

UNIVERSITÀ MEDITERRANEA DI REGGIO  
CALABRIA

---

---

FACOLTÀ DI GIURISPRUDENZA  
CORSO DI LAUREA MAGISTRALE IN GIURISPRUDENZA

**RACCOLTA DI MATERIALE DIDATTICO  
PER IL CORSO DI  
DIRITTO DELL'INFORMATICA**

MELCHIORRE MONACA

MODULO TECNICO

2018/2019

---

---

# Testi di riferimento

- Alessio Plebe, Melchiorre Monaca  
*Introduzione all'informatica delle conoscenze*  
Editori Riuniti University Press 2010 - ISBN13: 9788864732152
- Andrew S. Tanenbaum, David J. Wetherall  
*Reti di calcolatori* - V edizione  
Pearson 2011 - ISBN13: 9788871926407
- AgID - Agenzia per l'Italia Digitale - Firme Elettroniche  
<http://www.agid.gov.it/agenda-digitale/infrastrutture-architetture/firme-elettroniche>
- AgID - Agenzia per l'Italia Digitale - Posta Elettronica Certificata  
<http://www.agid.gov.it/agenda-digitale/infrastrutture-architetture/posta-elettronica-certificata>
- AgID - Agenzia per l'Italia Digitale - SPID  
<http://www.agid.gov.it/agenda-digitale/infrastrutture-architetture/spid>
- AgID - Agenzia per l'Italia Digitale - Cloud Computing  
<http://www.agid.gov.it/agenda-digitale/infrastrutture-architetture/cloud-computing>

# Indice

<b>Premessa</b>	<b>1</b>
<b>1 Reti di calcolatori</b>	<b>2</b>
1.1 Dagli albori militari alla fisica delle alte energie . . . . .	3
1.2 Stipulare intese su come comunicare . . . . .	6
1.2.1 Pensare a strati . . . . .	11
1.3 Dalle intese ad Internet . . . . .	15
1.3.1 Consegnare i dati . . . . .	16
1.3.2 Comunicazione efficace . . . . .	35
1.3.3 Le Applicazioni . . . . .	38
<b>2 Sicurezza</b>	<b>45</b>
<b>Bibliografia</b>	<b>46</b>

# Premessa

Con questa raccolta s'intende fornire allo studente una sorta di "traccia", un percorso che guidi lo studio attraverso le tematiche discusse a lezione, trasmettendo il "lessico minimo" indispensabile alla comprensione degli argomenti trattati nel modulo giuridico di questa stessa materia.

È stato evidentemente necessario semplificare - forse ai limiti del lecito - i contenuti qui riassunti: il lettore più interessato alla materia potrà approfondire gli argomenti di suo interesse sui testi di riferimento consigliati e sui lavori originali citati in bibliografia.

# Capitolo 1

## Reti di calcolatori

Una rete di telecomunicazioni è un sistema che fornisce servizi relativi al trasferimento di informazioni ad una popolazione di utenti distribuiti geograficamente. Le reti di telecomunicazioni sono vicine alla nostra esperienza quotidiana di uomini moderni: basti pensare alla rete telefonica, alla rete postale, alle reti per diffusione radio e TV, alle reti telematiche.

Alcune di queste reti sono di nuova concezione e quindi utilizzano tecnologie avanzate, tipicamente del settore elettronico (e in qualche caso anche della fotonica), mentre altre, come la rete postale, sono state in funzione per quasi due secoli e si basano su strumenti molto più tradizionali, quali i mezzi di trasporto.

Sappiamo inoltre che in tempi remoti sono esistite reti di telecomunicazioni basate su tecnologie diverse, come torri d'avvistamento e segnali luminosi o bandiere (i castelli della Valle d'Aosta, la Grande Muraglia Cinese), segnali di fumo (caratteristici degli indiani americani), o segnali acustici (i tam-tam della giungla). Inoltre, verso la fine del secolo scorso erano state attivate reti telegrafiche basate su segnalazioni ottiche, utilizzando tralicci su cui erano montati pannelli mobili azionabili dal basso e visibili da lontano.

È evidente una rilevante differenza tra le reti citate ad esempio: le reti per diffusione radio e TV, i segnali di fumo ed i tam-tam costituiscono reti a diffusione e unidirezionali: l'informazione viene distribuita da una sorgente

a chiunque disponga di un apparato ricevitore, quindi a ogni utente della rete, indipendentemente dalla sua identità. Non è inoltre possibile per la gran maggioranza degli utenti, che dispongono solo di un apparato ricevente, inviare informazioni ad altri. Le reti telematiche, la rete telefonica, il sistema postale, sono invece reti a selezione e bidirezionali: sono caratterizzate dalla possibilità per la sorgente dell'informazione di scegliere a quali interlocutori questa deve essere trasferita. In questo caso tutti gli utenti sono attrezzati sia per trasmettere sia per ricevere.

## 1.1 Dagli albori militari alla fisica delle alte energie

Internet, la rete delle reti che è divenuta il nostro principale strumento di ricerca, collaborazione ed interazione sociale, una realtà consolidata per il lettore oggi venticinquenne, ha radici lontane nella storia dell'Informatica. Riesaminare insieme i passi remoti della sua storia è un esercizio utile ed istruttivo su come stimoli storici e sociali, idee geniali e tecnologia possono essere fusi per creare qualcosa di unico e così influente da cambiare lo stile di vita di due terzi della popolazione mondiale.

Come spesso accade, il primo impulso alla realizzazione di un sistema d'interconnessione di calcolatori su scala geografica fu di origine militare. La "mamma" di Internet fu infatti ARPANET, una rete costruita negli anni '70 a scopo militare, pensata per condividere online il tempo di utilizzazione dei computer tra i diversi centri di elaborazione dati dell'ARPA (*Advanced Research Projects Agency*), che eseguivano ricerche scientifiche a lungo termine per conto del Dipartimento della Difesa degli Stati Uniti. Un'impresa non da poco, dato che all'epoca non esistevano standard per la costruzione di calcolatori ed i "supercomputer" erano isolati e basati su sistemi incompatibili tra loro.

Il progetto è principalmente dovuto alla caparbia di Bob Taylor, direttore della divisione informatica dell'ARPA, alle idee rivoluzionarie di Joseph

Licklider, all'epoca coordinatore dell'IPTO (*Information Processing Techniques Office*) [1, 2, 3] ed alla teorizzazione delle reti a commutazione di pacchetto<sup>1</sup>, proposta da Leonard Kleinrock [4] come argomento per la sua tesi di dottorato al Massachusetts Institute of Technology (MIT).

Tali e tanti sono gli eventi che si susseguono nel corso di un decennio, che si ritiene opportuno elencarne la sequenza:

- 1961 (luglio) Leonard Kleinrock del MIT pubblica *Information flow in large communication nets* [4] sulla teoria del *packet switching*
- 1962 (agosto) Licklider & Wesley Clark del MIT pubblicano *On-line man computer communication* [5] che forse può essere considerato il primo articolo sul concetto di internet
- 1962 (ottobre) J.C.R. Licklider diviene direttore dell'IPTO
- 1964 Leonard Kleinrock pubblica il libro *Communication net* [6], nel quale descrive rigorosamente il funzionamento di una rete basata sul *packet switching*
- 1964 (marzo) Paul Baran della RAND Corporation descrive, in una serie di memoranda per la USAF, una rete di comunicazione capace di resistere ad un attacco termonucleare basata sul *packet switching* [7]
- 1964 (settembre) Ivan Sutherland diviene il nuovo direttore dell'IPTO
- 1965 (ottobre) Lawrence Roberts (MIT) e Thomas Marrill (CCA) effettuano il primo collegamento con tecnologia *packet switching* fra il *TX-2* dei Lincoln Labs a Lexington e l'*AN/FSQ-32* della SDC a Santa Monica
- 1966 (agosto) Robert Taylor diventa il terzo direttore dell'IPTO ed assume Lawrence Roberts per coordinare il progetto
- 1967 (aprile) Wesley Clark suggerisce di utilizzare una sottorete di minicomputer, tutti uguali e compatibili tra di loro, dedicata esclusivamente alla ricezione e trasmissione dei dati. Suggerisce di chiamare questi computer IMP (*Interface Message Processors*). È la svolta concettuale che

---

<sup>1</sup>contrapposte alla commutazione di circuito, tipica dei sistemi di telefonia analogici

permetterà di superare i problemi legati all'eterogeneità dei computer dell'epoca

1967 (ottobre) Donald W. Davies, che lavora al National Physical Laboratory (UK), pubblica i risultati delle sue ricerche sul packet switching, svolte in modo del tutto indipendente dai ricercatori americani [8]

1967 (ottobre) Lawrence Roberts presenta il disegno della futura rete [9]

1968 (agosto) Lawrence Roberts invia a 140 società la "Request For Proposals" (RFP) per la realizzazione degli IMP della rete ARPANET; al bando rispondono la BBN e la Raytheon, ma non IBM e AT&T che giudicano il progetto improponibile: la fornitura è affidata alla BBN

1968 (ottobre) Leonard Kleinrock viene assunto al Network Measurement Center (UCLA)

1969 (aprile) restano ancora da definire le caratteristiche che deve avere l'interfaccia tra i singoli calcolatori e gli IMP, che sono pubblicate da Bob Kahn

1969 (aprile) Steve Crocker scrive il primo *Request For Comment* (RFC) che tratta l'*host-to-host protocol* [10]: da questo momento tutti i protocolli di rete saranno formalizzati in documenti di questo tipo e gli RFC saranno il principale strumento di collaborazione e sviluppo della comunità di ricerca nell'ambito delle reti di calcolatori [11].

Finalmente, nell'ottobre 1969 viene stabilito il primo collegamento da computer a computer fra l'Università della California di Los Angeles e lo Stanford Research Institute: nasce il primo link ARPANET. I centri collegati si susseguono con ritmo incalzante ed alla fine del 1971 la rete conta già 15 nodi, che diventano 37 alla fine del 1972; da allora la crescita è esponenziale. È il periodo nel quale viene formalizzato da Crocker il protocollo NCP (*Network Control Program*) [12], che stabilisce le regole per la connettività alla base di ARPANET.

La svolta determinante avviene però nel 1974 con la pubblicazione dell'RFC 675, dal significativo titolo "Specification of Internet Transmission Control



Program” [13], dove appare per la prima volta il termine “Internet”. Nel 1978 Cerf, Postel e Crocker aggiungono al TCP il protocollo IP (*Internet Protocol*) [14], mettendo a punto il definitivo modello su cui ancor oggi opera Internet, il TCP/IP [15].

Parallelamente nascono le prime applicazioni: Telnet, per la gestione remota di terminali, FTP per la trasmissione di file, E-Mail per la posta elettronica: sono tutte applicazioni basate sulla linea di comando, con un’interfaccia ostica e riservata al personale tecnico e ricercatore. Il primo tentativo di “interfaccia universale” alle risorse di rete è Gopher<sup>2</sup>, ma la vera rivoluzione arriva con l’implementazione dell’interfaccia grafica e del mouse nei sistemi operativi.

Questi due elementi rendono possibile la nascita del World Wide Web (1990), un sistema per la condivisione di informazioni in ipertesto, sviluppato da Tim Berners-Lee [16] presso il CERN di Ginevra, pensato per facilitare la condivisione di informazioni scientifiche nella comunità dei fisici nucleari. Il sistema si basa sul protocollo HTTP [17] e sul linguaggio HTML: per standardizzare quest’ultimo, Berners-Lee fonda il World Wide Web Consortium (W3C), che tuttora si occupa di definire gli standard per il web<sup>3</sup>.

Il resto è cronaca: l’avvento della “banda per tutti” (*broadband*) e, soprattutto, il DNS (Domain Name System), i motori di ricerca, i social network, la necessità di generare contenuti in modo collaborativo e semanticamente pregnante hanno portato alla nascita di quello che adesso è definito Web 2.0 [18].

