

# *Laboratorio di Reti di Computer*

---

## *Lezioni 1 e 2 LAN e Stack TCP/IP*

Claudio Covelli

[claudio.covelli@gmail.com](mailto:claudio.covelli@gmail.com)

Trento, 20 settembre 2006



# Agenda

## Obiettivi generali

- Spiegare il significato dei principali termini utilizzati nel mondo delle reti
- Capire gli aspetti essenziali del funzionamento delle reti, integrando teoria e pratica (saper e saper fare)
- Iniziare a progettare una semplice rete Ethernet/TCP/IP usando il sistema operativo Linux Ubuntu Dapper
- Riferimenti bibliografici:
  - Richard Stevens: [TCP/IP Illustrated, Volume 1: The Protocols](#), Addison-Wesley, 1994, ISBN 0-201-63346-9
  - [www.tcpipguide.com](http://www.tcpipguide.com)

# Agenda

## Introduzione alle reti

- Introduzione alle reti di tipo LAN
- **Laboratorio:** comandi ifconfig e ping
- Stack TCP/IP e modello ISO-OSI
- **Laboratorio:** approfondimento ifconfig e ping fra sottoreti; analisi dei pacchetti con Ethereal

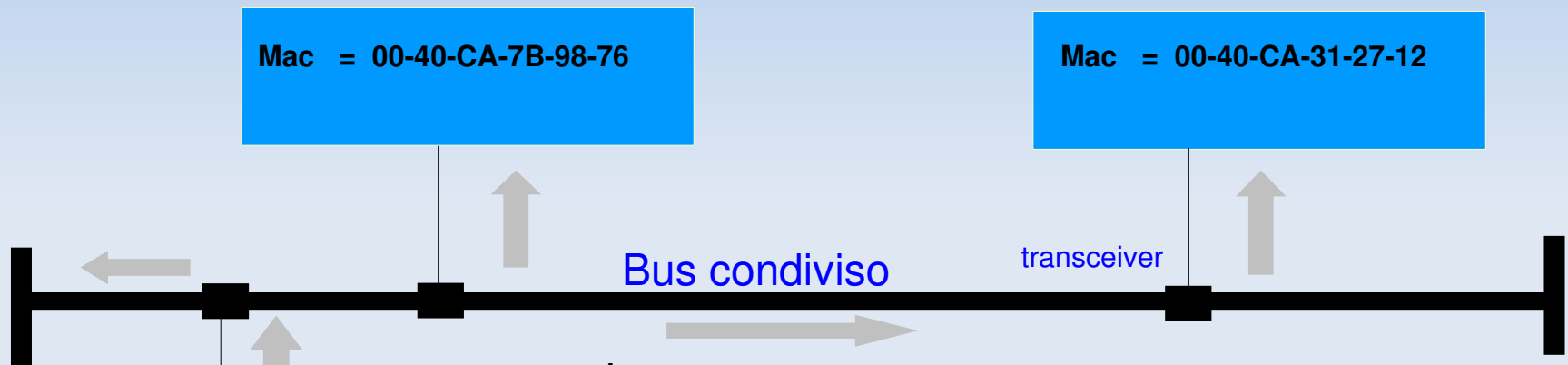
# LAN e Stack TCP/IP

## LAN (Local Area Network):

- ▶ Rete **privata** di computer “**fisicamente vicini**” (fino a qualche km), connessi mediante **schede di rete** ed opportuno cablaggio (hub, switch, cavi rame o fibra)
- ▶ Diverse tecnologie delle schede di rete (es. Tokenring, FDDI, **Ethernet**)
- ▶ Ogni scheda di rete ha un suo **indirizzo fisico**, assegnato dal costruttore (**Mac Address**). Per Ethernet 6 bytes.
- ▶ Ethernet nasce come sistema **broadcast** su canale (**bus**) condiviso (trasmissione simultanea a più stazioni)

# LAN e Stack TCP/IP

## Esempio di LAN a bus condiviso



- Il pacchetto, inviato da una stazione ad un certo destinatario, viene ricevuto da tutte le altre (**broadcasting**)
- Se pensiamo di ridurre la lunghezza del bus e di inserire una porta RJ45 al posto di ogni transceiver, otteniamo l'hub
- Lo switch è analogo all'hub, ma, al posto di un bus condiviso, si hanno connessioni fisiche **distinte** fra porta e porta

# LAN e Stack TCP/IP

## LAN

- La scheda di rete **mittente** invia un pacchetto (frame Ethernet) sul bus comune, specificando l'indirizzo fisico della scheda destinataria
- Solo la scheda di rete **avente indirizzo fisico corrispondente al destinatario** tratta il pacchetto; le rimanenti lo scartano
- L'invio del pacchetto su bus condiviso può provocare **collisioni**; **tale possibilità** aumenta all'incremento del numero di stazioni sulla LAN
- Esempi di rete che verranno ripresi durante il laboratorio (2 PC connessi con cavo incrociato/hub/switch)

# LAN e Stack TCP/IP

## LAN

- ♦ Una LAN può essere divisa, per motivi **prestazionali (collisioni)** ed **organizzativi**, in **sottoreti indipendenti**, fra loro collegate tramite **router** (vedi esempio seguente)
- ♦ Un router è un dispositivo costituito da **più schede di rete** (una per ogni sottorete collegata) e da **software** in grado di effettuare l'inoltro (forwarding) dei pacchetti fra le sue schede e quindi fra le diverse sottoreti ad esso connesse
- ♦ Il router si limita ad inoltrare i pacchetti da una sottorete all'altra, quando mittente e destinatario si trovano su due sottoreti diverse, da esso collegate
- ♦ In tal modo si limitano broadcasting e collisioni, mantenendo **logicamente** collegate **più sottoreti fisicamente distinte**

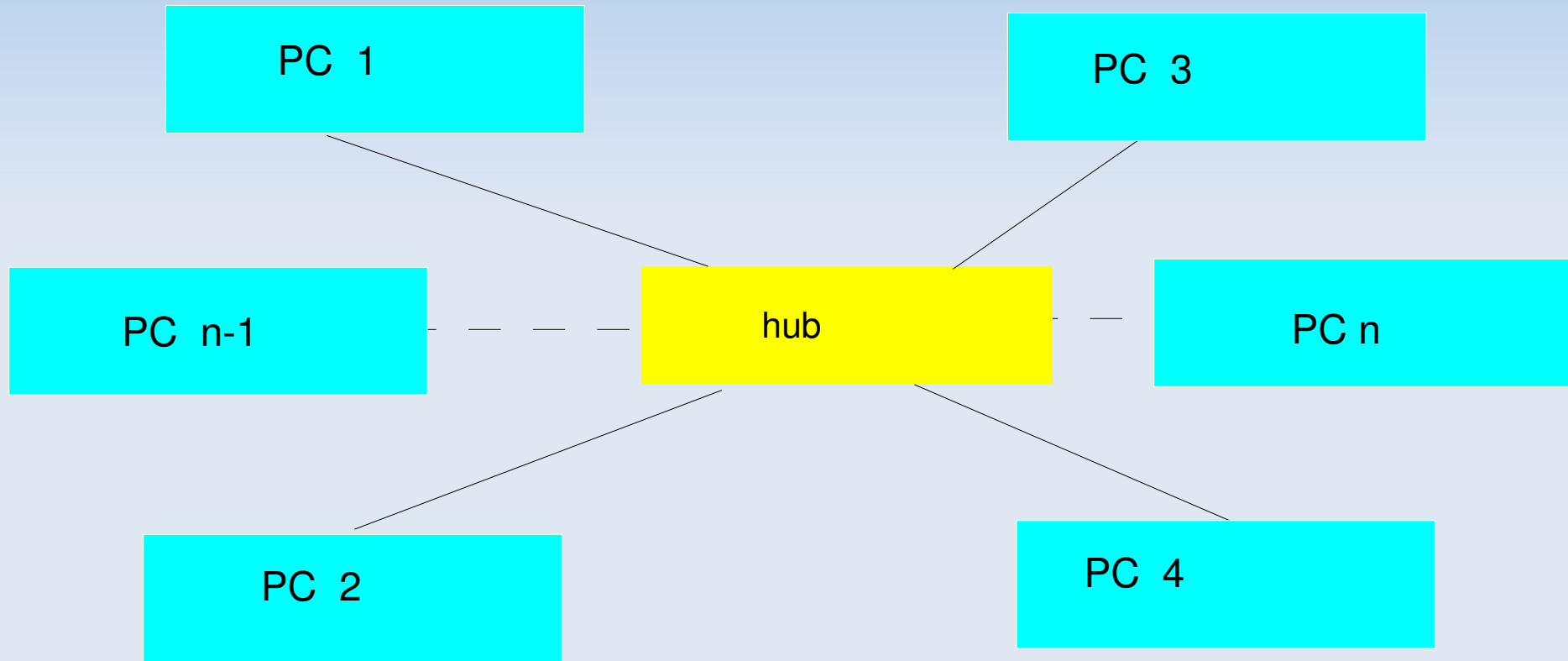
# LAN e Stack TCP/IP

## LAN

- ▶ I router si usano anche per collegare fra loro logicamente LAN differenti
- ▶ Internet è formato da LAN fra loro collegate mediante router

