GNU/Linux Intermedio

Settima lezione:

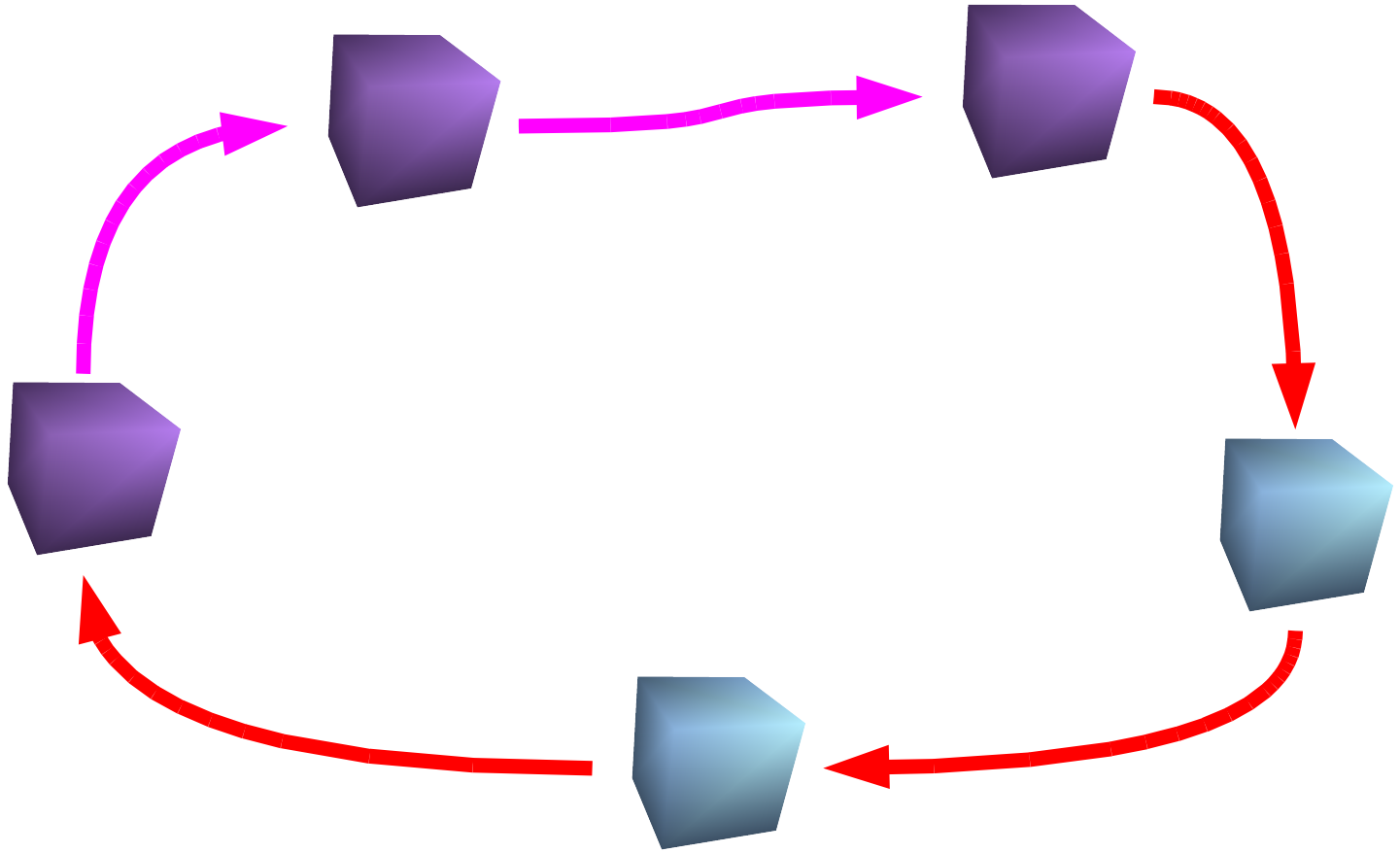
LA RETE

Cos'è la Rete?

- Una rete dati è un insieme di “cavi” (esistono reti wireless ma possiamo immaginare, per il momento, un funzionamento analogo) che mettono in comunicazione piu apparati, tra i quali i computer.
- La Rete, la “rete globale” è Internet.
  - Nasce negli anni sessanta alla ARPA (Advanced Research Projects Agency, creata nel 1958 dal DoD) con il nome di ARPAnet
  - Uno degli obiettivi principali che l'ARPA si poneva per ARPAnet, era quella di costruire una rete di comunicazione che mettesse in comunicazione le moltissime (e costose) macchine del DoD in giro per il mondo in modo da consentire loro uno scambio di informazioni
  - Nel fare questo, si cercava anche qualcosa che potesse sopravvivere ad un attacco nucleare (Guerra Fredda)

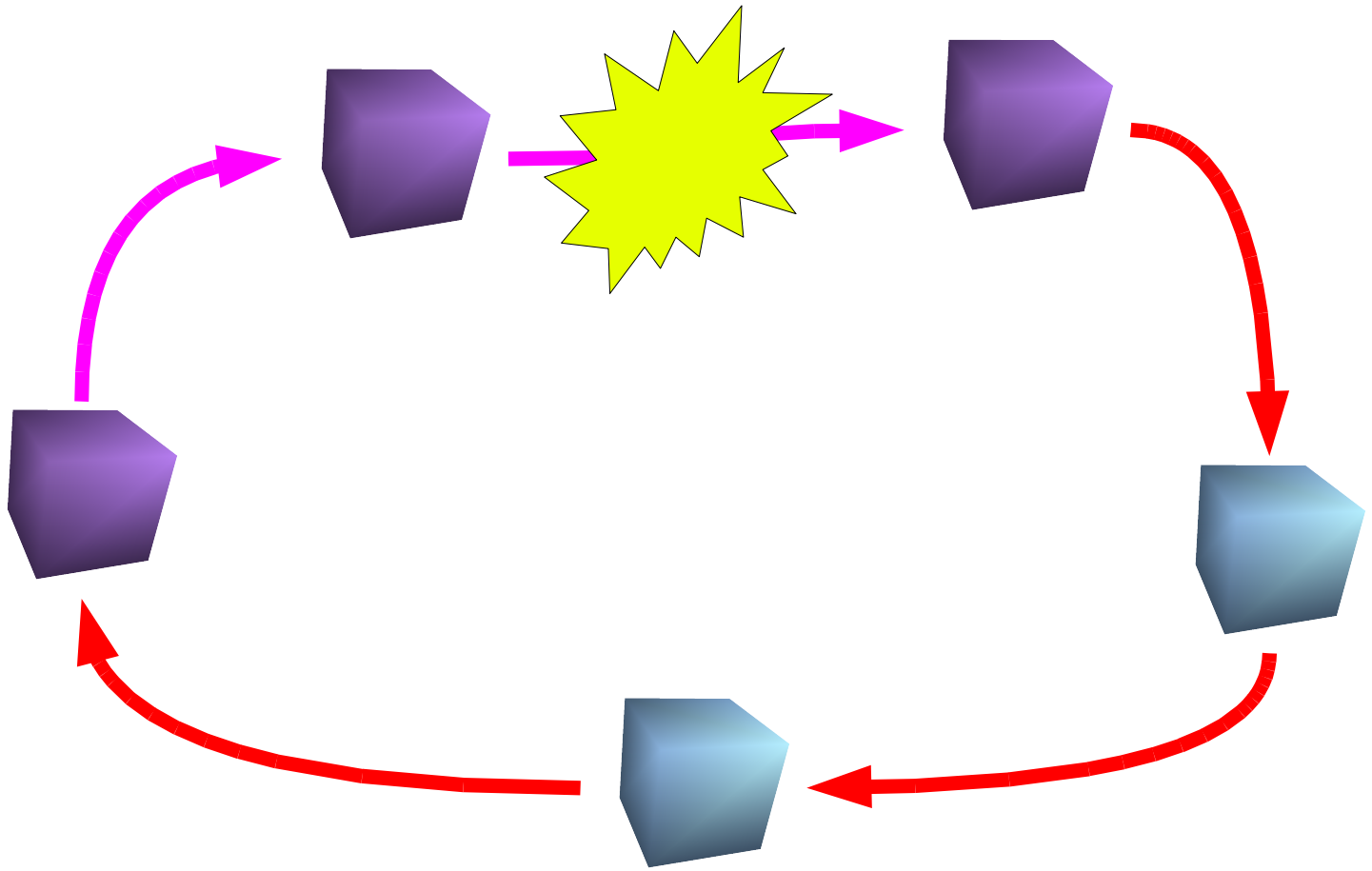
Che tipologia di rete?

- Resistere ad un attacco nucleare non è una cosa semplice
- Che tipo di rete scegliere?
- Una rete ad anello non funziona.



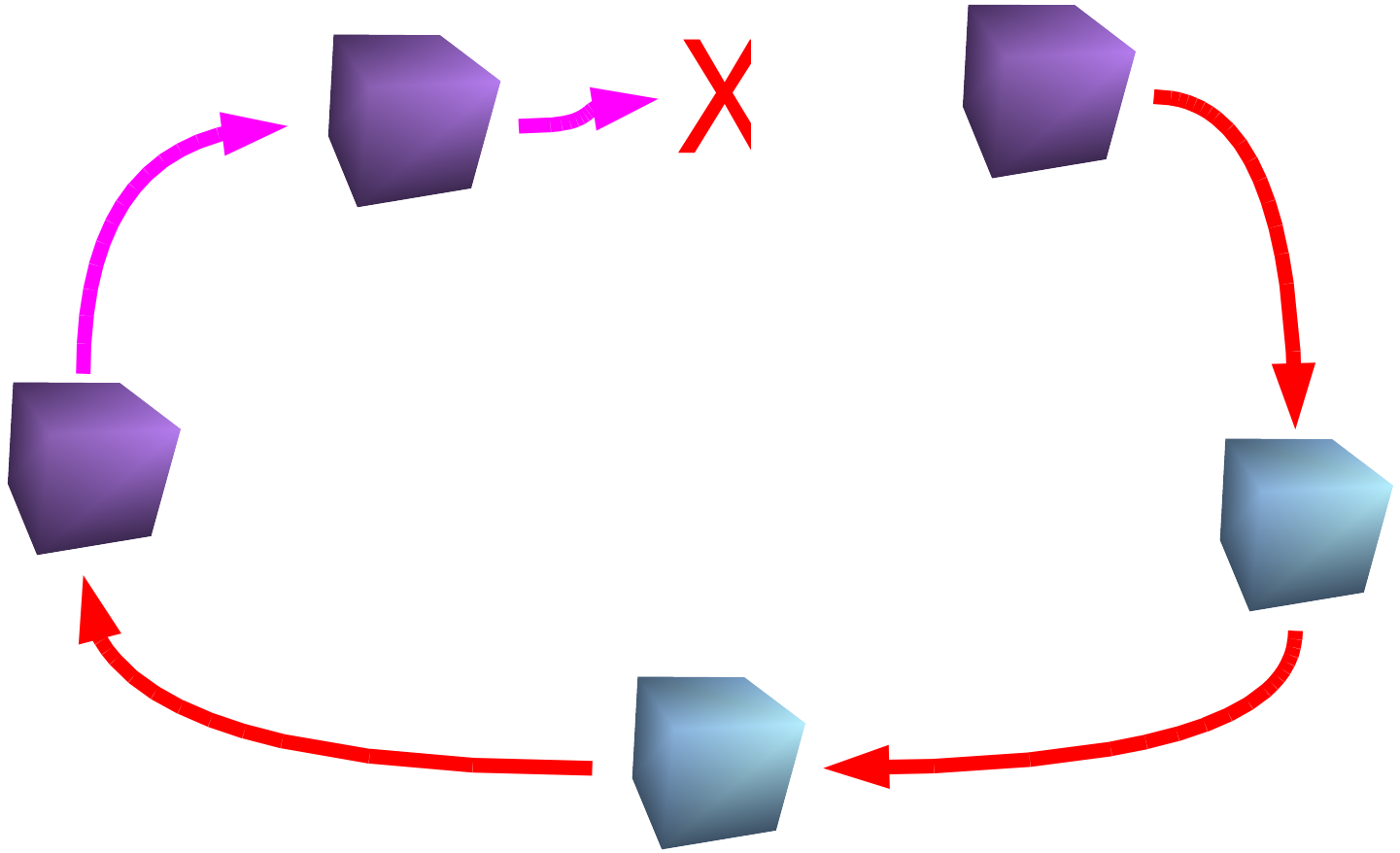
Che tipologia di rete?

