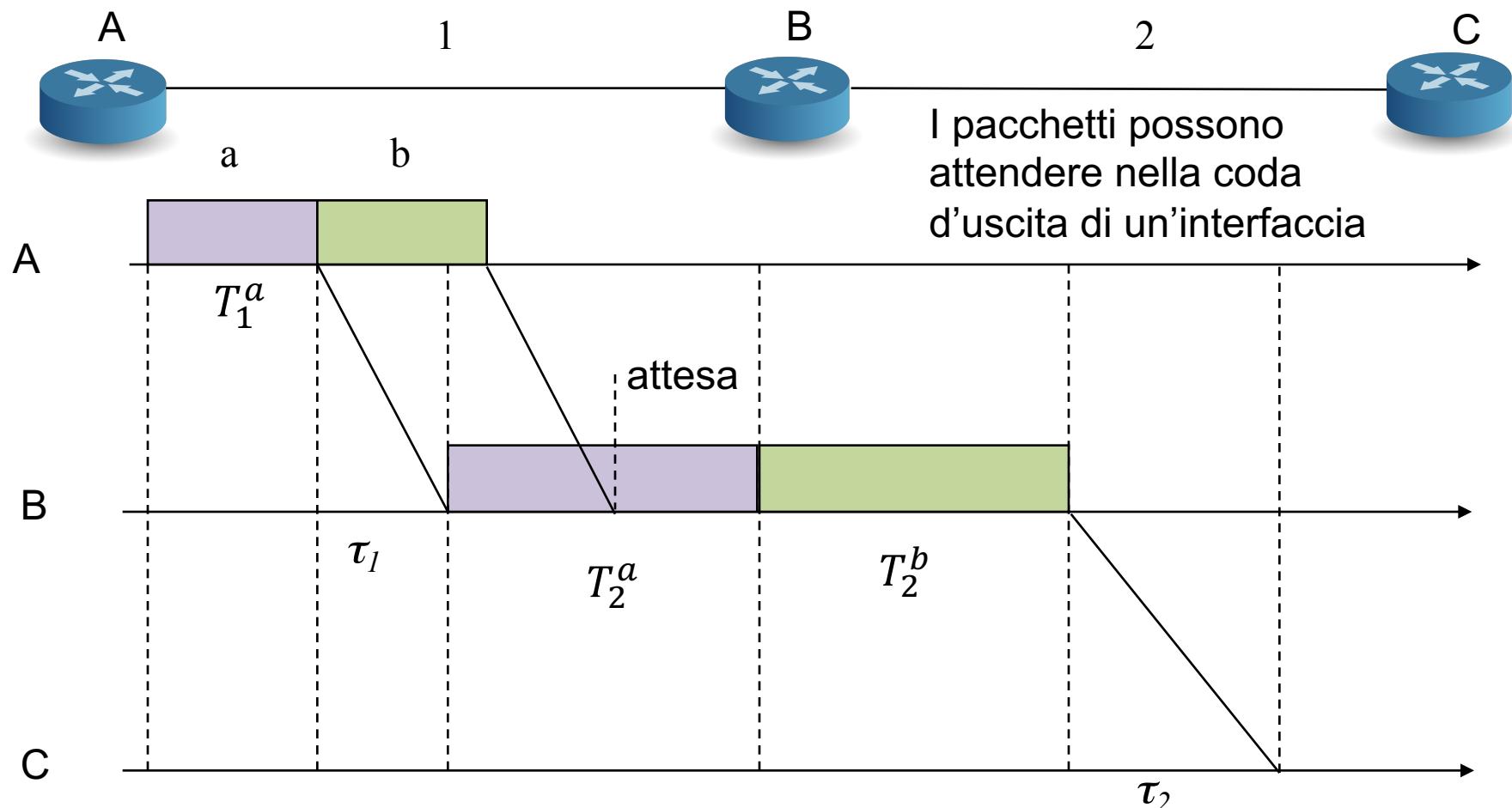
Tempo di elaborazione



Ritardo di accodamento

- Se la linea di uscita è occupata occorre aspettare in coda

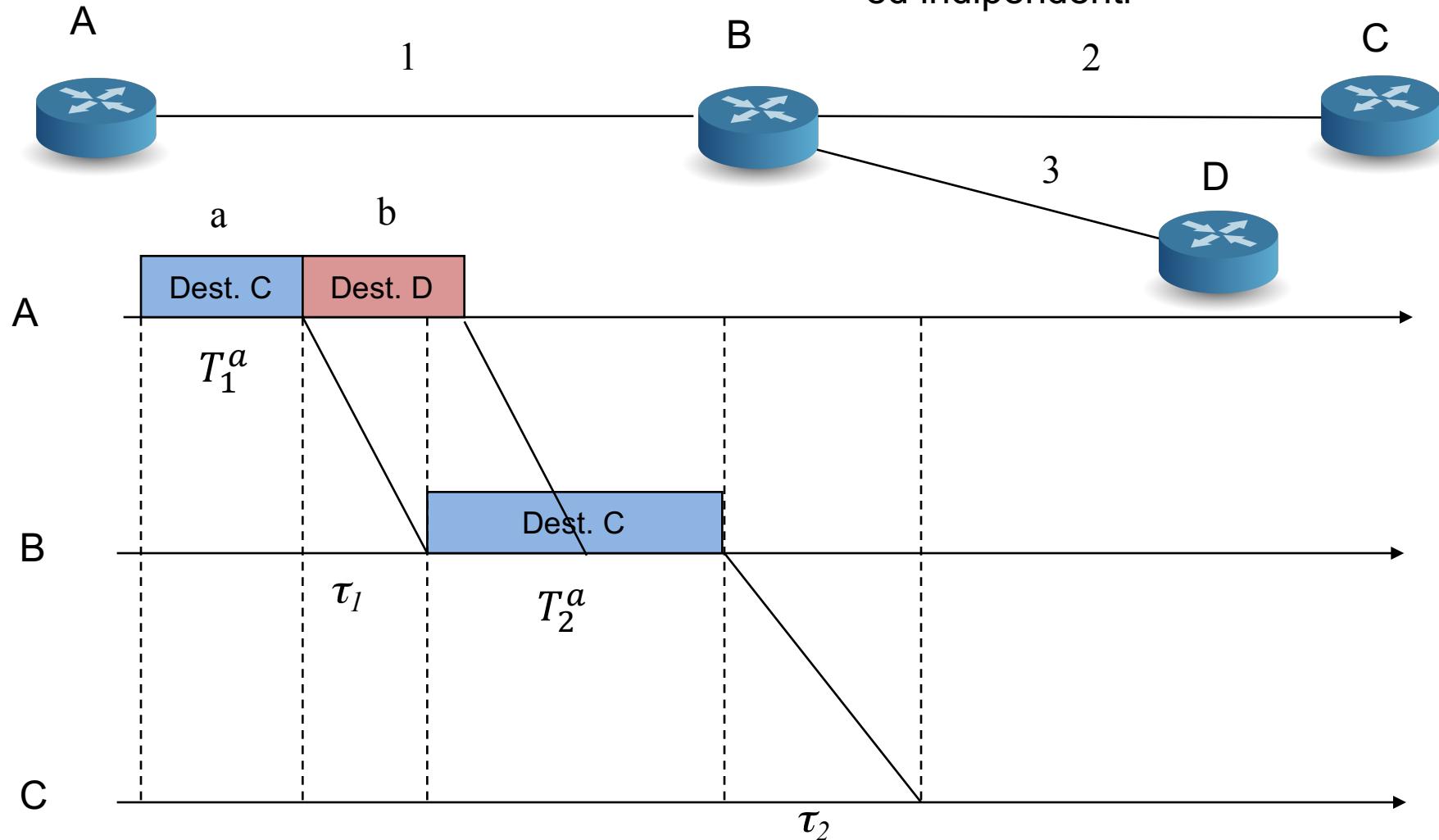
Esempio 1



Ritardo di accodamento

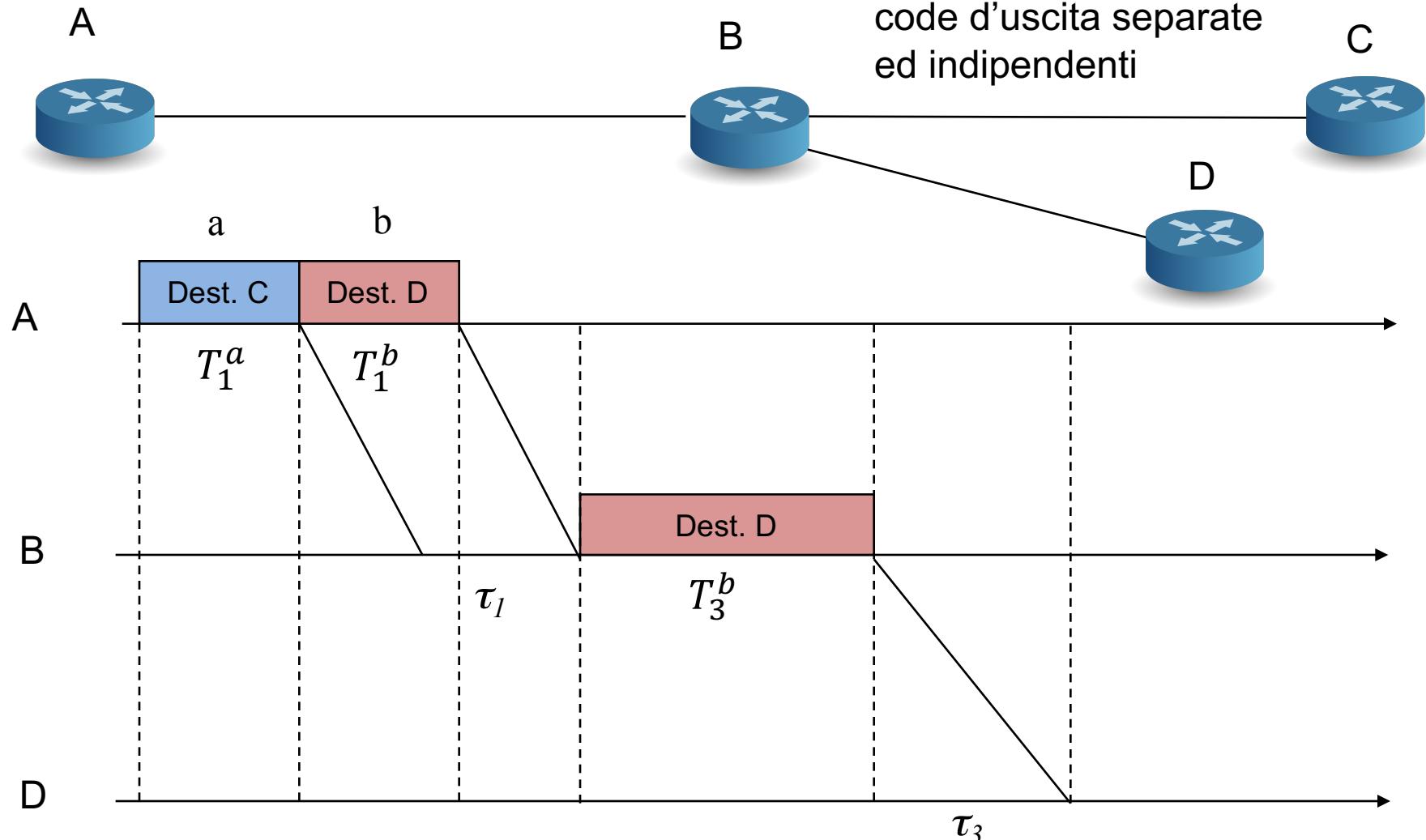
Esempio 2

Interfacce diverse hanno code d'uscita separate ed indipendenti



Ritardo di accodamento

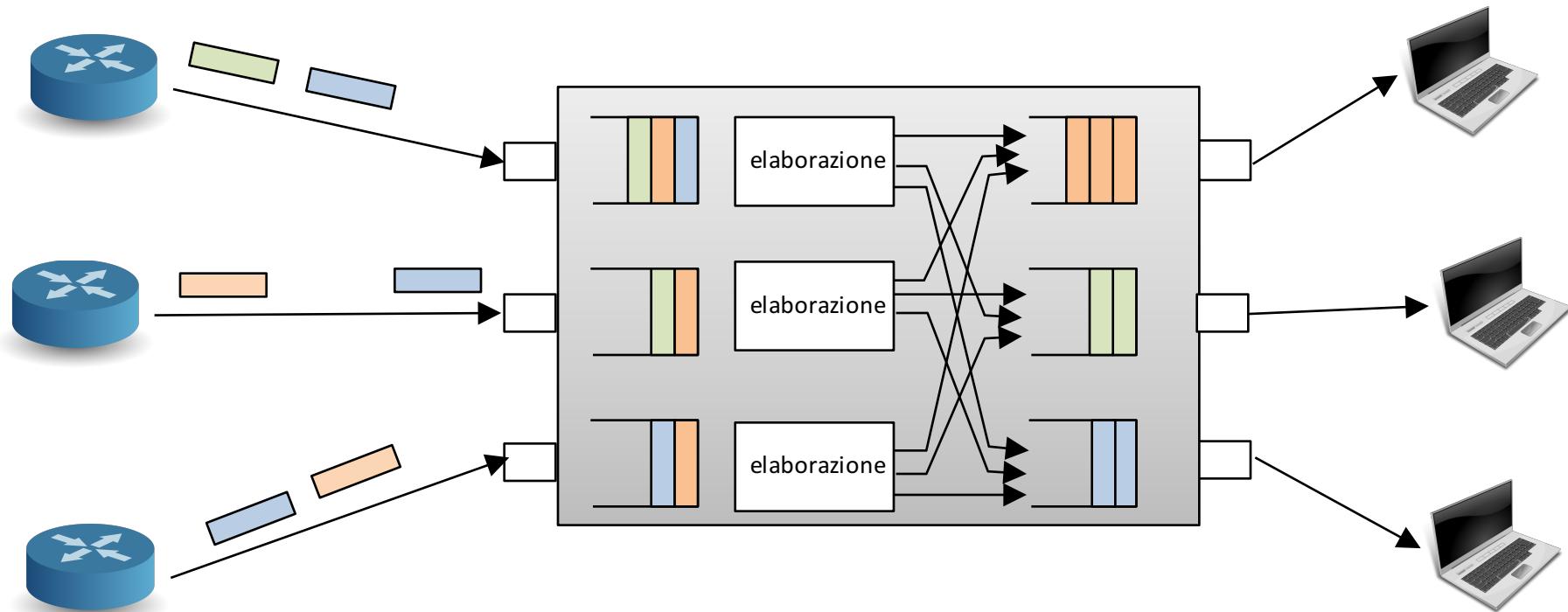
Esempio 2 (continua)



Ritardo di accodamento

Multiplazione statistica

Più in generale il **ritardo di accodamento** dipende dalla **multiplazione statistica** dovuto all'arrivo asincrono dei pacchetti alle code d'uscita (trasmissione)



Ritardo di accodamento

Multiplazione statistica

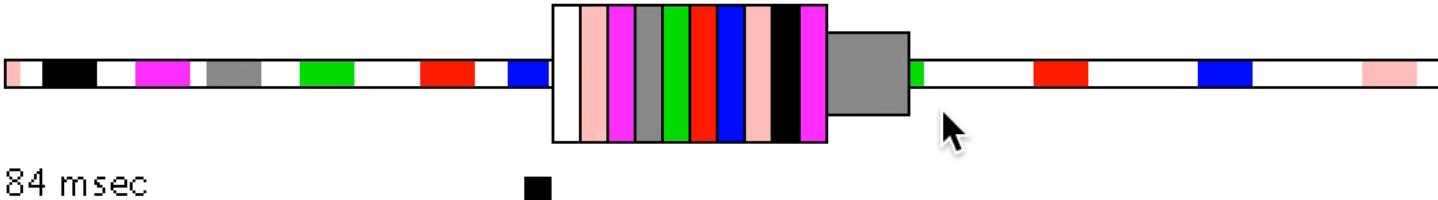
Più in generale il **ritardo di accodamento** dipende dalla **multiplazione statistica** dovuto all'arrivo asincrono dei pacchetti alle code d'uscita (trasmissione)

Emission rate Transmission rate



112 msec
0 packets dropped out of 32

Emission rate Transmission rate



84 msec
1 packets dropped out of 35



Ritardo di Accodamento

Del **ritardo di accodamento medio** T_a si possono fare dei modelli statistici basati sulla **teoria delle code**:

R = velocità del link [b/s]