## 1.2 Stipulare intese su come comunicare

Si vuole iniziare suggerendo una riflessione su cosa “si nasconde” dietro una semplice telefonata. Quando si solleva il microtelefono, prima di compiere

---

<sup>2</sup>la prima applicazione di rete basata su menu descrittivi a struttura gerarchica, realizzati mediante un’architettura di tipo client server

<sup>3</sup>una curiosità: il computer usato da Berners-Lee per realizzare il primo server web era basato sul sistema operativo NeXT, realizzato da Steve Jobs prima di rientrare alla Apple

qualsiasi azione, l'apparecchio telefonico controlla se è attivo il segnale di linea e restituisce un feedback sonoro: in modo totalmente trasparente per il chiamante, il telefono ha verificato l'esistenza di un collegamento fisico con la rete telefonica e la disponibilità di un canale per iniziare la comunicazione. Il passo successivo è la composizione del "numero di telefono", cioè dell'indirizzo del destinatario della telefonata. Val la pena notare che tale indirizzo è rigidamente codificato (infatti non può essere scelto dall'utente finale, ma viene assegnato dal provider dei servizi telefonici), è univoco (non possono esistere due utenti con lo stesso numero) ed ha una struttura gerarchica: un prefisso internazionale, un prefisso nazionale, un eventuale prefisso di centralino e, infine, l'identificativo del telefono chiamato.

Composto il numero, il segnale sarà instradato attraverso la rete telefonica mondiale e la richiesta di iniziare la comunicazione arriverà a destinazione: in altre parole, il telefono del destinatario squillerà e inizierà la conversazione. . .

. . . O no?

In realtà (pur supponendo che il telefono chiamato sia perfettamente funzionante) possono verificarsi almeno tre casi:

- l'essere umano che si vuole contattare non è in casa o non vuole rispondere
- l'apparecchio del destinatario è utilizzato per un'altra conversazione
- il destinatario risponde

Le tre situazioni saranno gestite in modo diverso. Nel primo caso, probabilmente, dopo un pò d'attesa, sarà chi chiama a chiudere la comunicazione; nel secondo caso si otterrà il tipico segnale di "occupato". Se invece si è fortunati, il destinatario, attivato il proprio microtelefono, risponderà col classico "... pronto..." o con un'altra frase convenzionale, a quel punto anche il chiamante, per iniziare la conversazione, userà un'espressione analoga. Durante tutta la telefonata, il sistema telefonico si occuperà di garantire che ciascun suono pronunciato giunga a destinazione velocemente, nella giusta sequenza e senza disturbi. Alla fine della telefonata sorge un problema, spesso causa

d'imbarazzo tra gli interlocutori: chi chiude per primo? Questa situazione capita, come si vedrà, anche nella gestione della connessione tra apparati fisici.

Infine, durante tutta la telefonata, non ci siamo minimamente preoccupati del tipo di telefono posseduto dal nostro interlocutore (fisso, cellulare, cordless, isdn, analogico): in realtà l'eterogeneità dei dispositivi da far comunicare aggiunge complessità e deve essere opportunamente gestita.

L'esempio precedente, sia pur semplicistico e limitato, evidenzia la complessità insita nel problema di mettere in comunicazione due entità remote (*end point*) variamente collegate.

Realizzare una rete di calcolatori richiede risorse e competenze in vari ambiti disciplinari, che vanno dalla progettazione e costruzione dei dispositivi fisici, al disegno e realizzazione del software che gestisce i servizi.

Pensare di codificare la materia in un blocco monolitico è dunque improponibile: è invece opportuno tentare di scomporre la problematica in una serie d'obiettivi specifici, il più possibile indipendenti l'uno dall'altro.

Il primo, ovviamente, è la connettività fisica tra gli apparati, necessaria affinché qualsiasi forma di comunicazione possa avvenire. Si noti, ancora una volta, la necessità di collegare alla stessa rete dispositivi di tipo diverso (per esempio PC e telefoni cellulari), su un ampio range di distanze (stampare un documento mediante la stampante dipartimentale è differente dal consultare un sito Web) e con esigenze di prestazioni diverse (lo *streaming* di un film richiede una "velocità" della rete molto maggiore rispetto alla trasmissione di un messaggio *e-mail*).

Un altro aspetto fondamentale è l'indirizzamento, cioè la necessità di poter identificare univocamente ciascun apparato (*host*) presente sulla rete. Come nel caso della telefonata, alla caratteristica d'univocità si deve associare la possibilità di conferire allo spazio d'indirizzamento una struttura gerarchica. Ancora, su una rete complessa, come, ad esempio, quella mondiale, è necessario poter determinare il percorso che i dati devono seguire per arrivare a destinazione nel modo più efficiente possibile: occorre cioè un

meccanismo di instradamento del traffico.

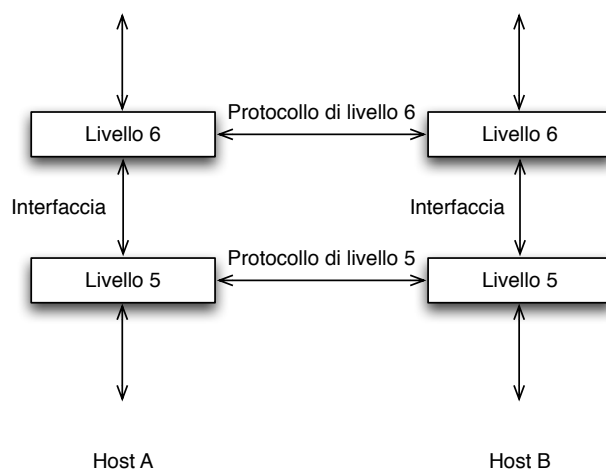
Tuttavia, il collegamento più veloce ed efficiente servirebbe a poco se i dati trasmessi non fossero correttamente interpretabili dal destinatario: occorre qualcosa che presieda al trasporto dei dati e che gestisca la connessione di modo che questi possano essere ricevuti in modo completo e senza errori da entrambe le parti. Infine occorre codificare le applicazioni che la rete deve erogare, i servizi, la loro interfaccia verso l'utente finale.

In sintesi (e semplificando) chi progetta reti di calcolatori dovrà tener conto di queste caratteristiche:

- Collegamento fisico
- Indirizzamento
- Instradamento
- Trasporto dei dati e gestione della connessione
- Applicazioni

Adesso si passerà a formalizzare quanto detto fin qui: per ridurre la complessità di progetto, le reti sono, in generale, organizzate a livelli, ciascuno costruito sopra il precedente. Lo scopo di un livello è offrire servizi ai livelli più alti, nascondendo i dettagli sull'implementazione. Le due macchine che devono comunicare prendono il nome di *host*: il livello  $n$  su un host “conversa” col livello  $n$  su un'altro host. Le regole e le convenzioni che governano la conversazione sono collettivamente indicate col termine protocollo di livello  $n$ . Si parla, in questo caso di “conversazione tra pari” e le entità (processi) che effettuano tale conversazione si chiamano *peer entity* (entità di pari livello).

In realtà non c'è un trasferimento diretto dal livello  $n$  di host  $A$  al livello  $n$  di host  $B$ . Ogni livello di host  $A$  passa i dati, assieme a delle informazioni di controllo, al livello sottostante; questo procedimento prende il nome di incapsulamento e ne discuteremo ampiamente più avanti. Questo meccanismo è evidenziato in Fig. 1.1. Al livello 1 c'è il mezzo fisico, attraverso il quale i dati vengono trasferiti da host  $A$  ad host  $B$ . Quando arrivano a host  $B$ , i

Figura 1.1: Comunicazione *peer to peer* tra livelli

dati sono trasmessi da ogni livello (a partire dal livello 1) a quello superiore, fino a raggiungere il livello  $n$ .

Fra ogni coppia di livelli adiacenti è definita un'interfaccia, che caratterizza le operazioni primitive che possono essere richieste *al* livello sottostante ed i servizi che possono essere offerti *dal* livello sottostante. Una buona progettazione delle interfacce tra i livelli consente di minimizzare la quantità e la complessità delle informazioni da trasferire e non preclude la possibilità di aggiornare l'implementazione del livello con l'evolversi della tecnologia.

Si noti che, in questo schema, un servizio definisce quali operazioni il livello è pronto ad eseguire per conto dei propri utenti (il livello superiore e quello inferiore), ma non dice nulla su come tali operazioni dovranno essere realizzate.

Di quest'ultimo aspetto si occupano i protocolli, cioè insiemi di regole che governano il formato ed il significato dei blocchi di informazione, dei pacchetti o dei messaggi che vengono scambiati dalle peer entity di un dato livello. Le entità utilizzano i protocolli per formalizzare le definizioni dei propri servizi. Esse sono libere di modificarli in futuro, purché non cambino i servizi erogati, implementando così la totale indipendenza della realizzazione di ciascun livello rispetto agli altri.

In questo modo si ottengono due vantaggi importanti: il primo è che possono dialogare fra loro anche host aventi caratteristiche (processore, sistema operativo, costruttore) diverse, il secondo è che ciascun livello si occuperà di un aspetto specifico in modo indipendente dall'implementazione dei livelli sottostanti; per esempio, una pagina web potrà essere interpretata correttamente dal *browser* indipendentemente dal fatto che il fruitore stia utilizzando un computer desktop collegato alla rete dell'Università o il suo portatile collegato via radio tramite un *hot spot* dell'aeroporto. L'insieme dei livelli e dei relativi protocolli di una specifica implementazione è detto architettura di rete.

### 1.2.1 Pensare a strati

Nelle pagine seguenti verranno esaminate in parallelo due formalizzazioni della struttura a strati fin qui presentata: un modello di riferimento, l'*ISO/OSI* ed un'architettura di rete, il *TCP/IP*. La sostanziale differenza tra i due è che il modello *ISO/OSI* si limita a specificare cosa dovrebbe fare ciascun livello, ma non specifica con precisione i servizi ed i protocolli che devono essere usati e, dunque, non può essere considerato un'architettura di rete. Tuttavia la sua rilevanza storica e concettuale lo rende il fulcro di ogni moderna implementazione di rete.

L'*OSI (Open Systems Interconnection) Reference Model* [19] è il frutto del lavoro della *ISO (International Standard Organization)*, ed ha lo scopo di:

- fornire uno standard per la connessione di sistemi aperti, cioè in grado di colloquiare gli uni con gli altri;
- fornire una base comune per lo sviluppo di standard per l'interconnessione di sistemi;
- fornire un modello rispetto a cui confrontare le varie architetture di rete.

Esso non include la definizione di protocolli specifici (che sono stati definiti successivamente, in documenti separati). I principi di progetto che sono stati seguiti durante lo sviluppo del modello OSI schematizzano fedelmente quanto esposto nel paragrafo precedente:

- ogni livello deve avere un diverso strato di astrazione;
- ogni livello deve avere una funzione ben definita;
- i limiti dei livelli devono essere scelti in modo da minimizzare il passaggio delle informazioni attraverso le interfacce;
- il numero dei livelli deve essere abbastanza ampio per permettere a funzioni distinte di non essere inserite forzatamente nel medesimo livello senza che sia necessario e abbastanza piccolo per permettere che le architetture non diventino pesanti e poco maneggevoli.

Sono individuati e formalizzati sette livelli, numerati a partire dal basso: fisico, data link, network, trasporto, sessione, presentazione, applicazione (Fig. 1.2).

I livelli più bassi, da quello fisico a quello di trasporto, si occupano della consegna dei dati tra gli host, mentre quelli più alti si occupano della loro elaborazione e realizzano perciò le applicazioni di rete. Non si esplicherà adesso il significato ed il ruolo del singolo livello: essi diverranno più chiari nel seguito della trattazione.

