SEZIONE

A Programmazione per la comunicazione in rete in C/C++ e Java

A1

Tecnologie e protocolli delle reti di computer

Internet è un sistema di informazioni globale che:

- *è logicamente interconnesso da uno spazio di indirizzamento unico e globale basato sul protocollo IP (Internet Protocol) o le sue successive estensioni o sviluppi;*
- *è in grado di supportare la comunicazione tramite la suite di protocolli TCP/IP (Transmission Control Protocol e Internet Protocol) o le sue successive estensioni o sviluppi e/o altri protocolli compatibili con IP;*
- *fornisce, utilizza o rende accessibili, sia pubblicamente sia privatamente, servizi di comunicazione di alto livello stratificati e basati sulla correlata infrastruttura qui descritta.*

USA *Federal Network Council*

Questa definizione della rete Internet è stata formulata nell'ottobre del 1995, quando i computer connessi alla «rete delle reti» erano circa dieci milioni; oggi, dopo più di 20 anni e con decine di miliardi di dispositivi connessi alla rete globale (non più solo computer, ma anche tablet, smartphone, smartwatch, ...), è ancora valida.

Il progetto di una rete di comunicazione a distanza tra computer – inizialmente denominata ARPANET – nasce in seno all'agenzia ARPA (*Advanced Research Projects Agency*) del Dipartimento della difesa USA, fondata nel 1958 come risposta americana al successo spaziale sovietico ottenuto con il lancio del primo satellite artificiale, lo Sputnik (FIGURA **1**).

ARPANET si sviluppò inizialmente interconnettendo università e centri di ricerca fino agli anni Ottanta del secolo scorso, quando la sua fusione con altre reti di computer nate nel frattempo sia negli USA sia in altri paesi realizzò di fatto la nascita di Internet, il cui nome – contrazione del termine *internetworking* – sottolinea come all'origine fosse di fatto una «rete di reti» anziché un'unica rete globale.

Le innumerevoli reti LAN (*Local Area Network*) che consentono a dispositivi mobili e computer di comunicare e di condividere risorse hardware e software nell'area limitata di uno o più edifici non sono oggi meno importanti dell'unica rete WAN (*Wide Area Network*) esistente a livello mondiale: praticamente tutte le tecnologie delle reti locali moderne derivano dal progetto *Ethernet* che Robert Metcalfe sviluppò presso il centro di ricerca di Palo Alto della Xerox Corporation negli anni Settanta del secolo scorso (FIGURA **2**).

Internet of Things

Una delle tendenze in atto che cambierà radicalmente l'uso della rete Internet è quella di integrarvi come nodi autonomi sensori (sensori ambientali, sensori di localizzazione di persone, veicoli e oggetti, telecamere, ...) e attuatori (dispositivi di controllo della climatizzazione e dell'illuminazione, elettrodomestici, veicoli, sistemi di controllo dell'accesso a determinate aree, ...). L'enorme quantità di dati che la presenza di sensori automatici renderà disponibile e la possibilità di operare a distanza su oggetti fisici probabilmente rivoluzionerà l'uso della rete da parte di persone, istituzioni e aziende e la crescita di Internet in termini di dispositivi connessi crescerà in modo ancora più esplosivo.

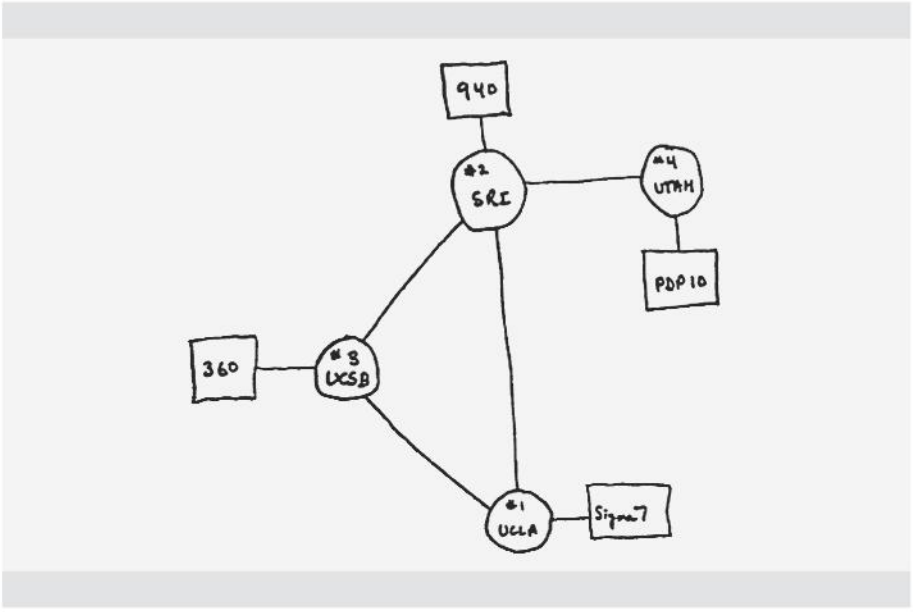


FIGURA **1** Nel dicembre del 1969 ARPANET interconnette tra loro 4 computer dell'Università della California a Los Angeles, dell'Istituto di ricerca dell'Università di Stanford, dell'Università della California a Santa Barbara e dell'Università dello Utah.

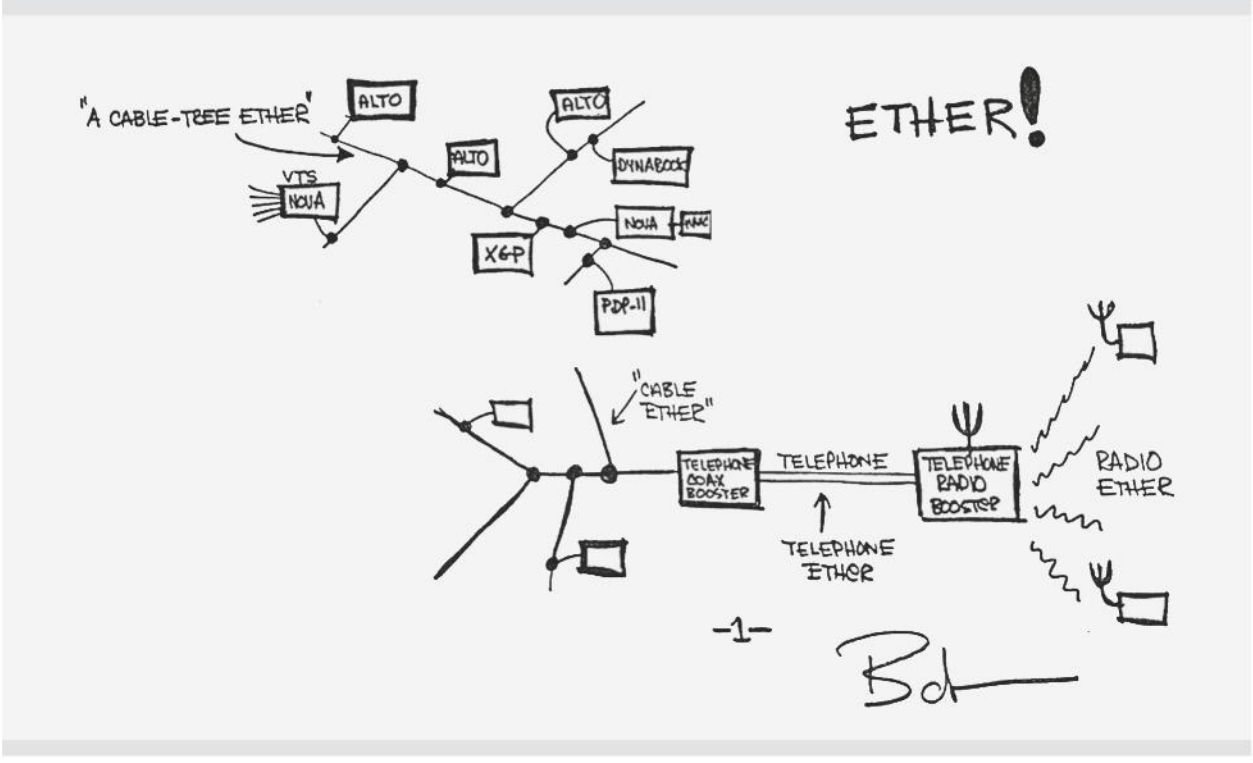


FIGURA **2** Il disegno del 1972 in cui Bob Metcalfe sintetizzò la sua visione della tecnologia *Ethernet* per le reti LAN (sia di tipo *wired*, «cablate», sia di tipo *wireless*, «senza fili»).

La rete *Ethernet* realizzata da Bob Metcalfe prevedeva l'uso di un cavo coassiale denominato *ether* («etere») cui si connettevano «a spina di pesce» tutti i computer che costituivano la rete stessa.