- Resistere ad un attacco nucleare non è una cosa semplice
- Che tipo di rete scegliere?
- Una rete ad anello non funziona.



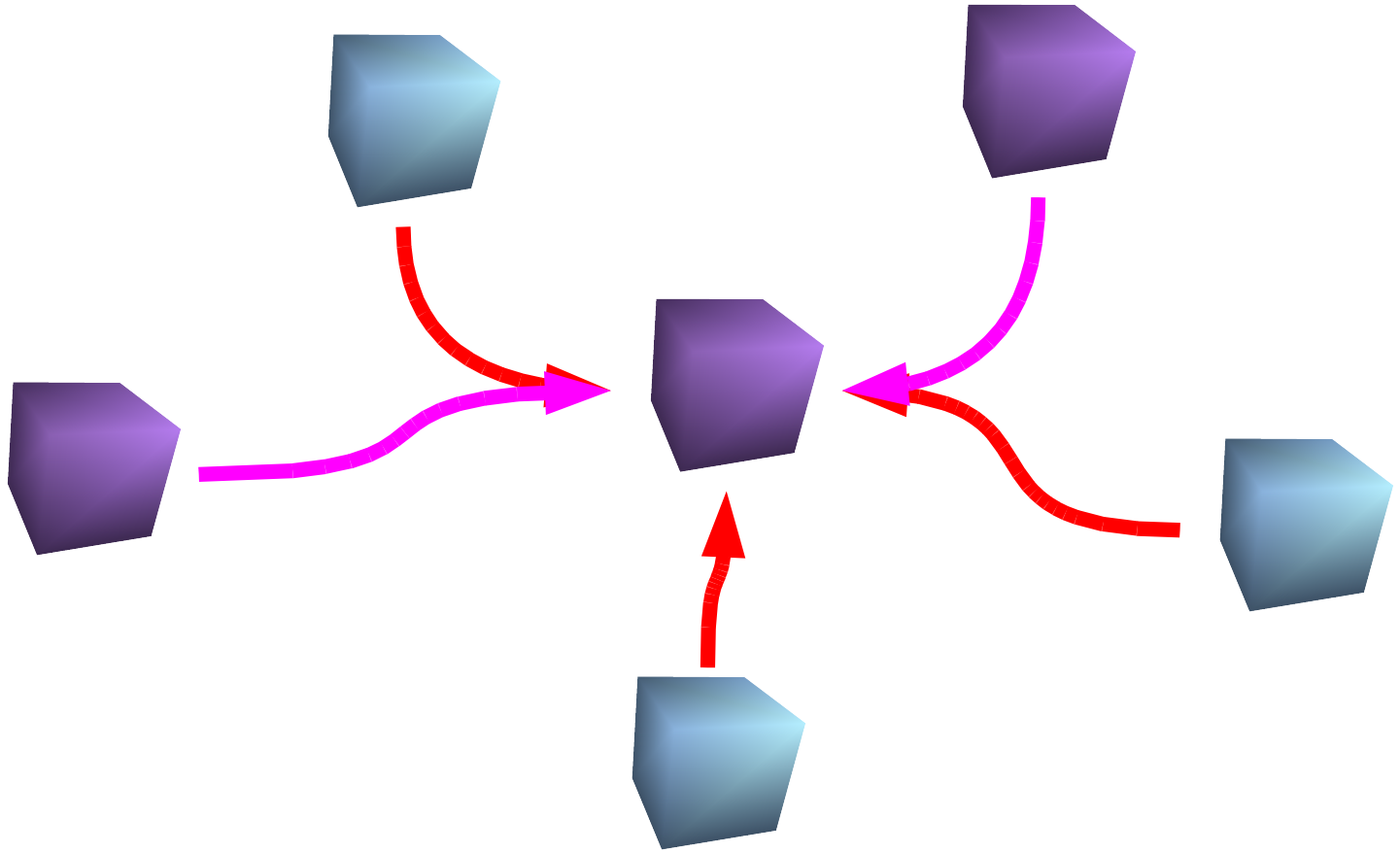
Che tipologia di rete?

- Resistere ad un attacco nucleare non è una cosa semplice
- Che tipo di rete scegliere?
- Una rete ad anello non funziona.



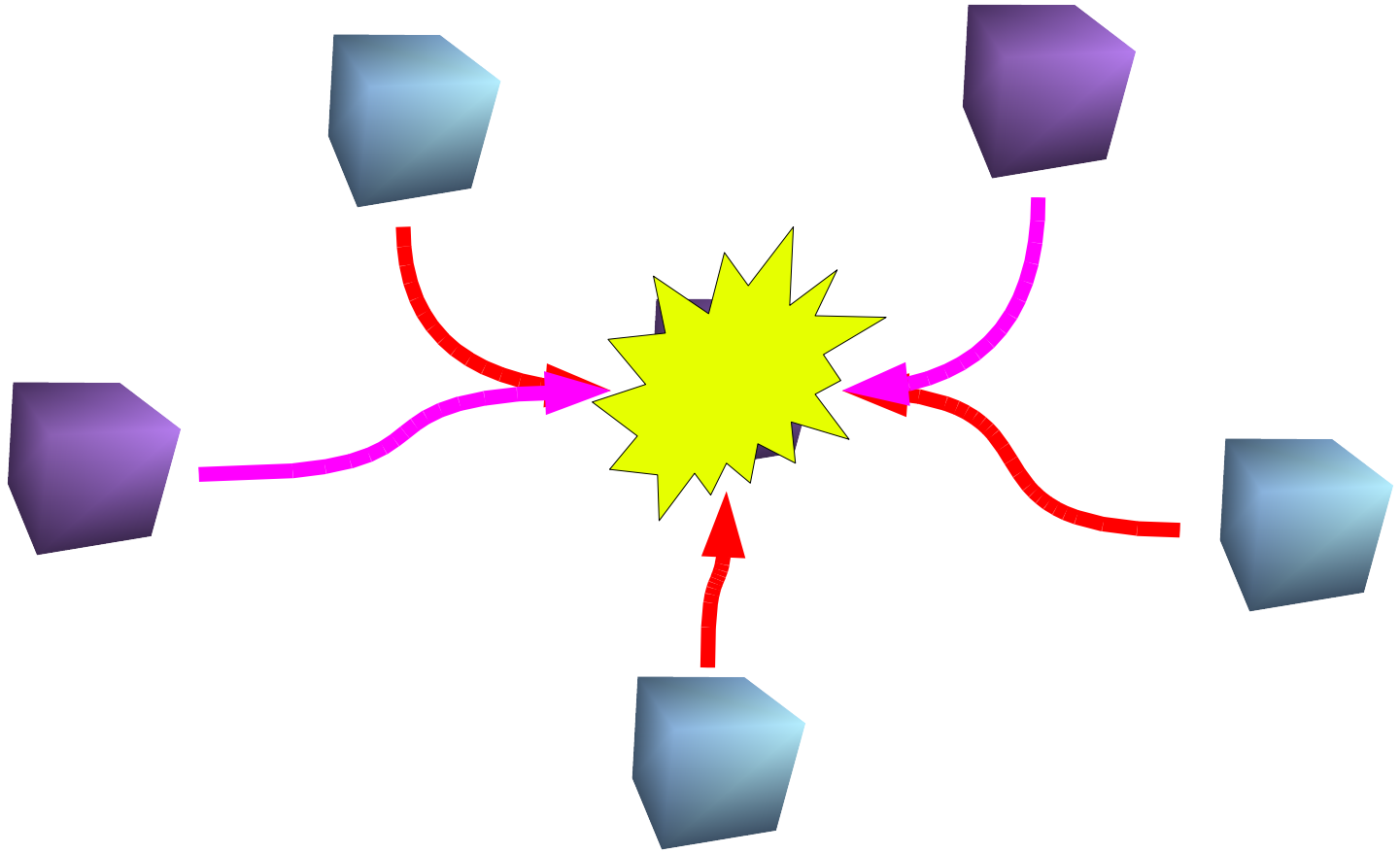
Che tipologia di rete?

- Resistere ad un attacco nucleare non è una cosa semplice
- Che tipo di rete scegliere?
- Una rete a stella ancora meno!



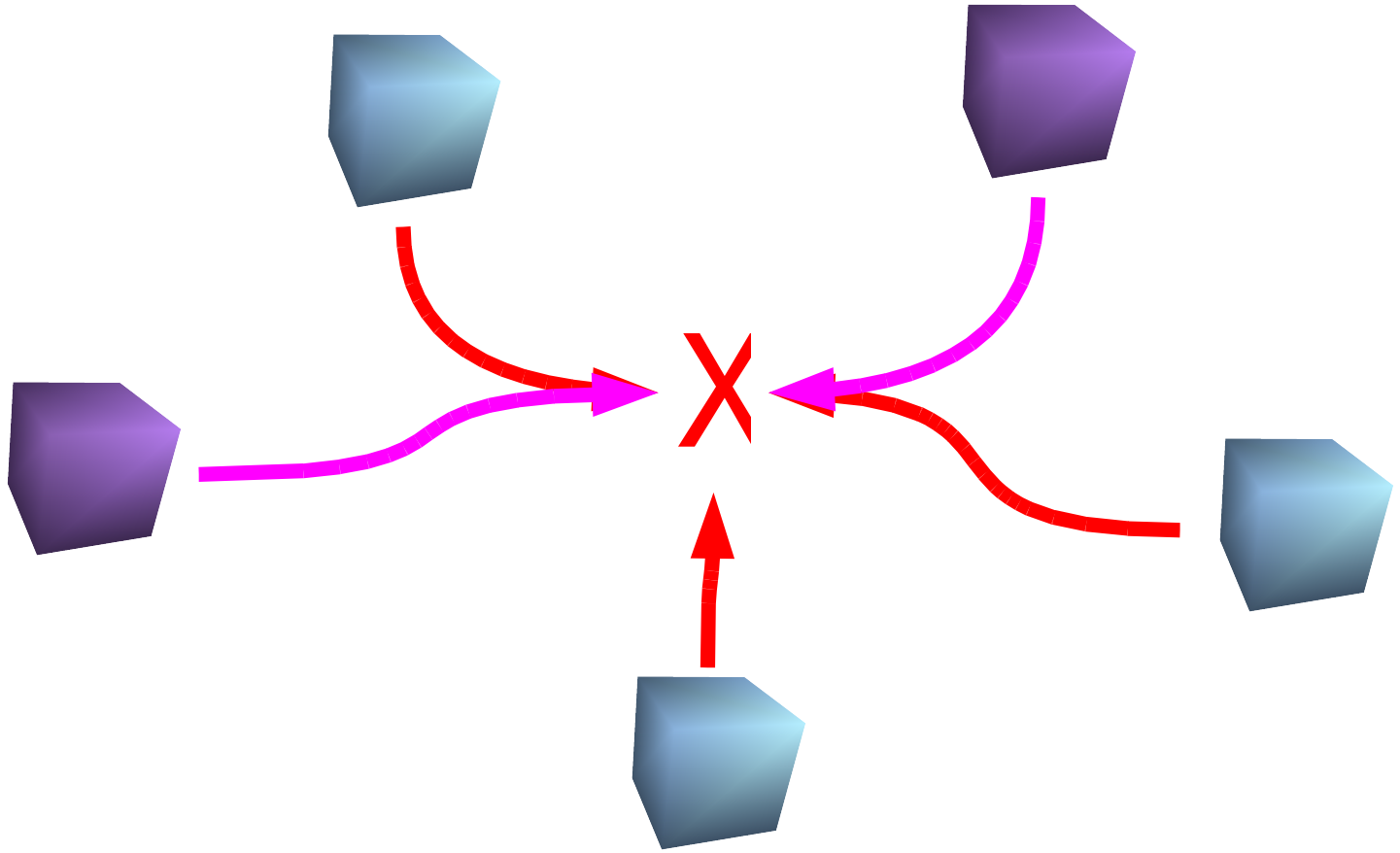
Che tipologia di rete?