Come avviene in pratica la comunicazione tra un livello e quello sottostante? Si supponga di dover spedire una lettera: redatto il messaggio su un foglio di carta, si metterà quest'ultimo in una busta, sulla quale viene scritto l'indirizzo del mittente e del destinatario. L'addetto della compagnia postale ritirerà la busta e la porterà al centro di smistamento della città di partenza, dove la lettera sarà messa in un sacco indirizzato alla città di destinazione. Il sacco sarà caricato via via sugli opportuni mezzi di trasporto (non importa quali e quanti) e giungerà al centro di smistamento della città di destinazione. Qui sarà aperto e la nostra busta sarà consegnata al postino per la consegna finale. Il postino leggerà l'indirizzo e consegnerà la lettera

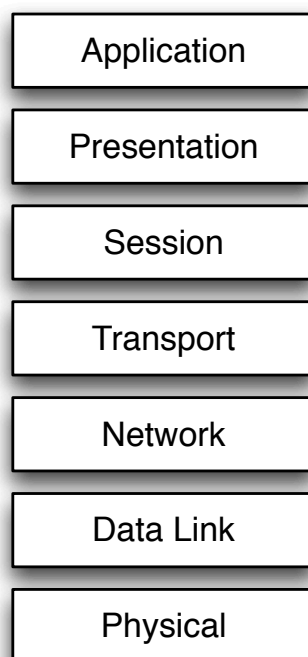


Figura 1.2: I livelli del modello di riferimento ISO/OSI



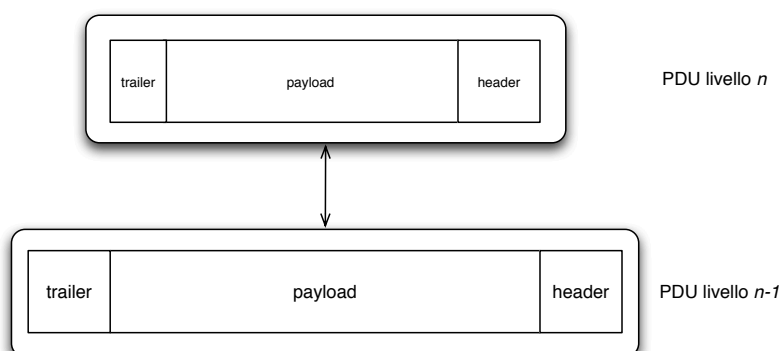


Figura 1.3: Incapsulamento dei dati nel livello sottostante

al destinatario. Il destinatario, letto l'indirizzo, aprirà la busta e leggerà il messaggio. È importante notare che soltanto il mittente ed il destinatario elaborano le informazioni contenute nella lettera, tutti gli altri protagonisti della consegna si limitano a leggere l'indirizzo sulla busta (o sul sacco) e reindirizzano la missiva alla tappa successiva.

Nelle reti di comunicazione avviene qualcosa d'analogo: i dati dell'applicazione vengono incapsulati nei livelli sottostanti fino ad arrivare al livello fisico; durante il percorso vengono "aperte" solo le "buste" relative ai livelli che si occupano dell'instradamento del messaggio e solo sull'host di destinazione i dati dell'applicazione vengono elaborati.

In altri termini, ciascun livello dell'host mittente incapsula i dati (*payload*) del livello superiore premettendo un'intestazione (*header*) ed, eventualmente, posponendo dei codici di controllo (*trailer*); a sua volta, il pacchetto così costruito diventa payload del livello sottostante (Fig. 1.3).

Lungo il percorso attraverso i nodi della rete vengono elaborati ed eventualmente modificati solo gli header dei livelli che si occupano della trasmissione. Soltanto sull'host di destinazione saranno elaborati gli header relativi ad ogni livello, fino alla consegna dei dati all'applicazione. Ad esempio, un *router*, che è un apparato che realizza l'instradamento dei dati, "aprirà" soltanto le "buste" fino al livello 3, che contiene le informazioni necessarie. Ciascun livello avrà una propria *Protocol Data Unit* (PDU), composta da

header, payload e trailer, che realizza l'incapsulamento. In particolare, è utile, per riferimento nel prosieguo della trattazione, conoscere i nomi delle PDU dei primi quattro livelli:

1. Livello fisico: *bit*
2. Livello data link: *frame*
3. Livello network: *pacchetto*
4. Livello trasporto: *TPDU* (*Transport Protocol Data Unit*)

## 1.3 Dalle intese ad Internet

Il TCP/IP (*Transmission Control Protocol / Internet Protocol*) [15] è l'architettura di rete che costituisce il fondamento della trasmissione dati in Internet.

Rispetto al modello ISO/OSI condensa i livelli fisico e data-link in un unico livello (*host-to-network*) ed i tre livelli applicativi in un più generico *application* (Fig. 1.4). Essendo nato principalmente per realizzare servizi Internet, formalizza in modo rigoroso i due livelli di networking (*Internet* e *transport*) definendone le funzionalità, lo spazio d'indirizzamento e le caratteristiche dei protocolli. Il livello più basso non è specificato nell'architettura, che prevede di utilizzare i protocolli disponibili per le varie piattaforme hardware e conformi agli standard, purché consentano agli host di inviare pacchetti IP sulla rete. Il livello applicativo è quello sul quale poggiano le applicazioni Internet che oggi tutti conoscono (posta elettronica, web, trasferimento file).

I protocolli che implementano le funzionalità del TCP/IP sono formalizzati in una serie di documenti denominati RFC (*Request For Comment*) [10, 11]: sono documenti aperti, nel senso che la descrizione di un protocollo esposta in ciascuno di essi potrà essere migliorata con la pubblicazione di un RFC successivo. Si descriverà di seguito sommariamente il ruolo di ciascun livello (usando la nomenclatura della pila ISO/OSI) evidenziando esempi d'implementazione nell'ambito del TCP/IP.

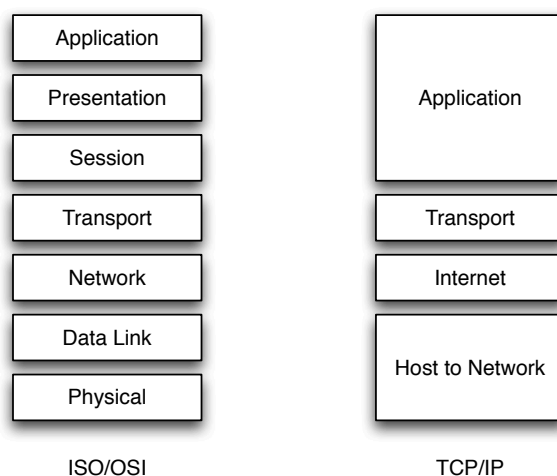


Figura 1.4: Lo stack TCP/IP

### 1.3.1 Consegnare i dati

Il livello fisico attiene la trasmissione di bit “grezzi” su un canale di comunicazione. Gli aspetti di progetto sono volti a garantire la congruenza dei bit ricevuti con quelli trasmessi; le specifiche sono, in massima parte, relative alle caratteristiche meccaniche, elettriche e procedurali delle interfacce di rete (componenti che connettono l’elaboratore al mezzo fisico) e alle caratteristiche del mezzo fisico stesso. La misura della “velocità” pura di una rete è data dalla quantità di dati che è possibile trasmettere nell’unità di tempo e spesso viene indicata con il nome di banda. L’unità di misura è il *bit/s* (attenzione, *bit* non *byte*), con i vari multipli: *kilo*, *Mega*, *Giga*. In teoria (ed in conformità col Sistema Metrico Internazionale) i multipli citati dovrebbero essere potenze di 10 (un kilo=1000), ma la matematica in base 2 e la tradizione portano spesso ad impiegare come moltiplicatore  $2^{10} = 1024$ .

In realtà, fatto comunque salvo il principio “. . . più banda c’è meglio è . . .”, spesso le prestazioni di una rete dipendono da altri fattori. Per esempio è auspicabile che una transazione bancaria vada, sia pur lentamente, a buon fine, piuttosto che precipitare rapidamente in uno stato imprevedibile. Oppure, nel caso di trasmissione di voce in tempo reale (una telefonata su Internet),

non è fondamentale che la banda sia elevata: è molto più importante che la banda richiesta per la telefonata (piccola, dell'ordine dei  $16Kb/s$ ) sia erogata con costanza nel tempo (*jitter* basso) di modo che le due parti possano percepire il parlato con fluidità. La destinazione d'uso della rete da realizzare e la sua estensione geografica giocano un ruolo fondamentale nei processi decisionali relativi alla progettazione; è allora utile suddividere le reti in categorie, ovviamente in modo del tutto indicativo. Una possibile classificazione è la seguente:

- PAN (*personal area network*): è una rete informatica utilizzata per permettere la comunicazione tra diversi dispositivi in ambito domestico; le tecnologie più utilizzate sono generalmente wireless (*WiFi*, *Bluetooth*), ma talvolta vengono usati anche cavi (*Ethernet*, *USB*, *FireWire*). Il classico esempio è costituito dal router WiFi, utilizzato per la connessione ad Internet.
- LAN (*local area network*): è costituita da computer collegati tra loro all'interno di un ambito fisico delimitato (un'azienda, un campus universitario); il cablaggio è costituito da un livello di distribuzione in fibra ottica e da punti d'accesso realizzati con cavo in rame. La banda tipica è dell'ordine del  $Gb/s$ .
- MAN (*metropolitan area network*): è un'infrastruttura in fibra ottica o, più raramente, wireless (*WiMax*) che realizza dorsali a larga banda che collegano i principali centri della vita sociale, politica e culturale della città.
- WAN (*wide area network*): realizza l'interconnessione tra le reti metropolitane, l'esempio tipico è Internet stessa.

La scelta del mezzo trasmissivo, cioè il supporto fisico che consente il collegamento degli host, è un elemento di fondamentale importanza nella realizzazione di una rete effettivamente funzionante; tale scelta è dettata sia da considerazioni tecniche (distanza massima tra due apparati, larghezza di banda, ostacoli di natura geografica), sia da motivazioni di carattere

economico-sociale (la diffusione del mezzo sul territorio, gli elevati costi e l'impatto sociale ed ambientale di alcune tecnologie).

Schematizzando, si usano mezzi fisici di quattro tipologie diverse: cavo elettrico, onde radio, fibra ottica, laser. Sulle classiche reti su doppino telefonico (dette anche POTS, *plain old telephone system*) è possibile realizzare reti con diverse tecnologie. Nel decennio scorso era frequente l'uso di *modem* per codificare segnali digitali sopra le comuni linee telefoniche analogiche: la connessione era *on-demand* e la velocità limitata a circa  $56Kb/s$ . Il grande vantaggio di questa tecnologia è che non richiede modifiche alla rete distribuita esistente. Una prima evoluzione furono le linee ISDN, costituite da due canali telefonici (in realtà ne serve un terzo, di controllo) in tecnologia digitale. La velocità massima di  $128Kb/s$  veniva raggiunta sfruttando due connessioni in parallelo su canali da  $64Kb/s$ . Ma la tecnologia che ha consentito la diffusione di massa (*broadband*) della connettività domestica è, senza alcun dubbio, l'ADSL (*asymmetric digital subscriber line*): essa richiede l'installazione di nuovi apparati di commutazione nelle centrali telefoniche, chiamati DSLAM, e l'utilizzo di filtri negli impianti telefonici domestici per separare le frequenze utilizzate per la trasmissione dati da quelle per la comunicazione vocale. La banda erogata è asimmetrica, tipicamente  $7Mb/s$  in *download* e  $384Kb/s$  in *upload*, ma ormai tutti gli operatori telefonici offrono collegamenti a velocità maggiore.