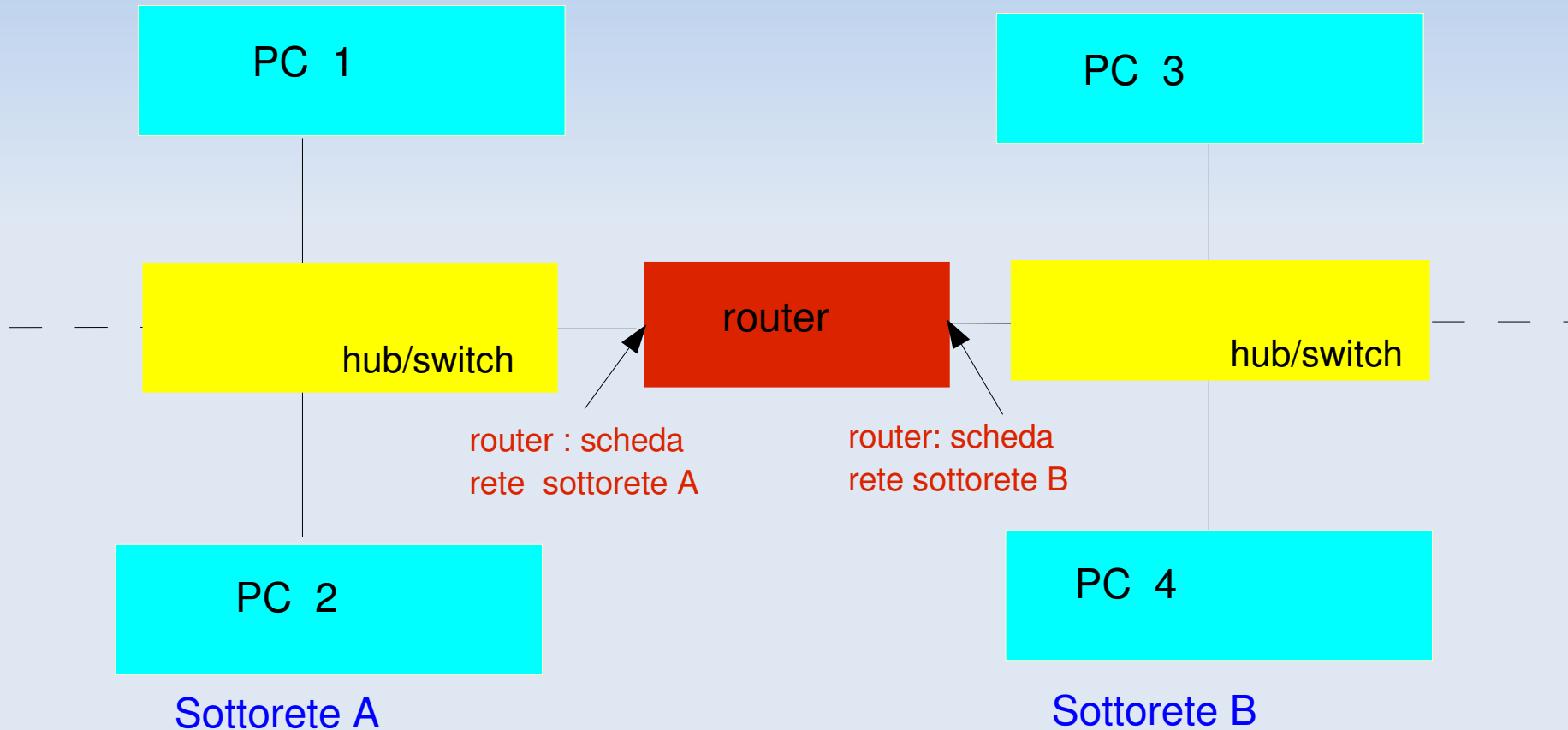
# LAN e Stack TCP/IP

Esempio di LAN unica



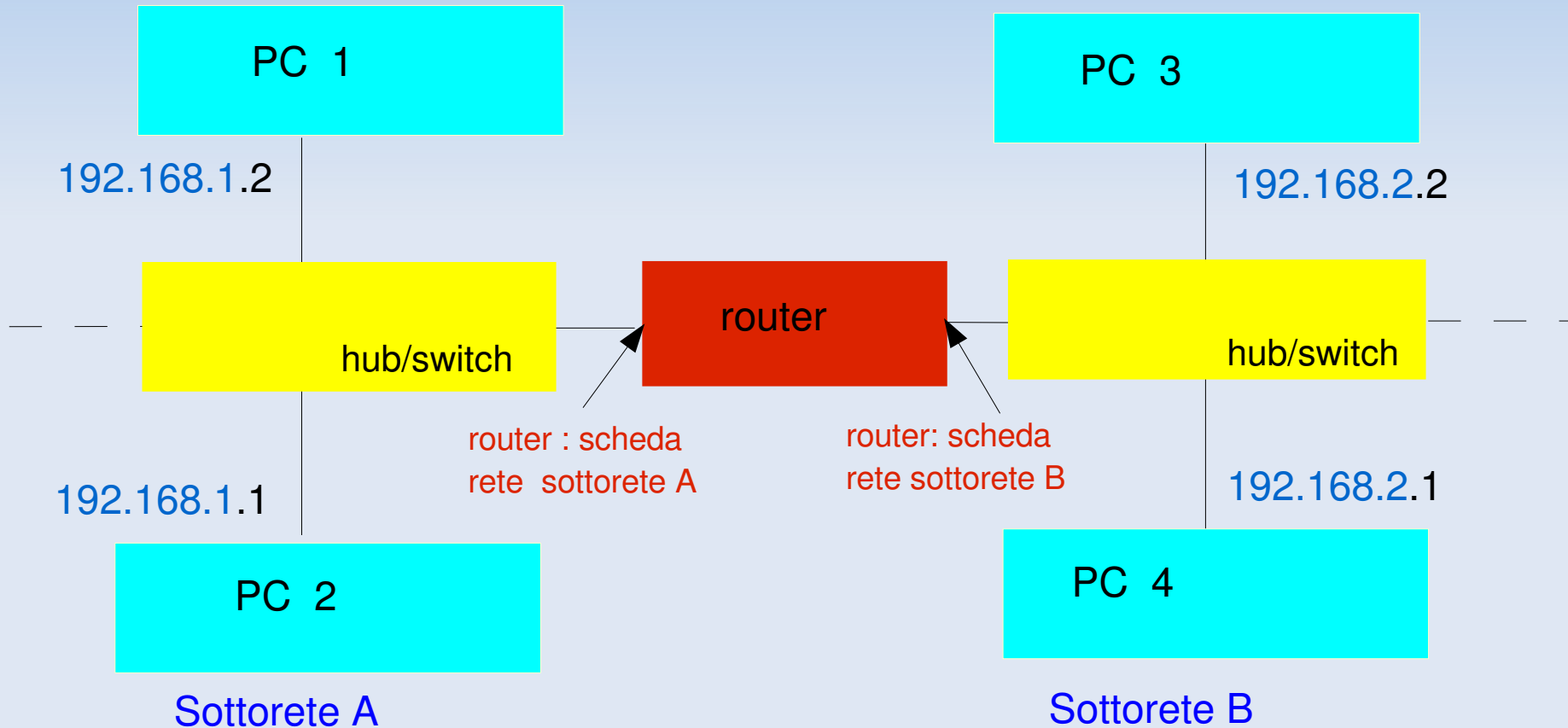
# LAN e Stack TCP/IP

Esempio di sottoreti fra loro connesse tramite router



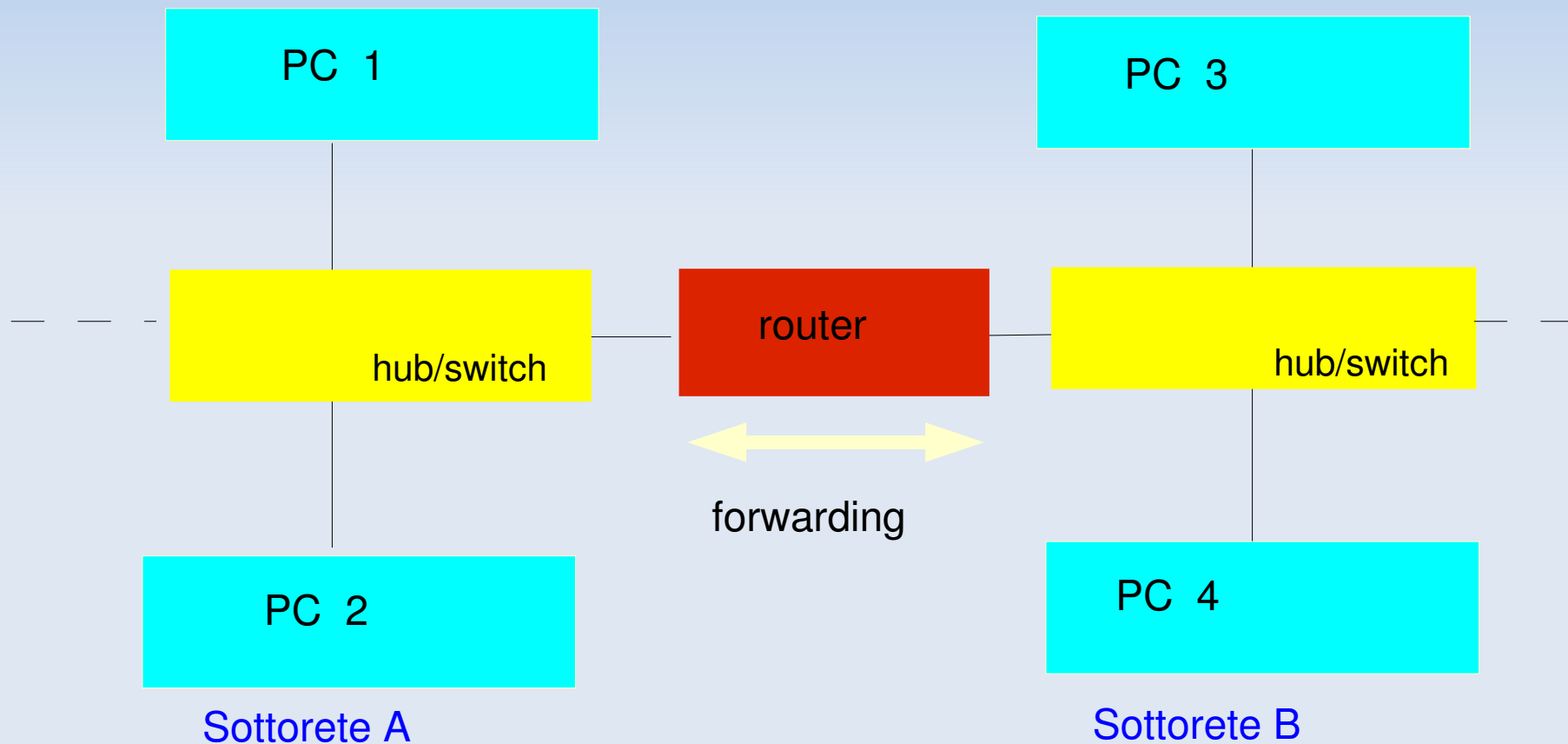
# LAN e Stack TCP/IP

Esempio di sottoreti fra loro connesse tramite router



# LAN e Stack TCP/IP

Esempio di sottoreti fra loro connesse tramite router



# LAN e Stack TCP/IP

## INDIRIZZI LOGICI

- ▶ Per individuare una sottorete, si usa un indirizzamento logico (**indirizzo IP**), i cui 32 bit sono divisi in due parti distinte:
  - 1) I bit iniziali, in numero variabile, rappresentano l'identificativo della sottorete (**net-id o net-prefix**)
  - 2) I rimanenti bit costituiscono l'identificativo del computer collegato alla sottorete (**host-id**)
- ▶ Il numero di bit che costituisce il **net-id** è fornito dal parametro **subnet-mask**

# LAN e Stack TCP/IP

## INDIRIZZI LOGICI

- ▶ I bit ad 1 di questo parametro rappresentano i bit dell'indirizzo IP che formano il net-id (vedi esempio)
- ▶ Anziché indicare la subnet-mask, si può anche specificare, a fianco dell'indirizzo IP, il numero di bit corrispondenti al net-id (es 192.15.32.2/20)

# LAN e Stack TCP/IP

Esempio suddivisione di indirizzo IP:

- ▶ Indirizzo IP 192.15.32.2 mask 255.255.255.0

IP        11000000.00001111.00100000.00000010

MASK 11111111.11111111.11111111.00000000

- ▶ Tale indirizzo contiene, nei 24 bits iniziali, il net-id 192.15.32
- ▶ Host-id = rimanenti 8 bit = 2

NOTA: host-id non può mai assumere il valore minimo (0) e massimo (tutti i bit ad 1)