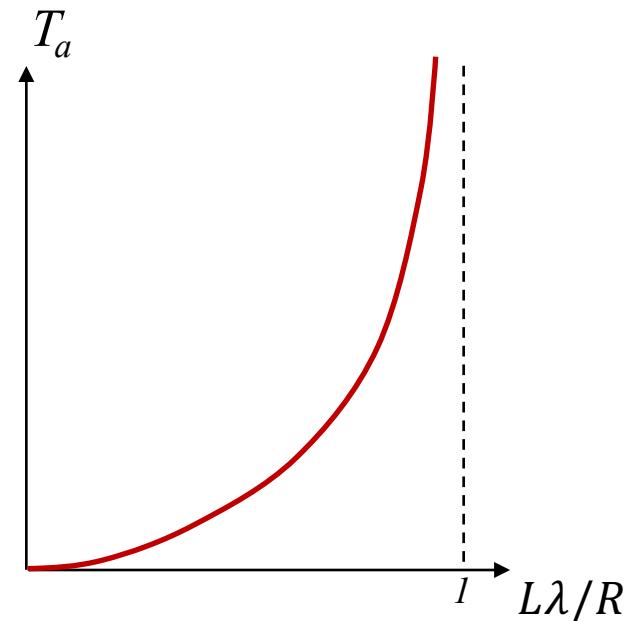
L = lunghezza pacchetto [bits]

λ = frequenza di arrivo dei pacchetti [pack/s]

Intensità di traffico = $L\lambda/R \sim 0$

$L\lambda/R \sim 0$: ritardo in coda piccolo

$L\lambda/R \rightarrow 1$: il ritardo tende all'infinito



Ritardo di Accodamento

Del **ritardo di accodamento medio** T_a si possono fare dei modelli statistici basati sulla **teoria delle code**:

R = velocità del link [b/s]

L = lunghezza pacchetto [bits]

λ = frequenza di arrivo dei pacchetti [pack/s]

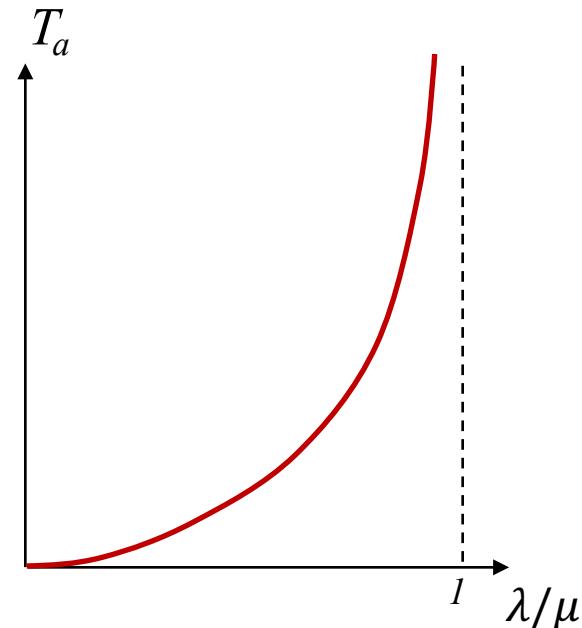
Definiamo:

μ = frequenza di trasmissione dei pacchetti [pack/s]

Si ha:

$$\mu = \frac{R}{L}$$

Si può mostrare che sotto alcuni condizioni sulla statistica degli arrivi e la distribuzione delle lunghezze dei pacchetti:

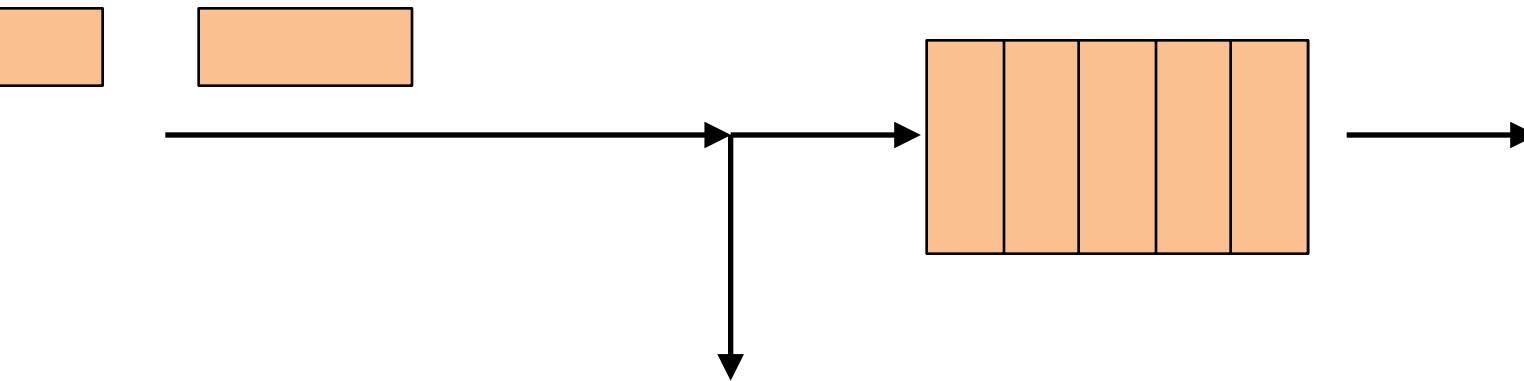


$$T_a = \frac{1}{\mu - \lambda} - \frac{1}{\mu}$$



Perdite di pacchetti in Internet

- Le code hanno **dimensioni limitate**
- In **congestione** (coda piena) i pacchetti che arrivano vengono **scartati**
- I pacchetti persi possono essere **ritrasmessi** o meno a seconda del livello/protocollo che gestisce l'evento di perdita (vedremo esempi a livello di linea e di trasporto).