Tra i candidati a sostituire il doppino per la distribuzione domestica dei servizi di telecomunicazioni, si possono citare le fibre ottiche e le infrastrutture della TV via cavo (diffusa soprattutto negli USA), il trasporto di dati sulla rete elettrica, le reti wireless e le reti satellitari (utili in aree disagiate). Per realizzare le LAN si usano in genere particolari cavi (UTP), costituiti da quattro doppini, ed interfacce di rete *Ethernet*: la particolare tecnica realizzativa li rende meno sensibili alle interferenze, consentendo di raggiungere velocità dell'ordine del  $Gb/s$ . Con tecnologie più costose, tipicamente utilizzate dai provider, si raggiungono velocità di  $40Gb/s$  per il singolo link su fibra ottica.

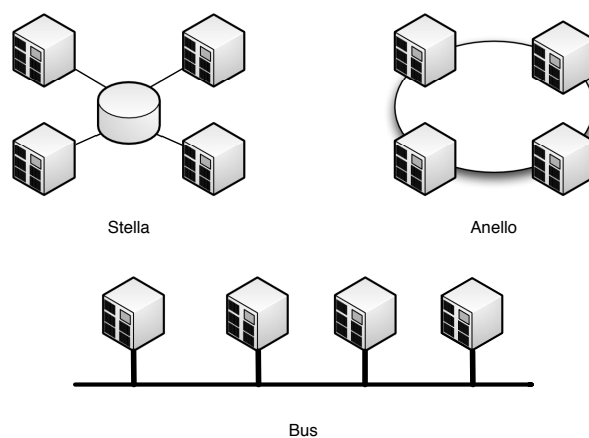


Figura 1.5: Topologia a stella, anello, bus

Il modo in cui i componenti di una rete sono collegati tra di loro, nel senso della disposizione ideale che questi hanno, viene definito generalmente attraverso quella che è nota come topologia di rete. Le reti punto a punto (*point-to-point*) consistono in un insieme di coppie di elaboratori connessi tra loro in vario modo (stella, anello, albero). Per passare da una sorgente ad una destinazione, l'informazione deve attraversare diversi elaboratori intermedi. Si ha una rete a stella quando tutti i componenti periferici sono connessi a un nodo principale in modo indipendente dagli altri; in tal modo tutte le comunicazioni passano per il nodo centrale e sono gestite completamente da questo. Si ha una rete ad anello quando tutti i nodi sono connessi tra loro in sequenza, in modo da formare un anello ideale, dove ognuno ha un contatto diretto solo con il precedente e il successivo; la comunicazione avviene (semplificando) a senso unico e ogni nodo ritrasmette i dati che non sono ad esso destinati al nodo successivo. Le reti *broadcast* (o *bus*) invece sono formate da un unico mezzo fisico, condiviso da più elaboratori, sul quale i messaggi inviati da un host vengono ricevuti da tutti gli altri. All'interno del messaggio vi è una parte relativa all'indirizzo del destinatario (elaborata a livello 2), in modo che tutte le altre macchine in ascolto possano scartare il messaggio in arrivo. Un esempio di una tale rete è la comune Ethernet [20].

Un problema tipico delle reti a bus è l'allocazione del canale trasmissivo.

Si pensi ad una normale conversazione tra esseri umani: capita talvolta che i due interlocutori inizino a parlare contemporaneamente. Di solito si genera una situazione d'imbarazzo che conduce ad un istante di silenzio, poi, dopo un intervallo casuale, uno dei due interlocutori riprende a parlare e la conversazione può aver luogo.

Analogamente, nel caso in cui il mezzo fisico è condiviso da più di due host, la trasmissione simultanea da parte di due di essi genera una sovrapposizione del segnale elettrico che inficia la trasmissione: è stata generata una collisione. Gli host che condividono il mezzo trasmissivo appartengono dunque allo stesso dominio di collisione: è evidente che maggiore è il numero di macchine appartenenti al dominio di collisione, più elevata è la probabilità che le collisioni abbiano luogo. Una buona regola nella progettazione delle reti è quindi far sì che i domini di collisione siano di dimensioni limitate.

Ciò non è possibile al mero livello fisico: occorre un protocollo, collocato nella parte bassa del livello 2, che consenta l'allocazione del canale trasmissivo all'host che vuole trasmettere. Si esamina ora il più diffuso, tipico delle reti ethernet: il CSMA/CD. CSMA/CD è l'acronimo inglese di *Carrier Sense Multiple Access with Collision Detection*, ovvero accesso multiplo tramite rilevamento della portante e delle collisioni. L'algoritmo è il seguente:

1. L'adattatore di rete sistema il messaggio in un buffer;
2. Se il canale è inattivo procede alla trasmissione, se è occupato attende prima di ritrasmettere;
3. Mentre trasmette, l'adattatore controlla la rete (è questo il vero e proprio collision detection), se non riceve segnali da altri adattatori considera il messaggio spedito, altrimenti è avvenuta una collisione, quindi va interrotta la trasmissione;
4. Se l'adattatore riceve, durante una trasmissione, un segnale da un altro adattatore, arresta la trasmissione e trasmette un segnale di disturbo (*jam*);

5. Dopo aver abortito la trasmissione attende un tempo casuale e ritrasmette.

Evidentemente un approccio di questo genere è poco efficiente perché comporta un elevato numero di ritrasmissioni, ma le reti a bus hanno il considerevole vantaggio dell'economicità e, per questo, sono ormai le più diffuse.

Quando si vogliono unire due o più reti (o anche degli elaboratori singoli) per formarne una sola più grande, occorre utilizzare dei nodi speciali connessi simultaneamente a tutte le reti da collegare. Il ripetitore è un componente che collega due reti fisiche intervenendo al primo livello ISO/OSI. In questo senso, il ripetitore non filtra in alcun caso i pacchetti dati, ma rappresenta semplicemente un modo per allungare un tratto di rete oltre il limite imposto dal singolo cavo passivo. Il ripetitore tipico è l'HUB, ovvero il concentratore di rete.

Da quanto detto risulta evidente che non può esistere un dispositivo di livello 1 in grado di interrompere un dominio di collisione; inoltre il lettore attento avrà notato che non è ancora emerso alcun tipo di meccanismo d'indirizzamento. Per ottenere questi risultati (ed altro ancora) occorre salire di livello.

Il *bridge* o *switch* è un dispositivo di livello 2 che mette in connessione due (o più) reti. Limitandosi a intervenire nei primi due livelli del modello ISO-OSI, il bridge è in grado di connettere tra loro solo reti fisiche dello stesso tipo. Il bridge più semplice duplica ogni frame nelle altre reti a cui è connesso; quello più sofisticato è in grado di determinare gli indirizzi dei nodi connessi nelle varie reti, ottimizzando il traffico. Nell'ottica dell'allocazione del canale trasmissivo, è interessante notare che l'inserimento di un bridge tra due segmenti di una rete a bus divide il dominio di collisione (Fig. 1.6).

Il livello data link ha il compito di offrire una comunicazione affidabile ed efficiente a due macchine adiacenti, cioè connesse fisicamente da un canale di comunicazione. Si occupa dunque di fare da tramite tra il livello 1 (fisico), che realizza la mera connettività ed il livello 3 (network), che instrada il



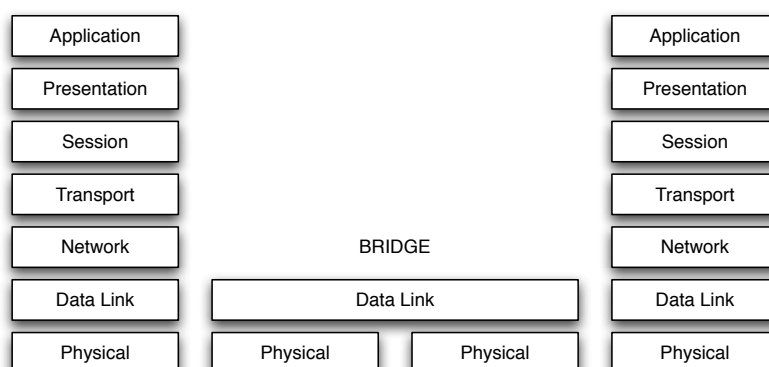


Figura 1.6: Collegamento a livello 2 mediante un bridge

traffico dati sulla rete geografica. È il livello sovrano delle LAN e attende alle seguenti incombenze principali:

- Frammentazione
- Controllo dell'errore
- Controllo di flusso
- Indirizzamento di livello 2

Un problema da non sottovalutare nella trasmissione dei dati a livello fisico è costituito dall'inaffidabilità del mezzo trasmissivo: per farsene un'idea basti pensare all'effetto che possono avere collisioni, interferenze e cadute di tensione sui collegamenti in rame. Per questo motivo è opportuno che il livello 2 si occupi di minimizzare il danno, organizzando i dati da trasmettere in piccoli "contenitori", i *frame*; in questo modo, un'eventuale problema comporterà la ritrasmissione di una piccola quantità d'informazione e non di tutto il contenuto della comunicazione. Questa operazione prende il nome di frammentazione.

Frammentati i dati, occorrerà prevedere un protocollo di controllo dell'errore, di solito basato sulla creazione di una *checksum*, una stringa generata con un opportuno algoritmo applicato al *payload* del frame. Questa sarà

calcolata dall'host trasmittente ed accodata al frame: l'host ricevente provvederà, ricevuto il frame, al ricalcolo della checksum, mediante il medesimo algoritmo, la confronterà con quella trasmessa e scarterà tutti i frame corrotti.

Per ogni frame correttamente ricevuto sarà inviata all'host trasmittente una "ricevuta di ritorno" (*acknowledgement* o, più semplicemente, *ack*): tutti i frame per i quali non arriverà, entro un tempo limite, un ack al mittente saranno ritrasmessi. Questo procedimento per il controllo del flusso è molto efficiente e consente, tra l'altro, di adeguare la velocità di trasmissione all'effettiva capacità di elaborazione del singolo host.

Infine, il livello 2 provvede a fornire una prima forma d'indirizzamento, in modo da evitare che host non coinvolti nella comunicazione siano comunque costretti ad elaborare a livelli più alti i dati ricevuti prima di scartarli.

Lo standard di livello 2 attualmente più diffuso è Ethernet. È una tecnologia nata molto presto, è più economica e facile da usare rispetto ai sistemi concorrenti, funziona bene e genera pochi problemi ed è adeguata all'utilizzo con TCP/IP; col passare del tempo lo standard si è aggiornato ed oggi consente velocità di trasmissione dell'ordine del *Gb/s*. Fornisce al livello di rete un servizio senza alcuna contrattazione iniziale ed il frame viene inviato nella LAN in modalità broadcast. Quando sarà ricevuto da tutti gli adattatori presenti sulla LAN, quello che vi riconoscerà il suo indirizzo di destinazione lo elaborerà (ed i dati saranno consegnati al livello 3), mentre tutti gli altri lo scarteranno. La gestione delle collisioni e dell'occupazione simultanea del canale di trasmissione viene gestita mediante il CSMA/CD.

Nelle reti più recenti si tende ad evitare completamente il problema delle collisioni, collegando ciascun host ad un bridge multiporta (*switch*) cosicché il dominio di collisione a cui appartiene ciascun host risulta essere popolato da due sole schede di rete: quella dell'host e la singola porta dello switch alla quale è collegato.