# LAN e Stack TCP/IP

## Riassunto: concetti essenziali

- L'indirizzo IP ed il valore di mask permettono un raggruppamento logico di indirizzi fisici: host che condividono il net-id appartengono alla stessa sottorete
- Net-id identifica la sottorete di appartenenza
- Host-id identifica il dispositivo di rete nell'ambito della sottorete di appartenenza
- Se due sottoreti hanno differente net-id, i pacchetti possono essere trasmessi, da una all'altra, solo tramite un router (lab. n. 3)

# LAN e Stack TCP/IP

## Riassunto: concetti essenziali

- ♦ L'host mittente:
  - ♦ quando la sottorete del destinatario è la stessa, invia il pacchetto alla scheda di rete destinataria, specificandone l'indirizzo fisico nel pacchetto (lab. n.1 e 2)
  - ♦ quando la sottorete è diversa:
    - ♦ Invia il pacchetto alla scheda di rete del router, specificandone l'indirizzo fisico nel pacchetto (lab. n.3)
    - ♦ Il pacchetto viene quindi inoltrato, dal software del router, alla scheda di rete collegata alla sottorete destinataria.
    - ♦ Questa trasmette il pacchetto al destinatario specificandone l'indirizzo fisico

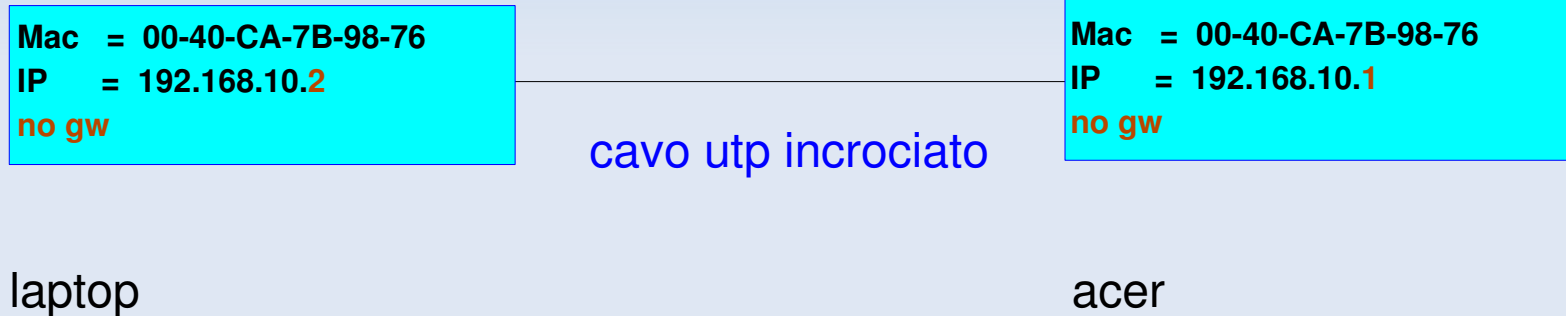
# LAN e Stack TCP/IP

Riassunto: concetti essenziali

DOMANDE?

# LAN e Stack TCP/IP

Laboratorio n. 1 : Unica Sottorete 192.168.10



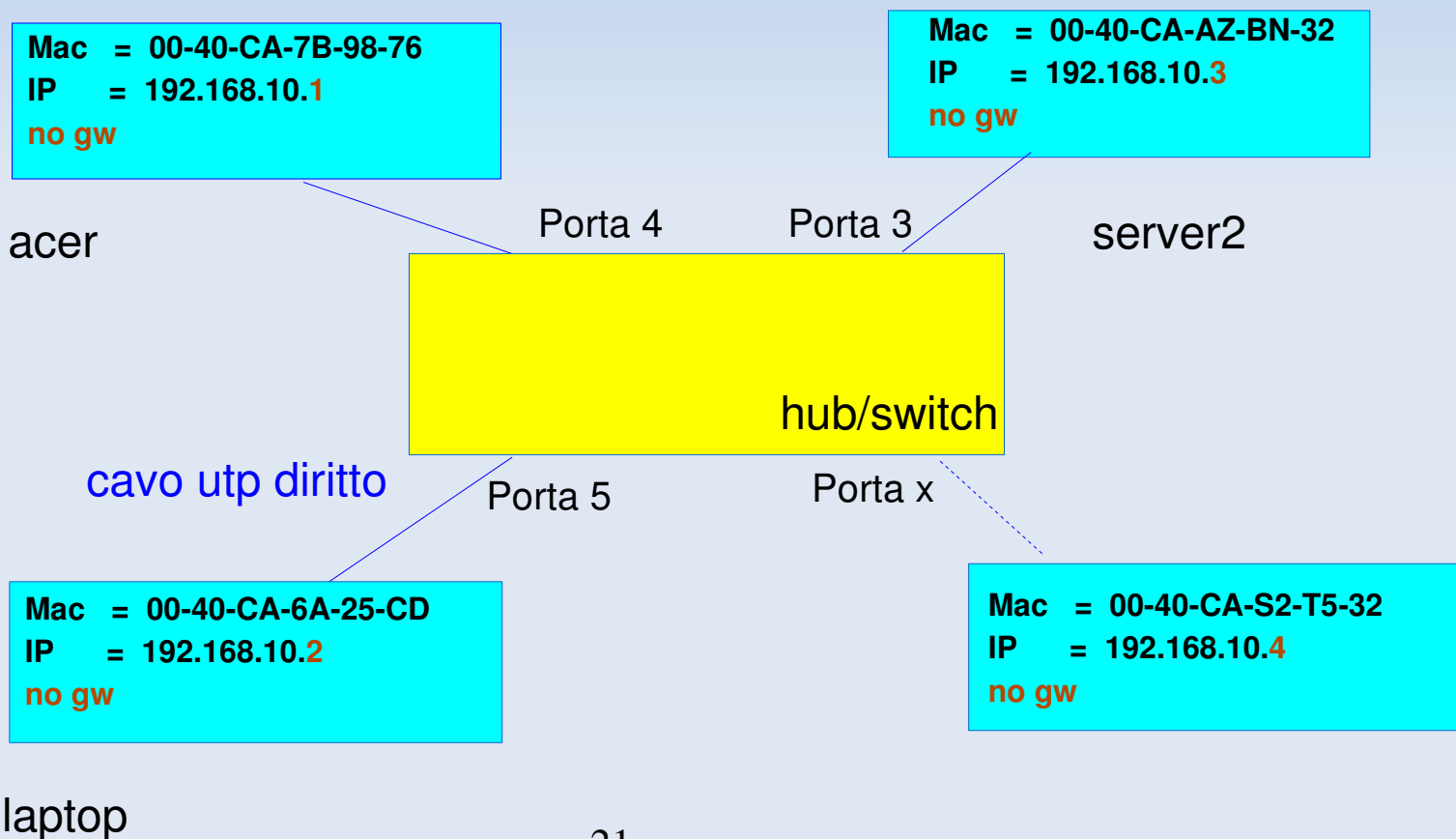
# LAN e Stack TCP/IP

## Laboratorio n. 1:

- ▶ Analisi dei comandi `ping` ed `ifconfig`
- ▶ Eseguire il comando `ping` fra due computer (NB: non specificare gateway)
- ▶ Visualizzare i pacchetti con `ethereal`
- ▶ Cambiare l'indirizzo di sottorete ad uno dei computer e verificare l'errore "Network is unreachabile" sul `ping`