Una **rete locale (LAN)** moderna è realizzata connettendo con una topologia «a stella» i computer che costituiscono la rete a un dispositivo centrale denominato ***switch***; nel caso di **reti locali *wireless* (WLAN)** il dispositivo al centro della stella di comunicazione radio è invece definito ***access-point*** (FIGURA **3**).

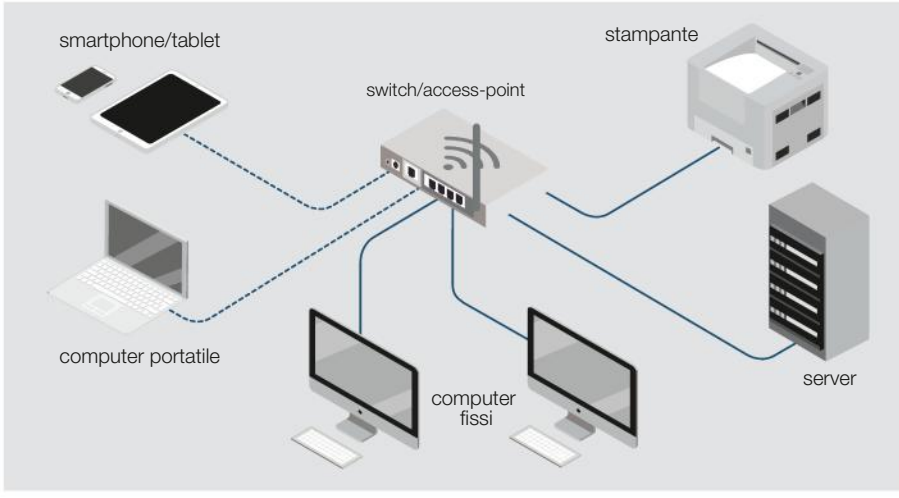


FIGURA **3** Una rete LAN è realizzata da computer e dispositivi interconnessi mediante uno *switch* (nel caso di una rete «cablata») o un *access-point* (nel caso di una rete «senza fili»).

La **rete geografica (WAN) Internet** interconnette tra loro milioni di reti LAN: il dispositivo che consente di collegare una rete LAN alla rete Internet è il ***modem***; i ***router*** hanno invece il compito di smistare il traffico dei dati nella rete (FIGURA **4**).

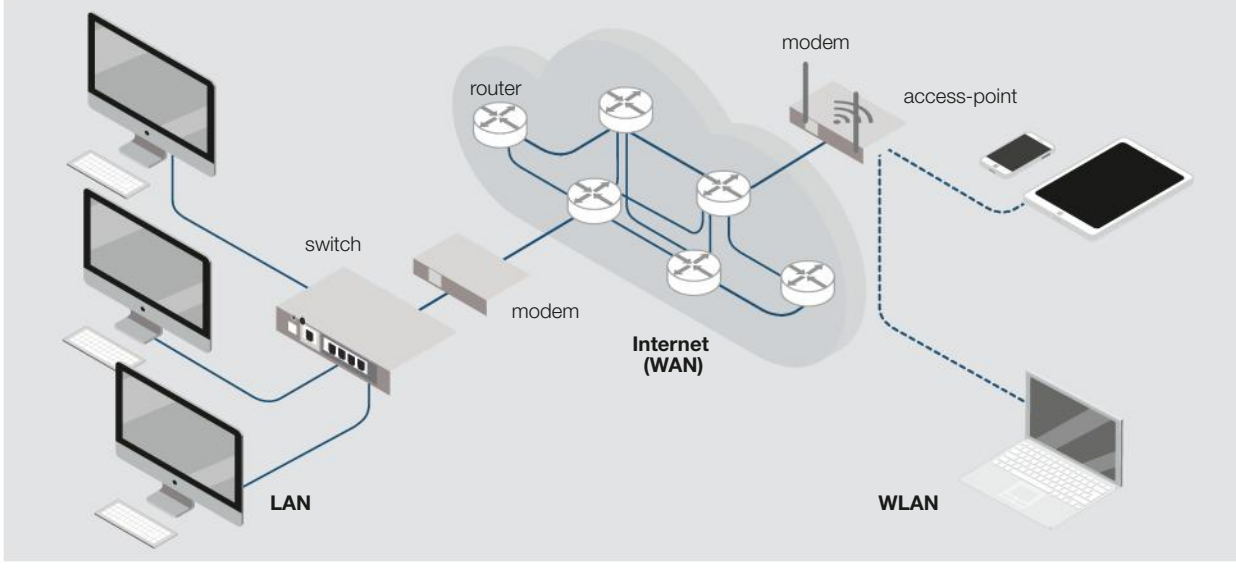


FIGURA **4** Internet è nota come la «rete delle reti» perché è una rete WAN che interconnette milioni di reti LAN domestiche, istituzionali, scolastiche, aziendali; data la sua complessità, Internet viene spesso rappresentata in forma semplificata mediante una nuvola.

Le reti di computer sia di tipo LAN (reti locali) sia WAN (reti geografiche) sono oggi il contesto di riferimento ineludibile per lo sviluppo di applicazioni software e di *APP* per dispositivi mobili.

Il modello di riferimento per l'accesso alle risorse presenti nella rete Internet (come i siti e i servizi web) è quello client/server: un'applicazione software del computer/dispositivo client – per esempio il *browser* o una *APP* – genera richieste al computer server che ospita la risorsa stessa e che trasmette le risposte all'applicazione client (FIGURA **5**).

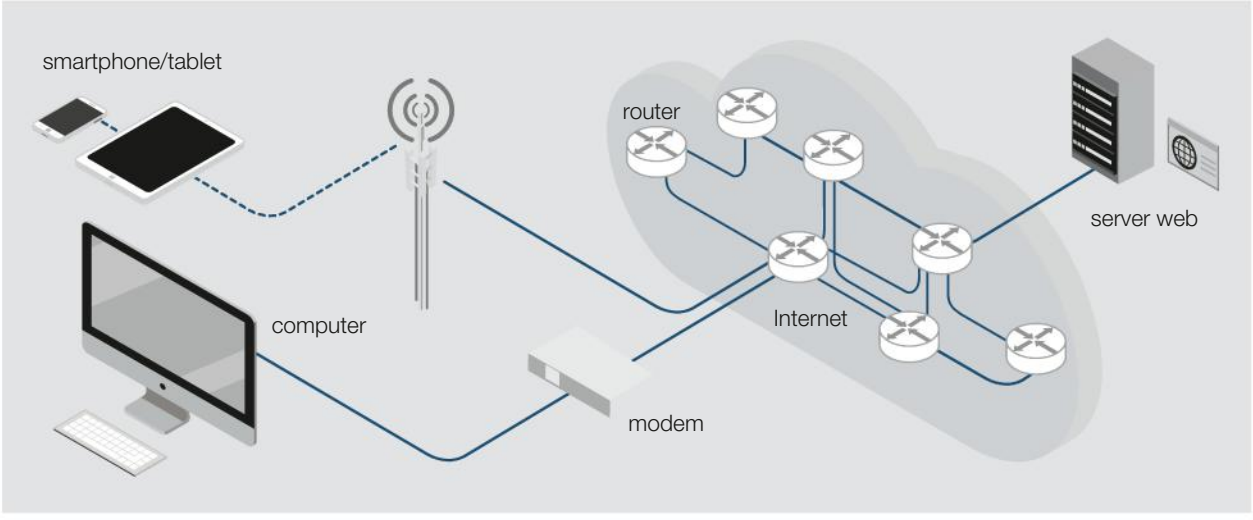


FIGURA **5** Le risorse web presenti nella rete Internet sono ospitate in computer server che rispondono alle richieste generate dalle applicazioni software del dispositivo client.

1 La tecnologia *packet-switching* e la rete Internet

La rete Internet che tutti utilizziamo quotidianamente è basata sulle tecnologie inizialmente sviluppate nei laboratori di ricerca militari e universitari degli USA nel periodo compreso tra la fine degli anni Sessanta e l'inizio degli anni Ottanta del secolo scorso. Quel periodo storico era caratterizzato dalla «guerra fredda» tra Stati Uniti e Unione Sovietica ed era ritenuto possibile che uno di questi due grandi paesi potesse subire da parte dell'altro un massiccio attacco missilistico con testate nucleari: scopo delle prime ricerche relative alla tecnologia *packet-switching*, su cui ancora oggi è basato il funzionamento della rete Internet, era quello di realizzare una rete di comunicazione militare che potesse continuare a funzionare, almeno parzialmente, anche dopo un attacco distruttivo come quello previsto quale inizio di una guerra atomica.