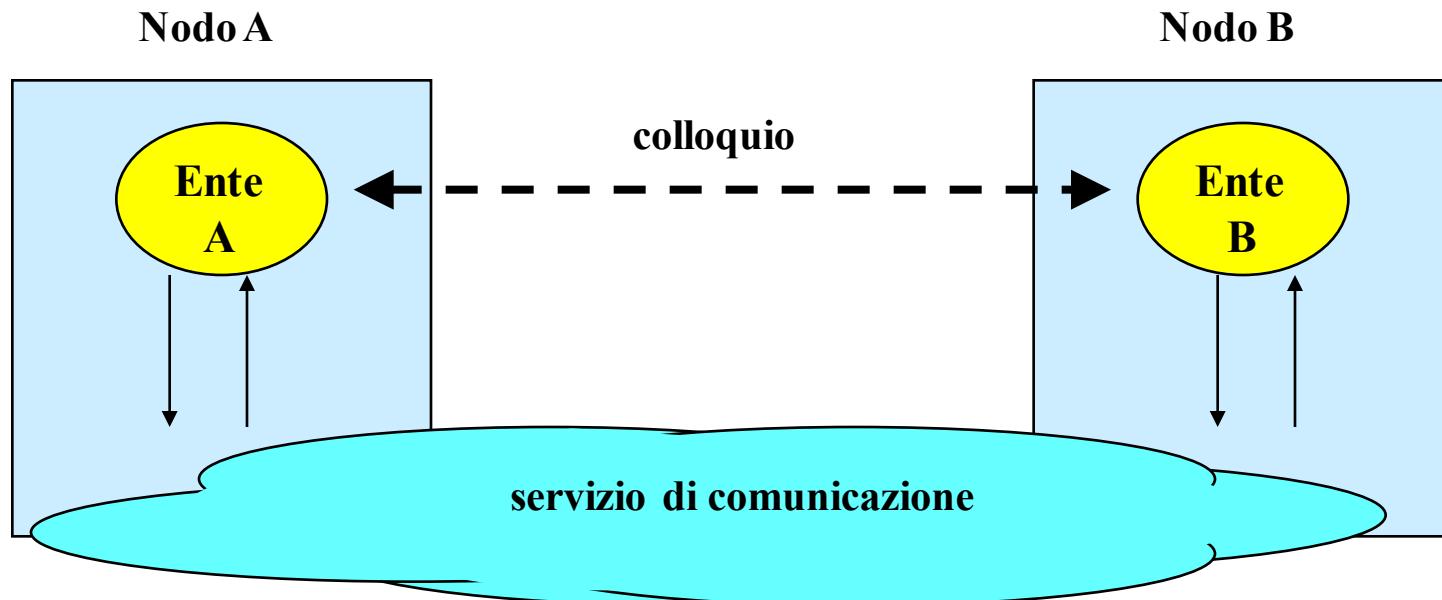


1d – Modelli di servizio e protocolli di comunicazione

**Servizio di comunicazione, protocolli,
primitive di servizio, architetture a strati,
funzioni di rete**

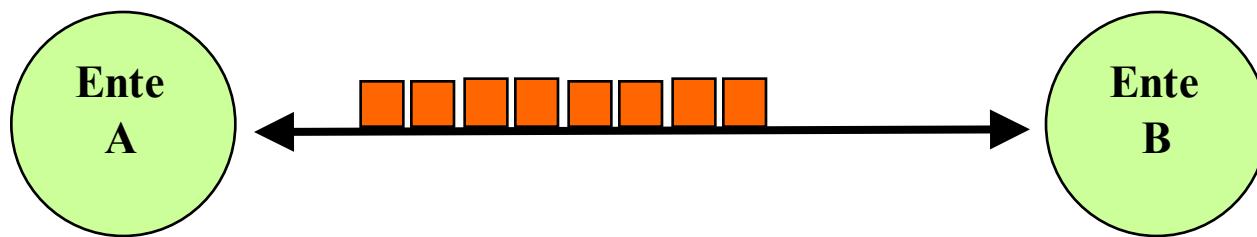
Il servizio di comunicazione

- Date due o più entità remote
- Possiamo descrivere il servizio di comunicazione per scambio di messaggi come un *“fornitore del servizio di trasporto dell'informazione”*



Il servizio di comunicazione

- Gestisce lo scambio di informazione fra due “entità”
- È in generale un servizio di trasferimento di unità informative:
 - parole
 - bit
 - gruppi di bit (trame o pacchetti)
 - file
 - flussi multimediali



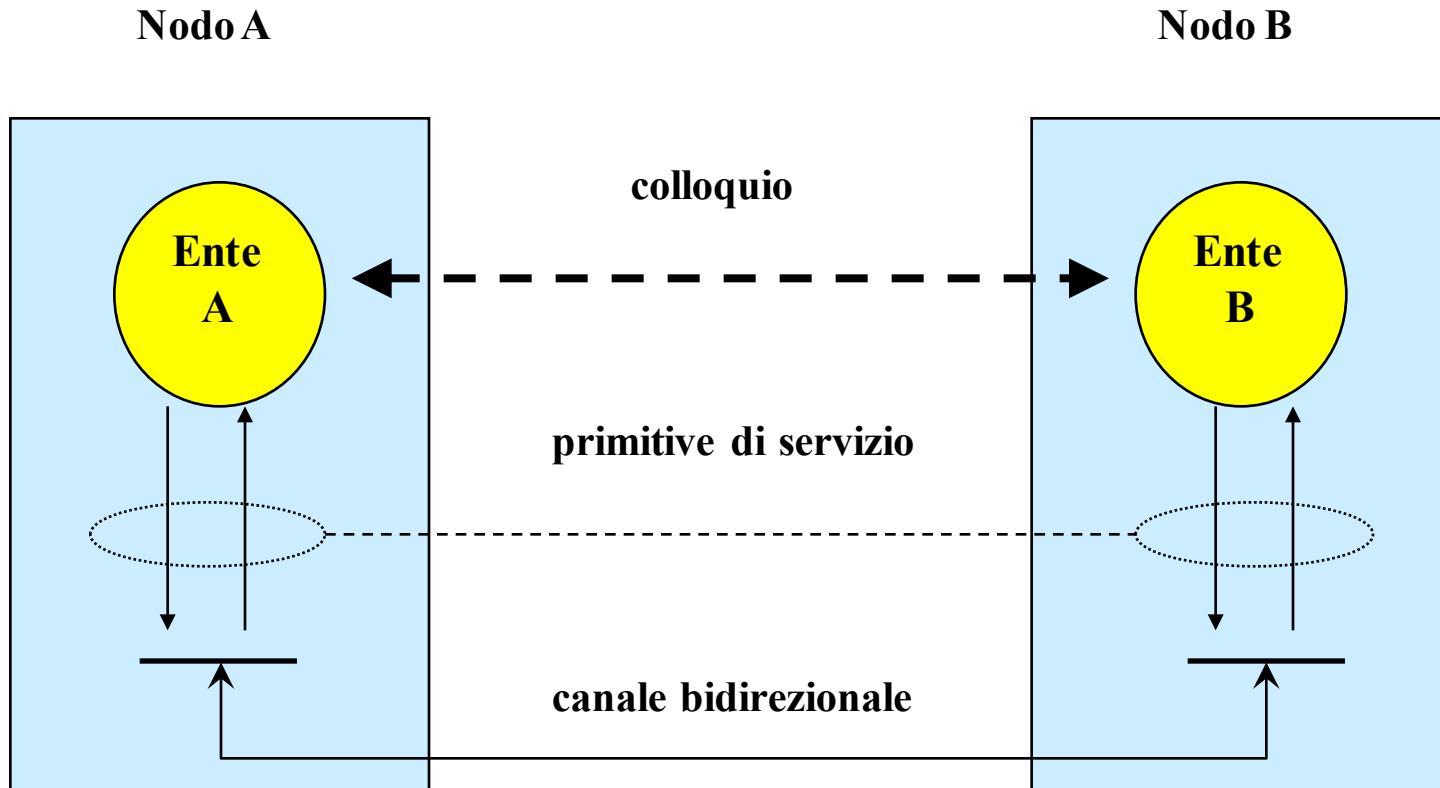
Primitive di servizio

- il servizio di comunicazione può essere descritto mediante delle **chiamate di servizio dette primitive di servizio**
- le primitive di servizio servono a descrivere il servizio, a richiederlo e a ricevere informazioni sul servizio dal fornitore
- le primitive di servizio sono caratterizzate da parametri tra cui:
 - informazione da trasferire
 - indicazione del destinatario
 - caratteristiche del servizio richiesto
 - ecc.

NOTA: le primitive di servizio dell'interfaccia socket in Python saranno trattate in laboratorio, dove vedrete meccanismi per creare le interfacce e passare i parametri tramite funzione.