Gli indirizzi sono tutti a 6 byte in quanto Ethernet definisce uno schema d'indirizzamento a 48 bit [21]: ogni nodo collegato, quindi, ha un indirizzo Ethernet univoco di questa lunghezza. Esso corrisponde all'indirizzo fisico

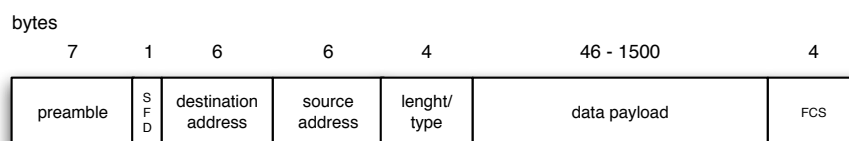


Figura 1.7: Struttura a blocchi di un *frame* Ethernet

della macchina ed è associato all'hardware (MAC *address*). Il MAC address viene, di solito, rappresentato in forma esadecimale: per esempio, il MAC address della scheda di rete del calcolatore col quale vengono redatte queste pagine è 00:0d:93:45:f4:22.

In figura 1.7 è mostrata la struttura a blocchi di un frame Ethernet: si noti la presenza del MAC address sorgente e del MAC address destinazione. Inoltre il payload del frame ha dimensioni massime di 1500 Bytes, quindi il protocollo frammenterà i dati ricevuti dal livello 3 in blocchi di questa dimensione.

Per poter realizzare la consegna dei dati da un protocollo di livello 3, come nel caso del protocollo IP che viene descritto più avanti, ad un protocollo di livello 2, occorre un modo per definire un abbinamento tra gli indirizzi di questo protocollo superiore e gli indirizzi fisici delle interfacce utilizzate effettivamente, secondo le specifiche del livello inferiore.

Le interfacce Ethernet hanno un sistema di indirizzamento composto da 48 bit. Quando con un protocollo di livello network si vuole contattare un nodo, identificato quindi da un indirizzo di livello 3, se non si conosce l'indirizzo Ethernet, ma ammettendo che tale nodo si trovi nella rete fisica locale, viene inviata una richiesta circolare (broadcast di livello 2, indirizzo di destinazione FF:FF:FF:FF:FF:FF) secondo il protocollo ARP (*Address Resolution Protocol*). La richiesta ARP è ascoltata da tutte le interfacce connesse a quella rete fisica e ogni nodo passa tale richiesta al livello 3, che quindi leggerà il payload del frame, in modo da verificare se l'indirizzo richiesto corrisponde al proprio.

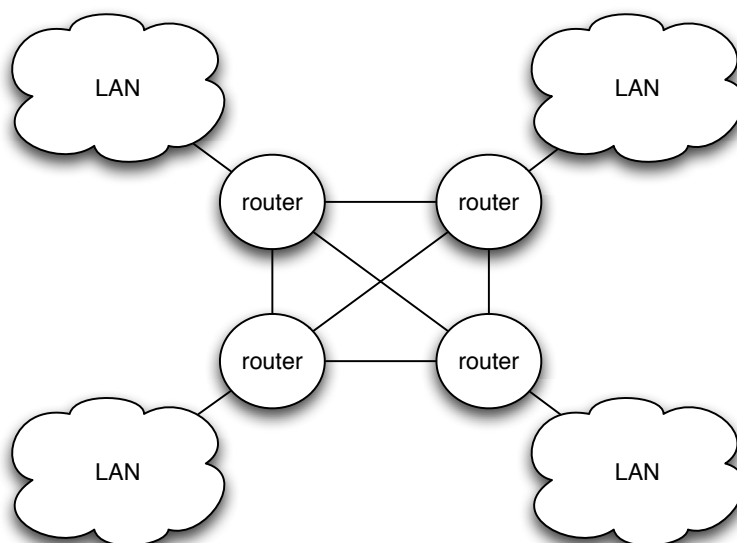


Figura 1.8: Internetworking a livello 3

In questo modo, soltanto il nodo associato all'indirizzo IP richiesto risponderà rivelando il proprio indirizzo Ethernet. Per praticità ogni nodo è in grado di conservare per un certo tempo le corrispondenze tra gli indirizzi di livello 2 e di livello 3, ottenute durante il funzionamento, mediante una tabella residente in memoria denominata *ARP table* o *ARP cache*.

Il livello network è incaricato di muovere i pacchetti dalla sorgente fino alla destinazione finale, attraversando tanti sistemi intermedi (*router*) della dorsale, anche su scala geografica: non a caso, nello stack TCP/IP questo livello prende il nome di *Internet* (Fig. 1.8). Ciò è molto diverso dal compito del livello data link, che è di muovere informazioni solo da un capo all'altro di un singolo canale di comunicazione.

Sintetizzando, il livello network si occupa di:

- gestire lo spazio di indirizzamento del livello 3
- conoscere la topologia della rete;
- scegliere di volta in volta il cammino migliore (*routing*);

- gestire le problematiche derivanti dalla presenza di più reti diverse (*internetworking*).

IP (*Inter-networking Protocol*) [14] è il protocollo di livello 3 della suite TCP/IP, nato per interconnettere reti eterogenee per tecnologia, prestazioni, gestione.

Gli indirizzi IP versione 4, cioè quelli tradizionali, sono composti da una sequenza di 32 bit, suddivisi convenzionalmente in quattro gruppetti di 8 bit, rappresentati in modo decimale e separati da un punto. Per esempio, l'indirizzo IP del computer (host) in cui risiede il file che contiene queste parole è il seguente:

82.55.113.23

ed è stato assegnato alla mia scheda ADSL dal provider di telecomunicazioni al quale sono connesso. Questo tipo di rappresentazione è definito come notazione decimale puntata. L'esempio seguente corrisponde all'indirizzo 1.2.3.4:

00000001.00000010.00000011.00000100

All'interno di un indirizzo del genere si distinguono due parti: l'indirizzo di rete e l'indirizzo del nodo particolare. Il meccanismo è simile a quello del numero telefonico in cui la prima parte del numero, il prefisso, definisce la zona ovvero il distretto telefonico, mentre il resto identifica l'apparecchio telefonico specifico di quella zona. Come per i numeri telefonici, sulla rete mondiale l'indirizzo IP di ogni singolo host non può essere duplicato, cioè non possono esistere due apparati di rete con lo stesso indirizzo; per questo motivo esistono delle organizzazioni a livello mondiale (INTERNIC, RIPE) che si occupano di rilasciare gli IP ai provider che ne fanno richiesta. In pratica viene rilasciato un indirizzo di rete in funzione del numero di nodi da connettere. In questo indirizzo una certa quantità di bit nella parte finale sono azzerati: ciò significa che quella parte finale può essere utilizzata per gli indirizzi specifici dei nodi.

Considerando l'esempio precedente, un possibile indirizzo di rete potrebbe essere 1.2.3.0, cioè:

00000001.00000010.00000011.00000000

In tal caso, si potrebbero utilizzare gli ultimi 8 bit (quindi  $2^8 = 256$  indirizzi) per i vari nodi. Ma l'indirizzo di rete non può identificare un nodo in particolare, quindi il numero di indirizzi possibili per gli host diventa 255.

Inoltre, un indirizzo in cui i bit finali lasciati per identificare i nodi siano tutti a uno, identifica, per convenzione del protocollo, un indirizzo broadcast, cioè un indirizzo per la trasmissione a tutti i nodi di quella rete. Nell'esempio precedente, 1.2.3.255

00000001.00000010.00000011.11111111

rappresenta simultaneamente tutti gli indirizzi che iniziano con

00000001.00000010.00000011

cioè che hanno lo stesso prefisso di rete.

In pratica, il livello 3 di tutti gli host della sottorete valuterà un pacchetto che ha come destinazione l'indirizzo di broadcast e passerà il payload di quel pacchetto al livello 4. Di conseguenza, un indirizzo broadcast non può essere utilizzato per identificare un singolo nodo ed il numero di indirizzi possibili per gli host dell'esempio scende a 254.

Il meccanismo utilizzato per distinguere la parte dell'indirizzo che identifica la rete è quello della maschera di rete o *netmask*. La maschera di rete è un numero di 32 bit, che viene abbinato all'indirizzo IP con l'operatore booleano AND<sup>4</sup>. La netmask sarà dunque costituita da tanti uno quanti sono i bit che si vuole dedicare alla parte di rete e da tutti zero per la parte host. Nell'esempio precedente, nel quale si sono usati 24 bit per la parte di rete, la netmask sarà:

11111111.11111111.11111111.00000000

---

<sup>4</sup>l'operatore AND fornisce in uscita 1 solo se i due valori in ingresso sono entrambi 1

cioè, in notazione decimale, 255.255.255.0 .

Il procedimento è il seguente: si definisce la netmask e la si applica all'indirizzo ip, il numero che si ottiene è l'indirizzo della rete: questa operazione non è soltanto una mera speculazione accademica, ma viene utilizzata in pratica dai router per calcolare la sottorete di destinazione dei singoli pacchetti al fine di instradare il traffico nel modo corretto. Nell'esempio precedente si ha:

```
00000001.00000010.00000011.00000100  1.2.3.4      (host)
11111111.11111111.11111111.00000000  255.255.255.0 (mask)
00000001.00000010.00000011.00000000  1.2.3.0      (net)
```

In base al valore dei primi bit, gli indirizzi IP vengono suddivisi in classi, ciascuna con una netmask convenzionale, per esempio la classe A è costituita da indirizzi il cui primo bit vale 0 e ha una netmask convenzionale di 8 bit, cioè 255.0.0.0. In tabella 1.1 è riportata la classificazione completa.

Classe	Leading bits	Inizio intervallo	Fine intervallo
A	0	0.0.0.0	127.255.255.255
B	10	128.0.0.0	191.255.255.255
C	110	192.0.0.0	223.255.255.255
D	1110	224.0.0.0	239.255.255.255
E		240.0.0.0	255.255.255.255

Tabella 1.1: Classi d'indirizzamento IP

Si può notare che l'esempio scelto (net 1.2.3.0 con 24 bit di parte di rete) non è conforme alla tabella, infatti l'indirizzo 1.2.3.0, convertito in cifre binarie, inizia con uno 0 e quindi appartiene alla classe A, che ha 8 bit dedicati alla rete. Non è un errore: estendendo la netmask si può suddividere una rete in sottoreti più piccole e ciò è molto utile, perché il numero di IP diversi ottenibili con 32 bit è grande, ma finito ( $2^{32}$ ) e non avrebbe senso assegnare, per esempio, una intera classe C (254 indirizzi utili) ad una rete

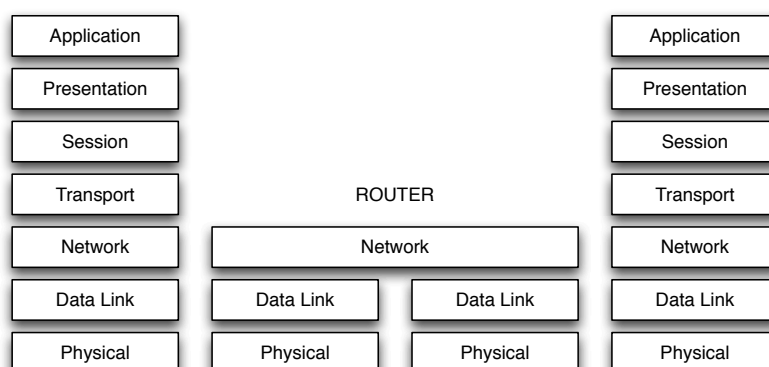


Figura 1.9: Instradamento a livello 3 mediante un router

di un piccolo ufficio con 4 computer<sup>5</sup>. L'operazione appena descritta prende il nome di *subnetting* e, purtroppo, non è supportata da tutti gli standard e protocolli. Gli ambienti in cui è possibile gestire il subnetting vengono definiti *classless*, mentre i contesti nei quali non si può derogare dalla rigida divisione in classi prendono il nome di *classfull*. In ambiente classless si può generalizzare la definizione di indirizzo di net e di broadcast di una sottorete: l'indirizzo della net sarà quello con tutti i bit della parte host posti a 0, mentre il broadcast avrà tutti i bit della parte host posti ad 1.