Le caratteristiche della tecnologia *packet-switching* hanno permesso la realizzazione di una rete WAN a cui oggi sono connessi oltre un miliardo di computer e dispositivi e il cui funzionamento non risente dei guasti e dei malfunzionamenti che sono inevitabili in un sistema di comunicazione globale così complesso.

La **tecnologia *packet-switching*** è alla base del funzionamento di tutte le reti di telecomunicazione moderne; essa prevede che un messaggio trasmesso da un dispositivo mittente a un dispositivo destinatario sia suddiviso in «pacchetti» di piccole dimensioni: ogni singolo pacchetto include l'«indirizzo» sia del dispositivo mittente sia del dispositivo destinatario (FIGURA **6**).

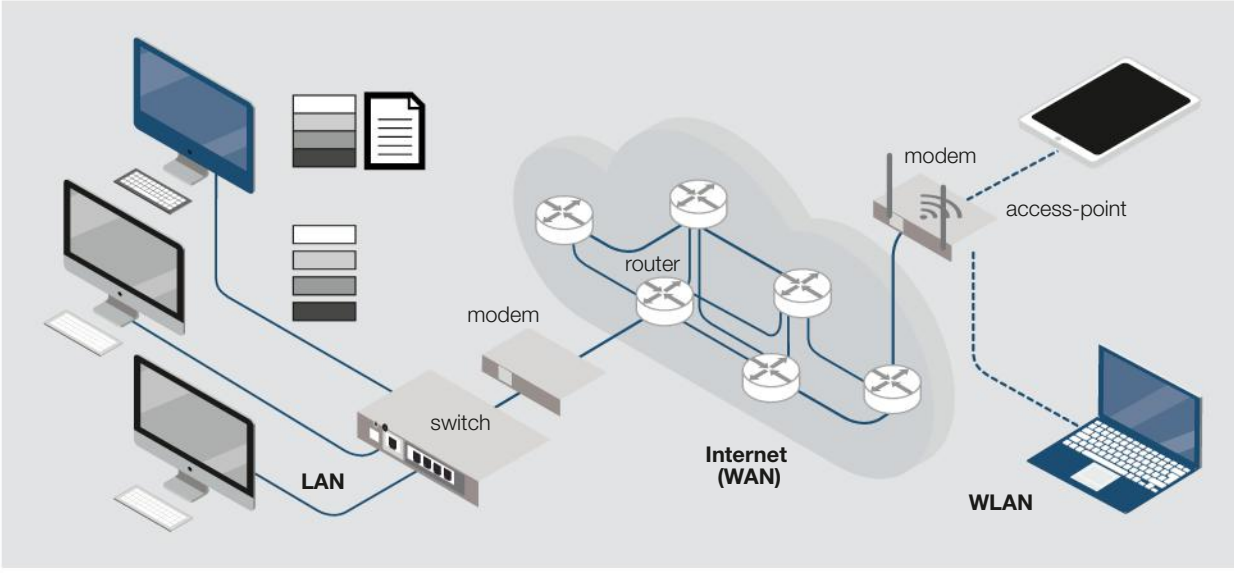


FIGURA **6** Nelle reti *packet-switching* il contenuto del messaggio viene suddiviso in pacchetti trasmessi in sequenza dal computer mittente.

I singoli pacchetti che costituiscono un messaggio possono percorrere cammini diversi nel loro trasferimento dal mittente al destinatario perché i *router* che incontrano nel loro percorso decidono come instradarli sui vari collegamenti in base alle informazioni che ricevono continuamente sullo stato di funzionamento della rete: il dispositivo destinatario ha il compito di «riordinare» eventuali pacchetti del messaggio che riceve in un ordine non corretto (FIGURE **7**, **8**).

La tecnologia *packet-switching* è stata fin dall'inizio concepita con l'idea di essere «robusta» rispetto ai guasti e ai malfunzionamenti degli apparati e dei collegamenti che costituiscono una rete di computer: anche se non frequentemente, un pacchetto che costituisce il frammento di un messaggio può essere perso senza compromettere l'integrità del messaggio ricevuto, ma solo un ritardo nella ricezione.

In questo caso, infatti, il computer destinatario del messaggio rileva l'incompletezza del messaggio ricevuto ed è in grado di richiedere al computer mittente solo il pacchetto, o i pacchetti, mancante/i.

La robustezza che garantisce il corretto trasferimento dei dati attraverso la rete Internet anche nel caso di *router* e collegamenti temporaneamente non funzionanti – situazione che in un sistema costituito da migliaia di componenti tecnologicamente complessi è in pratica continua – è

data dal fatto che il verificarsi di un guasto viene immediatamente rilevato dal *router* direttamente connesso con l'apparato malfunzionante e questa informazione viene trasmessa agli altri *router*, iniziando dai più prossimi, che modificano i criteri con cui instradano i pacchetti che ricevono inoltrandoli su cammini considerati momentaneamente più affidabili o veloci.

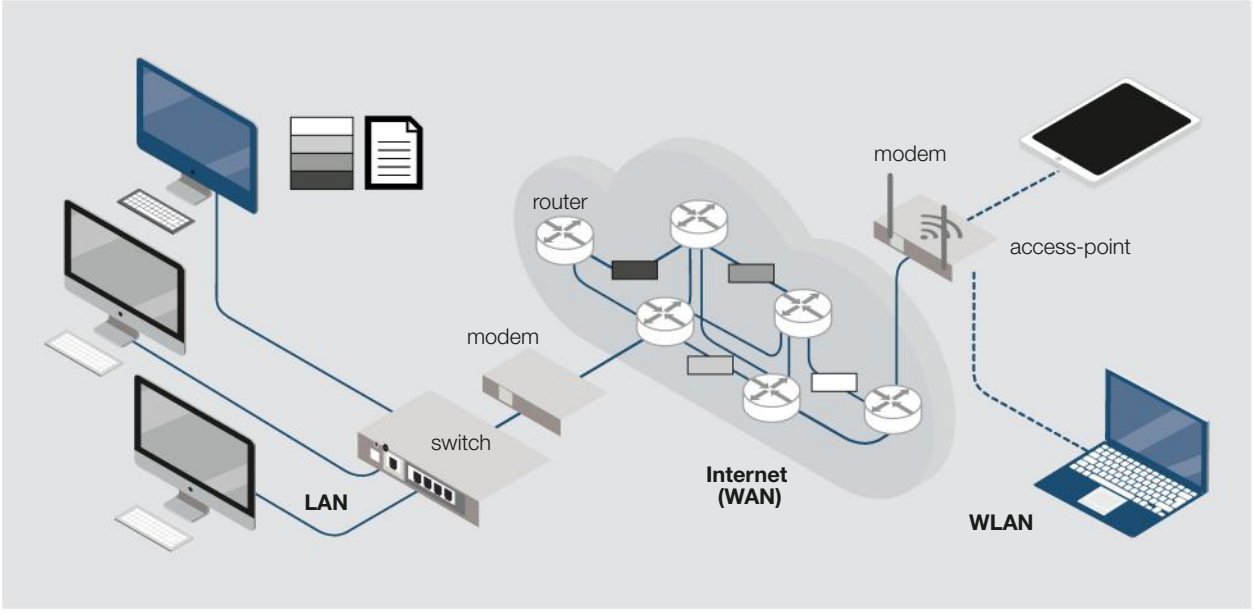


FIGURA **7** I singoli pacchetti possono percorrere cammini diversi nel loro trasferimento attraverso la rete: ogni singolo *router* li instrada sul cammino che, al momento della ricezione, ritiene migliore per inoltrarli verso la destinazione.