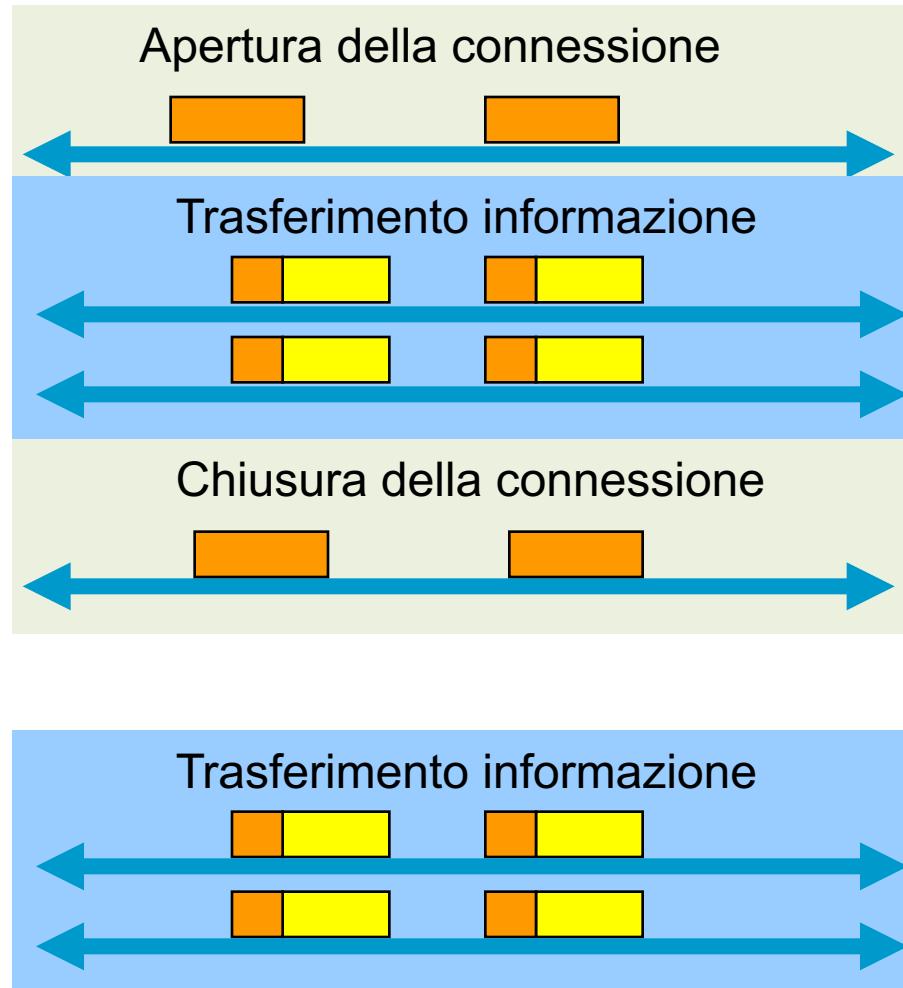


Primitive di servizio



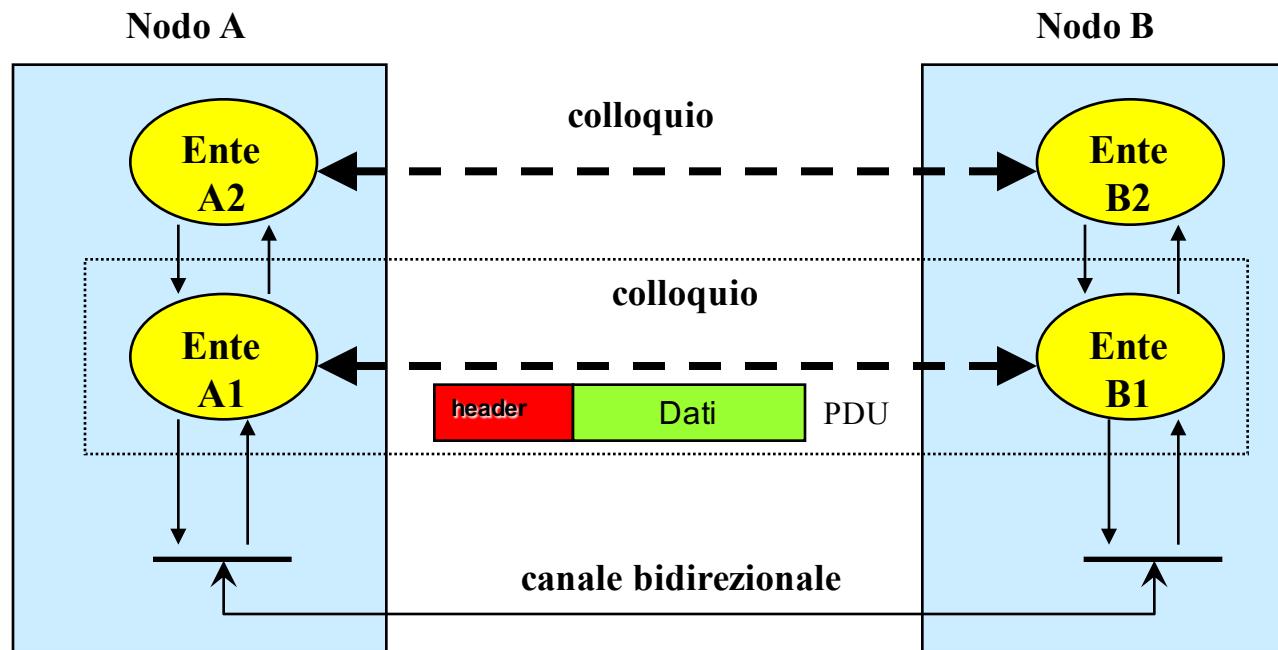
Caratteristiche del servizio di comunicazione

- **modalità a connessione**
 - instaurazione della connessione
 - trasferimento dell'informazione
 - rilascio delle connessione
- **modalità senza connessione**
 - una sola fase



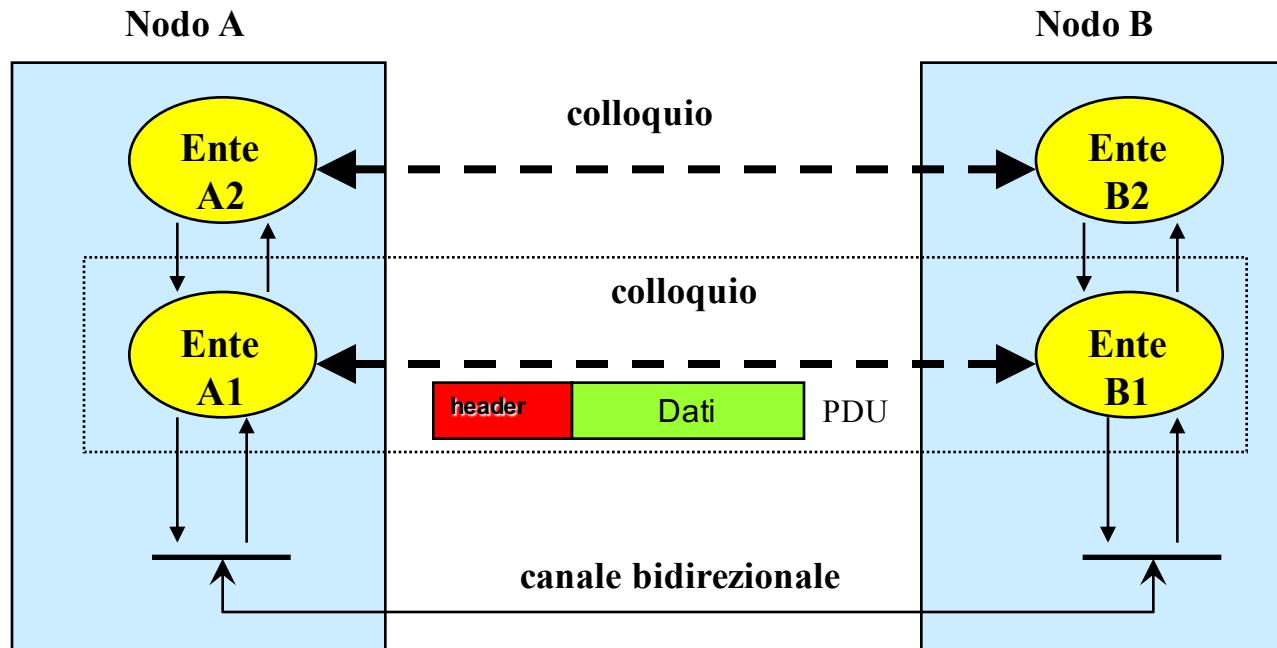
Livelli

- Le entità che colloquiano in un servizio di comunicazione possono anche offrire un servizio di comunicazione a entità terze, dette di **livello superiore**



Funzioni dei livelli

- Il servizio di comunicazione offerto al livello superiore è più ricco e complesso grazie alle funzioni implementate dal livello inferiore



Protocolli di comunicazione

- Le entità di un livello collaborano per fornire il servizio di comunicazione al livello superiore e si scambiano messaggi mediante il servizio offerto dal livello inferiore
- **Protocollo:**
 - Insieme delle regole che gestiscono il colloquio tra entità dello stesso livello
 - formato dei messaggi
 - informazioni di servizio
 - algoritmi di trasferimento
 - ecc.



Packet Data Units (PDU)

- Un protocollo utilizza per il colloquio tra entità dello stesso livello delle **unità di trasferimento dati** dette PDU

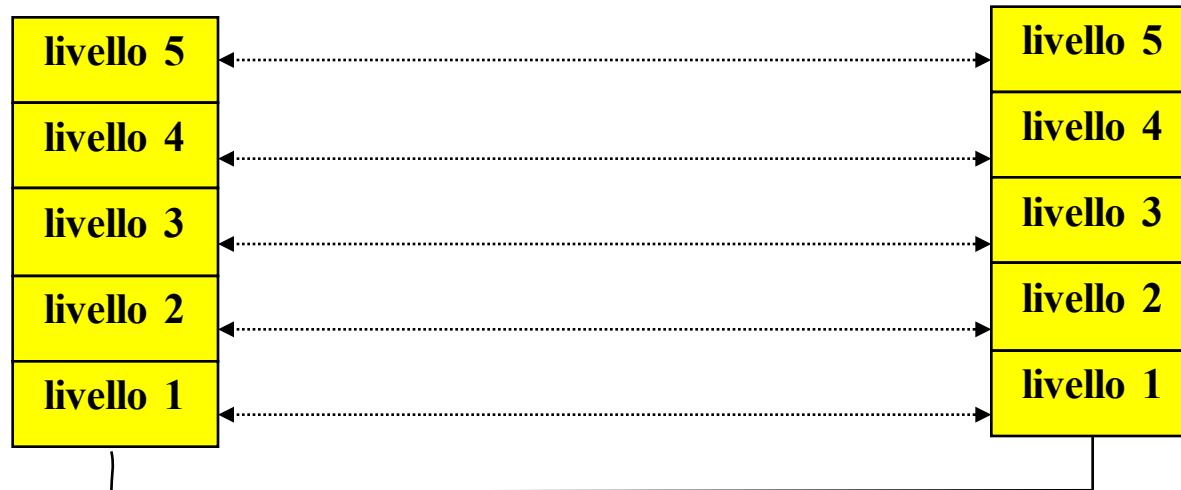
informazione di servizio
necessaria al
coordinamento tra le
entità

informazione vera e
propria ricevuta dai
livelli superiori

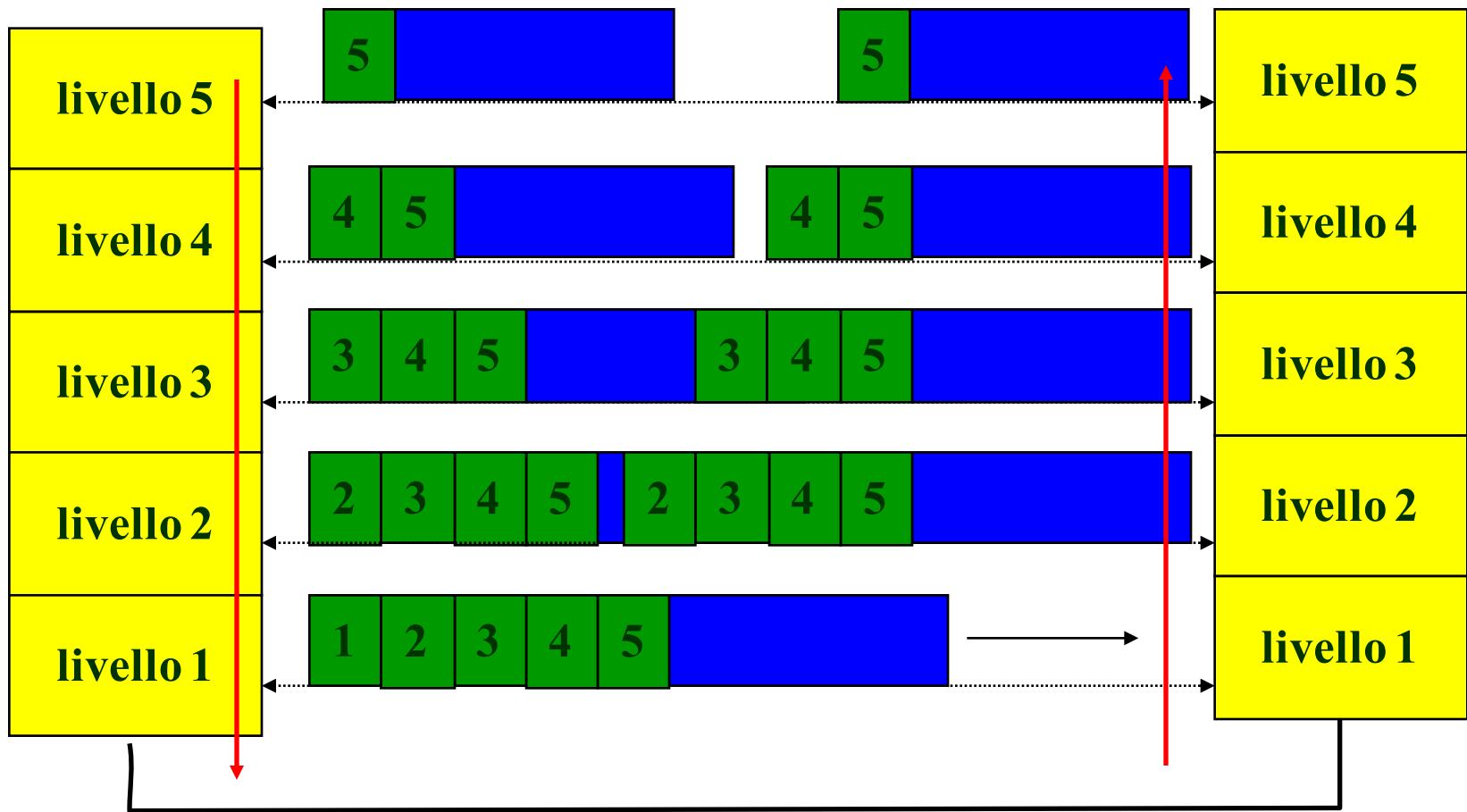


Architettura a strati

- I servizi di comunicazione complessi possono essere articolati a strati
 - da un livello che garantisce solo il trasporto dei bit
 - a un livello dove sono definite complessi servizi caratterizzati da molti parametri e funzionalità