Un pacchetto IP avrà un header che conterrà l'indirizzo sorgente e l'indirizzo di destinazione, oltre a vari campi di controllo sui quali non ci soffermeremo. L'header sarà analizzato dal livello 3 dell'host: se il pacchetto ha un indirizzo di destinazione uguale quello del nodo che lo sta esaminando (oppure la destinazione è l'indirizzo di broadcast della sottorete) il payload del pacchetto verrà consegnato al livello 4 dell'host; in tutti gli altri casi il pacchetto sarà scartato. Per interconnettere due (o più) reti, intervenendo al terzo livello del modello ISO-OSI, è necessario un *router*; esso è in grado di *instradare* i pacchetti IP indipendentemente dal tipo di reti fisiche connesse effettivamente (Fig. 1.9).

<sup>5</sup>con una parte di rete di 29 bit si ottengono  $2^3 - 2 = 6$  indirizzi utili, resta al lettore l'incombenza di calcolare il valore della netmask opportuna



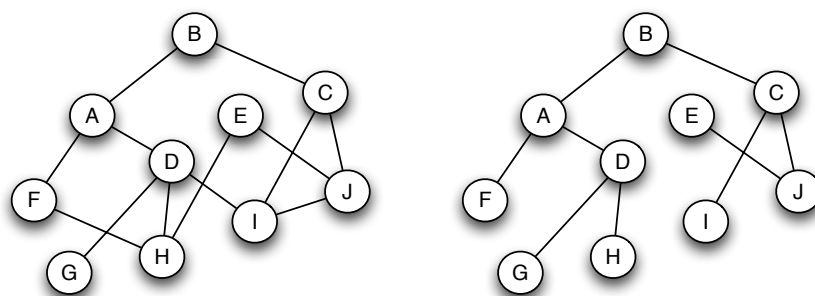


Figura 1.10: Costruzione del Sink Tree

L'instradamento dei pacchetti attraverso le reti connesse al router avviene in base a una tabella di instradamento che può anche essere determinata in modo dinamico, in presenza di connessioni ridondanti; questo procedimento prende il nome di *routing*. Un *algoritmo di routing* è quella parte del software di livello network che decide su quale linea d'uscita instradare un pacchetto che è arrivato al router. Da un algoritmo di routing desideriamo:

- correttezza (deve muovere il pacchetto nella giusta direzione);
- semplicità (l'implementazione non deve essere troppo complicata);
- robustezza (deve funzionare anche in caso di cadute di linee e/o guasti dei router e di riconfigurazioni della topologia);
- stabilità (deve convergere, e possibilmente in fretta);
- ottimalità (deve scegliere la soluzione globalmente migliore).

Esiste un cosiddetto principio di ottimalità per cui se il router  $j$  è nel cammino migliore fra  $i$  e  $k$ , allora anche il cammino ottimo fra  $j$  e  $k$  è sulla stessa strada. Se così non fosse, ci sarebbe un altro cammino fra  $j$  e  $k$  migliore di quello che è parte del cammino ottimo fra  $i$  e  $k$ , ma allora ci sarebbe anche un cammino migliore fra  $i$  e  $k$ . Una diretta conseguenza è che l'insieme dei cammini ottimi da tutti i router a uno specifico router di destinazione costituisce un albero, detto *sink tree* per quel router (Fig. 1.10).

```

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 172.31.255.253 to network 0.0.0.0

C   192.168.101.0/24 is directly connected, Vlan1
C   192.167.101.0/24 is directly connected, Vlan1
O   192.167.106.0/24 [110/5] via 172.31.200.22, 00:49:56, Vlan902
    192.167.107.0/24 is variably subnetted, 3 subnets, 3 masks
O   192.167.107.0/27 [110/6] via 172.31.200.86, 00:49:56, Vlan910
O   192.167.107.64/26 [110/18] via 172.31.200.86, 00:49:56, Vlan910
O   192.167.107.128/25 [110/5] via 172.31.200.38, 00:49:56, Vlan904
O   192.167.105.0/24 [110/3] via 172.31.200.70, 00:49:56, Vlan908
O   192.167.110.0/24 [110/5] via 172.31.200.13, 00:49:56, Vlan901
.....

```

Figura 1.11: Esempio di tabella di routing

In sostanza, gli algoritmi di routing calcolano i sink tree relativi a tutti i possibili router di destinazione, e quindi instradano i pacchetti esclusivamente lungo tali alberi. Ciascun router della rete conserverà una tabella di routing, generata a partire dal sink tree, che conterrà, per ogni sottorete di destinazione conosciuta dal router, l'indicazione della "rotta" (*route*) da far seguire ai pacchetti, cioè l'interfaccia tramite la quale instradare il traffico o l'IP del *next hop* (il salto successivo) sul percorso per raggiungere la sottorete di destinazione (Fig. 1.11). Eventualmente, se il protocollo è evoluto, la tabella di routing conterrà altre informazioni, come il "peso" della rotta (metrica), il protocollo di routing che l'ha comunicata al router, una marca temporale, etc.

Un'ultima osservazione: un router, specie se realizza un nodo di Internet, non può conoscere direttamente rotte per ogni destinazione possibile, perché la tabella di routing dovrebbe avere dimensioni (e quindi un'occupazione di RAM) enormi ed i tempi di aggiornamento sarebbero improponibili.

Il problema è di facile soluzione: alle rotte verso specifiche destinazioni si aggiunge anche la rotta verso la sottorete 0.0.0.0, che, convenzionalmente, indica la destinazione “sconosciuta”. Questa particolare rotta viene indicata come “*default route*” o “*last resort*”: attraverso di essa il router instraderà tutto il traffico diretto a destinazioni non esplicitamente presenti nella tabella di routing.

Quando la rete cresce fino contenere decine di migliaia di nodi, diventa troppo gravoso mantenere in ogni router la completa topologia. Il routing va quindi impostato in modo gerarchico, come succede nei sistemi telefonici. La rete viene divisa in zone (spesso dette regioni): all’interno di una regione vale quanto visto finora, cioè i router (detti router interni) sanno come arrivare a tutti gli altri router della regione; viceversa, quando un router interno deve spedire qualcosa a un router di un’altra regione sa soltanto che deve farlo pervenire a un particolare router, detto router di confine (*border router*), che possiederà la rotta per il next hop verso la destinazione.

Il modo più semplice per implementare il routing è stabilire a priori i percorsi ottimali sulla rete e scrivere la tabella di routing direttamente nella configurazione dei router. Questo approccio, di tipo statico, ha il vantaggio della semplicità (non bisogna implementare sui router un software che calcoli le rotte, questo calcolo è già stato fatto dall’amministratore di rete), ma richiede che l’amministratore conosca completamente la topologia della rete, in modo da poter calcolare i percorsi migliori. Inoltre l’amministratore dovrà materialmente scrivere la routing table d’ogni router e cambiarla esplicitamente ogni volta che è apportata una modifica alla topologia (un nuovo nodo, un guasto, la caduta di un circuito, un cambiamento negli indirizzi IP). Uno schema di questo tipo si adatta bene a piccole realtà, collegate a reti più grandi mediante un solo router di frontiera, che avrà anche il ruolo di last resort gateway. Nelle moderne reti si usano algoritmi dinamici, che si adattano automaticamente ai cambiamenti della rete. Questi algoritmi non sono eseguiti solo all’avvio della rete, ma rimangono in esecuzione sui router durante il normale funzionamento e aggiornano le rotte ad intervalli

temporali regolari o a seguito di variazioni di topologia.

Gli algoritmi di routing dinamico sono sostanzialmente di due tipi: *distance vector* e *link state*. Nel primo ogni router mantiene una tabella (*vector*) contenente un elemento per ogni altro router destinazione. Ogni elemento della tabella contiene la “distanza” (numero di hop, ritardo, ecc.) che lo separa dal router in oggetto e la linea in uscita da usare per arrivarci. Il protocollo distance vector più diffuso è il RIP. Per i suoi vicini immediati il router stima direttamente la distanza dei collegamenti corrispondenti, mandando speciali pacchetti ECHO e misurando quanto tempo ci mette la risposta a tornare. A intervalli regolari ogni router manda la sua tabella a tutti i vicini, e riceve quelle dei vicini. Quando un router riceve le nuove informazioni, calcola una nuova tabella scegliendo, fra tutte, la concatenazione migliore con quest’ordine: se stesso → vicino immediato → router remoto di destinazione.

Ovviamente, la migliore è la concatenazione che produce la minore somma di distanza fra il router stesso ed un suo vicino immediato (viene dalla misurazione diretta) e la distanza fra quel vicino immediato ed il router remoto di destinazione (viene dalla tabella ricevuta dal vicino immediato). L’algoritmo distance vector routing funziona piuttosto bene, ma è molto lento nel reagire alle cattive notizie, cioè quando un collegamento va giù. Ciò è legato al fatto che i router non conoscono la topologia della rete e basano le loro scelte solo sulle tabelle che vengono loro fornite dai router adiacenti.

Si è cercato di ovviare con un approccio diverso, che ha dato origine al link state routing. L’idea di base è che ogni router controlla lo stato dei collegamenti fra se stesso e i suoi vicini immediati (misurando il ritardo di ogni linea) e distribuisce tali informazioni a tutti gli altri; sulla base di tali informazioni, ogni router ricostruisce localmente la topologia completa dell’intera rete e calcola il cammino minimo verso tutti gli altri. I passi da seguire sono:

1. scoprire i vicini e identificarli;
2. misurare il costo (ritardo o altro) delle relative linee;

3. costruire un pacchetto con tali informazioni;
4. mandare il pacchetto a tutti gli altri router;
5. dopo aver ricevuto gli analoghi pacchetti che arrivano dagli altri router, costruire la topologia dell'intera rete;
6. calcolare il cammino più breve verso tutti gli altri router
7. redigere la tabella di routing e copiarla nella RAM.

Quando il router si avvia, invia un pacchetto HELLO su tutte le linee in uscita. In risposta riceve dai vicini i loro indirizzi (univoci in tutta la rete). Inviando vari pacchetti ECHO, misurando il tempo di arrivo della risposta (diviso 2) e mediando su vari pacchetti si deriva il ritardo della linea. Si costruisce un pacchetto con identità del mittente, numero di sequenza del pacchetto, età del pacchetto, lista dei vicini con i relativi ritardi. La costruzione e l'invio di tali pacchetti si verifica tipicamente a intervalli regolari o quando accade un evento significativo (es.: una linea va giù o torna su). La distribuzione dei pacchetti è la parte più delicata, perché errori in questa fase possono portare qualche router ad avere idee sbagliate sulla topologia, con conseguenti malfunzionamenti. Combinando tutte le informazioni arrivate, ogni router costruisce il sink tree della subnet e calcola il cammino minimo verso tutti gli altri router (l'algoritmo si chiama SPF, *Shortest Path First* [22]); con queste informazioni costruisce la propria tabella di routing.

Il vantaggio di questo approccio è che, all'occorrenza di una variazione dello stato della rete, è sufficiente che il router che la percepisce direttamente mandi una segnalazione a tutti gli altri, che, in modo autonomo, provvederanno a modificare le proprie tabelle di routing in conseguenza dell'evento segnalato. Il link state routing è molto usato attualmente su reti di grandi dimensioni: esempi di protocolli che implementano algoritmi di tipo link state sono: OSPF (*Open Shortest Path First*), che è il più usato, ed IS-IS (*Intermediate System-Intermediate System*), progettato per DECnet e poi adottato da OSI. La sua principale caratteristica è di poter gestire indirizzi

di diverse architetture (OSI, IP, IPX) per cui può essere usato in reti miste o multiprotocollo.