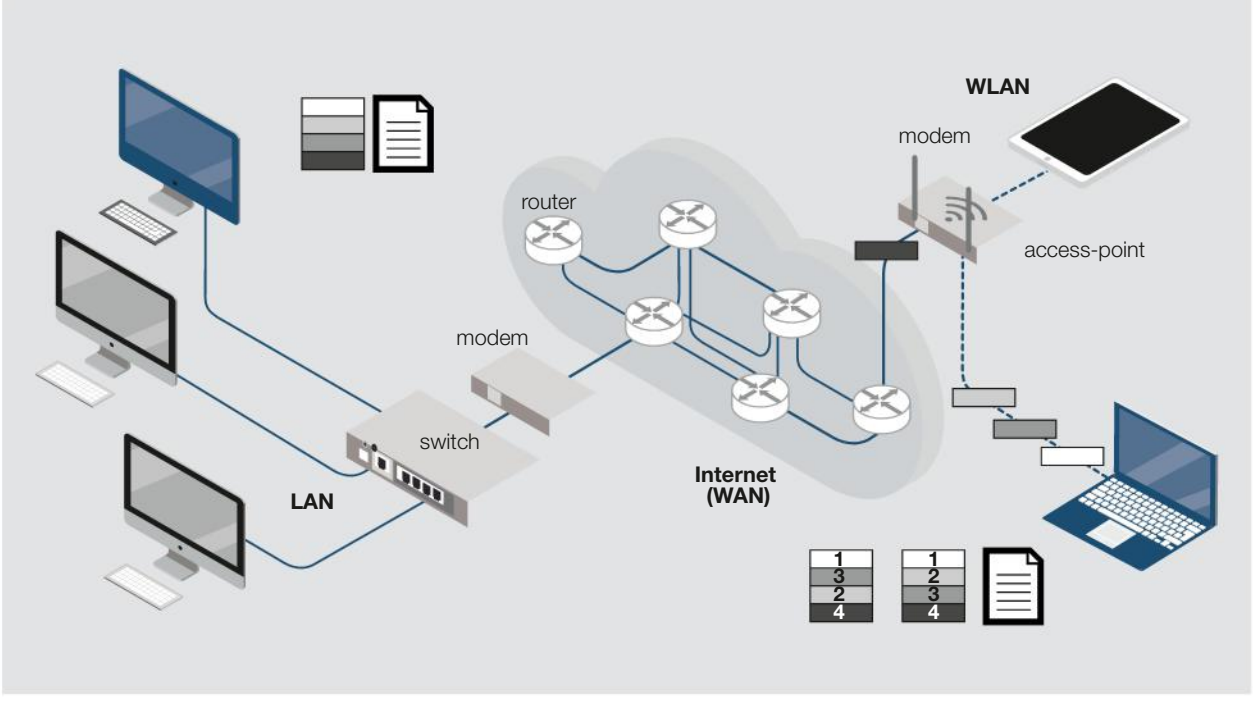


FIGURA **8** Pacchetti che hanno percorso cammini diversi con tempi di trasferimento diversi possono giungere a destinazione ordinati in modo non corretto: è compito del computer di destinazione riordinarli per ricostruire correttamente il messaggio originale.

Questo comportamento, oltre a realizzare una valida gestione dei malfunzionamenti, permette anche di risolvere automaticamente il problema ricorrente della congestione del traffico di dati in una zona di una rete: al verificarsi di ritardi e di perdite di dati i *router* devieranno un numero sempre maggiore di pacchetti verso parti della rete meno congestionate.

ESEMPIO Il malfunzionamento di un *router* comporta la perdita di alcuni pacchetti di un messaggio; il computer destinatario rileva la non integrità del messaggio ricevuto e richiede al computer mittente la ritrasmissione dei soli pacchetti mancanti (FIGURE **9**, **10**, **11**).

L'infrastruttura di telecomunicazione di una rete WAN come Internet è estremamente varia: essa è costituita da cavi sottomarini, canali satellitari, ponti radio terrestri, ... Ciascuno di questi collegamenti consente la trasmissione dei dati da un luogo all'altro della Terra e nei punti di snodo dei milioni di collegamenti della rete Internet sono presenti dei *router* (FIGURA **12**).