# LAN e Stack TCP/IP

Laboratorio n. 2 : Unica Sottorete 192.168.10



# LAN e Stack TCP/IP

## Laboratorio n. 2:

- ▶ Eseguire il comando ping fra due computer (NB: non specificare gateway)
- ▶ Visualizzare i pacchetti con ethereal
- ▶ Cambiare l'indirizzo di sottorete ad uno dei computer e verificare l'errore “Network is unreachabile” sul ping

# LAN e Stack TCP/IP

- Vediamo ora come avviene la **trasmissione fra due computer**, appartenenti o meno a reti differenti (ad esempio un client ed un server sulla stessa sottorete oppure su due sottoreti/reti diverse)
- Vedremo che essa è basata su **pacchetti** che contengono i dati delle applicazioni client/server, arricchiti via via di informazioni specifiche dai **vari strati software sottostanti coinvolti nella loro trasmissione**
- Quali sono questi strati software?

# LAN e Stack TCP/IP

- Quando un programma client si collega ad un processo server imposta una socket (ossia un canale di comunicazione client-server), utilizzato in modo analogo ad un file (es.)
- La socket si basa su 5 parametri:
  - Indirizzo IP mittente
  - Porta mittente (numero progressivo assegnato alla socket; ephemeral port)
  - Indirizzo IP destinatario
  - Porta destinatario (generalmente di tipo well-known)
  - Protocollo (TCP oppure UDP)

# LAN e Stack TCP/IP

- Impostata la socket, il client invia e riceve messaggi con modalità simili alla lettura/scrittura su file
- La socket evita al programma client di doversi curare di tutti i dettagli implementativi del processo di comunicazione

# LAN e Stack TCP/IP

```
/* esempio di socket client TCP in java */
protected String host;
protected int port;
protected DataInputStream in;
protected DataOutputStream out;

protected Socket connect () throws IOException {
    System.err.println ("Connessione a " + host + ":" + port + "...");
    Socket socket = new Socket (host, port);
    System.err.println ("Connessione avvenuta.");
    out = new DataOutputStream (socket.getOutputStream ());
    in = new DataInputStream (socket.getInputStream ());

    out.writeBytes ("GET " + file + " HTTP/1.0\r\n\r\n");
    .....
    return socket;
}
```

# LAN e Stack TCP/IP

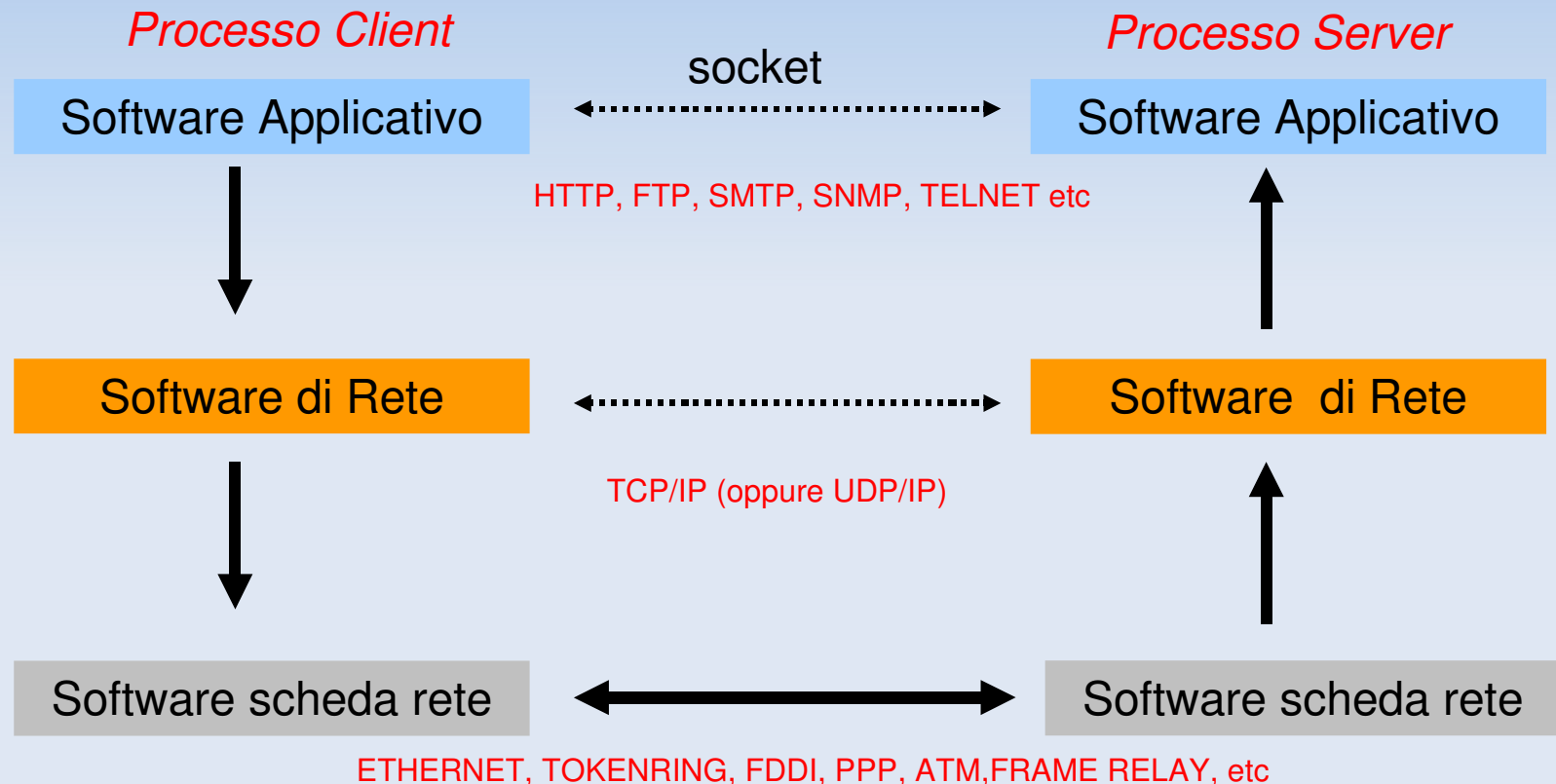
```
// Esempio di socket lato server
System.out.println ("Server in partenza sulla porta " + port);
ServerSocket server = new ServerSocket (port);
System.out.println ("Server partito sulla porta " +
    server.getLocalPort() );
System.out.println ("In attesa di connessioni...");
Socket client = server.accept ();
System.out.println ("Richiesta di connessione da " +
    client.getInetAddress () );
InputStream i = client.getInputStream ();
OutputStream o = client.getOutputStream ();
PrintStream p = new PrintStream (o) ;
p.println("BENVENUTI.")
```

# LAN e Stack TCP/IP

- La socket pertanto costituisce **un meccanismo di astrazione** che consente ai processi applicativi di ignorare le problematiche connesse alla comunicazione (es. controllo errori di trasmissione, ritrasmissione pacchetti etc)
- Di tali problematiche si fanno infatti carico strati di software inferiori, che interagiscono con la socket e sono inclusi nel sistema operativo (TCP/IP oppure UDP/IP)
- A loro volta TCP/IP ed UDP/IP utilizzano il **software della scheda di rete** per la comunicazione a livello LAN (ossia la comunicazione nella sottorete di appartenenza della scheda di rete, basata sui MAC Address)

# LAN e Stack TCP/IP

Un primo modelli di comunicazione fra due computer



La vera e propria trasmissione dei dati avviene a livello fisico; negli strati soprastanti la trasmissione dei dati avviene solo a livello software e nell'ambito della stessa macchina

# LAN e Stack TCP/IP

- Vediamo ora cosa succede quando due stazioni, ad esempio un client ed un server, si scambiano dati **su un'unica sottorete TCP/IP (ossia appartengono ad una LAN Ethernet identificata da un unico network Id)**
- I concetti esposti valgono anche nel caso di più reti fra loro collegate; in tal caso però sono necessari dei **router** in grado di instradare i pacchetti fra le varie LAN collegate, in base all'Id di rete contenuto nei pacchetti
- I concetti esposti valgono, **da un punto di vista generale**, anche nel caso di altre tecnologie diverse da TCP/IP (es. UDP/IP) ed Ethernet (es. TokenRing), ma l'attenzione viene posta ora, per comodità di esposizione, solo su queste

# LAN e Stack TCP/IP

In **invio**:

1. Un primo strato (applicativo) gestisce i messaggi applicativi che il processo mittente (**client**) deve inviare al processo destinatario (**server**)