### 1.3.2 Comunicazione efficace

Fin qui si sono esaminati livelli, dispositivi e protocolli che, sinergicamente, ci consentono di realizzare la *connettività*, cioè la possibilità di scambiare dati tra due host, indipendentemente dalla loro distanza. Quel che ora occorre è uno strato che simuli la connessione diretta tra le due macchine, un cavo virtuale che nasconda al livello applicativo la complessità della rete fisica sottostante e che si occupi di organizzare e consegnare i dati affinché siano fruibili dalle applicazioni. Il livello di Trasporto fa proprio questo, nello specifico sovrintende alle seguenti operazioni:

- offre servizi ai livelli applicativi;
- controlla la connessione;
- controlla il flusso dei dati;
- riordina le TPDU (Transport Protocol Data Unit).

La PDU di questo livello prende il nome di TPDU (Transport PDU) e l'indirizzamento, nel TCP/IP, è gestito mediante i numeri di porta.

Porta	Protocollo	Applicazione
21	FTP	File transfer
23	Telnet	Remote Login
25	SMTP	e-mail
80	HTTP	World Wide Web
110	POP3	Remote e-mail- access

Tabella 1.2: Indirizzi del livello di trasporto

Come si vede dalla tabella in Fig. 1.2, ciascun numero di porta identifica un servizio applicativo diverso: il *port number* è quindi il tramite con i

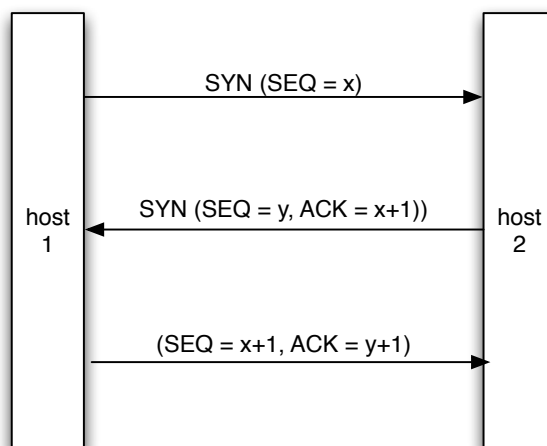


Figura 1.12: Three-way handshake

livelli applicativi superiori. Un servizio applicativo sarà quindi identificato dall'indirizzo IP dell'host che lo eroga e dal numero di porta che identifica il servizio stesso. Questa coppia di valori prende il nome di *socket*.

Un altro obiettivo del livello di trasporto è il riordino dei TPDU: il livello network si occupa, infatti, soltanto della consegna dei pacchetti, ma non garantisce che essi siano consegnati nell'ordine con il quale sono stati trasmessi. Ai fini della corretta ricostruzione dell'informazione è dunque necessario che il livello 4 si occupi di caricare il payload dei pacchetti ricevuti in un buffer e li consegni al livello 5 nell'ordine corretto. A questo livello è di fondamentale importanza la distinzione tra due tipologie di servizi: servizi affidabili orientati alla connessione (tipici di questo livello), servizi *datagram* senza gestione della connessione (poco usati in questo livello).

Il TCP [15] è il protocollo di livello 4 che si occupa di realizzare una connessione nell'ambito della suite TCP/IP. Il protocollo TCP è stato progettato per fornire un flusso di byte affidabile, da sorgente a destinazione, su una rete eterogenea e si occupa di:

- accettare dati dal livello application;

- spezzarli in segment, il nome usato per i TPDU (dimensione massima 64 Kbyte, tipicamente circa 1.500 byte);
- consegnarli al livello network, eventualmente ritrasmettendoli;
- ricevere segmenti dal livello network;
- rimetterli in ordine;
- consegnare i dati, in ordine, al livello application.

Esso realizza la connessione mediante un *three-way handshake*: l'host 1 invia all'host 2 un messaggio contenente la richiesta di iniziare la connessione ed un numero (sequence number) di sequenza; l'host 2 risponde con una "ricevuta di ritorno", contenente, a sua volta, un segnale di inizializzazione (SYN) con un proprio sequence number ed un segnale di acknowledgement (ACK) con il sequence number contenuto nel messaggio dell'host 1, aumentato di una unità. Come si vede dalla figura 1.12, il processo continua con una operazione analoga da parte dell'host 1, finché la connessione è attivata. Durante tutta la connessione i due host si scambieranno periodici messaggi di acknowledgement, in modo da effettuare il controllo della connessione. Il rilascio della connessione avviene mediante un messaggio di conclusione (FIN) al quale segue, normalmente, un messaggio di conferma della chiusura. Per evitare problemi alla disconnessione, spesso si stabilisce un tempo di timeout dopo il quale la connessione viene comunque considerata chiusa.

Il livello transport della suite TCP/IP fornisce anche UDP, un protocollo non connesso e non affidabile, utile per inviare dati senza stabilire connessioni (ad esempio per applicazioni client-server, streaming video). L'header di un segmento UDP è molto semplice e contiene essenzialmente la porta sorgente, la porta destinazione, la lunghezza del segmento ed una checksum, il calcolo della quale può essere disattivato, tipicamente nel caso di traffico in tempo reale (come voce e video) per il quale è in genere più importante mantenere un'elevato tasso di arrivo dei segmenti piuttosto che evitare i rari errori che possono accadere.



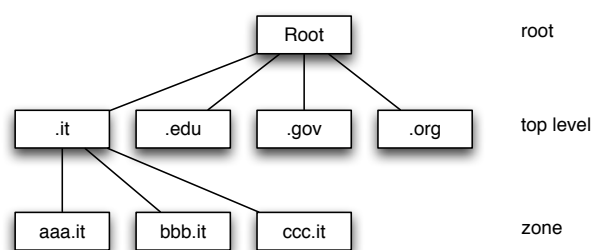


Figura 1.13: Gerarchia DNS

### 1.3.3 Le Applicazioni

In questo modello di architettura, sopra il livello transport c'è il livello application, nel quale viene effettivamente svolto il lavoro utile per l'utente. In questo livello si trovano diverse tipologie di oggetti:

- protocolli di supporto a tutte le applicazioni, come il DNS (*Domain Name System*, RFC 1034 e 1035);
- protocolli di supporto ad applicazioni di tipo standardizzato, come FTP (*File Transfer Protocol*, RFC 959) per il trasferimento di file, SMTP e POP3 (*Simple Mail Transfer Protocol*, RFC 821 e *Post Office Protocol*, RFC 1225) per la posta elettronica, HTTP (*HyperText Transfer Protocol*, RFC 1945) alla base del World Wide Web (WWW);
- applicazioni proprietarie.

## DNS

Il DNS è un servizio di livello applicativo che mette in relazione gli indirizzi IP con delle stringhe di testo strutturate in modo gerarchico: i nomi di dominio (*domain name*), organizzati ad albero in *domini*, *sottodomini* (eventualmente altri sottodomini di livello inferiore...), fino ad arrivare a identificare il nodo desiderato<sup>6</sup> (Fig. 1.13).

<sup>6</sup>www.unime.it è l'host di nome www, appartenente al dominio unime.it

Il servizio, corrispondente alla porta 53 sia TCP (*zone transfer*) che UDP (*request*), è erogato da nodi (*name server*) che si occupano di risolvere i nomi in indirizzi numerici IP e viceversa.

La gerarchia dei server rispecchia abbastanza fedelmente quella dei nomi di dominio, implementando almeno tre livelli: i *root server*, i *top level domain server*, i server di zona; una siffatta struttura permette di delegare al “proprietario” del dominio la gestione dei nomi degli host della propria zona (*hostname*), mantenendo comunque possibile la risoluzione del nome da ogni punto della rete mediante richieste ai nodi superiori dell’albero.

## E-MAIL

La posta elettronica (*e-mail*) è uno dei servizi per i quali è nata la rete; è basata sul paradigma della posta tradizionale ed il suo utilizzo ne ripropone i passaggi, dalla composizione del messaggio, alla spedizione, all’iter di consegna, alla ricezione e lettura. Il servizio è realizzato mediante una serie di protocolli, ciascuno dedicato ad una particolare funzionalità; il cuore del sistema è il protocollo SMTP (*simple mail transfer protocol*, porta 25 TCP), che agisce da *mail transport agent* (MTA) e si occupa di ricevere i messaggi e trasportarli sulla rete fino al server (MTA) del destinatario.

La gestione delle *mailbox* moderne è demandata ai protocolli POP3 (*post office protocol*, porta 110 TCP) ed IMAP<sup>7</sup> (*Internet Message Access Protocol*, porta 143 TCP), che agiscono da *mail user agent*, consentendo all’utente finale di comporre, leggere, eliminare e spedire messaggi (via MTA).

Il singolo messaggio di posta elettronica ha un formato piuttosto semplice, che prevede un’intestazione (*header*) ed un corpo (*body*), separati da una linea vuota. Le linee che compongono l’header sono codificate in modo stringente, perché contengono i campi necessari al corretto instradamento del messaggio:

- *To*: indirizzo e-mail di uno o più destinatari, nella forma, a tutti nota, user@dominio.

---

<sup>7</sup>la gestione e-mail via web, alla quale siamo abituati, è realizzata tramite questo protocollo

- *From*: indirizzo e-mail del mittente
- *Cc*: indirizzo di uno o più destinatari per conoscenza;
- *Bcc*: come *Cc*, ma gli indirizzi non sono visibili al destinatario
- *Subject*: Argomento del messaggio

ed altri campi di gestione del messaggio, la cui descrizione non è qui essenziale.

Per ragioni storiche e di compatibilità, il formato previsto per i messaggi di posta elettronica è di “solo testo”, costituito dai caratteri che compongono il codice ASCII; ciò renderebbe impossibile inviare allegati (immagini, filmati, documenti), in genere codificati in formato binario. Per superare questa limitazione viene utilizzato lo standard MIME (*Multipurpose Internet Mail Extension*, RFC 1341 e 1521), che si occupa di codificare (*encode*), allegare (*attach*) e decodificare (*decode*) gli allegati.

## WORLD WIDE WEB

Il servizio, nato nel 1989 al CERN di Ginevra [17], è erogato dal protocollo HTTP (*hypertext transfer protocol*, porta 80 TCP) e prevede un’architettura basata su client (i *browser*) che interrogano *pagine*, scritte in linguaggio HTML, rese disponibili da server (*siti*) distribuiti su tutta la rete mondiale. L’architettura prevede uno spazio d’indirizzamento basato su URL (*Uniform Resource Locator*), sequenze di caratteri che identificano univocamente una risorsa, nella forma:

```
protocol://<user:pass@>host[:port]></path><?query>
```

In tabella 1.3 sono riportati alcuni esempi.

Gli unici elementi indispensabili sono il protocollo e l’*hostname* del server da raggiungere; i campi opzionali *user* e *password* servono all’autenticazione per la consultazione di risorse private, il campo *port* è la porta del TCP (eventualmente diversa dalla 80 standard), il *path* è il percorso diretto tramite il quale è possibile raggiungere una risorsa senza passare dalla pagina iniziale

Nome	Uso	Esempio
http	Hypertext (html)	http://www.unime.it
ftp	FTP	ftp://files.unime.it
gopher	Gopher	gopher://gopher.unime.it/libs
mailto	e-mail	mailto:user@unime.it

Tabella 1.3: Esempi di URL

(*home page*) e, infine, il campo *query* è utile per passare parametri a pagine dinamiche, costruite a richiesta, mediante linguaggi di programmazione (per esempio il Python), basandosi sui dati presenti in un datatabase.

Questa forma di gestione degli indirizzi è molto potente e permette di collegare contenuti presenti su pagine e server diversi mediante *hyperlink*, basati sulla URL della risorsa da referenziare; ciò consente la “navigazione Internet” (*browsing*) con le modalità da tutti sperimentate nella vita quotidiana.