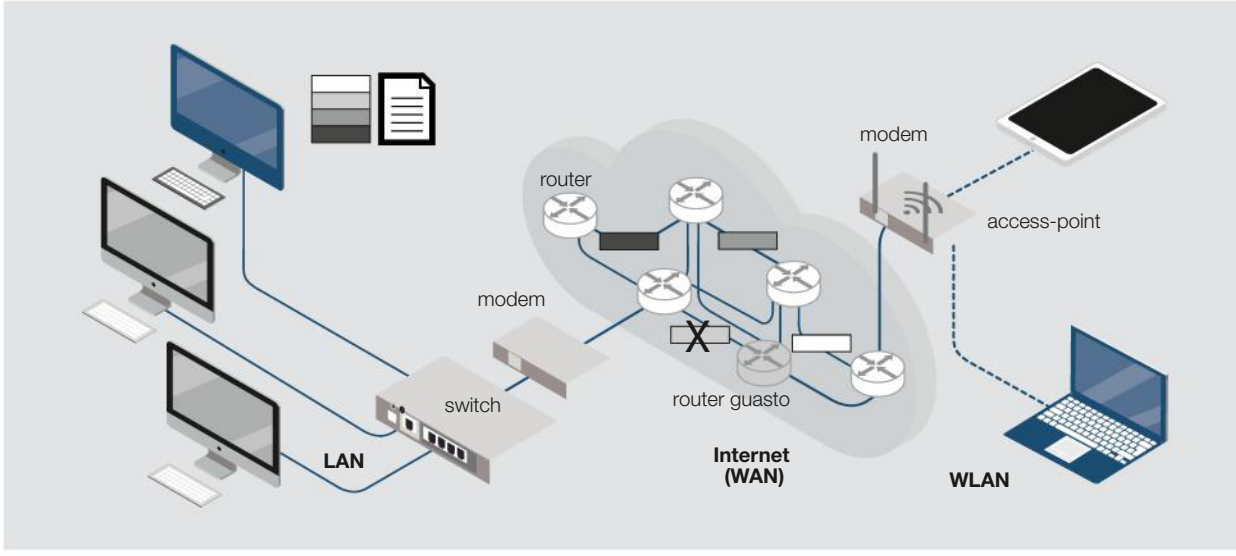


FIGURA **9**

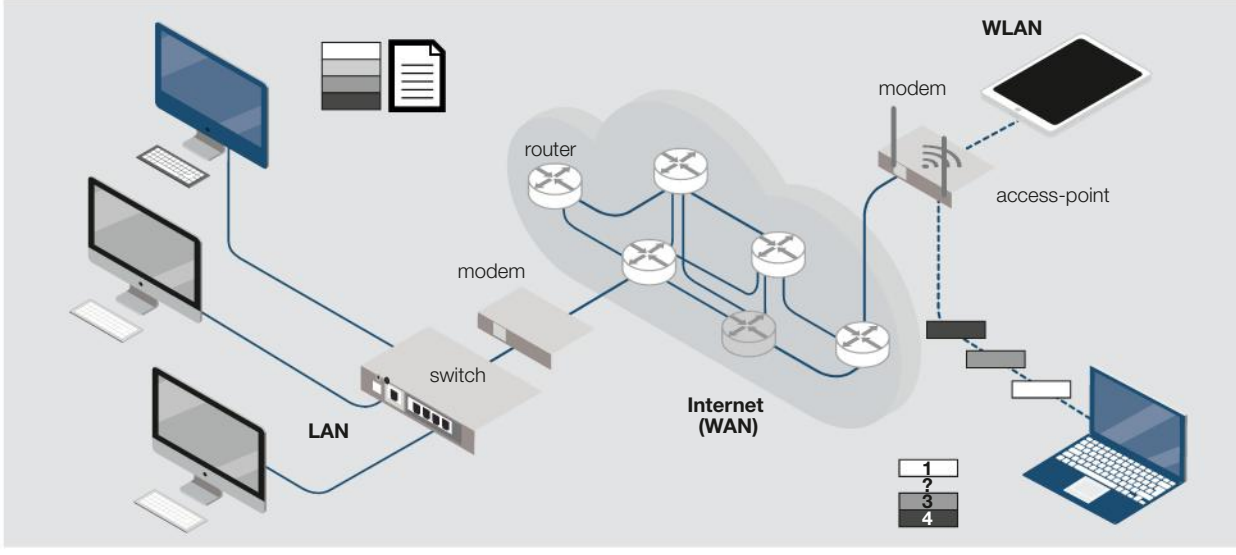


FIGURA **10**

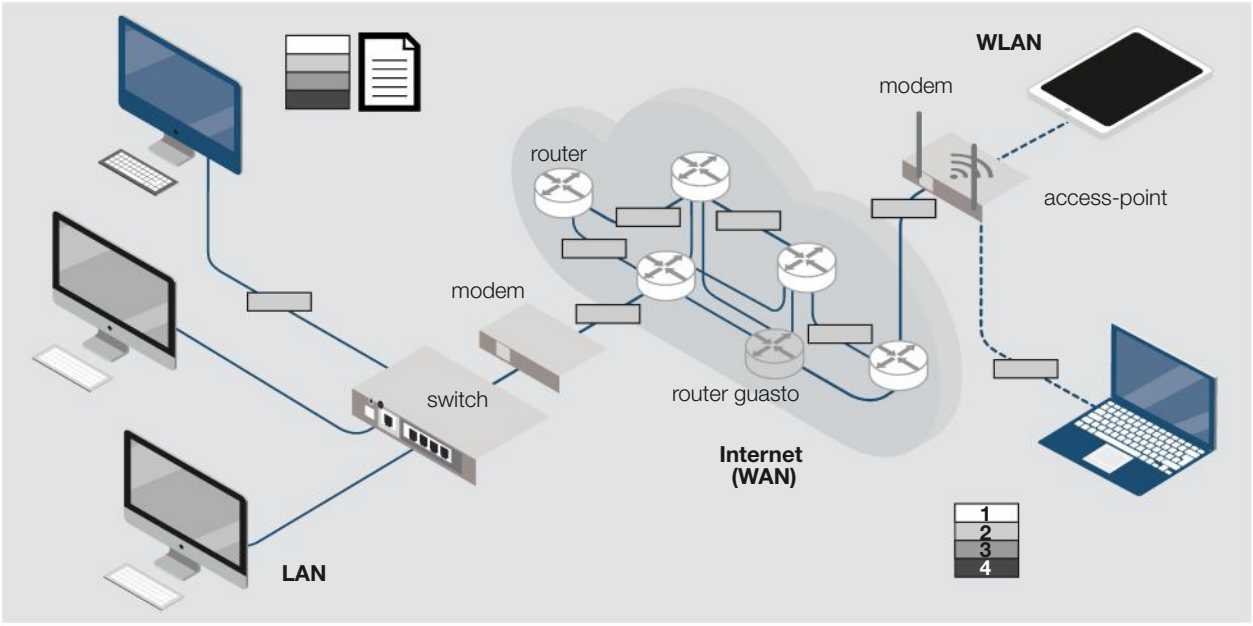


FIGURA **11**

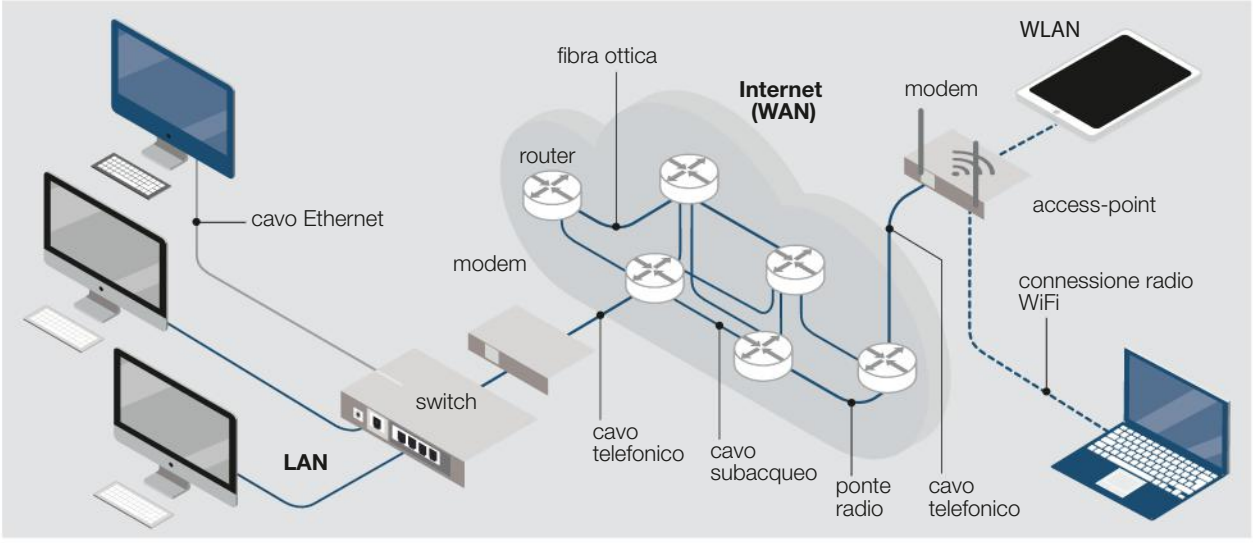


FIGURA **12** Un singolo pacchetto nel suo transito attraverso la rete Internet, dal dispositivo di origine fino al dispositivo destinatario, percorre tratti realizzati con tecnologie hardware di telecomunicazione diverse.

2 Lo standard *Ethernet* per le reti LAN *wired* e *wireless*

Per quanto l'affidabilità delle moderne tecnologie di realizzazione delle reti LAN sia grandemente superiore a quella delle reti WAN, anche esse adottano la modalità di funzionamento *packet-switching*. Le reti LAN moderne sono tutte realizzate in base allo standard *Ethernet* e possono essere di due tipi:

- ***wired*** o **cablata**: utilizzano cavi in rame o fibre ottiche; ogni dispositivo è connesso a uno *switch* e la velocità di trasmissione, come la massima

lunghezza di un collegamento, varia in funzione delle caratteristiche dei cavi e degli apparati di rete;

- ***wireless*** o «**senza fili**»[1]: utilizzano le onde radio nelle bande di frequenza comprese tra 2,4 e 2,5 GHz o tra 5,7 e 5,9 GHz; ogni dispositivo comunica con un *access-point* e la velocità di trasmissione e la distanza raggiungibile dipendono dalle caratteristiche dei dispositivi e degli apparati di rete.

1. Reti di questo tipo sono commercialmente note come WiFi.

OSSERVAZIONE In realtà molte reti LAN sono miste: alla sezione cablata sono connessi – oltre a computer e altri dispositivi – alcuni *access-point* che consentono l'accesso *wireless* realizzando una WLAN (*Wireless* LAN).

Tutte le reti di tipo *Ethernet* cablate condividono alcune caratteristiche tra cui la struttura del pacchetto, denominato ***frame***, che viene trasmesso[2] (TABELLA **1**).

2. La struttura del *frame* di una rete *wireless* è più complessa, ma lo schema di indirizzamento fisico dei dispositivi è lo stesso, permettendo la realizzazione di reti miste *wired*/*wireless*.

TABELLA **1**

Preambolo	SOF	Indirizzo fisico destinazione	Indirizzo fisico mittente	Lunghezza dati	Dati	FCS
7 byte	1 byte	6 byte	6 byte	2 byte	46-1500 byte	4 byte

I primi 8 byte (Preambolo e SOF, *Start Of Frame*) sono una sequenza fissa di 64 bit usati per sincronizzare il dispositivo che riceve il *frame*; gli ultimi 4 byte (FCS, *Frame Check Sequence*) rappresentano un codice a 32 bit calcolato a partire dal contenuto del *frame* stesso che permette al dispositivo che riceve il *frame* di verificare la correttezza dei dati ricevuti rispetto a quelli trasmessi. I dati che un singolo *frame Ethernet* può contenere variano da un minimo di 46 a un massimo di 1500; il campo «Lunghezza dati» del *frame* ne specifica la quantità: nel caso che i dati da trasmettere siano inferiori a 46 byte viene specificato il numero esatto e i byte non necessari fino al 46-esimo sono ugualmente trasmessi, ma con tutti i loro bit impostati al valore «0».

Gli **indirizzi fisici** riportati in un *frame Ethernet* sono sequenze di 48 bit (sono sempre presenti l'indirizzo del dispositivo mittente che genera e trasmette il *frame* e quello del dispositivo di destinazione del *frame*; nel caso che il *frame* sia destinato a tutti i dispositivi della rete esso è costituito da una sequenza di 48 bit impostati al valore «1»): i produttori di dispositivi che si possono connettere a una rete LAN o WLAN predefiniscono in fase di produzione indirizzi univoci per ciascuno di essi.

I dispositivi connessi in una rete LAN o WLAN ricevono in alcuni casi anche *frame* destinati ad altri dispositivi, ma in base al proprio indirizzo fisico selezionano solo i *frame* a loro destinati.

OSSERVAZIONE Gli apparati *switch* che ricevono un *frame* lo ritrasmettono su un collegamento specifico, se è noto che il dispositivo avente l'indirizzo di destinazione è presente su quel tratto della rete; in caso contrario lo trasmettono su tutti i collegamenti. Nel corso del loro fun-

Gli standard IEEE per le reti LAN e WLAN

Le reti *Ethernet* cablate prevedono una varietà di connessioni fisiche (cavi in rame di varia tipologia e fibre ottiche con caratteristiche differenziate), ma sono tutte realizzate in accordo con i vari emendamenti apportati allo standard originale IEEE-802.3 del 1983 (IEEE è l'acronimo di *Institute of Electrical and Electronics Engineers*, un'associazione professionale internazionale che opera nel campo delle tecnologie elettriche, elettromagnetiche ed elettroniche).
Le reti LAN *wireless* prevedono una varietà di frequenze (principalmente 2,4 GHz e 5,7 GHz) e modulazioni radio, ma sono tutte realizzate in accordo con i vari emendamenti apportati allo standard originale IEEE-802.11 del 1999.

zionamento gli apparati *switch* costruiscono una tabella di associazione degli indirizzi dei dispositivi presenti sui tratti della rete corrispondenti ai vari collegamenti ispezionando gli indirizzi mittenti dei *frame* che ricevono da ogni collegamento[3]: questo semplice meccanismo di «autoapprendimento» rende nella maggior parte dei casi lo *switch* un apparato molto efficiente nel filtrare il traffico dei pacchetti di dati di una rete LAN.