I messaggi vengono inviati dal processo mittente allo strato software sottostante detto strato di **trasporto** (transport), tramite la socket

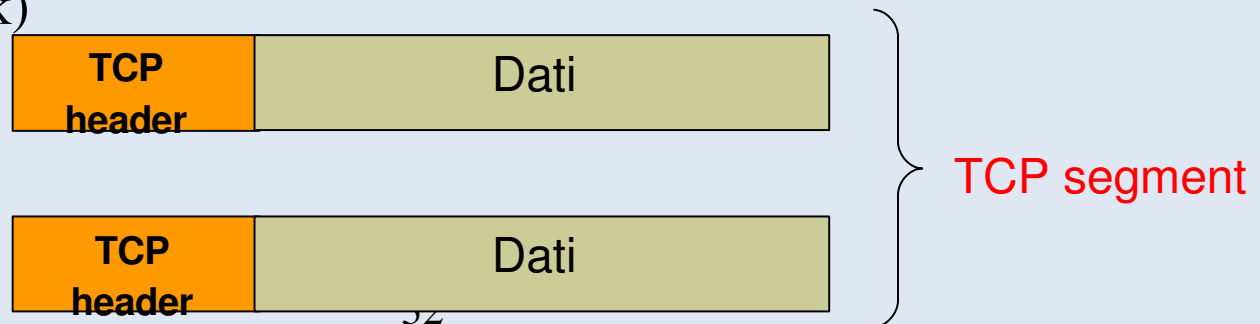


Dati applicativi

# LAN e Stack TCP/IP

- Il secondo strato (**trasport;TCP**)
  - ↳ Riceve i messaggi dallo strato applicativo
  - ↳ **Li suddivide in pacchetti** (TCP segment)
  - ↳ **Associa ad ogni pacchetto un header** (TCP header) che contiene informazioni necessarie alla **corretta consegna** del pacchetto al processo di destinazione (fra i più importanti, **le porte** del processo client e quella del processo server ed **il numero di sequenza**).

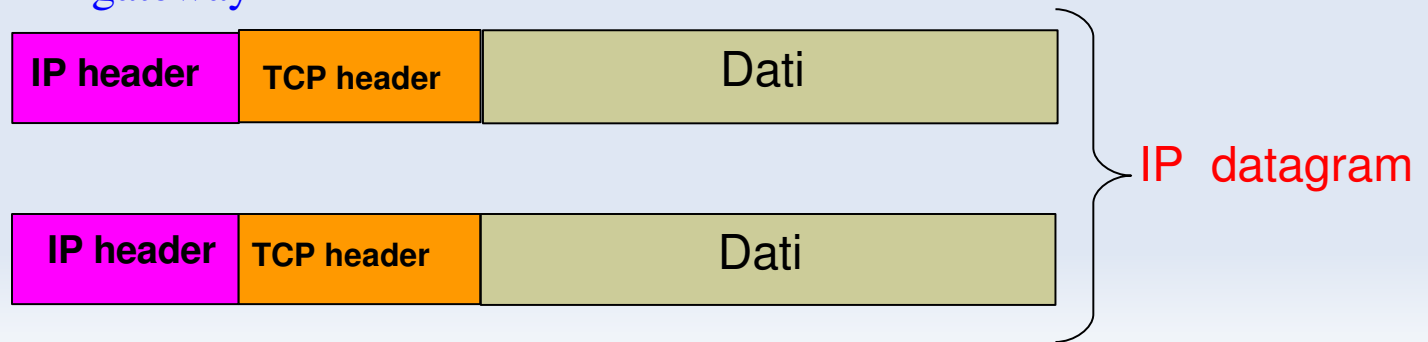
I pacchetti vengono quindi passati allo strato sottostante detto di rete (network)



# LAN e Stack TCP/IP

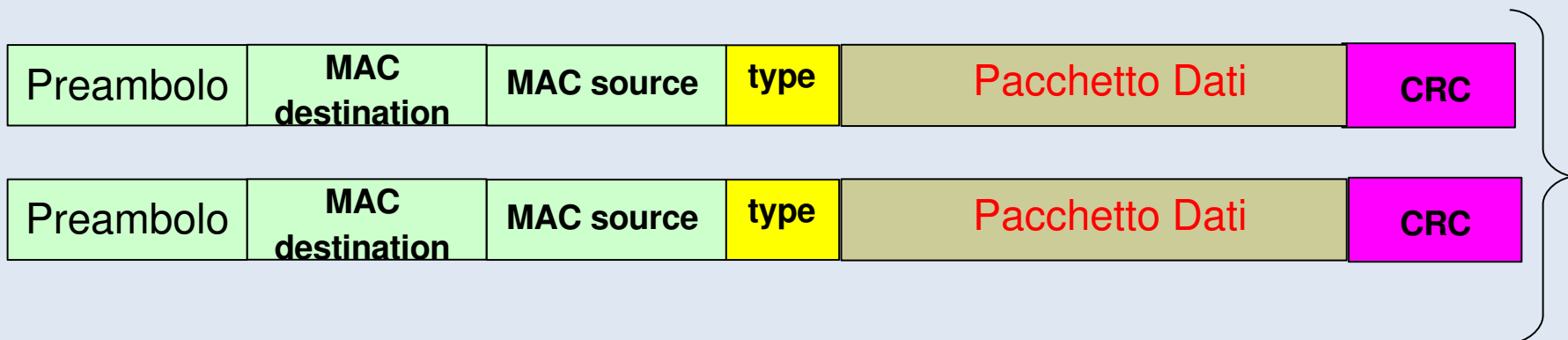
## 3. Il terzo strato (**network;IP**):

- Riceve i messaggi dallo strato di trasporto
- Può, se necessario **frammentare ulteriormente i pacchetti** (se la lunghezza del singolo pacchetto è differente fra diverse sottoreti)
- Associa ad ogni pacchetto (IP datagram) un header (IP header) che contiene informazioni necessarie alla consegna del pacchetto all'host di destinazione (fra i più importanti, **indirizzo IP mittente e destinatario**).
- Mappa l'indirizzo IP del destinatario in indirizzo fisico (Mac), tramite ARP. Qualora il destinatario sia in sottorete/rete IP differente, **l'indirizzo fisico del destinatario viene posto eguale a quello del gateway**

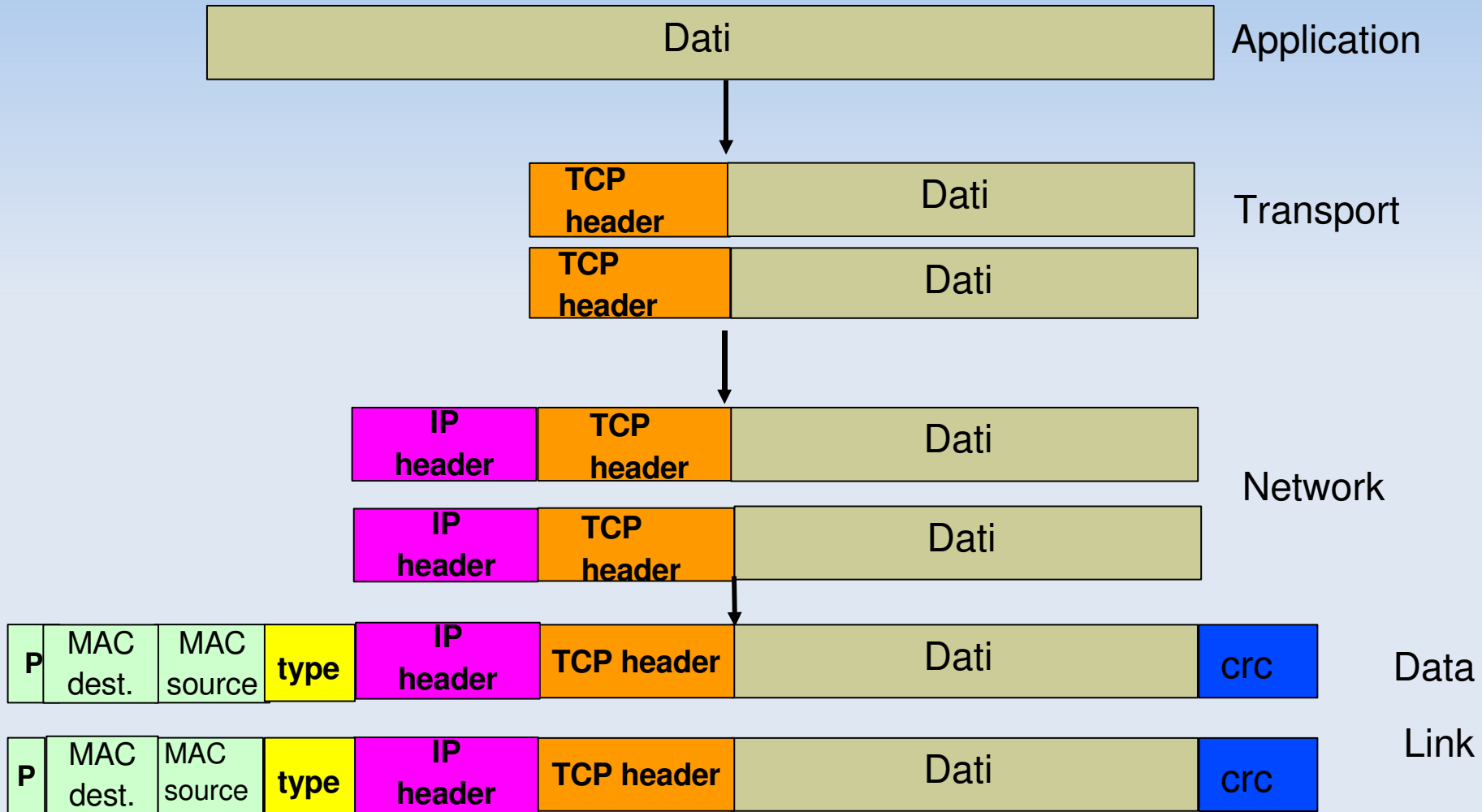


# LAN e Stack TCP/IP

- Il quarto strato (**data link;ETHERNET**)
  - Riceve i pacchetti dallo strato di rete
  - Associa ad ogni pacchetto un header che contiene informazioni necessarie per la trasmissione dei pacchetti all'interno della stessa sottorete.
  - Invia il pacchetto (frame Ethernet) all'indirizzo fisico del destinatario