- Resistere ad un attacco nucleare non è una cosa semplice
- Che tipo di rete scegliere?
- Una rete a stella ancora meno!



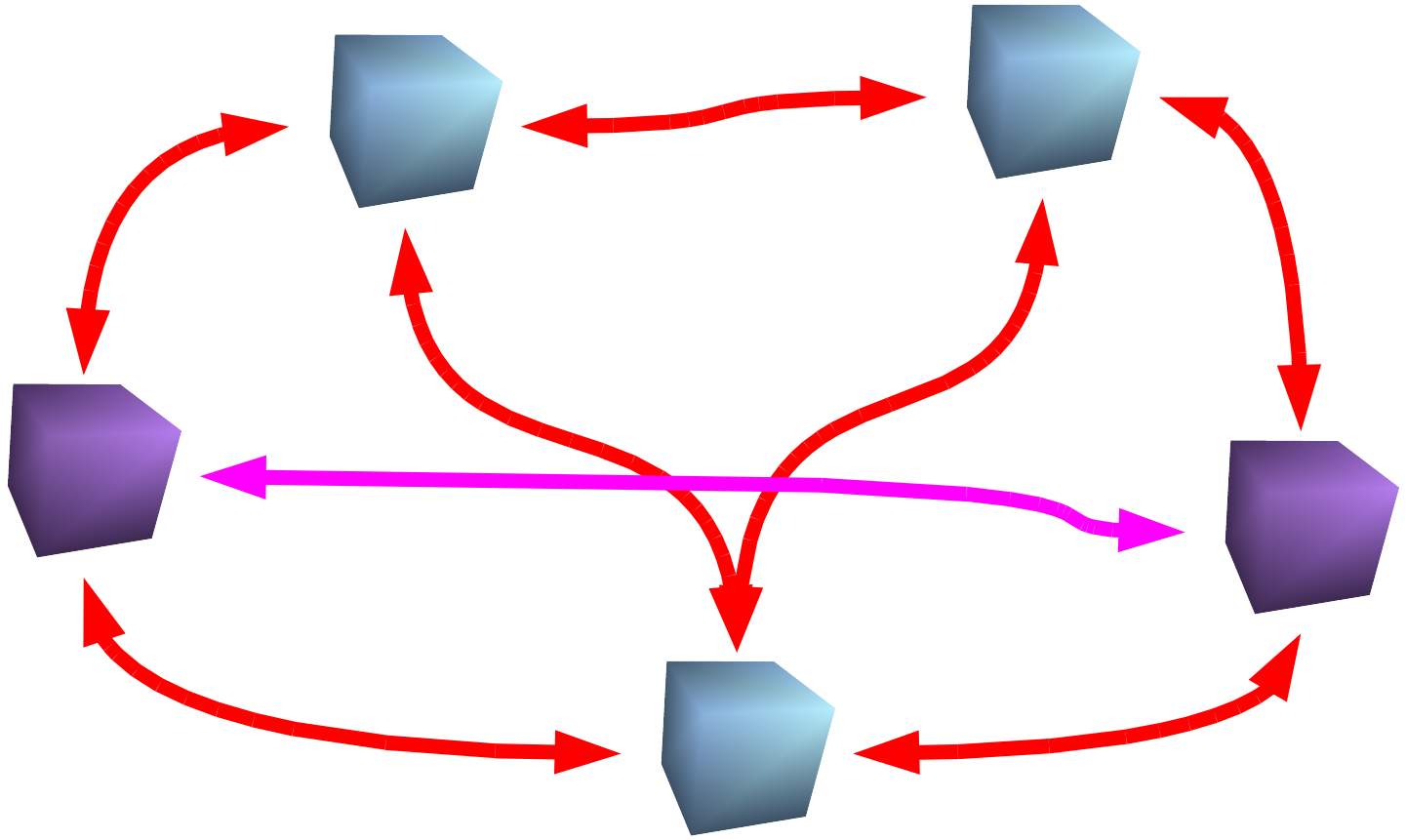
Che tipologia di rete?

- Resistere ad un attacco nucleare non è una cosa semplice
- Che tipo di rete scegliere?
- Una rete a stella ancora meno!



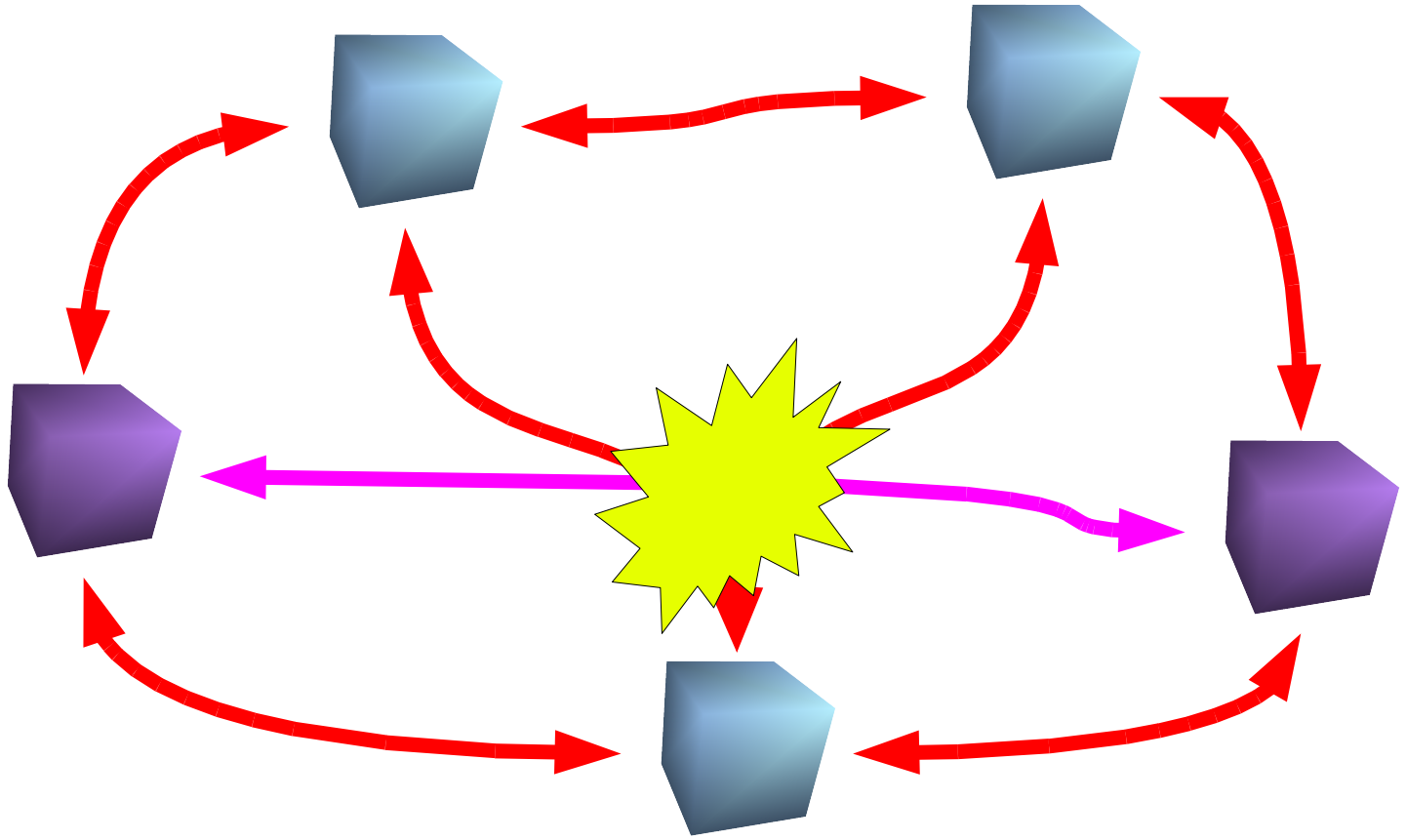
Che tipologia di rete?

- Resistere ad un attacco nucleare non è una cosa semplice
- Che tipo di rete scegliere?
- Si è allora scelta una rete “magliata”



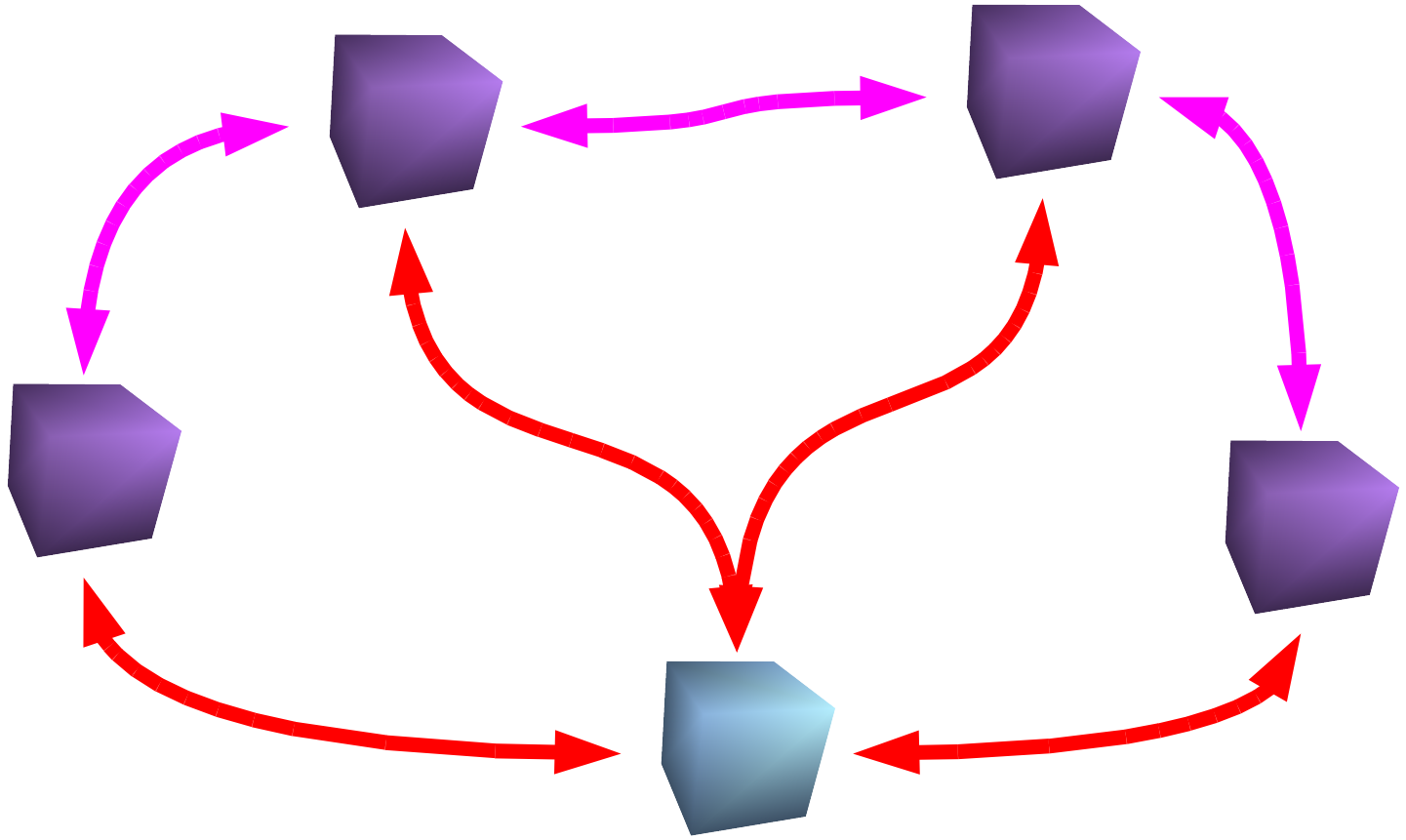
Che tipologia di rete?