3. La topologia «a stella» delle reti *Ethernet* impedisce infatti che un dispositivo possa essere contemporaneamente presente su due distinti tratti di una rete LAN.

L'adozione degli apparati *switch* per la realizzazione delle reti LAN di tipo *Ethernet* ha reso l'evento di «collisione» di due *frame*, che si verifica nel caso di trasmissione simultanea da parte di due diversi dispositivi sullo stesso collegamento[4], un evento estremamente raro. Lo standard *Ethernet* implementa comunque la tecnica **CSMA-CD** (*Carrier Sense Multiple Access-Collision Detection*) che eredita dalle sue origini:

- un dispositivo prima di trasmettere un *frame* verifica che non vi siano trasmissioni già in corso;
- se il collegamento risulta disponibile, il dispositivo inizia la trasmissione del *frame*; dato che più dispositivi, trovando il collegamento disponibile, possono iniziare simultaneamente la trasmissione di un *frame*, la trasmissione viene monitorata allo scopo di rilevare una possibile collisione;
- nel caso che sia rilevata una collisione, il dispositivo interrompe la trasmissione e attende un tempo di durata casuale prima di ritentare una nuova trasmissione del *frame*, in modo da minimizzare la possibilità di una nuova collisione;
- dopo tre tentativi di trasmissione che si risolvono in una collisione la trasmissione del *frame* non viene più ritentata ed è considerata fallita.

4. Una collisione può verificarsi solo su un collegamento mediante cavo in rame: le fibre ottiche sono per loro natura collegamenti «punto-punto» tra due apparati.

OSSERVAZIONE Le reti *wireless* adottano una tecnica **CSMA-CA** (*Carrier Sense Multiple Access-Collision Avoidance*), in cui i dispositivi richiedono esplicitamente all'*access-point* il permesso (RTS, *Request To Send*) di trasmissione e attendono una specifica conferma (CTS, *Clear To Send*) prima di trasmettere un *frame*. In questo modo sono evitate le collisioni.

Lo standard *Ethernet* prevede soluzioni hardware estremamente diversificate ma espone un modello unico di trasmissione da dispositivo a dispositivo di un *frame* nell'ambito di una rete locale.

3 Il modello OSI dell'ISO e lo *stack* di protocolli TCP/IP

La rete Internet è uno dei sistemi tecnologici più complessi mai realizzati dall'uomo, inoltre non esiste un'unica azienda o organizzazione proprietaria di questa rete WAN: apparati e collegamenti appartengono ad aziende di telecomunicazione diverse e il loro funzionamento congiunto dipende dall'e-

sistenza di normative tecniche comuni a cui tutti – produttori di hardware e sviluppatori di software – si attengono: i cosiddetti «protocolli di rete».

Un **protocollo di rete** è un insieme di regole tecniche, rigorosamente definite e formalmente documentate, che consente la progettazione e la realizzazione di apparati di rete (*router*, *switch*, *access-point*, *modem*, schede di rete *wired* e *wireless* per i computer e per altri dispositivi, ...) e di applicazioni software che – come il *browser* – consentono di comunicare con altri computer o dispositivi connessi mediante una rete LAN e/o WAN.

La complessità realizzativa di una rete di computer è da sempre stata affrontata organizzandone in livelli gerarchici – ciascuno dei quali con specifiche funzionalità – l'architettura hardware/software.

Nel 1994 l'Organizzazione Internazionale per la Standardizzazione (ISO, *International Standard Organization*) ha pubblicato lo standard OSI (*Open System Interconnection*) che definisce un modello di generica architettura di rete organizzata in 7 livelli gerarchici (TABELLA **2**).

TABELLA **2**

Livello	Funzione
7. Applicazione	Comunicazione dati tra i processi in esecuzione su due computer della rete.
6. Presentazione	Rappresentazione dei dati in un formato comune tra i processi in esecuzione su due computer della rete.
5. Sessione	Gestione della sessione di comunicazione tra i processi in esecuzione su due computer della rete.
4. Trasporto	Trasferimento di pacchetti (denominati «segmenti» a questo livello) tra i processi in esecuzione su due computer della rete; il livello di trasporto può rendere la comunicazione affidabile gestendo gli errori (pacchetti persi o in sequenza non ordinata) del livello di rete.
3. Rete	Trasferimento dei pacchetti attraverso la rete, dal computer di origine fino al computer di destinazione; a questo livello la trasmissione dei dati non è affidabile (per esempio, in caso di malfunzionamento o di congestione i *router* di una rete WAN scartano i pacchetti che non sono in grado di gestire).
2. *Data-link*	Trasmissione/ricezione di un pacchetto di bit (denominato *frame* a questo livello) tra due dispositivi direttamente connessi (per esempio su una rete LAN).
1. Fisico	Trasmissione/ricezione dei segnali elettrici, ottici o a radiofrequenza che rappresentano i singoli bit; include la definizione delle caratteristiche meccaniche, elettriche, ottiche ed elettromagnetiche dei collegamenti di rete (cavi, prese, modulazioni, ...).

I sistemi operativi rendono disponibili allo sviluppatore API per la gestione della comunicazione di rete che costituiscono l'interfaccia del livello di trasporto: le funzionalità dei livelli di sessione, presentazione e applicazione non risultano in molti casi distinte, essendo implementate dal codice dell'applicazione software. Per questo motivo viene spesso impiegato un modello semplificato avente solo 5 livelli in cui i livelli 5, 6 e 7 del modello OSI sono unificati in un solo livello applicativo:

- Applicazione;

- Trasporto;
- Rete;
- *Data-link*;
- Fisico.

I protocolli di rete sono definiti per ogni livello del modello gerarchico, escluso il livello fisico per il quale sono definite le caratteristiche funzionali e realizzative dei collegamenti e dei segnali; nella TABELLA **3** ne sono riportati alcuni tra i più noti.

TABELLA **3**

Livello	Protocollo
7. Applicazione 6. Presentazione 5. Sessione	**HTTP** (*Hyper-Text Transfer Protocol*) **SMTP** (*Simple Mail Transfer Protocol*) **POP** (*Post Office Protocol*) **FTP** (*File Transfer Protocol*) ...
4. Trasporto	**TCP** (*Transmission Control Protocol*) **UDP** (*User Datagram Protocol*)
3. Rete	**IP** (*Internet Protocol*)
2. *Data-link*	**IEEE-802.3** (*wired Ethernet*) **IEEE-802.11** (*wireless Ethernet*) **PPP** (*Point-to-Point Protocol*) **HDLC** (*High-level Data-Link Control*) ...

OSSERVAZIONE I protocolli del livello *data-link* dipendono dalla tecnologia di telecomunicazione adottata (LAN/WAN, *wired*/*wireless*, ...) e i protocolli del macrolivello applicativo (applicazione, presentazione e sessione) dipendono dal tipo di operazione che l'utente – uomo o programma – richiede di effettuare al software che comunica in rete (accesso a una pagina o servizio web, lettura o inoltro di una e-mail, download o upload di un file, ...). Ma i protocolli dei livelli intermedi di rete e di trasporto sono indipendenti sia dalle tecnologie impiegate, sia dalle operazioni effettuate e i protocolli indicati nella tabella si sono da tempo affermati come gli unici effettivamente utilizzati sia per le reti WAN sia per le reti LAN[5].

I protocolli dei livelli intermedi di trasporto e di rete denominati rispettivamente TCP e IP sono gli stessi per tutte le reti LAN e WAN moderne: sono stati sviluppati nel corso degli anni Settanta del secolo scorso da Vinton Cerf e Robert Kahn presso i laboratori dell'università di Stanford per la realizzazione di Internet e ne sono ancora oggi alla base del funzionamento.
Dal punto di vista tecnologico la rete Internet è infatti fondata sullo *stack* («pila», per sottolineare la struttura gerarchica articolata in livelli) di protocolli TCP/IP.

I tre protocolli hanno funzioni diverse, ma complementari, che sono riepilogate nella TABELLA **4** a pagina seguente.

5. Se si escludono alcuni protocolli di supporto (per esempio il protocollo ICMP, *Internet Control Message Protocol*, a livello di rete), o aventi finalità particolari (per esempio il protocollo SCTP, *Stream Control Transmission Protocol*, a livello di trasporto), i protocolli IP a livello di rete e TCP e UDP a livello di trasporto sono praticamente gli unici effettivamente utilizzati sulle reti LAN *wired*/*wireless* e sulla rete WAN Internet.