# LAN e Stack TCP/IP



# LAN e Stack TCP/IP

Modello standard di rappresentazione di comunicazione TCP/IP

application

HTTP, FTP, SMTP, etc

transport

TCP, UDP

network

IP, ARP, RARP, ICMP

Data link

**Ethernet**, Tokenring,  
PPP, ATM, altri

Mittente

# LAN e Stack TCP/IP

Nel destinatario avviene il processo inverso !!!

application

HTTP, FTP, SMTP, etc

transport

TCP, UDP

network

IP, ARP, RARP, ICMP

Data link

**Ethernet**, Tokenring,  
PPP, ATM, altri

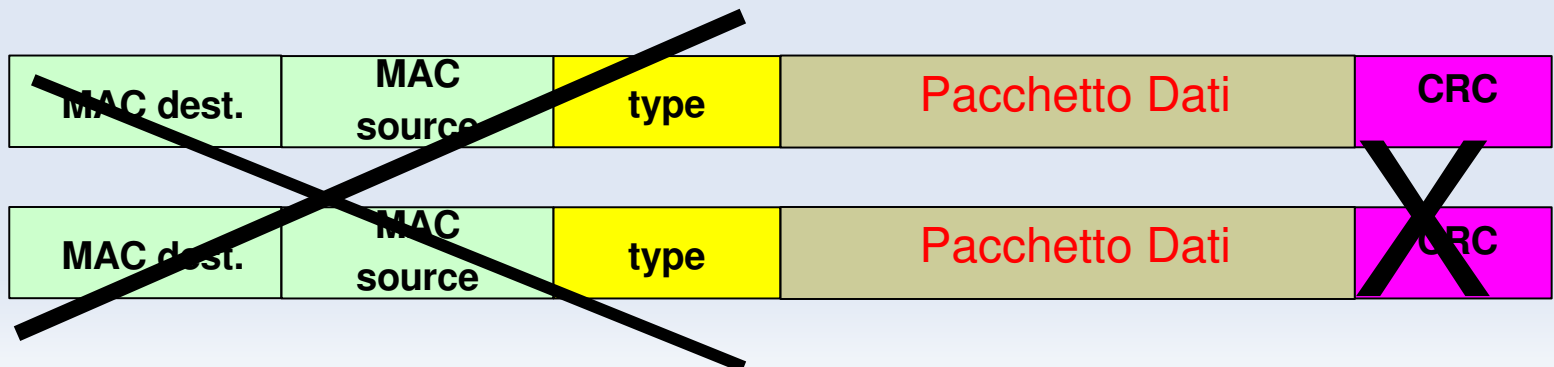
Destinatario

# LAN e Stack TCP/IP

Vediamo ora cosa succede invece dal lato del destinatario

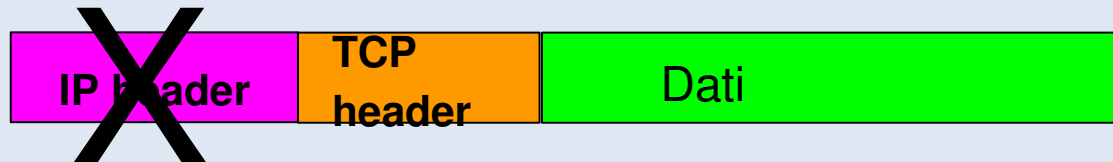
In **ricezione** avviene il **processo simmetrico**:

1. Il destinatario:
  - ▶ Controlla che **l'indirizzo fisico destinatario** corrisponda a quello della sua scheda di rete
  - ▶ Ricalcola CRC e richiede la ritrasmissione del frame alla scheda Ethernet mittente, se CRC diverso da quello calcolato dal mittente
  - ▶ Toglie gli header dello strato data link ed invia il pacchetto allo strato superiore (network)



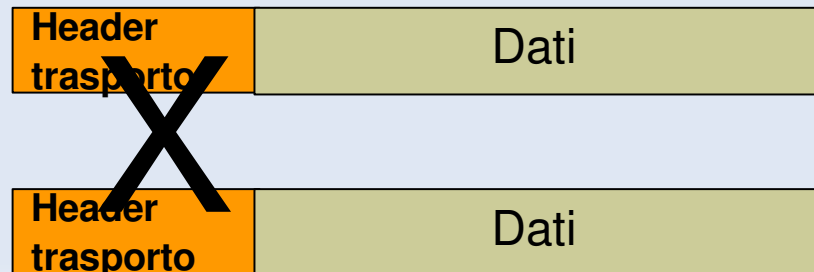
# LAN e Stack TCP/IP

2. Il secondo strato (IP;network):
- Riceve i messaggi dallo strato data link
  - Toglie l'header dello strato di rete
  - Effettua i controlli di merito (es. checksum)
  - Se non richiesto routing (stessa sottorete), passa il pacchetto allo strato superiore (TCP; trasporto)
  - Se richiesto routing (sottorete IP destinatario diversa da sottorete IP mittente), passa il pacchetto ad un'altra delle sue schede di rete incaricata dell'instradamento



# LAN e Stack TCP/IP

- Il terzo strato (TCP; **trasporto**)
  - ↳ Riceve i messaggi dallo strato network
  - ↳ Notifica il corrispondente strato del mittente della corretta ricezione del pacchetto (**acknowledgment del numero di sequenza**)
  - ↳ Ricompone il messaggio originario ordinando la sequenza dei pacchetti ricevuti
  - ↳ Controlla che non sia stato perso alcun pacchetto e che non ve ne siano di doppi
  - ↳ Passa infine il messaggio così ricostruito alla socket lato server di porta corrispondente



# LAN e Stack TCP/IP

4. Il quarto strato (applicativo)
  - Riceve il messaggio trasmessogli dallo strato di trasporto
  - Ne effettua il trattamento.

Generalmente risponde al mittente con un altro messaggio (ad esempio una pagina HTML) ed in tal caso si ripercorre la strada già vista all'incontrario



Dati applicativi

# LAN e Stack TCP/IP

## In sintesi:

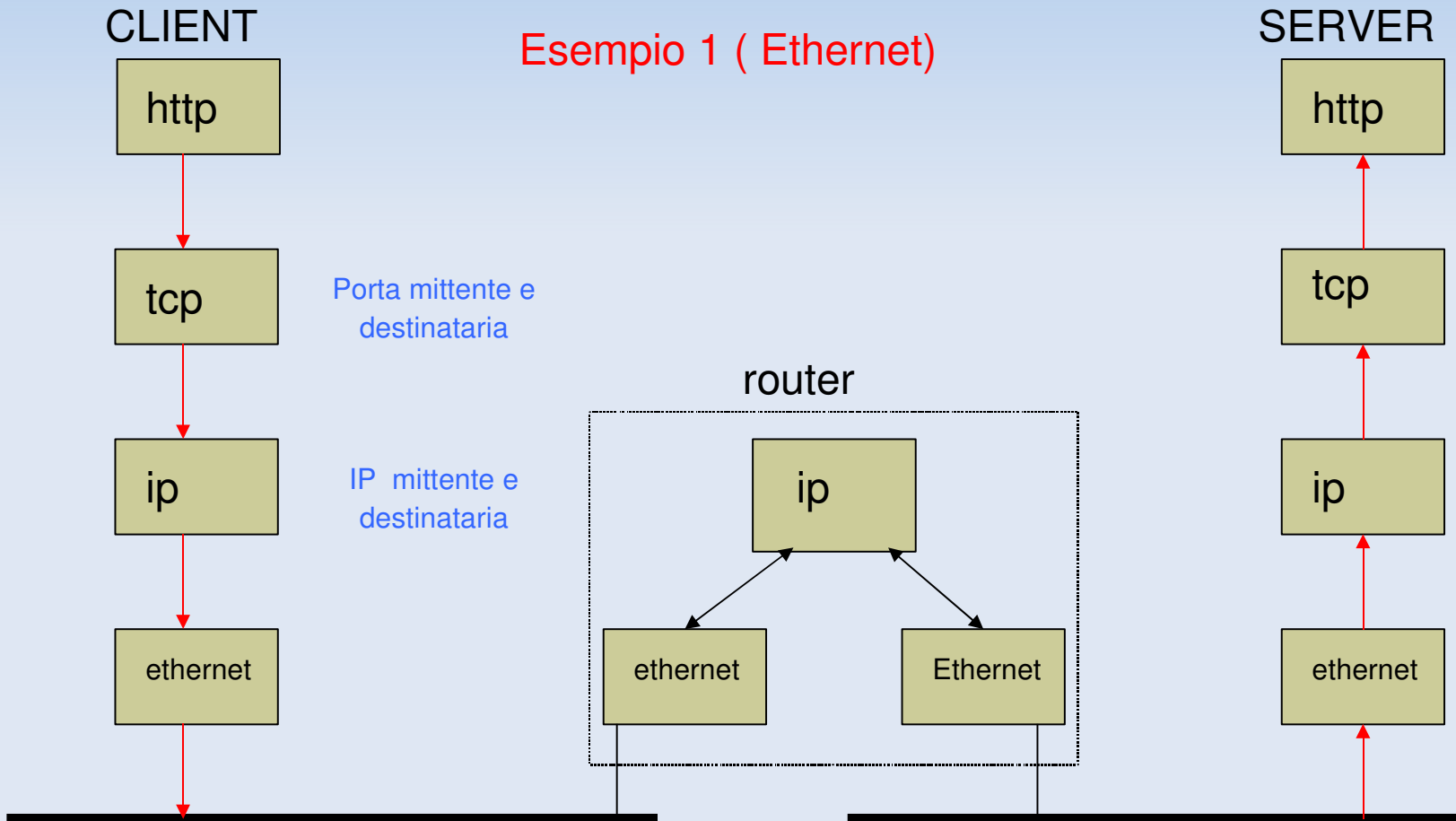
- ◆ Esiste un primo livello (**data link**) che consente la trasmissione fisica di tali pacchetti all'interno della stessa sottorete
- ◆ Tale livello utilizza gli indirizzi fisici (**Mac Address**) per l'invio dei pacchetti all'interno della stessa LAN
- ◆ Esiste un secondo livello (**network**) che gestisce l'instradamento dei pacchetti fra le varie sottoreti, in base all'indirizzo logico (indirizzo IP) del destinatario