Architettura completa



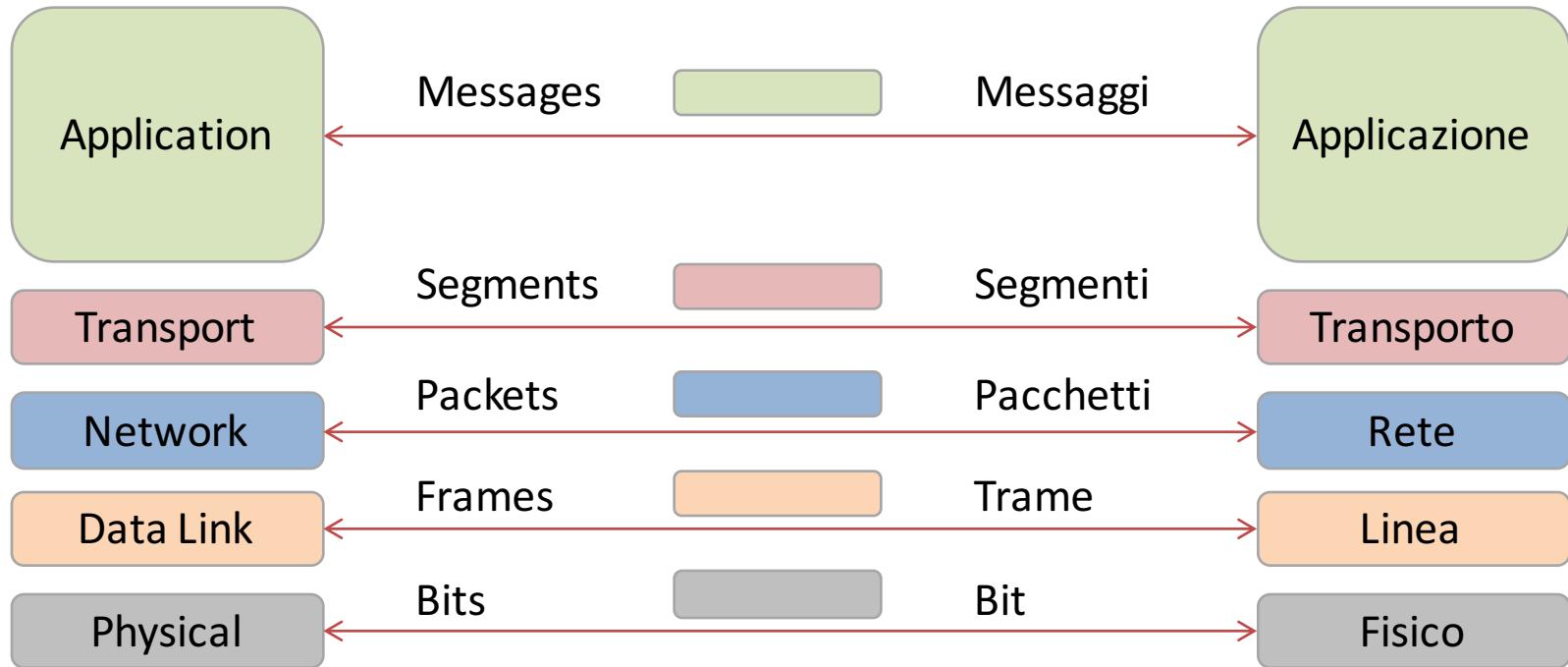
Perchè un'architettura a strati?

Sistemi complessi:

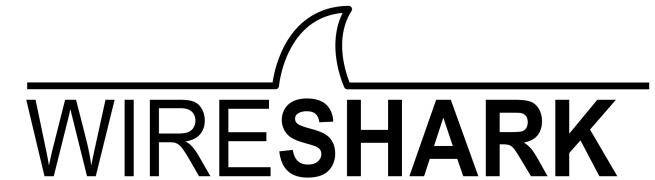
- **Facile identificazione dei servizi (implementazione, discussione)**
- **Facile gestione ed update**
 - Cambiamenti in un livello sono trasparenti agli altri
- **Q: quando la suddivisione in livelli può essere pericolosa?**



Modello a livelli di Internet (TCP/IP)



Wireshark



test.pcap - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.0.2	Broadcast	ARP	Who has 192.168.0.2? Gratuitous /
2	0.299139	192.168.0.1	192.168.0.2	NBNS	Name query NBSTAT <00><00><00>
3	0.299214	192.168.0.2	192.168.0.1	ICMP	Destination unreachable (Port unreachable)
4	1.025659	192.168.0.2	224.0.0.22	IGMP	V3 Membership Report
5	1.044366	192.168.0.2	192.168.0.1	DNS	Standard query SRV _ldap._tcp.nbg
6	1.048652	192.168.0.2	239.255.255.250	UDP	Source port: 3193 Destination port: 255.255.255.250
7	1.050784	192.168.0.2	192.168.0.1	DNS	Standard query SDA nb10061d.w004.
8	1.055053	192.168.0.1	192.168.0.2	UDP	Source port: 1900 Destination port: 1900
9	1.082038	192.168.0.2	192.168.0.255	NBNS	Registration NB NB10061D<00>
10	1.111945	192.168.0.2	192.168.0.1	DNS	Standard query A proxyconf.w004.
11	1.226156	192.168.0.2	192.168.0.1	TCP	3196 > http [SYN] Seq=0 Len=0 MSS=
12	1.227282	192.168.0.1	192.168.0.2	TCP	http > 3196 [SYN, ACK] Seq=0 Ack=1

Frame 11 (62 bytes on wire, 62 bytes captured)
Ethernet II, Src: Netgear_2d:75:9a (00:0b:5d:20:cd:02), Dst: Netgear_2d:75:9a (00:09:5b:2d:75:9a)
Internet Protocol Version 4, Src: 192.168.0.2 (192.168.0.2), Dst: 192.168.0.1 (192.168.0.1)
Transmission Control Protocol, Src Port: 3196 (3196), Dst Port: http (80), Seq: 0, Len: 0
Source port: 3196 (3196)
Destination port: http (80)
Sequence number: 0 (relative sequence number)
Header length: 28 bytes
Flags: 0x0002 (SYN)
Window size: 64240

0000 00 09 5b 2d 75 9a 00 0b 5d 20 cd 02 08 00 45 00 ..-[u...]E.
0010 00 30 18 48 40 00 08 06 61 2c c0 a8 00 02 c0 a8 .0.HR... a,.....
0020 00 01 0c 7c 00 50 3c 36 95 f8 00 00 00 70 02 ..|..P<6p.
0030 fa f0 27 e0 00 00 02 04 05 b4 01 01 04 02

File: "D:/test.pcap" 14 KB 00:00:02 P: 120 D: 120 M: 0

Passare ad attività con wireshark per analisi livelli protocollari



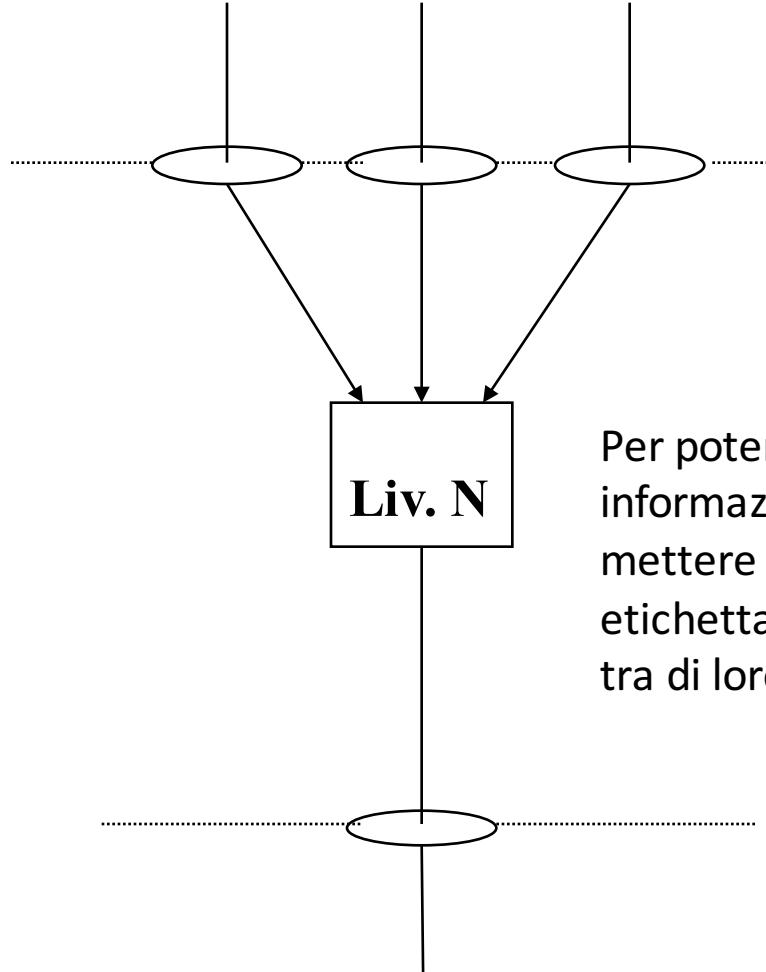
Funzioni

- **Molteplici sono le funzioni che possono essere svolte da uno livello**
- **Vediamo alcuni esempi importanti concentrandoci sui principi di base, poi i dettagli saranno ripresi più avanti**
 - Esempi:
 - Funzione di Multiplazione e De-multiplazione
 - Funzione di controllo d'errore
 - Funzione di instradamento