## SEARCH ENGINES

Fin dagli inizi del Web un utente si trova più frequentemente a voler cercare informazioni di cui ignora la provenienza, piuttosto che accedere ad una pagina di cui possiede già la URI. Quest’ultimo è tutt’altro che un evento raro, ancora oggi la navigazione comprende sempre accedere a pagine usuali, in cui si ha interesse a verificare l’aggiornamento delle informazioni contenute, si pensi al sito di una Facoltà universitaria o ancor più alle versioni on-line di quotidiani. Purtroppo il primo caso, in cui si ha in mente il genere di conoscenze desiderate ma non la loro fonte, perlomeno non sotto forma di URI, è di importanza prevalente nella fruizione della rete.

Una maniera inizialmente pratica di affrontare il sistema fu quella degli elenchi di risorse, pagine Web di per se scarse di contenuto, ma ricche di link ad altre pagine, selezionate manualmente ed eventualmente corredate singolarmente di un breve commento. è una soluzione che richiede un grande sforzo umano, ed è impiegata esclusivamente per settori ristretti, dove an-

cora oggi costituisce la forma più precisa e preziosa di ausilio alla ricerca. È evidentemente impraticabile come modo generalizzato di cercare nel Web. Il metodo che si è andato invece affermando è quello dei cosiddetti motori di ricerca, *search engine*, sistemi che l'utente interroga mediante una serie di parole, che reputa significative dei contenuti che cerca, e restituiscono elenchi di link a pagine selezionate automaticamente, sulla base di una semplice verifica del contenere quelle parole (dette *keywords*, parole chiave). Attualmente esiste un migliaio circa di diversi motori di ricerca, ma in questa sezione si prenderà in esame solo quello che da alcuni anni si è andato affermando come il più popolare dei motori di ricerca: Google. La sua fama è ben motivata risultando effettivamente il più efficiente sulla base di valutazioni oggettive [23].

Ci sono comunque degli elementi del sistema Google che sono comuni alla maggior parte degli altri motori di ricerca. Ciò che l'utente vede abitualmente, la pagina con la form per scrivere parole, è l'interfaccia del componente *searcher* del sistema, quello che effettua le ricerche per gli utenti. In realtà questo sottosistema non cerca proprio nulla su internet, ma funziona totalmente su database interni al motore di ricerca, generati dagli altri componenti. L'insieme degli archivi interni, denominato semplicemente *index*, è costruito dal software che si chiama appunto *indexer*, il quale pure lavora completamente off-line, su un altro enorme database, il *repository*, dove sono immagazzinate in formati compressi tutte le pagine Web conosciute dal motore di ricerca. Infine il *repository* è costruito dal *crawler*, il programma che effettivamente recupera copie delle pagine da internet.

Quest'ultima operazione è di gran lunga la più lenta, perché richiede l'effettiva navigazione attraverso la rete. È effettuata da tante copie dello stesso programma che lavorano in parallelo su computer distribuiti, raggiungendo velocità dell'ordine di migliaia di documenti scaricati al secondo. Pur con questo ritmo, se si tiene conto che attualmente il volume di documenti controllati da Google è di diversi miliardi, il tempo di attraversamento dell'intera rete è di decine di giorni. Mediamente uno stesso sito è rivisitato dai crawler

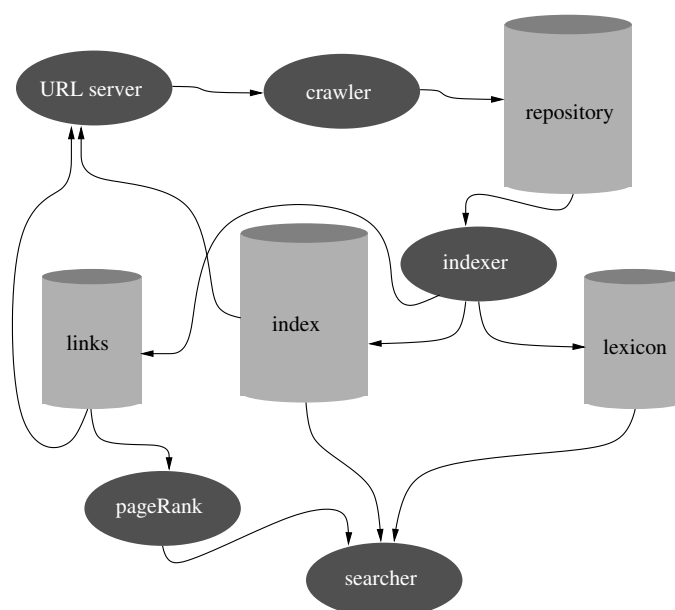


Figura 1.14: Uno schema generale dell'architettura di Google, o di un motore di ricerca analogo.

circa una volta al mese.

L'indexer passa in rassegna tutte le pagine archiviate nel repository e ne estrae diverse caratteristiche importanti per la ricerca. Anzitutto produce le liste dei cosiddetti *hits*, che sono parole con abbinato il numero di volte in cui compaiono nel documento, e la loro posizione. Inoltre riserva un trattamento particolare al testo contenuto nei tag **A**, che viene immagazzinato come parziale descrizione della pagina cui rimanda il link, e naturalmente viene inoltre archiviata la URI del link stesso. Queste informazioni sono quelle che poi, processate dal *URL Server*, andranno ad informare il crawler su dove reperire da internet le pagine. Ma un uso particolare in Google è quello che verrà spiegato al paragrafo successivo. Uno schema complessivo del sistema è in Fig. 1.14.

Il searcher tipicamente non effettua nulla di complesso, ma semplicemente cerca nell'index tutte le pagine che contengono le parole scritte dall'utente. Vi sono pochissimi accorgimenti opzionali che possono raffinare la ricerca, del tipo di imporre che le parole siano consecutive. Il risultato è che abitualmente

una ricerca sia esaurita da diverse migliaia di documenti. Aggiungere parole chiave non sempre migliora la situazione, e facilmente si passa da un numero molto elevato a nessun documento, quando le parole sono evidentemente poco compatibili. La strategia che può cambiare radicalmente la soddisfazione nella ricerca di conoscenze è nell'assegnare in qualche modo un punteggio alle pagine trovate, e presentarle nell'ordine corrispondente. Siccome chi naviga difficilmente si prende la briga di esaminare il contenuto di più di qualche decina di pagine recuperate, è evidente quanto sia essenziale che compaiano al primo posto quelle, tra molte migliaia, veramente rilevanti.

Le strategie dei primi motori di ricerca erano basate su semplici valutazioni rispetto alle parole cercate: per esempio aveva maggior punteggio una pagina che conteneva un numero maggiore di hit o un rapporto tra hit e numero complessivo di parole vantaggioso. Data l'importanza capitale per siti commerciali di comparire in cima agli elenchi dei motori di ricerca, criteri di questo genere hanno innescato rapidamente distorsioni aberranti, come pagine web ingigantite da parole invisibili (per es. scritte in colore bianco su sfondo bianco), ripetute migliaia di volte. Lo stesso fenomeno si potrebbe ripetere per qualunque criterio estratto dalla pagina stessa, è evidente che un'oggettività non poteva che scaturire da elementi esterni al documento, non controllabili da chi progetta la propria pagina.

Così come l'intero Web è derivato da un primo uso accademico, anche il metodo introdotto da Google, denominato *PageRank* [24], è preso in prestito dal modello di valutazione degli articoli scientifici in sede accademica: tramite le citazioni che riceve l'articolo stesso. Analogamente, il metodo PageRank assegna un punteggio ad ogni pagina Web in base a quante altre pagine hanno link a essa diretti, pesando il contributo di questi link mediante il punteggio di quelle altre pagine. In altre parole, un documento è ritenuto importante se ci sono altri documenti in cui compare come link, ma lo è ancor di più se questi documenti sono importanti.

# Capitolo 2

## Sicurezza

Per lo studio delle problematiche riguardanti la Sicurezza Informatica, la Firma digitale e la Posta Elettronica Certificata lo studente è **invitato ad usare come traccia le slides del corso**, approfondendo i contenuti mediante la lettura dei seguenti documenti:

- AgID - Agenzia per l'Italia Digitale - Firme Elettroniche  
<http://www.agid.gov.it/agenda-digitale/infrastrutture-architetture/firme-elettroniche>
- AgID - Agenzia per l'Italia Digitale - Posta Elettronica Certificata  
<http://www.agid.gov.it/agenda-digitale/infrastrutture-architetture/posta-elettronica-certificata>
- AgID - Agenzia per l'Italia Digitale - SPID  
<http://www.agid.gov.it/agenda-digitale/infrastrutture-architetture/spid>
- AgID - Agenzia per l'Italia Digitale - Cloud Computing  
<http://www.agid.gov.it/agenda-digitale/infrastrutture-architetture/cloud-computing>



# Bibliografia

- [1] Joseph C.R. Licklider. Man-computer symbiosis. *IRE Transactions on Human Factors*, 1:4–11, 1960.
- [2] J. C. R. Licklider. Man-computer partnership. *International Science and Technology*, May 1965.
- [3] J. C. R. Licklider and Robert W. Taylor. The computer as a communication device. *Science and Technology*, April 1968.
- [4] Leonard Kleinrock. *Information Flow in Large Communication Nets*. Proposal for a ph.d. thesis, MIT, May 1961.
- [5] J. C. R. Licklider and Wesley Clark. On-line man-computer communication. In *Proceedings of the May 1-3, 1962, spring joint computer conference San Francisco, California, AIEE-IRE '62 (Spring)*, pages 113–128, New York, NY, USA, May 1962. ACM.
- [6] Leonard Kleinrock. *Communication Nets: Stochastic Message Flow and Design*. McGraw-Hill, New York, 1964.
- [7] Paul Baran. On distributed communications. Memoranda for United States Air Force Project, RAND Corporation, Santa Monica (CA), august 1964.

- 
- [8] D. W. Davies, K. A. Bartlett, R. A. Scantlebury, and P. T. Wilkinson. A digital communication network for computers giving rapid response at remote terminals. In *Proceedings of the first ACM symposium on Operating System Principles*, SOSP '67, pages 2.1–2.17, New York, NY, USA, 1967. ACM.
- [9] Lawrence G. Roberts. Multiple computer networks and intercomputer communication. In *Proceedings of the first ACM symposium on Operating System Principles*, SOSP '67, pages 3.1–3.6, New York, NY, USA, October 1967. ACM.
- [10] S. D. Crocker. Host software, 1969.
- [11] E. Krol. Hitchhikers guide to the internet, 1989.
- [12] S. D. Crocker. New host-host protocol, 1970.
- [13] V. Cerf, Y. Dalal, and C. Sunshine. Specification of internet transmission control program, 1974.
- [14] J. Postel. Internet protocol, 1981.
- [15] J. Postel. Internet message protocol, 1980.
- [16] T. J. Berners-Lee, R. Cailliau, and J.-F. Groff. The World-Wide Web. *Computer Networks and ISDN Systems*, 25:454–459, 1992.
- [17] T. Berners-Lee, R. Fielding, and H. Frystyk. Hypertext transfer protocol – http/1.0, 1996.
- [18] D. DiNucci. Fragmented future. *Print*, 53(4):32, 1999.
- [19] C. ISO. Information technology - open systems interconnection - basic reference model. International Standard 7498, ISO/IEC, United States, November 1994.
- [20] A. G. Malis. Arpanet 1822l host access protocol, 1981.

- 
- [21] J. Postel and J. K. Reynolds. Standard for the transmission of ip datagrams over ieee 802 networks, 1988.
- [22] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, December 1959.
- [23] Yi Shang and Longzhuang Li. Precision evaluation of search engines. *World Wide Web: Internet and Web Information Systems*, 5:159–173, 2002.
- [24] Sergey Brin and Lawrence Page. The anatomy of a large-scale hyper-textual Web search engine. *Computer Networks and ISDN Systems*, 30:107–117, 1998.