# LAN e Stack TCP/IP

## In sintesi:

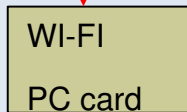
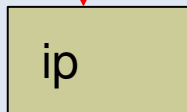
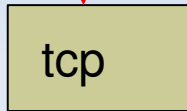
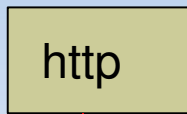
- ▶ Esiste un terzo livello (**transport**) che si fa carico di controllare che tutti i pacchetti trasmessi siano stati correttamente ricevuti dal destinatario, di notificare il mittente della corretta ricezione e di consegnare i pacchetti al processo destinatario
- ▶ Infine esiste un quarto livello (**applicativo**) che consente di inviare messaggi da un client ad un server, **conoscendo indirizzi IP e porte mittenti e destinatarie**. L'invio e la ricezione di tali messaggi è gestito dai livelli sopra descritti

# LAN e Stack TCP/IP



# LAN e Stack TCP/IP

CLIENT

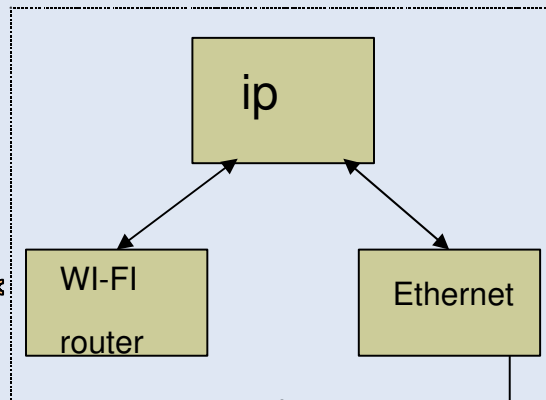


Porta mittente e  
destinataria

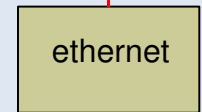
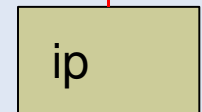
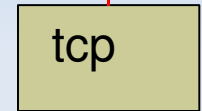
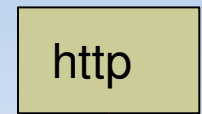
IP mittente e  
destinataria

Esempio 2 (WI-FI ed Ethernet)

router



SERVER



# LAN e Stack TCP/IP

Mappatura Stack TCP/IP -> Stack OSI

Application

HTTP, FTP, SMTP, etc

**7 Application**

**6 Presentation**

**5 Session**

Transport

TCP, UDP

**4 Transport**

Network

IP, ARP, RARP, ICMP

**3 Network**

Data Link

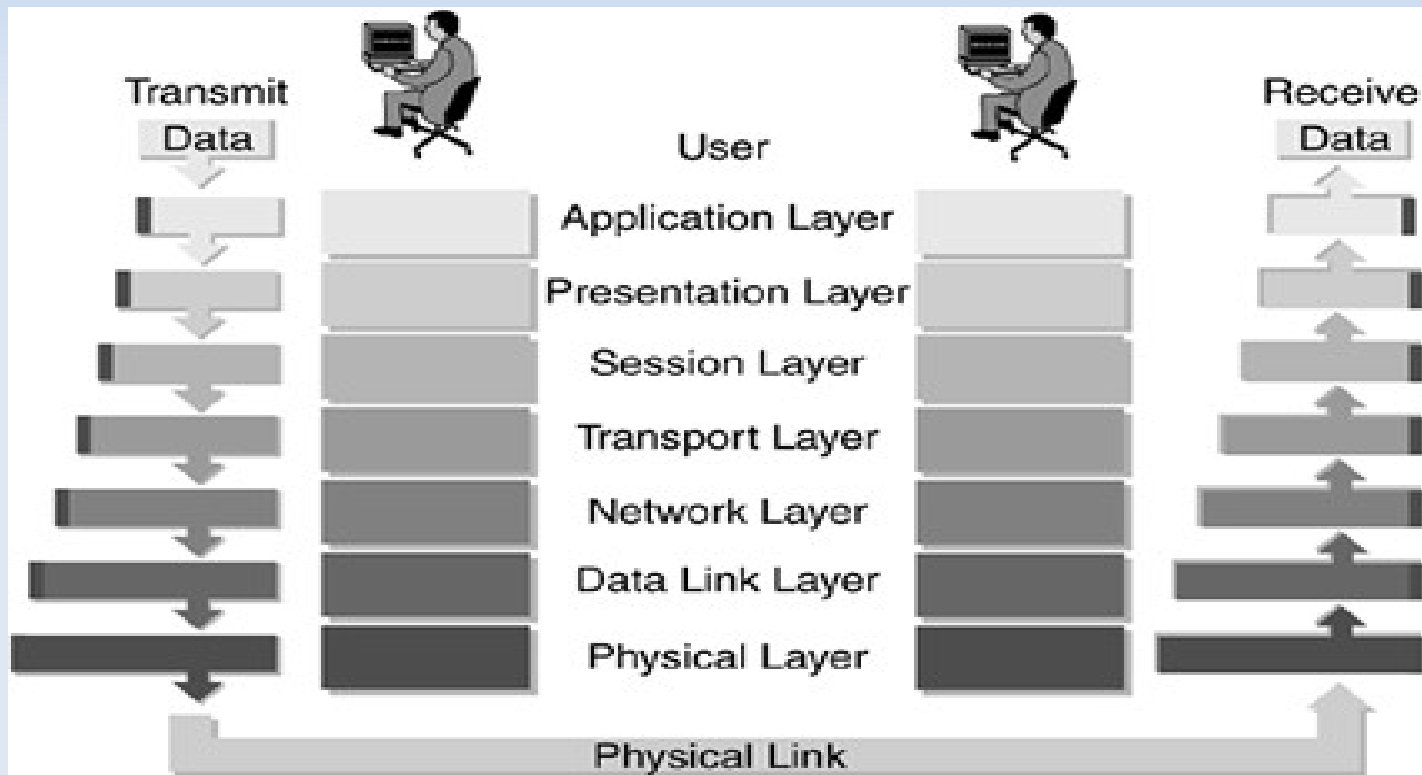
Ethernet, PPP,  
altri

**2 Data Link (LLC,MAC)**

**1 Physical**

# LAN e Stack TCP/IP

## Stack ISO-OSI



# LAN e Stack TCP/IP

## Layer 7—Application Layer

- E' il livello dove operano le applicazioni utente
- Esempi : browser, client di posta

# LAN e Stack TCP/IP

## Layer 6— Presentation Layer

- In ricezione, converte i dati ricevuti dal mittente in un formato comprensibile alle applicazioni di livello 7
- Ad esempio un flusso di byte viene trasformato in un documento Open Office od un immagine Jpeg
- In trasmissione, è il livello responsabile della cifratura dei dati
- Spesse volte le funzionalità del livello 6 sono demandate al livello applicativo

# LAN e Stack TCP/IP

## Layer 5— Session Layer

- E' il livello software responsabile della gestione delle sessioni ossia del riconoscimento di appartenenza, dei pacchetti trasmessi o ricevuti, allo stesso utente (es. autenticazione)
- Spesse volte le funzionalità del livello 5 sono demandate al livello applicativo

# LAN e Stack TCP/IP

## Layer 4— Transport Layer

- E' il livello responsabile della corretta trasmissione e ricezione dei pacchetti
- Nel caso di non corretta ricezione dei pacchetti chiede la loro ritrasmissione

# LAN e Stack TCP/IP

## Layer 3— Network Layer

- E' il livello responsabile del corretto instradamento dei pacchetti attraverso le varie reti che collegano il mittente al destinatario
- E' il livello al quale appartengono i router e gli switch che operano usando gli indirizzi IP al posto dei MAC Address

# LAN e Stack TCP/IP

## Layer 2— Data Link Layer

- E' il livello responsabile della trasmissione/ricezione di un pacchetto all'interno di una singola rete
- E' il livello al quale appartengono i bridge e gli switch