- Resistere ad un attacco nucleare non è una cosa semplice
- Che tipo di rete scegliere?
- Si è allora scelta una rete “magliata”



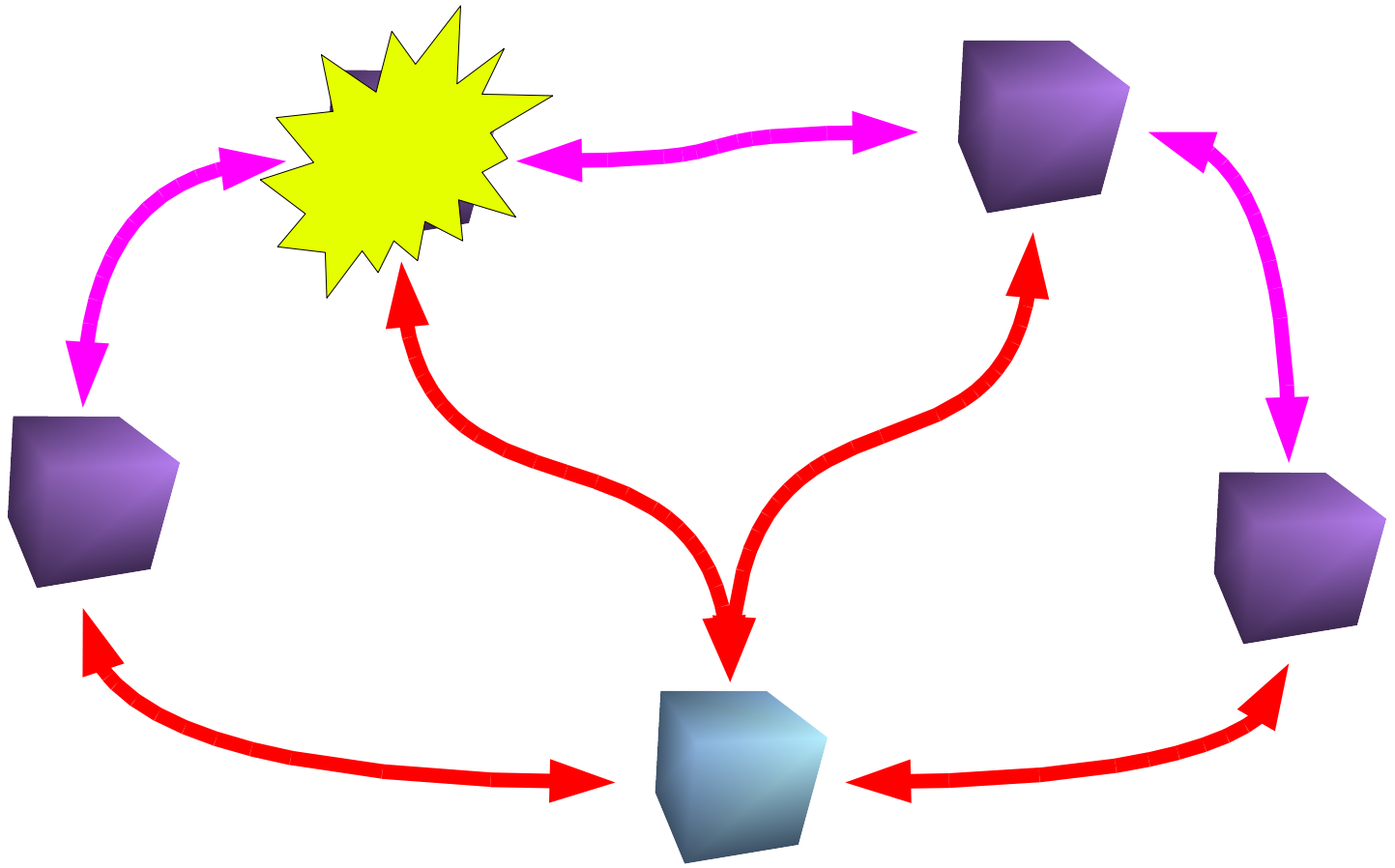
Che tipologia di rete?

- Resistere ad un attacco nucleare non è una cosa semplice
- Che tipo di rete scegliere?
- Si è allora scelta una rete “magliata”



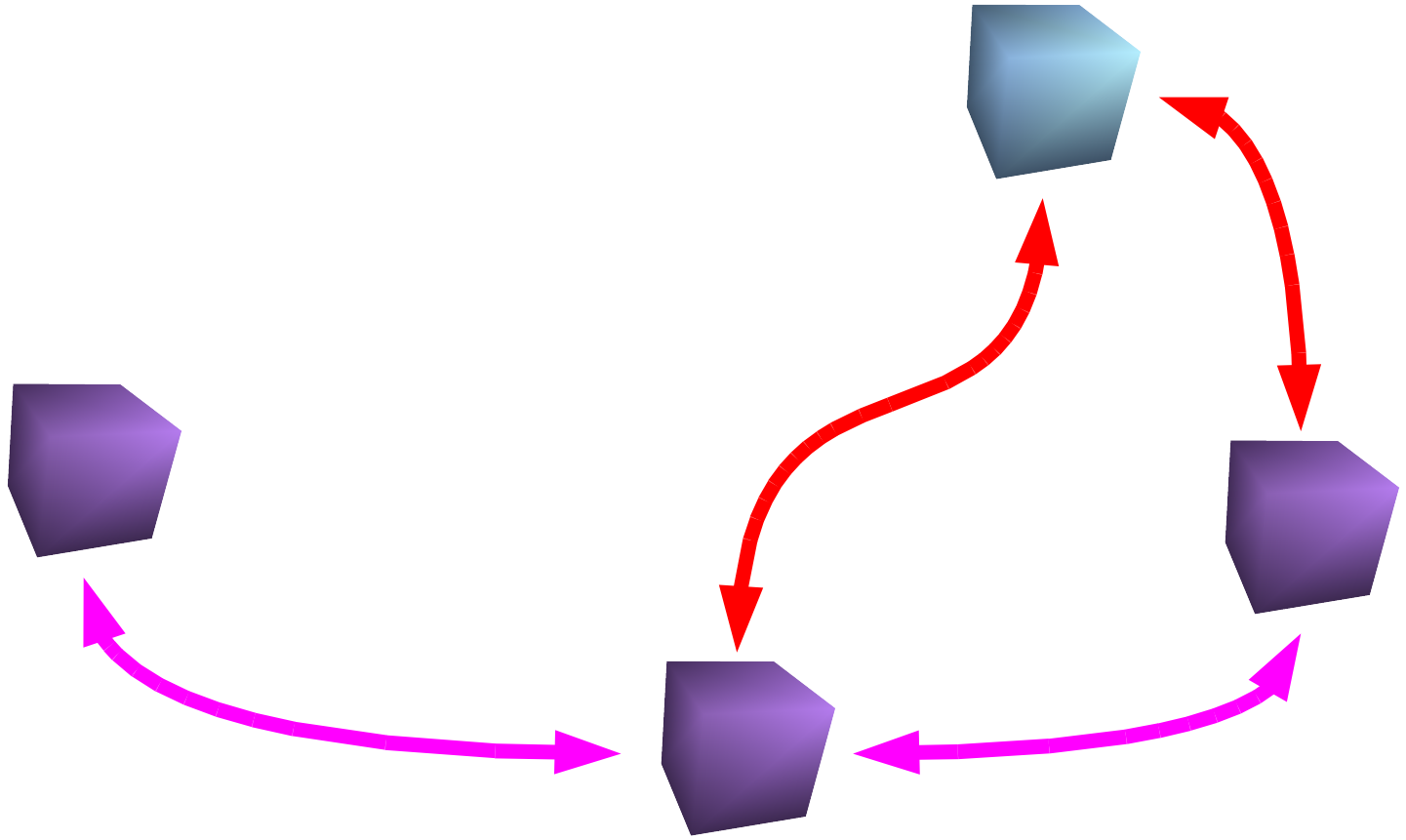
Che tipologia di rete?

- Resistere ad un attacco nucleare non è una cosa semplice
- Che tipo di rete scegliere?
- Si è allora scelta una rete “magliata”



Che tipologia di rete?

- Resistere ad un attacco nucleare non è una cosa semplice
- Che tipo di rete scegliere?
- Si è allora scelta una rete “magliata”



Gli albori: ARPAnet

- I primi due nodi della nuova rete ARPAnet, furono, nel 1969, l'Università della California e lo Stanford Research Institute
- La prima applicazione che abbia mai girato su internet è stata una sessione di telnet
- A dicembre 1969 si aggiunsero le Università di Santa Barbara e dello Utah
- A partire dal 1970, la crescita della rete fu estremamente veloce: in 3 anni diventarono 37
- La “liberazione” dalla DoD avviene nel 1983, con la nascita di MILNET, che relega Internet a “rete di ricerca”
- Il funzionamento di Internet è ancora oggi praticamente immutato dal 1978, da quando TCP fu affiancato da IP

Lo stack: i livelli

- La comunicazione tra i vari host, avviene tramite una serie (stack) di “protocolli di rete”, linguaggi cioè che i vari host sono in grado di parlare
- Tra i protocolli piu importanti troviamo certamente i protocolli IP e TCP, che danno il nome a tutto lo “stack”
- I protocolli vengono organizzati a “livelli”, ognuno dei quali esegue un'astrazione su quello inferiore
- Piu protocolli possono gestire uno stesso livello
  - Il livello piu basso si chiama “livello fisico”, ed è solitamente gestito direttamente dal driver dell'interfaccia di rete, che modula il segnale sul cavo (o tramite un'antenna)
  - Il livello immediatamente superiore (livello 2) è il “livello data-link”, spesso gestito dal protocollo Ethernet