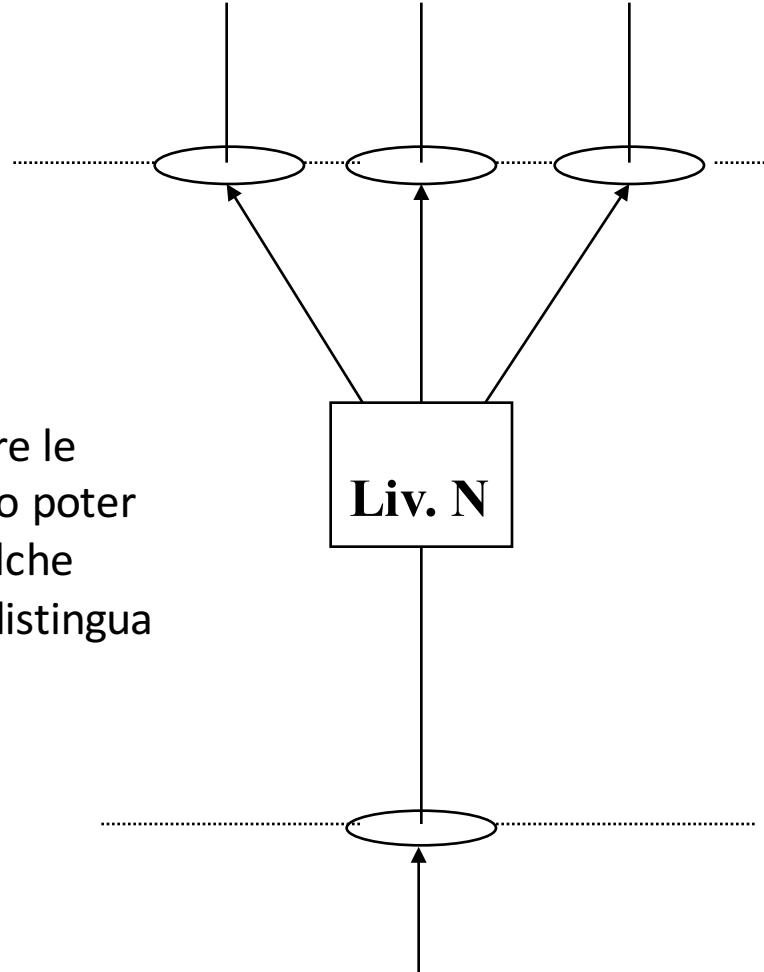


Esempio: Funzione Multiplazione

- Più livelli superiori possono condividere lo stesso servizio di comunicazione

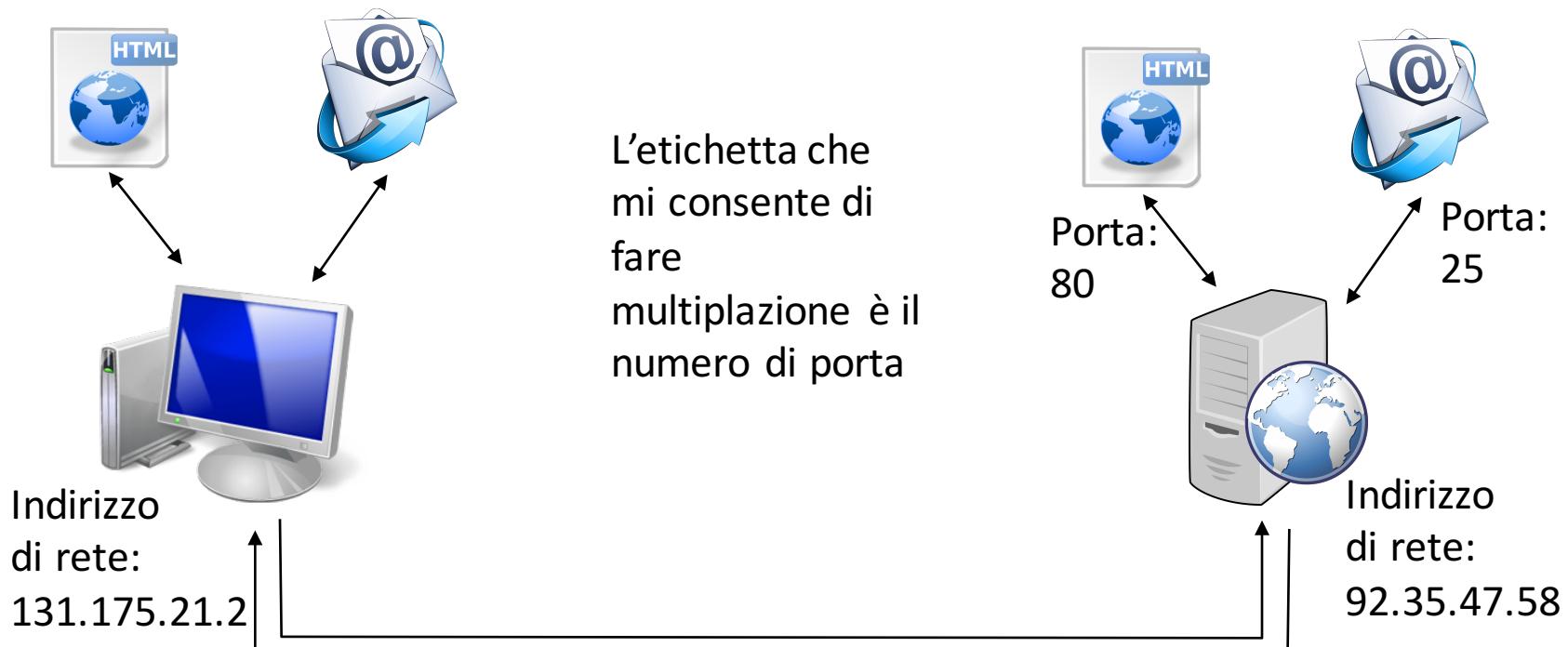


Per poter separare le informazioni devo poter mettere una qualche etichetta che le distingua tra di loro



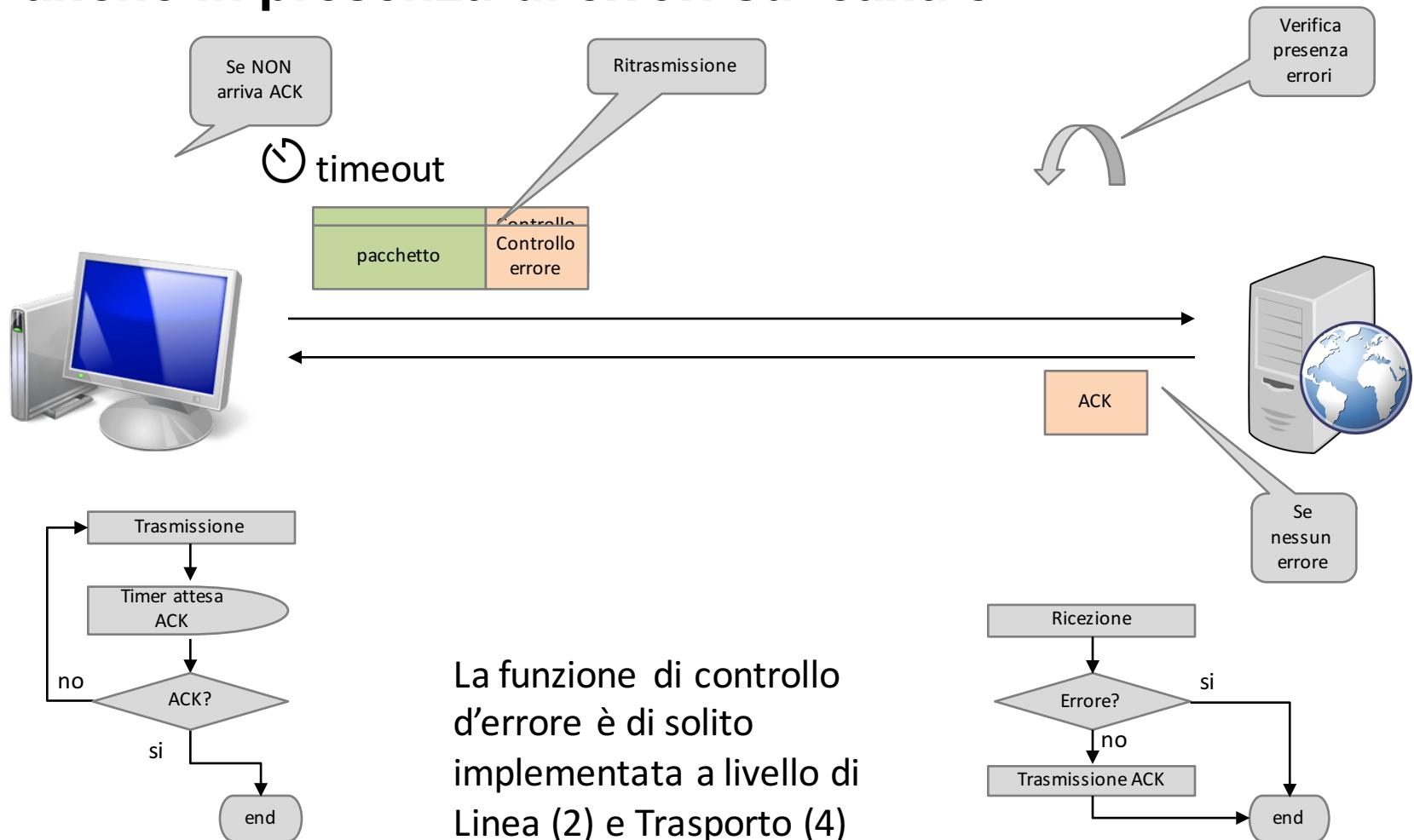
Esempio: Funzione Multiplazione

- Più applicazioni condividono la stessa interfaccia di rete (identificata da un suo indirizzo di rete)
- Le applicazioni sono distinte in base a un numero di porta



Esempio: Funzione di controllo d'errore

- E' possibile garantire affidabilità delle comunicazioni anche in presenza di errori sul canale