TABELLA **4**

Livello	Protocollo/i	Funzioni
Trasporto	**TCP**	Creazione della connessione virtuale tra i due computer che partecipano alla comunicazione; a questo livello avviene la rilevazione di eventuali pacchetti ricevuti fuori ordine, o andati persi. La gestione di queste situazioni realizza una trasmissione dei dati affidabile priva di errori, anche se al livello sottostante sono sporadicamente persi alcuni pacchetti.
	UDP	Il protocollo UDP, al contrario di TCP, non garantisce una trasmissione affidabile dei dati e viene impiegato in situazioni in cui la perdita di pacchetti, o l'arrivo al destinatario in ordine diverso da quello di trasmissione, non costituisce un problema.
Rete	**IP**	IP è l'unico protocollo utilizzato al livello di rete, sia sulle reti LAN sia sulle reti WAN. Si occupa del trasferimento dei pacchetti attraverso la rete, dal computer di origine fino al computer di destinazione; a questo livello la trasmissione dei dati non è affidabile. Al livello di rete ogni computer, dispositivo e apparato è identificato da un indirizzo numerico noto come «indirizzo IP»: questo numero deve essere unico, nella stessa rete LAN o WAN non ne può esistere un altro uguale. Sono oggi utilizzate due diverse versioni del protocollo IP: la versione 4 e la versione 6. Esse possono essere impiegate contemporaneamente sulla stessa rete LANG WAN.

OSSERVAZIONE Nonostante l'esistenza di due protocolli alternativi per il livello di trasporto si è affermato il nome di *stack* TCP/IP che fa riferimento al protocollo di trasporto maggiormente utilizzato dalle applicazioni software.

Ogni protocollo a ogni livello del modello gerarchico aggiunge a ogni singolo «pacchetto» di dati che gestisce un'intestazione (*header*) contenente le informazioni necessarie alla sua gestione (per esempio, a livello di rete, gli indirizzi IP mittente e destinatario del pacchetto).

Il funzionamento congiunto di più protocolli a vari livelli gerarchici è basato sulla tecnica di **incapsulazione** dei singoli pacchetti costituiti ai livelli più alti del modello nei pacchetti dei livelli più bassi come dati.

Lo schema di FIGURA **13** illustra la tecnica di incapsulazione:

- al livello dell'applicazione i dati sono frazionati e ogni frazione viene trasmessa invocando il protocollo TCP;
- al livello di trasporto il protocollo TCP aggiunge la propria intestazione configurando un segmento che trasmette invocando il protocollo IP;
- al livello di rete il protocollo IP aggiunge la propria intestazione configurando un pacchetto che trasmette invocando il protocollo del livello *data-link*;
- trattandosi di una rete LAN al pacchetto IP viene aggiunta l'intestazione del frame *Ethernet* (e in questo caso anche, in coda, il *Frame Control Sequence*) prima della modulazione effettiva dei singoli bit sul mezzo fisico di trasmissione.

La gestione dei protocolli della rete Internet

Pur non esistendo un'organizzazione centralizzata che gestisce la rete Internet, gli standard tecnici necessari al funzionamento e allo sviluppo della rete sono proposti, documentati e formalizzati dalla **IETF** (*Internet Engineering Task Force*). La IETF è un ente internazionale che si occupa di coordinare l'evoluzione dei protocolli di rete esistenti (in particolare TCP/IP) e lo sviluppo di nuovi protocolli.
A testimonianza del ruolo di coordinamento della IETF, i documenti di specifica tecnica degli standard e dei protocolli mantengono ancora oggi la denominazione originale di **RFC**, *Request For Comments* («Richiesta di commenti»).

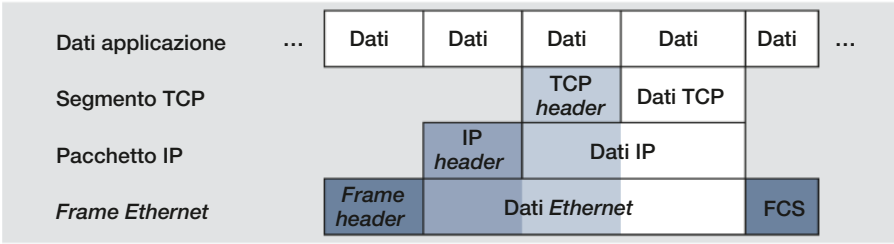


FIGURA **13**

Quando il dispositivo destinatario riceve il *frame* estrae il pacchetto IP, dal quale estrae il segmento TCP, dal quale estrae i dati destinati all'applicazione.

Nel corso del «viaggio» dal dispositivo di origine al dispositivo di destinazione un pacchetto IP può essere estratto e incapsulato più volte dal *frame* che lo contiene. Per esempio, un pacchetto IP generato su una rete locale e destinato a un server raggiungibile mediante la rete Internet, al momento in cui giunge al *modem* che interconnette la LAN con la WAN viene estratto dal *frame Ethernet* e incapsulato nel *frame* ADSL, con cui viaggia fino alla centrale telefonica locale, dove viene nuovamente estratto per essere incapsulato nel *frame* proprio della tecnologia di telecomunicazione utilizzata dall'operatore che fornisce il servizio di accesso Internet (ISP, *Internet Service Provider*).

OSSERVAZIONE La comunicazione che si stabilisce al livello applicativo è lo scopo dell'intero schema di comunicazione, ma è assolutamente virtuale, essendo reale solo la comunicazione che avviene al livello più basso tra i vari apparati hardware che costituiscono la rete LAN/WAN utilizzata.

4 Il livello di rete e il protocollo IP

Il **protocollo IP** ha il compito di trasferire un pacchetto dal computer mittente al computer destinatario: a questo scopo ogni computer presente sulla stessa rete deve disporre di un indirizzo IP univoco[6].

Gli indirizzi per la rete Internet sono rilasciati sotto il controllo dell'authority IANA (*Internet Assigned Numbers Authority*), che opera a sua volta per conto di ICANN (*Internet Corporation for Assigned Names and Numbers*), che è l'ente responsabile dell'assegnazione degli indirizzi IP e dei nomi di dominio per la rete Internet a livello mondiale.

L'**indirizzo IP versione 4** (IPv4) di un dispositivo è un numero a 32 bit che viene considerato suddiviso in due parti:

- l'indirizzo che identifica la rete – normalmente una rete LAN – cui appartiene il dispositivo (da un minimo di 1 bit a un massimo di 30 bit iniziando dal bit più significativo);
- l'indirizzo del dispositivo all'interno della rete (da un minimo di 2 bit a un massimo di 31 bit iniziando dal bit meno significativo).

Tutti i dispositivi presenti sulla stessa rete condividono la parte di rete del proprio indirizzo IP.

6. Questo è vero per gli indirizzi «pubblici» dei dispositivi presenti sulla rete Internet e per gli indirizzi «privati» di una rete LAN, ma in reti LAN distinte gli indirizzi sono spesso ripetuti. La tecnologia che consente a indirizzi non univoci di originare e ricevere pacchetti IP che attraversano la rete Internet – nota come NAT/PAT, *Network Address Translation/Port Address Translation* – è al di fuori dello scopo di questa trattazione.

Dispositivi con la parte di rete dell'indirizzo IP diversa appartengono a reti distinte e possono comunicare solo attraverso un *router*. I *modem* che interconnettono le reti LAN alla rete Internet svolgono quindi anche la funzione di *router*.

I *router* utilizzano la parte di rete dell'indirizzo IP di destinazione per inoltrare i singoli pacchetti sul cammino che ritengono più favorevole in base alle informazioni disponibili.

Il numero di bit della parte di rete di un indirizzo IP è definito dalla sua ***netmask***, una sequenza di 32 bit in cui i bit meno significativi sono posti a «0» e i più significativi sono posti a «1» in numero corrispondente ai bit dedicati alla parte di rete dell'indirizzo.

ESEMPIO

L'indirizzo IP

11000000 10101000 00000000 00000001

con *netmask*

11111111 11111111 11111111 00000000

ha i 24 bit più significativi che costituiscono la parte di rete dell'indirizzo e i rimanenti 8 bit meno significativi che rappresentano l'indirizzo del dispositivo nella rete.

Data la difficoltà di rappresentare gli indirizzi IP in formato binario, essi sono normalmente scritti come 4 numeri separati dal simbolo «.» corrispondenti al valore dei 4 byte che costituiscono l'indirizzo (i 4 numeri non possono quindi assumere un valore superiore a 255).

ESEMPIO

L'indirizzo IP dell'esempio precedente viene normalmente scritto come

192.168.0.1

con *netmask*

255.255.255.0

Sulle reti LAN è pratica comune utilizzare le *netmask* riportate nella TABELLA **5**.