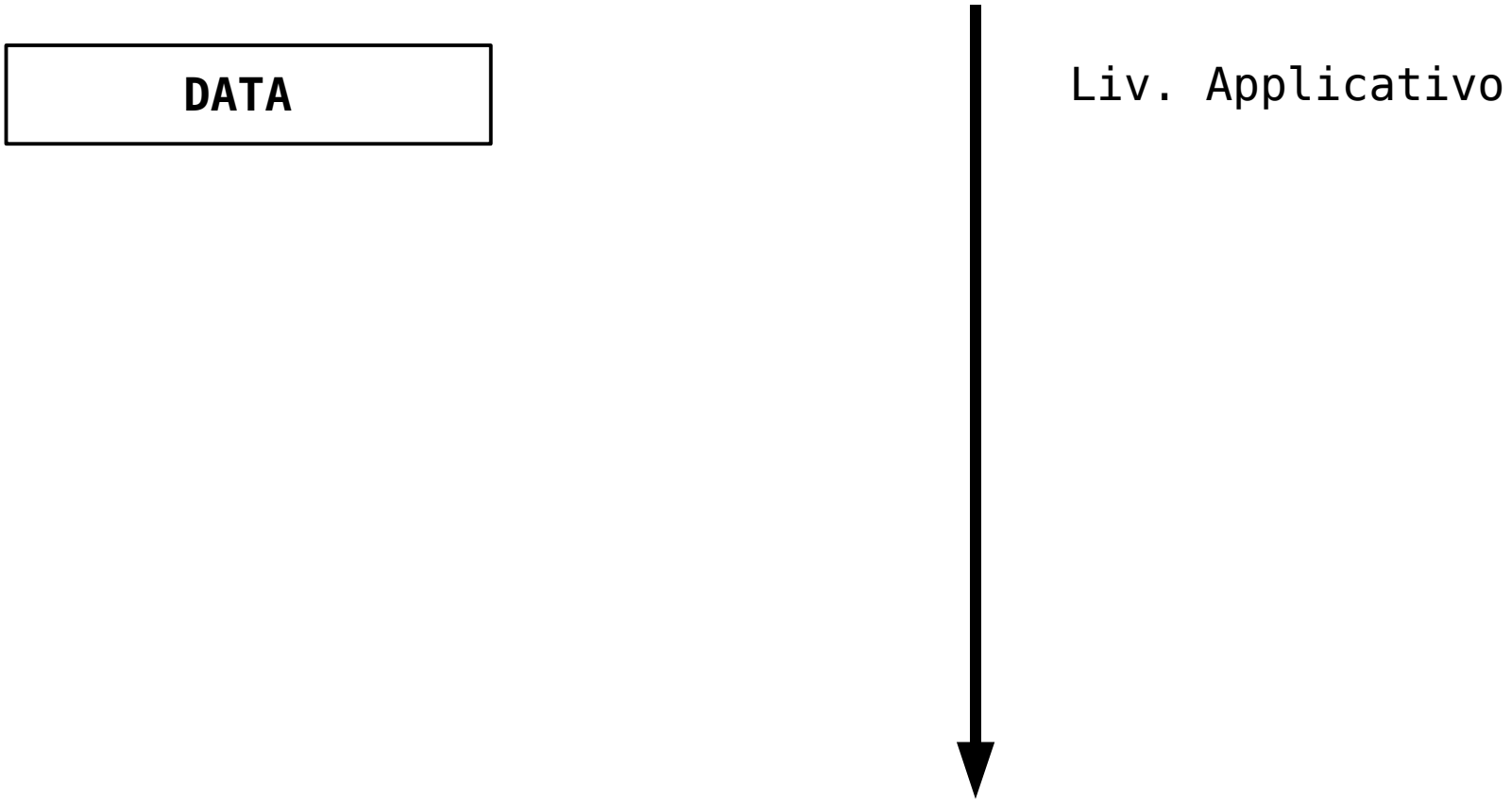
Lo stack: i livelli

- La comunicazione tra i vari host, avviene tramite una serie (stack) di “protocolli di rete”, linguaggi cioè che i vari host sono in grado di parlare
- Tra i protocolli piu importanti troviamo certamente i protocolli IP e TCP, che danno il nome a tutto lo “stack”
- I protocolli vengono organizzati a “livelli”, ognuno dei quali esegue un'astrazione su quello inferiore
- Piu protocolli possono gestire uno stesso livello
  - Il piu in alto, troviamo il “livello di rete”, che viene gestito nella quasi totalità dei casi dal protocollo IP
  - Salendo di un altro gradino troviamo i protocolli del livello “trasporto”, quasi sempre incarnati dai protocolli UDP e TCP
  - Infine, troviamo il “livello applicativo”, le applicazioni

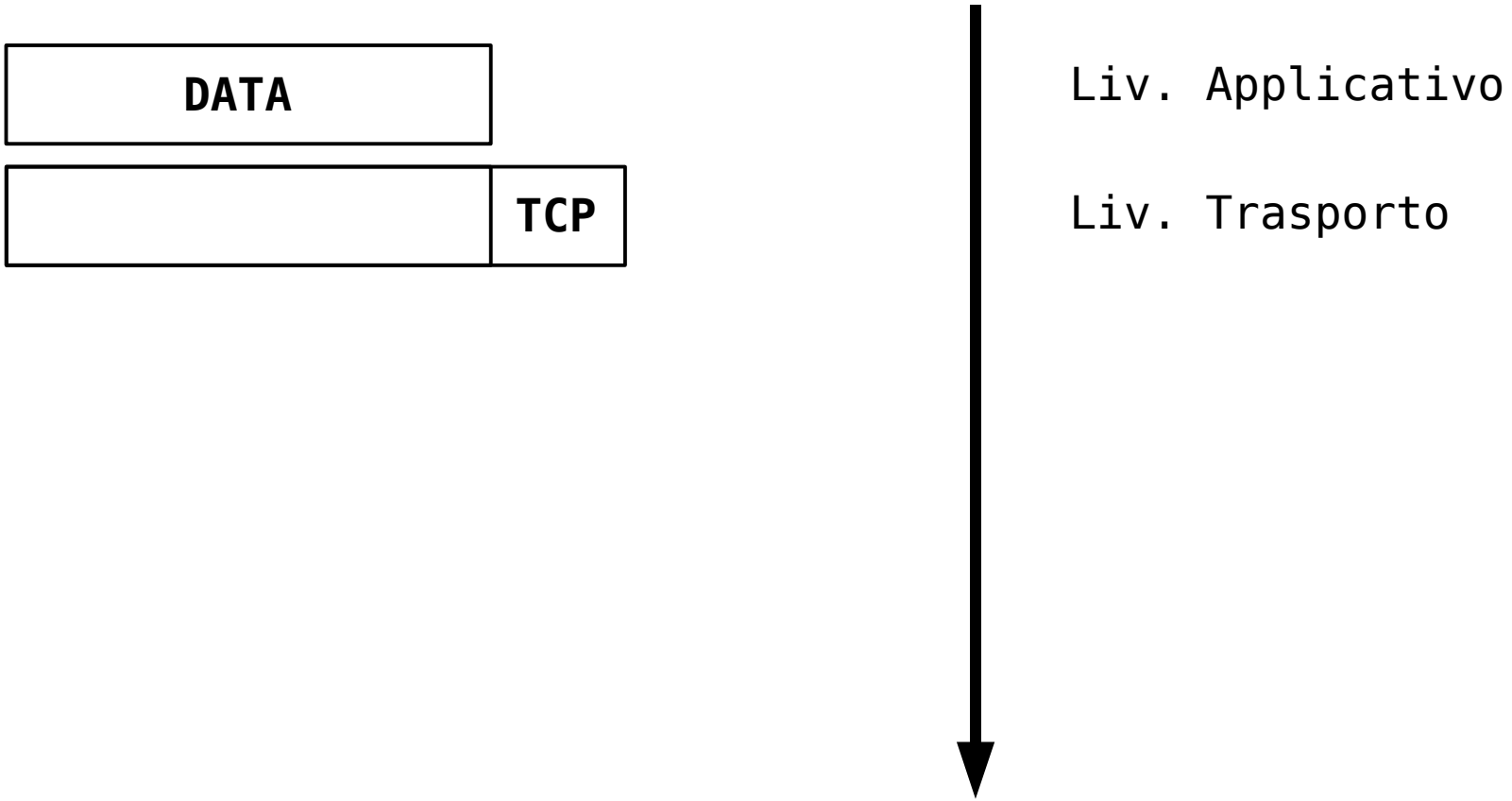
Trasmissione

- Ogni livello inserisce una “intestazione”, detta “header”, che contiene alcune informazioni necessarie a far arrivare il pacchetto al destinatario (indirizzo mittente e destinatario, proprio come per la posta cartacea)
- Una volta che il sistema destinatario avrà ricevuto il frame (tramite la propria interfaccia di rete, gestendolo a livello data-link), procederà un passo alla volta “estraendo” livello dopo livello il pacchetto originale.
- Una volta che questo giungerà al livello “applicativo”, il sistema (server) otterrà una richiesta espressa nel protocollo di livello applicativo che è in grado di interpretare.
- Una volta generata la risposta, il pacchetto verrà inviato indietro nello stesso modo.