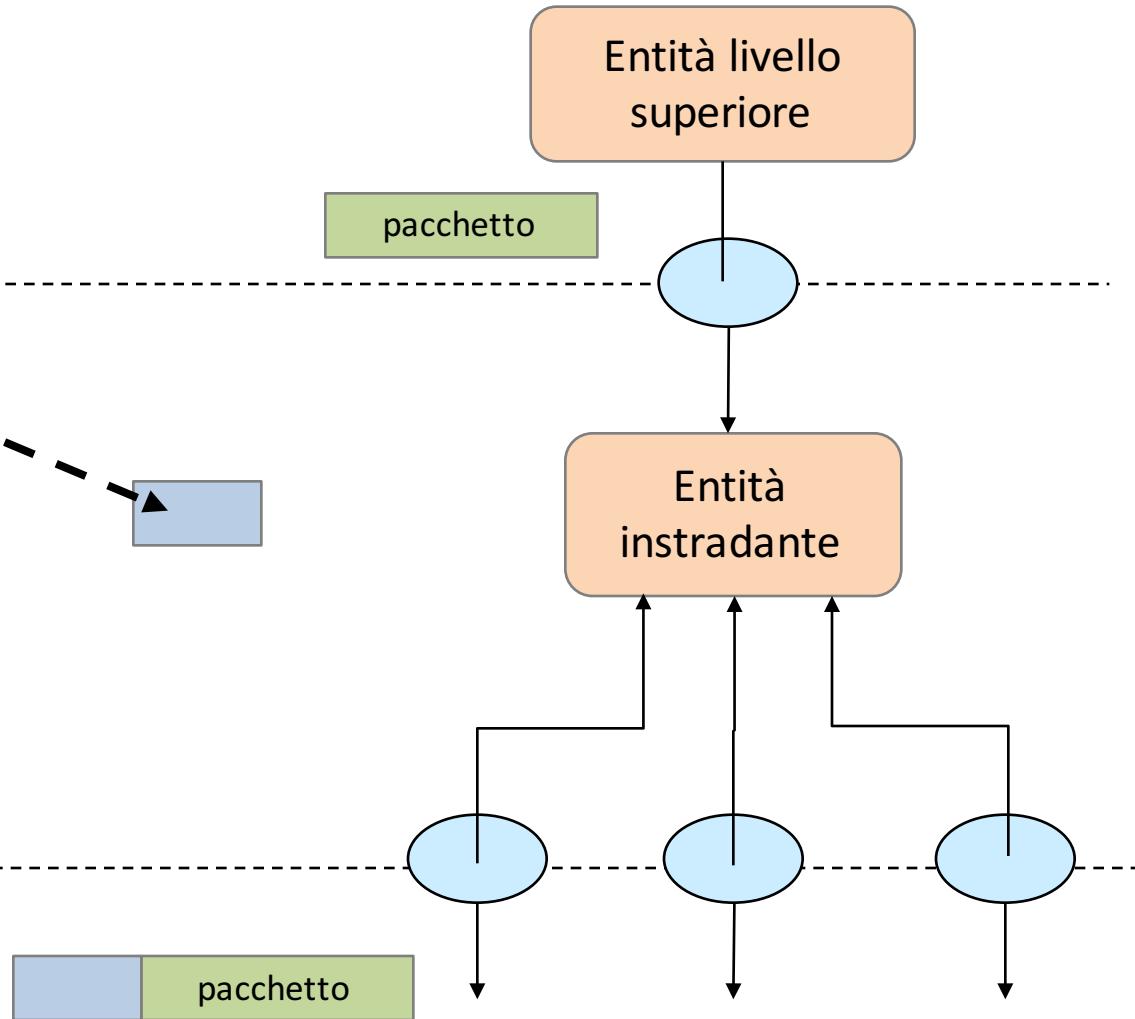


Esempio: Funzione di instradamento (routing)

Il pacchetto può arrivare dal livello superiore passato col parametro **INDIRIZZO**

L'indirizzo e viene scritto nell'header perché possa essere instradato da altri nodi

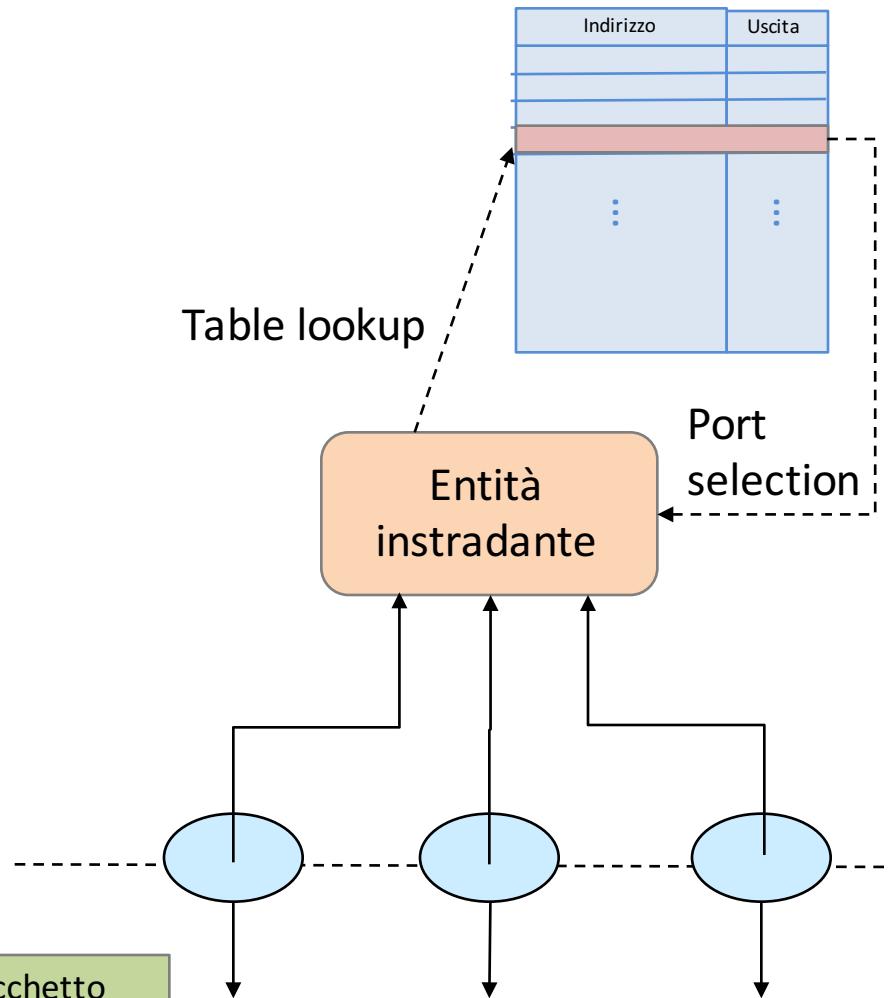
Il pacchetto può arrivare da una porta d'ingresso



Esempio: Funzione di instradamento (routing)

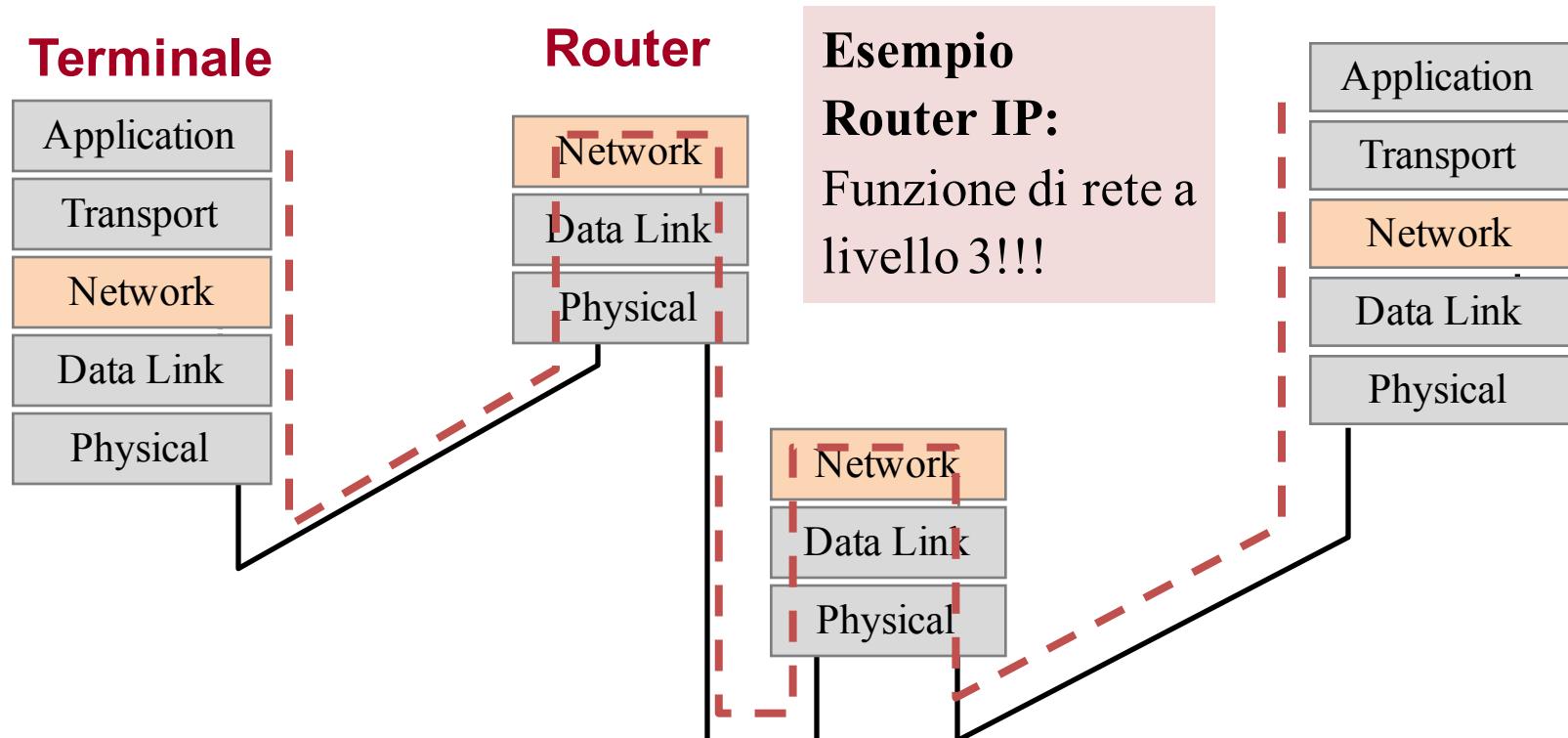
L'entità instradante decide dove inoltrare (forwarding) il pacchetto sulla base di una tabella di instradamento che viene consultato usando l'indirizzo come parametro di ricerca

OSSERVAZIONE: Se il pacchetto arriva da una porta d'ingresso, non sale mai più in alto nei livelli salvo che non sia giunto alla destinazione finale



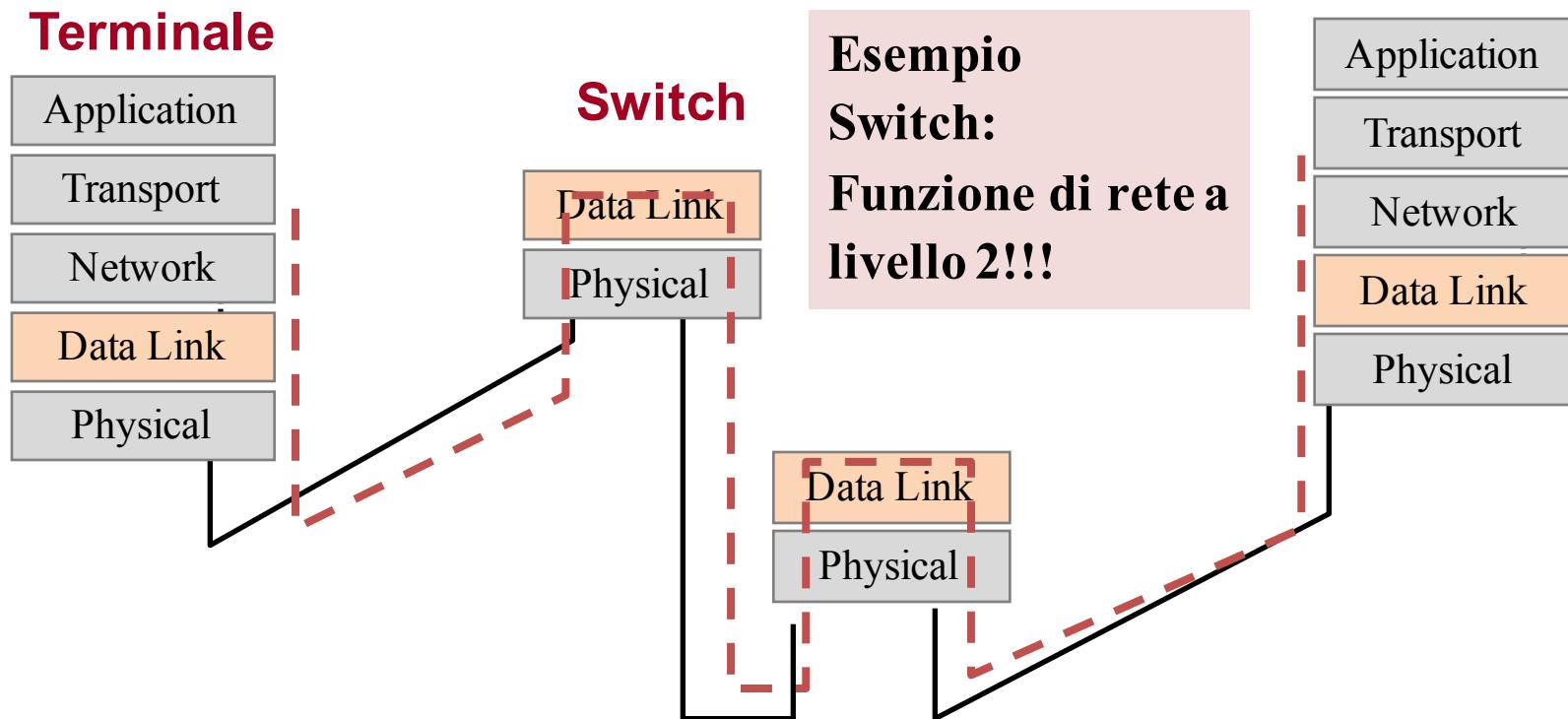
Esempio: Funzione di instradamento (routing)