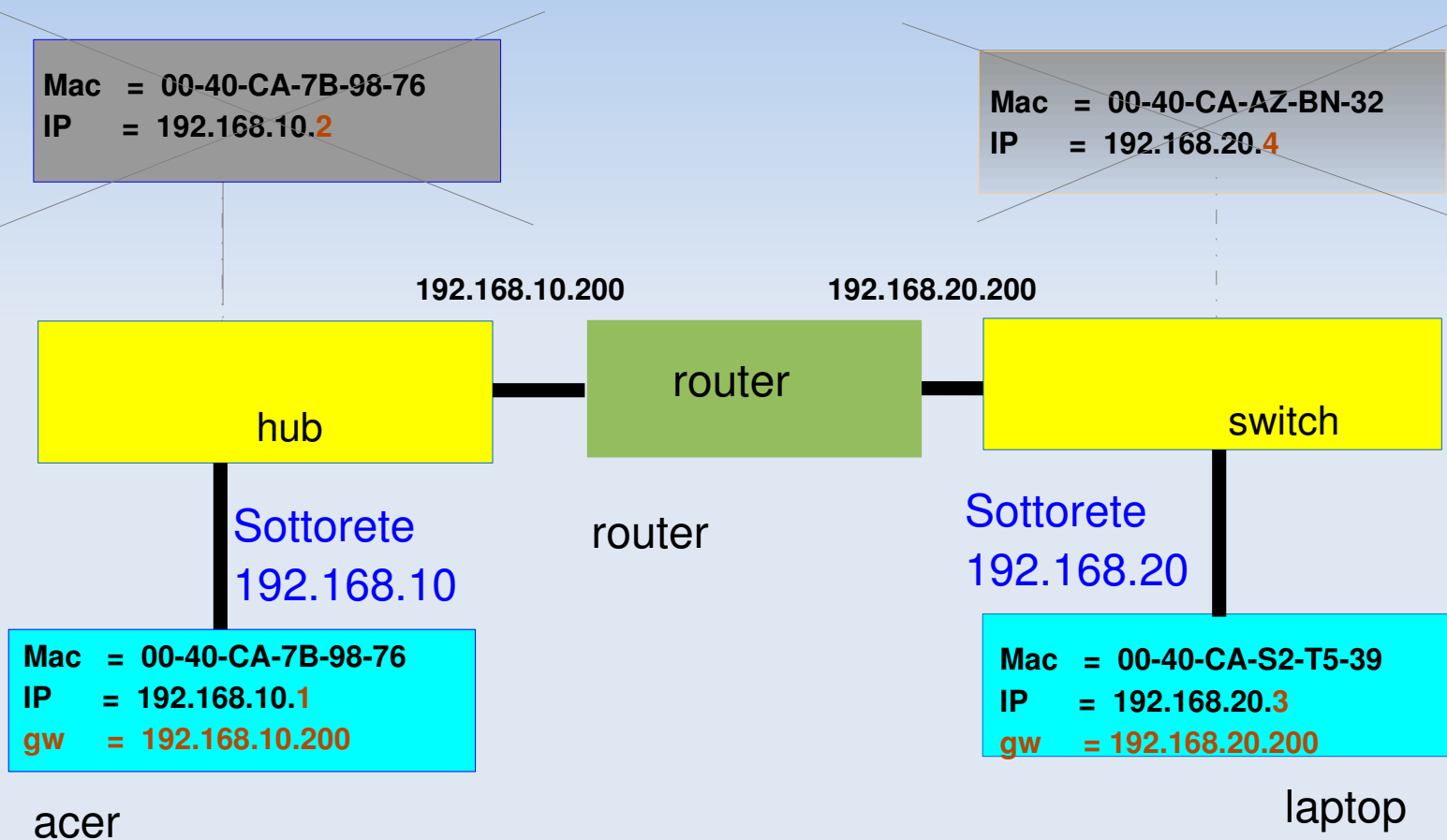
# LAN e Stack TCP/IP

## Layer 1— Physical Layer

- E' il livello responsabile della trasmissione/ricezione fisica di un pacchetto (a livello di segnale elettrico o luminoso)
- E' il livello al quale appartengono gli hub

# LAN e Stack TCP/IP

Laboratorio n. 3: Due sottoreti 192.168.10 ed 192.168.20



# LAN e Stack TCP/IP

## Laboratorio n.3:

- ▶ Abilitare routing su macchina Linux “router”  
echo 1 > /proc/sys/net/ipv4/ip\_forward
- ▶ Verificare schede rete e tabella di routing (route -n)
- ▶ Collegare hub a scheda 192.168.10.200
- ▶ Collegare ad hub computer windows con scheda 192.168.10.1 e verifica ping con 192.168.10.200
- ▶ Collegare switch a scheda 192.168.20.200
- ▶ Collegare a switch PC con scheda 192.168.20.3 e verifica ping con 192.168.20.200
- ▶ Verificare tabella di route
- ▶ Ping da 192.168.20.3 a 192.168.10.1

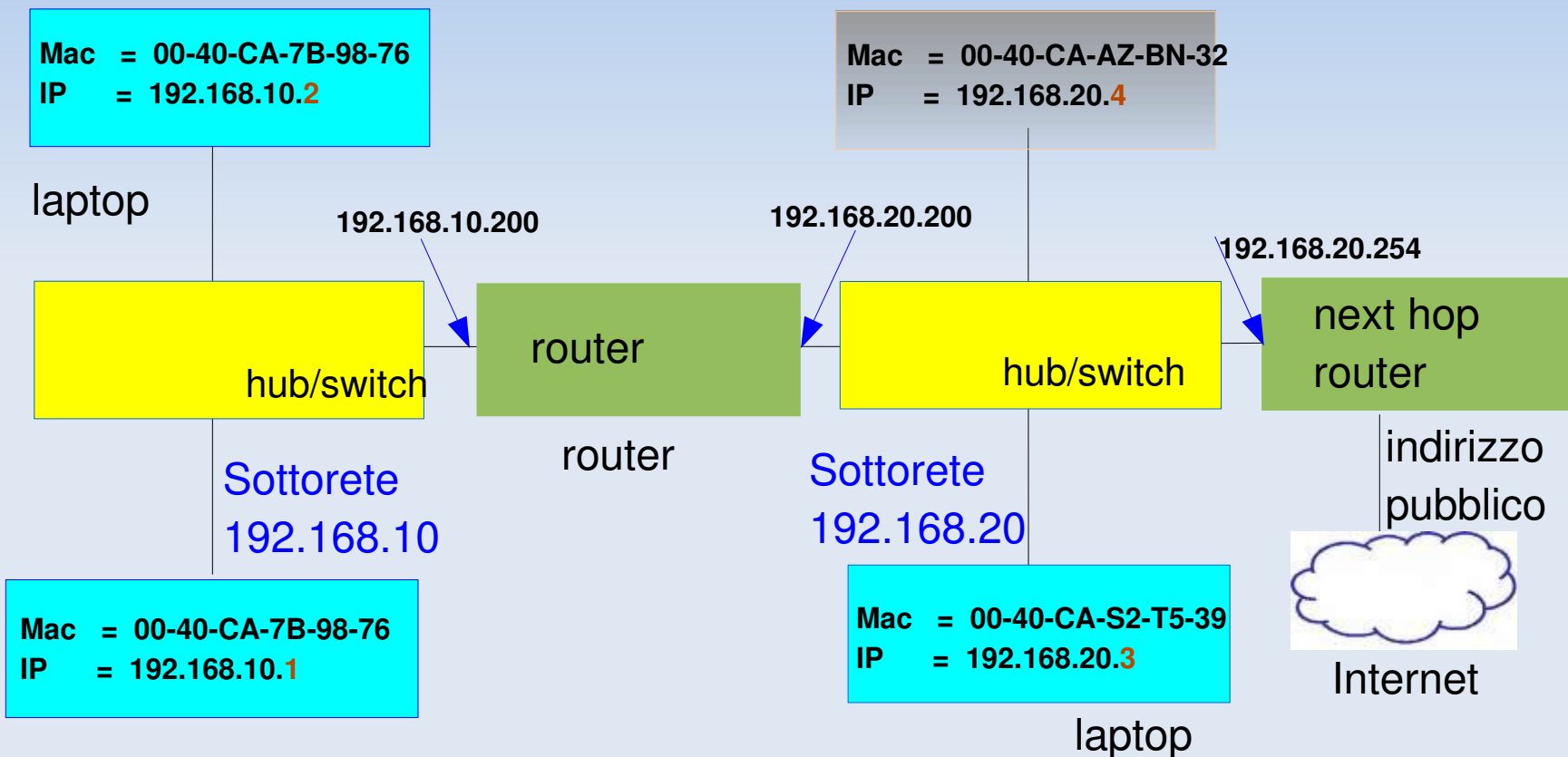
# LAN e Stack TCP/IP

## Laboratorio n.3:

- ♦ Ethereal su router
- ♦ Ping -R da 192.168.20.3 a 192.168.10.1 su laptop
- ♦ Comando traceroute da laptop (tracert in windows)
- ♦ Togliere ip forwarding e riprovare

# LAN e Stack TCP/IP

Laboratorio n. 3: Due sottoreti 192.168.10 ed 192.168.20



Domanda: e se un pacchetto fosse  
destinato ad Internet ?

# LAN e Stack TCP/IP

## Laboratorio 4

- Analisi dello stack TCP/IP mediante Ethereal