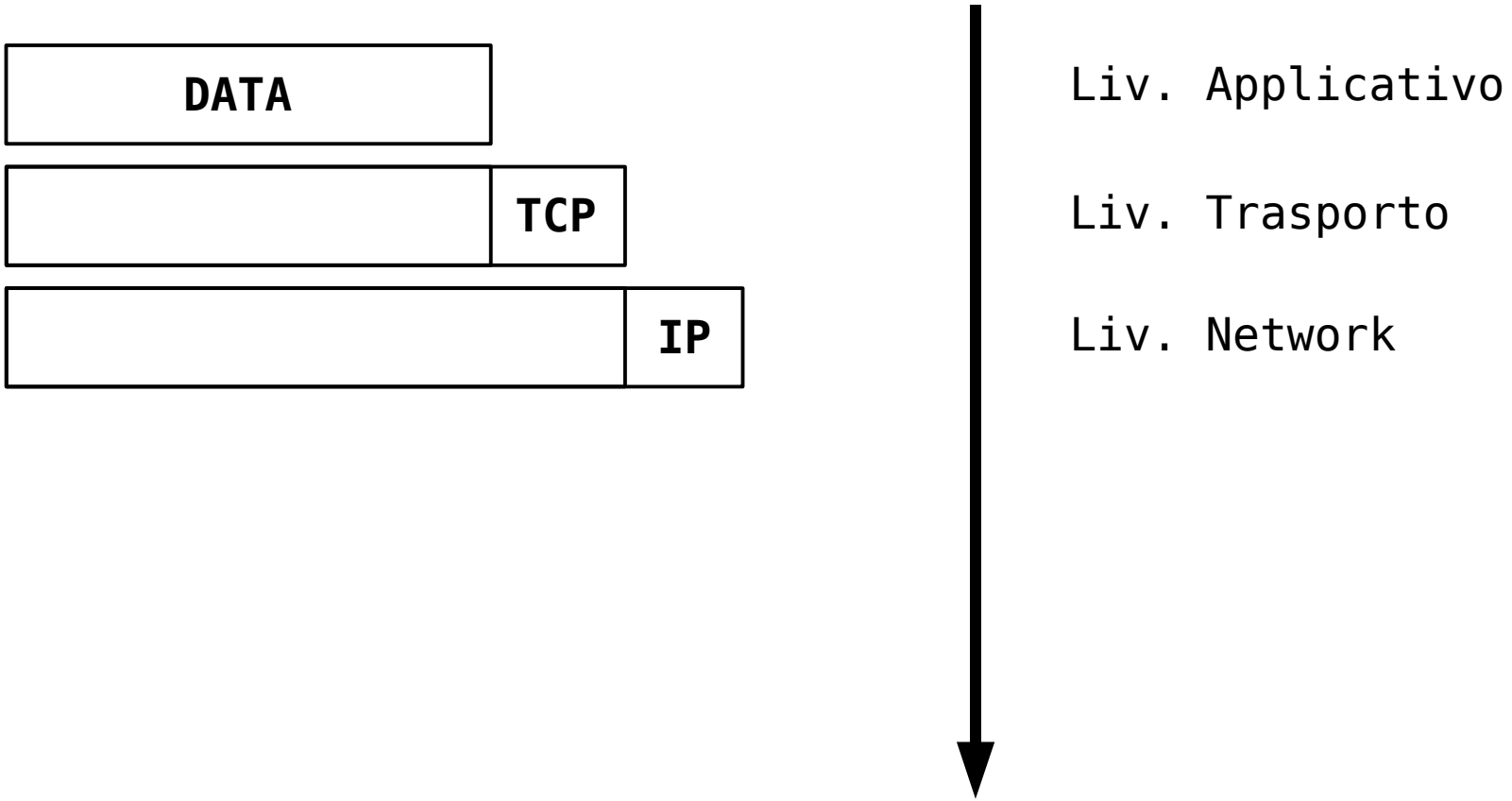
Imbustamento



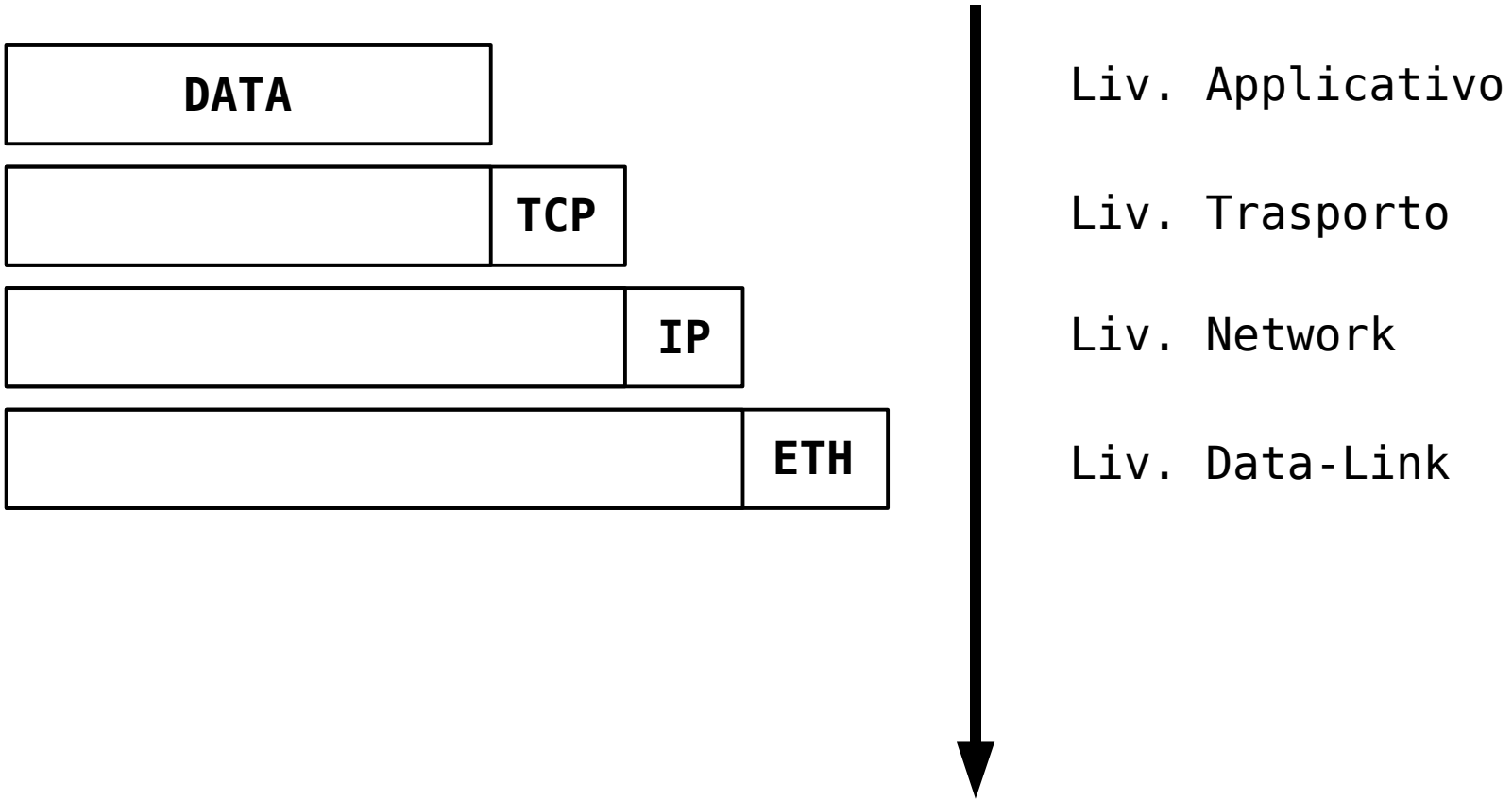
Imbustamento



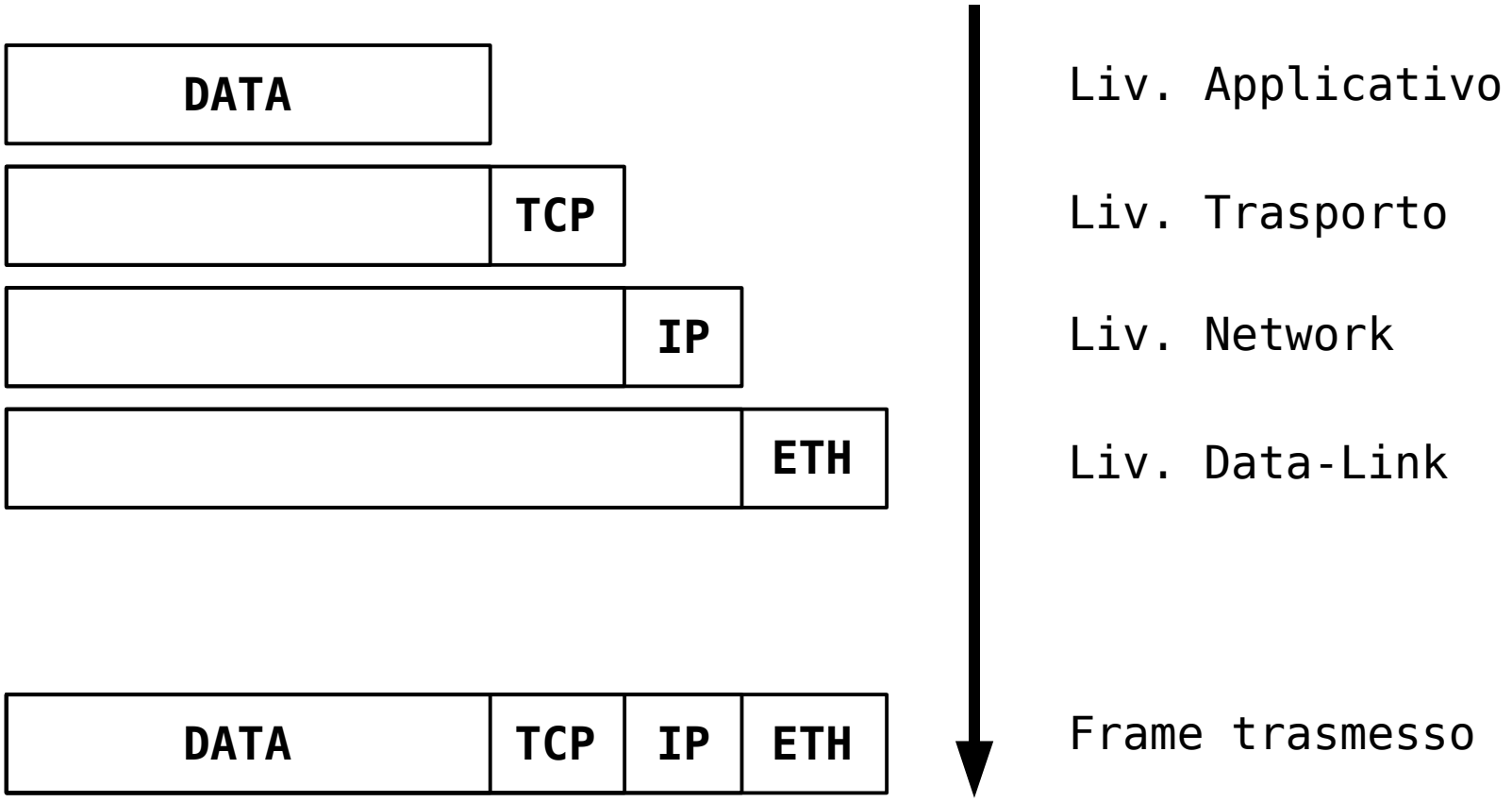
Imbustamento



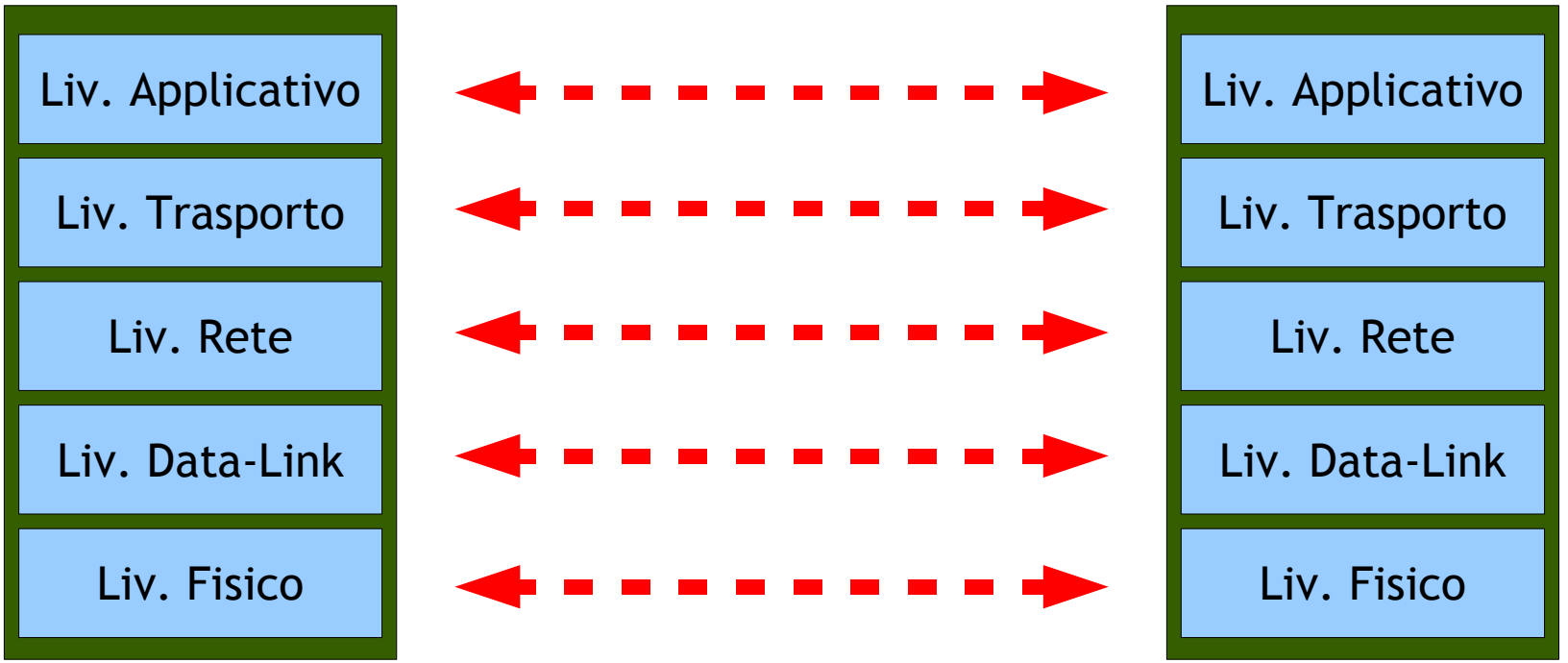
Imbustamento



Imbustamento

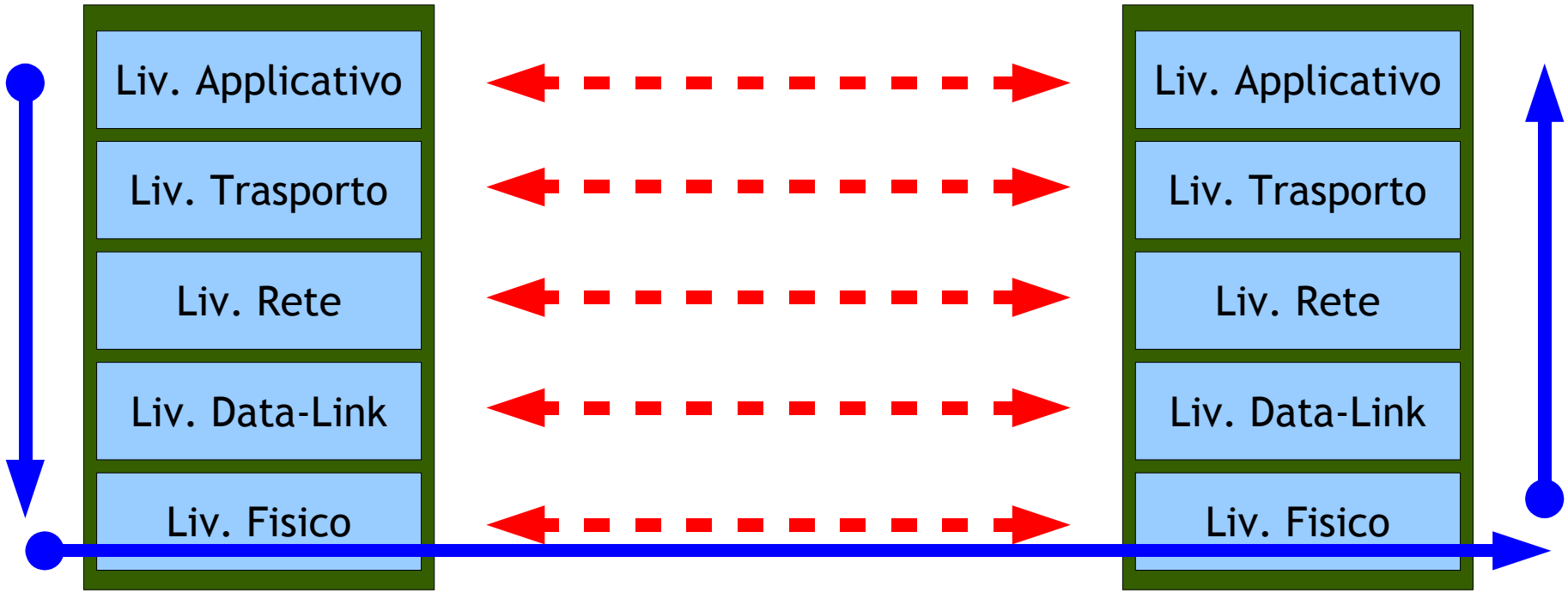


Apparentemente...



In questo modo, ogni livello ha l'impressione di comunicare, utilizzando un protocollo comune, con il proprio “pari” sull'host destinatario...

... realmente



... anche se in realtà la comunicazione avviene tramite un imbustamento di un protocollo dentro l'altro.

Comunicazione standard

- Grazie all'inserimento dell'indirizzo del destinatario nell'header del pacchetto, è possibile che ogni sistema sul percorso prescelto verso il destinatario sia in grado di inoltrare lo stesso sul canale migliore per raggiungerlo.
- Grazie all'inserimento dell'indirizzo del mittente nell'header del pacchetto, è possibile per il destinatario sapere a chi inviare la risposta.
- Questo meccanismo si replica per ogni livello dello “stack”
- Questa strutturazione, divenuta uno standard “de facto” con il nome di “TCP/IP”, venne poi definita rigorosamente dall'introduzione del modello “ISO/OSI”, che introduce due ulteriori livelli tra i livelli “trasporto” e “applicativo”, “sessione” e “presentazione”, che per il momento tralasceremo.

Perchè tanti protocolli?

- Ma perchè tutti questi protocolli?
  - Perchè esistono reti diverse
    - Su reti di tipologia diversa, si possono cambiare alcuni protocolli (i livelli piu bassi) mantenendo intatti gli altri (quelli superiori).
    - Reti diverse infatti, possono avere richieste completamente diverse:
      - PPP
      - Tokern-Ring
      - Ethernet Bus
  - KISS (Keep it simple)
    - Ogni livello ha una sua specifica funzione
    - “Un cammino è fatto di piccoli passi”
    - Piu le cose sono semplici, meglio funzionano

Indirizzamento

- L'indirizzamento piu interessante, è quello che si applica a livello 3, in quanto il primo a fare astrazione dalla tipologia fisica di rete, e quindi quello che viene poi di fatto utilizzato per indirizzare tutti i sistemi di Internet: **IP**
- IP prevede l'assegnazione di un numero di 32 bit ad ogni sistema connesso.
- Per facilitarne la lettura viene raccolto in ottetti e questi tradotti in decimale
  - 213.102.1.21
  - 192.168.1.10
  - 10.0.0.45

Netmask

- Internet è però (l'abbiamo già accennato) un insieme di reti diverse.
- Affinchè sia possibile consegnare a destinazione un pacchetto, è necessario sapere a quale rete appartiene.
- Per fare questo, è necessario affiancare a IP un altro numero, che consenta di identificare, a partire dal primo, la rete di appartenenza di quel determinato IP: **la netmask**
- La netmask non è altro che un numero di 32 bit, composto da una fila di 1 e terminato da 0 (in realtà è cosi solo nella pratica)
  - 255.255.255.0 = 11111111.11111111.11111111.0000000
  - 255.255.0.0 = 11111111.11111111.00000000.0000000
  - 255.255.255.224 = 11111111.11111111.11111111.1110000

Netmask

- La parte composta da 1 della netmask individua la parte di indirizzo utilizzata per individuare la “rete” (parte rete), mentre la parte composta da 0 è lo spazio di indirizzamento dedicato agli host (parte host)
- Associando netmask e IP, possiamo catalogare gli indirizzi
- Gli IP che hanno la parte “host” composta da 0, vengono detti “indirizzi di rete”. Vengono utilizzati nelle tabelle di routing per identificare una rete.
  (**192.168.12.0 / 255.255.255.0**)
- Gli IP che hanno la parte “host” composta da 1, vengono detti “indirizzi di broadcast”. Vengono utilizzati per indirizzare “tutti gli host” della rete (un pacchetto di broadcast viene consegnato a tutti gli host della rete).
  (**192.168.12.255 / 255.255.255.0**)

Indirizzi riservati

- Oltre a broadcast ed indirizzi di rete, ci sono altri tipi di indirizzi che non possono essere utilizzati per indirizzare sistemi “pubblici” su internet. I piu interessanti sono:
  - `127.0.0.0 / 255.0.0.0 ( 1 rete da 16.777.216 host )`