- I nodi di rete non implementano i livelli superiori a quello di instradamento
- La funzione di instradamento può essere implementata in vari livelli in base alle circostanze (vedremo in dettaglio)
- Esempio: **Router IP**



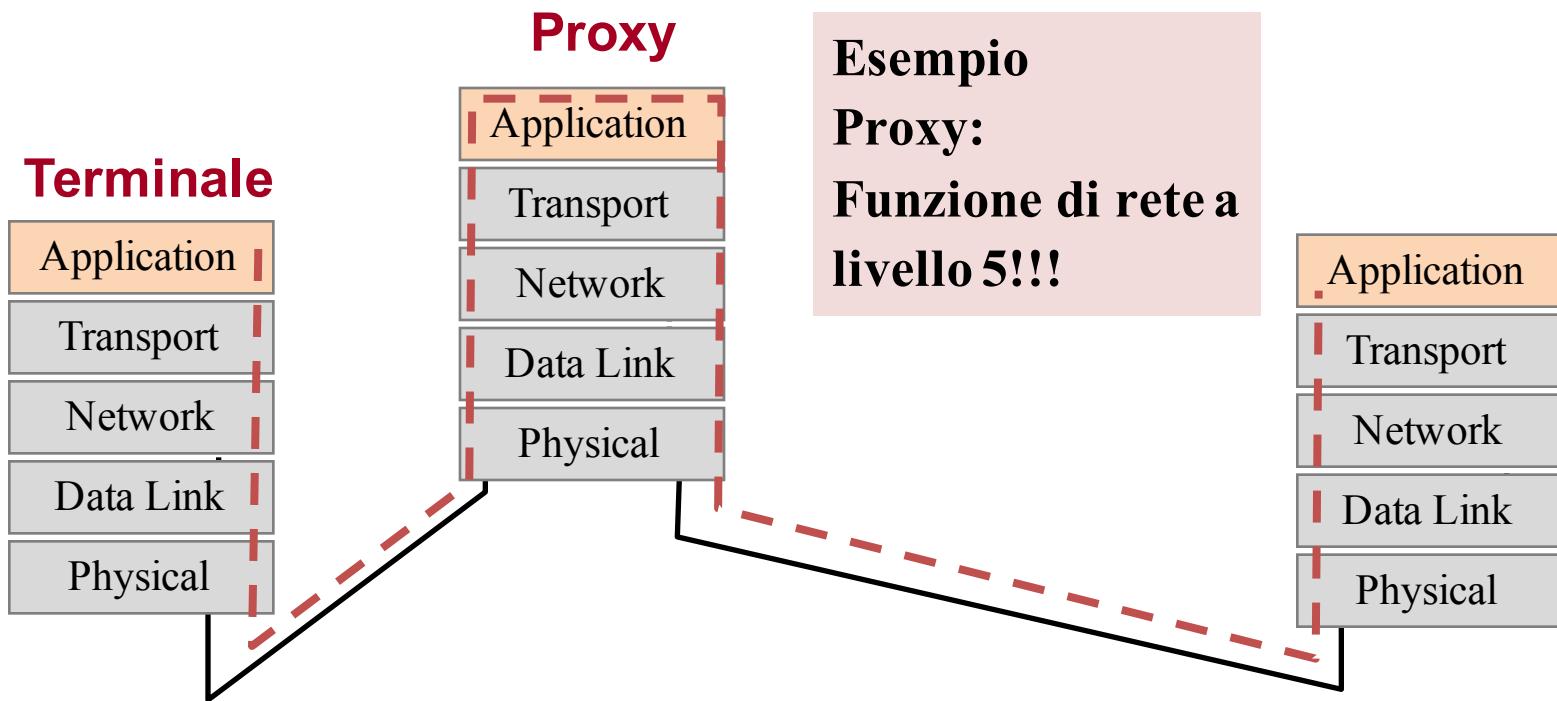
Esempio: Funzione di instradamento (routing)

- Esempio: LAN Switch



Esempio: Funzione di instradamento (routing)

- Esempio: proxy



Esempio: Funzione di instradamento (routing)

- Scelta dell'uscita avviene sulla base delle informazioni memorizzate in una tabella

tabella di routing	
indirizzo	uscita

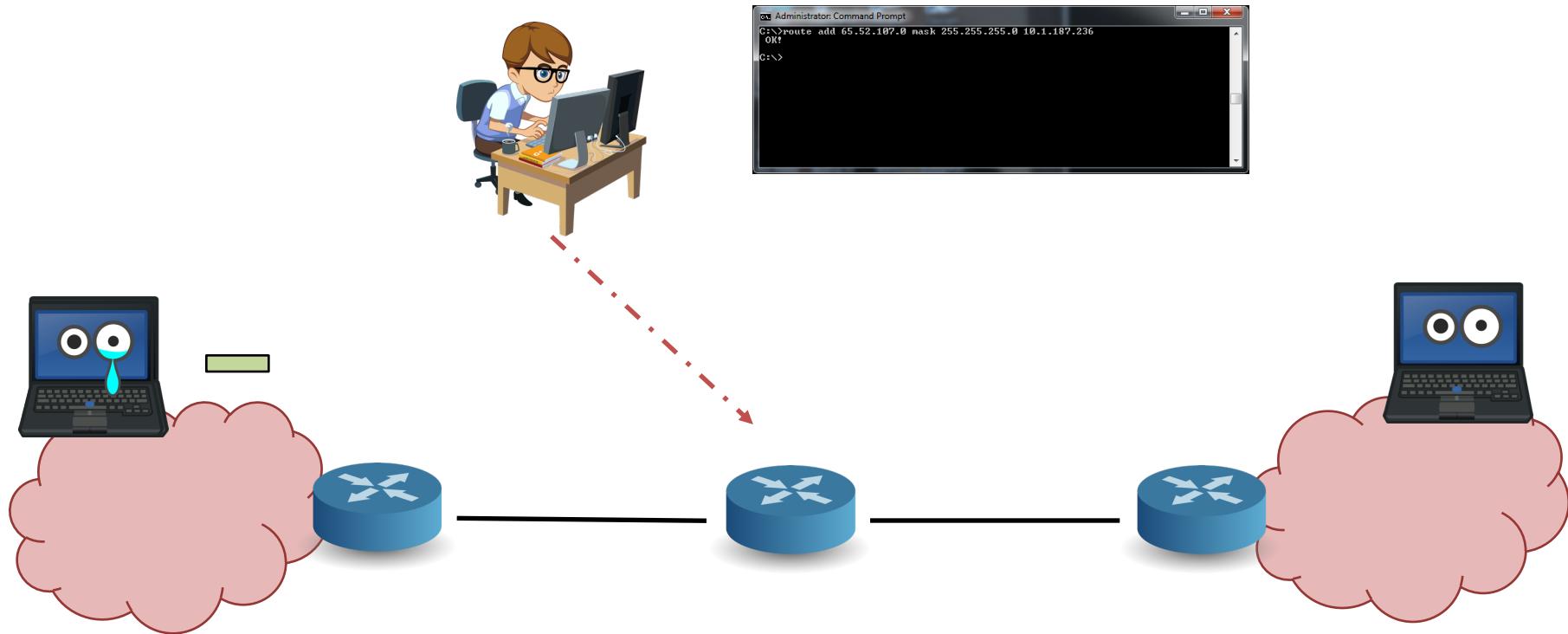
Ma chi scrive le tabelle di
instradamento



Esempio: Funzione di instradamento (routing)

“Human Defined Networking”

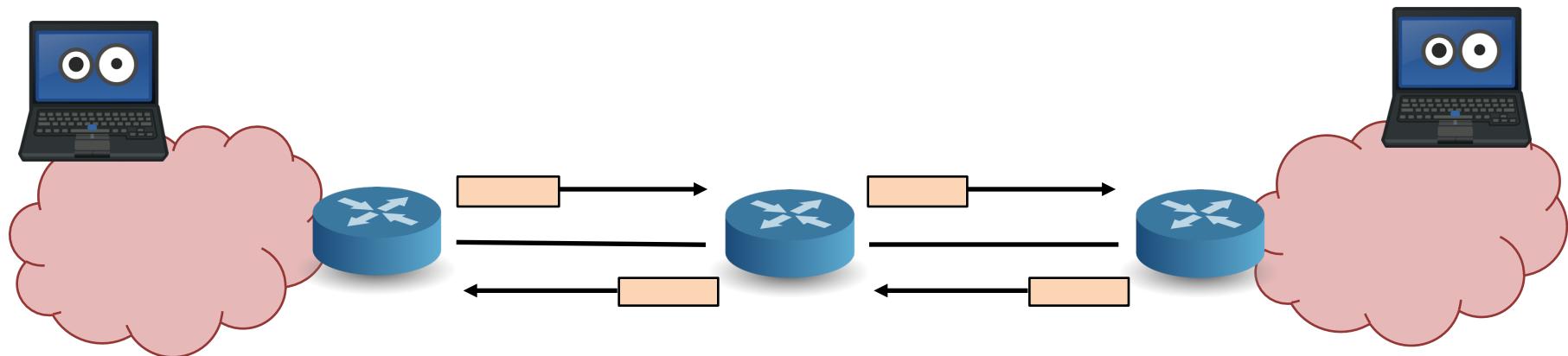
- Le tabelle di instradamento possono essere **scritte a mano** (faremo molti esercizi sulle tabelle di instradamento)
- Nei router IP le rotte configurate manualmente sono dette **rotte statiche**



Esempio: Funzione di instradamento (routing)

Protocolli di instradamento distribuiti

- L'approccio classico è però basato su uno **scambio di informazioni tra i router** che consente di compilare le tabelle di instradamento in modo **automatico e distribuito**
- Esempio:
 - MAC learning (si veda livello di linea)
 - Instradamento per cammini minimi (si veda livello IP)



Esempio: Funzione di instradamento (routing)

Software Defined Networking

- L'approccio emergente denominato **SDN** consente di operare come nell'approccio manuale, ma con un'**applicazione software centralizzata** che compila e modifica le tabelle di instradamento