    - Viene utilizzato da ogni sistema per identificare se stesso
  - `10.0.0.0 / 255.0.0.0 ( 1 rete da 16.777.216 host )`
  - `172.16.0.0 / 255.255.0.0 ( 1 rete da 1.048.576 host )`
  - `192.168.0.0 / 255.255.0.0 ( 255 reti da 65.536 host )`
    - Vengono utilizzati per “indirizzamento privato o locale”
- A questi si aggiungono altre classi riservate per vari motivi, che vengono tenute aggiornate all'indirizzo

`http://www.iana.org/assignments/ipv4-address-space`

La tabella di routing

- Internet è, abbiamo detto, una rete di reti.
- L'interconnessione tra le diverse reti, viene fatta da sistemi appositamente configurati detti “router”
- Il meccanismo del routing è piuttosto semplice.
- Viene impostata una tabella (tabella di routing) che viene ordinata per lunghezza della netmask delle sue righe. (piu una netmask è lunga, piu è precisa la definizione che da)

Destination	Gateway	Genmask	Interface
213.12.19.0	*	255.255.255.0	eth0
10.0.0.0	*	255.0.0.0	eth1
default	213.12.19.254	0.0.0.0	eth0

Il routing

- Quando un nuovo pacchetto viene ricevuto, vengono eseguite le seguenti operazioni:
  - Verifica che non sia destinato al sistema
    - Nel qual caso, viene semplicemente passato alle applicazioni preposte a gestirlo
  - Vengono verificate una per una le entry della routing table.
    - Supponiamo che il pacchetto ricevuto sia indirizzato a 10.0.1.200 e di avere la seguente routing table:

```
Destination         Gateway         Genmask         Interface
  213.12.19.0       *               255.255.255.0   eth0
  10.0.0.0          *               255.0.0.0       eth1
  default           213.12.19.254   0.0.0.0         eth0
```

Il routing

- Quando un nuovo pacchetto viene ricevuto, vengono eseguite le seguenti operazioni:
  - Verifica che non sia destinato al sistema
    - Nel qual caso, viene semplicemente passato alle applicazioni preposte a gestirlo
  - Vengono verificate una per una le entry della routing table.
    - Supponiamo che il pacchetto ricevuto sia indirizzato a 10.0.1.200 e di avere la seguente routing table:
      - Applicando la netmask 255.255.255.0 a 10.0.1.200 non si ottiene 213.12.19.0. Andiamo avanti.

Destination	Gateway	Genmask	Interface
10.0.19.0	*	255.255.255.0	eth0
10.0.0.0	*	255.255.0.0	eth1
default	10.0.0.1	0.0.0.0	eth0

Il routing

- Quando un nuovo pacchetto viene ricevuto, vengono eseguite le seguenti operazioni:
  - Verifica che non sia destinato al sistema
    - Nel qual caso, viene semplicemente passato alle applicazioni preposte a gestirlo
  - Vengono verificate una per una le entry della routing table.
    - Supponiamo che il pacchetto ricevuto sia indirizzato a 10.0.1.200 e di avere la seguente routing table:
      - Applicando la netmask 255.0.0.0 a 10.0.1.200 si ottiene 10.0.0.0. Inoltriamo il pacchetto direttamente all'host, sull'interfaccia eth1

Destination	Gateway	Genmask	Interface
213.12.19.0	*	255.255.255.0	eth0
10.0.0.0	*	255.0.0.0	eth1
default	213.12.19.254	0.0.0.0	eth0

Il routing

- Quando un nuovo pacchetto viene ricevuto, vengono eseguite le seguenti operazioni:
  - Verifica che non sia destinato al sistema
    - Nel qual caso, viene semplicemente passato alle applicazioni preposte a gestirlo
  - Vengono verificate una per una le entry della routing table.
    - Se il pacchetto fosse destinato ad una rete non prevista nella routing table? Ad esempio 20.30.40.50?

```
Destination          Gateway          Genmask          Interface
  213.12.19.0        *                255.255.255.0    eth0
  10.0.0.0           *                255.0.0.0        eth1
  default            213.12.19.254    0.0.0.0          eth0
```

Il routing

- Quando un nuovo pacchetto viene ricevuto, vengono eseguite le seguenti operazioni:
  - Verifica che non sia destinato al sistema
    - Nel qual caso, viene semplicemente passato alle applicazioni preposte a gestirlo
  - Vengono verificate una per una le entry della routing table.
    - Se il pacchetto fosse destinato ad una rete non prevista nella routing table? Ad esempio 20.30.40.50?
      - Non matcha con la prima riga.

Destination	Gateway	Genmask	Interface
213.12.19.0	*	255.255.255.0	eth0
10.0.0.0	*	255.0.0.0	eth1
default	213.12.19.254	0.0.0.0	eth0

Il routing

- Quando un nuovo pacchetto viene ricevuto, vengono eseguite le seguenti operazioni:
  - Verifica che non sia destinato al sistema
    - Nel qual caso, viene semplicemente passato alle applicazioni preposte a gestirlo
  - Vengono verificate una per una le entry della routing table.
    - Se il pacchetto fosse destinato ad una rete non prevista nella routing table? Ad esempio 20.30.40.50?
      - Non matcha con la seconda riga.

```
Destination          Gateway          Genmask          Interface
  213.12.19.0        *                255.255.255.0    eth0
  10.0.0.0           *                255.0.0.0        eth1
  default            213.12.19.254    0.0.0.0          eth0
```

Il routing

- Quando un nuovo pacchetto viene ricevuto, vengono eseguite le seguenti operazioni:
  - Verifica che non sia destinato al sistema
    - Nel qual caso, viene semplicemente passato alle applicazioni preposte a gestirlo
  - Vengono verificate una per una le entry della routing table.
    - Se il pacchetto fosse destinato ad una rete non prevista nella routing table? Ad esempio 20.30.40.50?
      - Matcha per forza con la terza riga (netmask 0.0.0.0, la piu breve ma comprende TUTTI gli host), e viene inoltrata al default gateway

```
Destination          Gateway          Genmask          Interface
  213.12.19.0        *                255.255.255.0    eth0
  10.0.0.0           *                255.0.0.0        eth1
  default            213.12.19.254    0.0.0.0          eth0
```

I “core routers”

- Tramite questo meccanismo, replicato su ogni router connesso ad internet, è possibile che ognuno di essi funga un po da “ufficio postale”
  - Conosce gli host connessi sulle sue interfaccie e il modo per recapitare loro i pacchetti (sia la tecnologia della rete che il tipo di indirizzamento)
  - Per i pacchetti che non sono destinati a quegli host, ha un default gateway che ne saprà qualcosa di piu (per i “core router”, il default gateway è proprio quello che gestisce quella rete)
- I “core router” di Internet si scambiano costantemente informazioni relative alle reti connesse tramite appositi protocolli di sincronizzazione

Il livello applicativo

- E' il livello che si interfaccia direttamente (tramite i programmi applicativi) con l'utente.
- E' il livello che sfrutta tutte le potenzialità offerte dai livelli inferiori
- I protocolli piu utilizzati, oggi, sono probabilmente i seguenti:
  - POP3 – ricezione della posta elettronica
  - SMTP – invio della posta elettronica
  - HTTP – navigazione delle pagine web
- Questi protocolli hanno delle varianti protette da crittografia SSL, POP3S, SMTPS, HTTPS, purtroppo ancora non sufficientemente diffuse (si comincia a vedere qualcosa con HTTPS per via dell'ecommerce e ebancking)

POP3

- **Nome:** Post Office Protocol (versione 3)
- **Porta:** 110
- Si tratta del protocollo piu usato per consentire agli utenti di scaricare la posta da un server dedicato alla gestione della stessa (solitamente in associazione ad un server SMTP per la ricezione).
- Dopo aver stabilito la connessione, vi è una fase di autenticazione (username + password) che identificano una singola casella di posta.
- Tramite il comando LIST il client visualizza l'elenco delle email disponibili, tramite il comando RETR [N], scarica il messaggio N.
- Cancellazione dei messaggi tramite DELE [N]

POP3

S:+OK <22593.1129980067@example.com>
C:USER pippo
S:+OK
C:PASS pluto
S:+OK
C:LIST
S:+Ok
  1 817
  2 124
  .
C:RETR 1
S:+OK
  Return-Path: <pippo@example.org>
  Delivered-To: pippo@example.org
  Date: Sat, 22 Oct 2005 13:24:54 +0200
  From: Mario Rossi <mario@rossi.org>
  Subject: xxxx
  Content-Type: text/plain; charset=ISO-8859-1
  testo messaggio
C:DELE 1
S:+OK
C:QUIT
S:+OK

SMTP

- **Nome:** Simple Mail Transfer Protocol
- **Porta:** 25
- E' il protocollo standard utilizzato per spedire le email. Sia dal client al server, sia tra server e server.
- Non prevede autenticazione (viene implementata con metodi alternativi, come il POP-before-smtp)
- I server possono essere configurati però per accettare email solo da/per determinati hosts (quelli gestiti direttamente da lui), quindi per non fare il “relay”
- Ogni nuova email viene inserita in una “coda” che poi il server cerca di smaltire
- Un server remoto, magari non disponibile, potrebbe portare il server a fare piu tentativi di consegna

SMTP

S: 220 www.example.com ESMTP Postfix
C: HELO mydomain.com
S: 250 Hello mydomain.com
C: MAIL FROM: sender@mydomain.com
S: 250 Ok
C: RCPT TO: friend@example.com
S: 250 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: Subject: messaggio di prova
C: From: sender@mydomain.com
C: To: friend@example.com
C:
C: Ciao,
C: questa è una prova.
C: .
S: 250 Ok: queued as 12345
C: QUIT
S: 221 Bye

HTTP

- **Nome:** Hyper Text Transfer Protocol (versione 1.1)
- **Porta:** 80
- E' il protocollo standard per il trasferimento di pagine e files fruibili tramite un server web.
- In associazione ad HTML, è entrato nella storia: WWW
- Può prevedere meccanismi di autenticazione
- E' stateless. Una volta scaricati tutti i componenti di una pagina, la connessione viene chiusa automaticamente.
  - Per ovviare a questo problema, la prima soluzione ideata sono i cookies: coppie “variabile-valore” salvate dal browser e spedite al server ad ogni connessione
  - Nel “web-2.0” si sfrutta poi una tecnologia basata su JavaScript chiamata AJAX che consente di fare richieste asincrone ad un server web, “trasparentemente” all'utente

HTTP

C: GET / HTTP/1.1
C: Connection: Keep-Alive
C: User-Agent: Mozilla/5.0 (compatible; Konqueror/3.2; Linux) (KHTML, like Gecko)
C: Accept: text/html, image/jpeg, image/png, text/*, image/*, */*
C: Accept-Encoding: x-gzip, x-deflate, gzip, deflate, identity
C: Accept-Charset: iso-8859-1, utf-8;q=0.5, *;q=0.5
C: Accept-Language: en
C: Host: www.google.it
C:
S: HTTP/1.1 200 OK
S: Cache-Control: private
S: Content-Type: text/html
S: Set-Cookie:
PREF=ID=54d952b78ec3ba4d:TM=1172437864:LM=1172437864:S=QLFiepSB5OwNhJHJ;
expires=Sun, 17-Jan-2038 19:14:07 GMT; path=/; domain=.google.it
S: Server: GWS/2.1
S: Transfer-Encoding: chunked
S: Date: Sun, 25 Feb 2007 21:11:04 GMT
S:
S: <html><head><meta http-equiv="content-type" content="text/html; charset=ISO-88...

HTTP

- E' possibile inviare dati ad un webserver:
  - POST
    - Le informazioni vengono inserite tra gli header della richiesta HTTP
  - GET
    - Le informazioni vengono aggiunte all'URL:
      - http://www.google.it/?var=val&var1=val
- A partire da queste, il webserver può generare (tramite l'uso di programmi “lato-server”, scritti ad esempio in PHP) pagine differenti, basandosi sui dati ricevuti (ad esempio delle form).
- Ci sono poi anche PUT e DELETE (!!!)

Il Modem

- Quando vogliamo connetterci ad internet sfruttando un modem “dialup”, ci troviamo di fronte ad un problema che va ad aggiungersi a quelli insiti nell'uso della tecnologia stessa, che è quello del driver del modem.
- A causa della grande diffusione di Windows e della necessità di costruire apparati sempre meno “intelligenti”, nella produzione di modem interni, si è spesso “delegato” al sistema operativo (ed in particolare ad un driver software appositamente scritto) gran parte delle funzionalità del modem, riducendo l'hardware all'osso.
- Questo modem (win-modem o soft-modem), hanno per lungo tempo causato numerosi problemi agli utenti GNU/Linux in quanto non vi erano ne driver ne specifiche per scriverne uno funzionante.

Il modem

- Con il passare degli anni, sono stati scritti una serie di driver che oggi coprono la quasi totalità dei win-modem.
- Il problema non riguarda i modem “esterni”, perchè questi hanno “intelligenza a bordo” e parlano via seriale (usb)
- Il problema va ripetendosi con i modem ADSL esterni di tipo USB (il problema qui sta nel protocollo di comunicazione che non sempre è seriale [sic...])
- Una volta che il modem viene riconosciuto dal sistema, il grosso dei problemi sono risolti:

Si tratta solo di configurarlo :P

Il modem

- Per fare questo esistono
  - sia applicazioni grafiche
    - KPPP
    - gnome-ppp
  - Sia applicazioni testuali
    - wvdial
- Spesso i programmi da interfaccia grafica poi si avvalgono di quelli testuali per il riconoscimento e la configurazione del modem
- I dati solitamente richiesti all'utente sono:
  - Numero di telefono da chiamareù
  - Username cominicata dall'ISP
  - Relativa password

Il modem

- Il programma poi dovrà essere in grado di rilevare autonomamente (o chiedere all'utente in caso di fallimento) la device del modem e la velocità con cui comunicare con lui
- Wvdial è piuttosto semplice da utilizzare:
  - wvdialconf [nomefile]
    - Si occupa di cercare il modem e creare il file di configurazione [nomefile]
  - wvdial
    - Esegue la connessione vera e propria
- E' necessario editare i campi “phone”, “username” e “password” del file di configurazione inserendo i dati corretti comunicatici dall'ISP.

pppd

- Si usa poi un daemon per gestire la connessione ppp (connessione/riconnessione): pppd
- File di configurazione in `/etc/ppp/*`
  - `options`
    - Contiene la configurazione generale
  - `options.ttyXX`
    - Nel caso si possiedano piu modem, è possibile specificare opzioni diverse a seconda del modem
- Per supportare le autenticazioni di tipo PAP/CHAP, ci si avvale poi di altri files: `pap-secrets` e `chap-secrets` in cui si possono specificare, su colonne diverse, le varie credenziali.
- Per PPPoE esiste `/etc/ppp/pppoe.conf` che solitamente poi usa PAP o CHAP per l'autenticazione.

In LAN

- La configurazione di Ethernet, non necessitando di autenticazioni, è piu semplice.
- Si tratta solitamente di assegnare indirizzo IP, netmask, default gateway e server DNS all'interfaccia desiderata:
  - Se la netmask è facilmente derivabile dal tipo di indirizzo IP (usando le classi) è possibile ometterla, altrimenti, dovremo specificarla
    - `ifconfig eth0 192.168.1.2 netmask 255.255.255.248`
- Bisogna poi impostare il default gateway
  - `route add default gw 192.168.1.1`
- Ed il DNS in /etc/resolv.conf
  - `nameserver 192.168.1.1`

DHCP

- Se è disponibile un server DHCP in rete locale, è sufficiente eseguire il programma messo a disposizione dalla nostra distribuzione:
  - dhclient {interfaccia facoltativa}
  - dhcpcd {interfaccia facoltativa}
  - pump
- Una volta eseguita la configurazione, il client dhcp diventa poi un demone per rinnovare la configurazione ad intervalli regolari (leases)

Wireless

- Per le reti Wireless, oltre alla normale configurazione IP+routing/dhcp, è necessario anche gestire la parte di rete, gestendo il nome della rete ed eventuali estensioni crittografiche
- Impostare l'ESSID (nome della rete)
  - iwconfig eth2 essid “OpenLabs”
  - iwconfig eth2 essid any
- Trovare un access point
  - iwlist eth2 scan
- Password WEP
  - iwconfig eth2 essid “OpenLabs” key “s:password”
- Per gestire la parte di WPA e WPA2 si utilizza un programma a parte, chiamato “wpa_supplicant”

Configurazione

- Per rendere definitiva la configurazione di un'interfaccia di rete, solitamente, si utilizza un file di configurazione.
- Questo cambia però da distribuzione a distribuzione.
- Normalmente si trova tra gli script di avvio, in quanto viene incluso direttamente (è un po piu “grezzo”) oppure vi è un file di configurazione a parte che viene poi elaborato da uno script
- Debian ricade nella seconda tipologia
  - `/etc/init.d/networking`
  - `/etc/network/interfaces`
- E' inoltre possibile definire delle “zone” di configurazione, e gestirle automaticamente tramite script realizzati appositamente.

Configurazione

```
# This file describes the network interfaces
# available on your system
# and how to activate them. For more information, see
# interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
        address 81.29.194.7
        netmask 255.255.255.0
        gateway 81.29.194.1

auto eth1
iface eth1 inet dhcp
```