TABELLA **5**

Netmask	**Bit indirizzo di rete**	**Bit indirizzo dispositivo**	**Massimo numero di dispositivi indirizzabili nella rete**
255.0.0.0	8	24	16 777 214
255.255.0.0	16	16	65 534
255.255.255.0	24	8	254

Il numero di dispositivi che si possono indirizzare in una rete è pari al numero di indirizzi distinti che si possono configurare con il numero di bit a disposizione meno 2. Si tratta degli indirizzi di dispositivo composti da tutti «0» e da tutti «1» che hanno un significato speciale: nel primo caso è

Protocolli di *routing*

Il protocollo IP è di fatto l'unico protocollo impiegato nelle LAN e nelle WAN a livello di rete. Nelle reti WAN come Internet, il trasferimento di un pacchetto IP dalla sua origine alla sua destinazione avviene per mezzo dei *router* che lo instradano sul percorso ritenuto «migliore». In realtà i *router* mantengono delle tabelle interne con i percorsi su cui instradare i pacchetti indirizzati ad alcune reti di destinazione e un percorso di *default* per le reti di destinazione ignote. Questo meccanismo apparentemente ingenuo funziona in modo efficace ed efficiente perché i *router* si scambiano continuamente informazioni sulle reti di destinazione dei pacchetti, propagando la conoscenza della topologia e dello stato della rete. I protocolli di *routing* (come RIP, *Routing Information Protocol*, o OSPF, *Open Short Path First*) hanno lo specifico compito di propagare in tempo reale da un *router* all'altro lo stato della rete.

Protocollo IP versione 6

Gli indirizzi IP a 32 bit caratterizzano il protocollo IP versione 4 (IPv4). La nuova versione 6 del protocollo IP (IPv6), che da alcuni anni affianca la versione precedente, prevede, oltre a importanti differenze nella struttura dello *header* finalizzate a rendere più efficiente il *routing* dei pacchetti, indirizzi IP di 128 bit.

Protocollo IPsec

Per la realizzazione di VPN (*Virtual Private Network*), cioè di reti che utilizzano l'infrastruttura pubblica della rete Internet, ma il cui traffico dati, anche se intercettato, non possa essere manipolato o interpretato, si utilizza il protocollo IPsec (IP *security*) in luogo del protocollo IP.
Il protocollo IPsec autentica e cifra ogni singolo pacchetto trasmesso; la verifica dell'integrità, della provenienza e la decodifica del pacchetto sono effettuate dal dispositivo destinatario.

l'indirizzo stesso della rete, mentre nel secondo caso è l'indirizzo di *broadcast* da utilizzare se un pacchetto è destinato a tutti i dispositivi presenti nella rete.

ESEMPIO

Dato l'indirizzo IP di rete

192.168.0.0

con *netmask*

255.255.255.0

i dispositivi presenti nella rete possono assumere indirizzi IP compresi tra

192.168.0.1 e 192.168.0.254

L'indirizzo

192.168.0.255

è l'indirizzo di *broadcast* che identifica come destinatari tutti i dispositivi della rete.

Protocollo di configurazione DHCP

Nelle reti LAN di grandi dimensioni e in particolare nelle reti WLAN, in cui l'accesso di dispositivi *wireless* non è facilmente controllabile, la configurazione manuale degli indirizzi IP e delle relative *netmask* risulta non essere pratica. In una rete LAN è possibile configurare un server per il protocollo DHCP (*Dynamic Host Configuration Protocol*) a cui i dispositivi non configurati inviano in *broadcast* una richiesta per ottenere una configurazione valida (contenente in particolare indirizzo IP e *netmask*, ma anche indirizzo IP del *gateway* e del server DNS).

Il problema di «risoluzione dell'indirizzo» si presenta su una rete LAN quando un dispositivo deve inoltrare un pacchetto per il quale è specificato l'indirizzo IP di destinazione: per incapsulare il pacchetto IP nel *frame Ethernet* è necessario conoscere l'indirizzo fisico del dispositivo di destinazione. In questo caso interviene il protocollo ARP (*Address Resolution Protocol*), che inoltra in *broadcast* a tutti i dispositivi della rete LAN una richiesta contenente l'indirizzo IP da risolvere; a questa richiesta risponde – se esiste – solo il dispositivo che ha l'indirizzo IP specificato, fornendo in questo modo il proprio indirizzo fisico. L'associazione tra indirizzo IP e indirizzo fisico viene mantenuta dal sistema operativo in una tabella temporanea, in modo da non dover ripetere questa interrogazione ogni volta che viene spedito un pacchetto allo stesso destinatario.

Nel caso che l'indirizzo IP di destinazione non appartenga alla stessa rete del dispositivo mittente (cosa che si verifica quando la parte di rete dell'indirizzo mittente è diversa da quella dell'indirizzo di destinazione), il protocollo ARP risolve l'indirizzo del cosiddetto *gateway*, che è normalmente il dispositivo *router*/*modem* che interfaccia la rete LAN alla rete Internet.

La configurazione di un computer o di un dispositivo connesso a una rete LAN o WAN deve di conseguenza prevedere i tre seguenti parametri:

- indirizzo IP;
- *network*;
- indirizzo IP del *gateway*.

L'indirizzo IP speciale (indirizzo di *loopback*)

127.0.0.1

rappresenta sempre il dispositivo stesso di origine: un pacchetto destinato a questo indirizzo non transita sulla rete, ma viene ricevuto dal dispositivo stesso che lo ha trasmesso.

L'intestazione (*header*) di un pacchetto IP versione 4 presenta una struttura piuttosto complessa, come mostra a pagina seguente la FIGURA **14**, mentre la TABELLA **6** ne descrive solo i campi fondamentali.

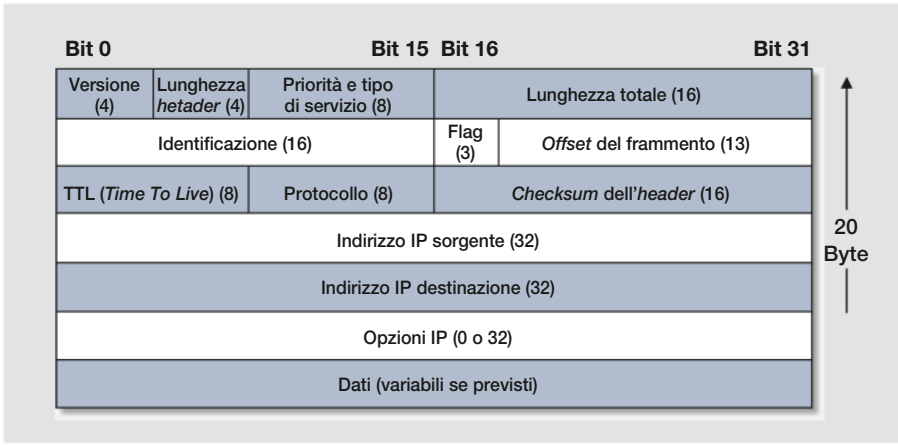


FIGURA **14**

TABELLA **6**

Versione	È la versione del protocollo IP: questo *header* è quello della versione 4, la nuova versione 6 del protocollo ha un *header* di tipo diverso.
Priorità e tipo di servizio	Usato per evidenziare i pacchetti che hanno alta priorità di trasmissione (per esempio pacchetti che contengono dati di *streaming* multimediali o conversazioni in tempo reale).
Lunghezza totale	È la dimensione in byte del pacchetto.
TTL (*Time To Live*)	Un eventuale errore dei *router* potrebbe non garantire la consegna di un pacchetto al dispositivo di destinazione. Questo valore (spesso impostato inizialmente a 128) viene decrementato ogni volta che il pacchetto viene instradato da un *router*; se il valore raggiunge 0 allora il pacchetto viene distrutto per evitare che il pacchetto vaghi in eterno da un *router* all'altro.
Protocollo	Usato per identificare il protocollo del livello di trasporto (per il protocollo TCP assume il valore 6, mentre per il protocollo UDP assume il valore 17).
Indirizzo IP sorgente	Sono gli indirizzi IP rispettivamente del dispositivo mittente e del dispositivo destinatario del pacchetto.
Indirizzo IP destinazione	

La presenza dell'indirizzo mittente nell'intestazione del pacchetto IP, oltre a permettere al dispositivo destinatario del pacchetto di rispondere, consente a un *router* che dovesse eventualmente distruggere un pacchetto di segnalare l'evento indicandone il motivo (per esempio perché la rete o il computer di destinazione non sono raggiungibili) al dispositivo mittente. A questo scopo viene utilizzato il protocollo di supporto ICMP.

Ciascuno di noi utilizza quotidianamente il *browser* del proprio computer o di un dispositivo mobile, ma la connessione al computer server che contiene le pagine web di interesse non avviene digitandone l'indirizzo IP, ma un indirizzo simbolico definito URL, *Uniform Resource Locator*.

ESEMPIO

I seguenti riferimenti a siti web sono URL:

http://www.zanichelli.it, http://www.istruzione.it, http://www.wikipedia.org, http://www.google.com, http://www.cern.ch, ...

Il prefisso «http» di questi URL fa ovviamente riferimento al protocollo applicativo HTTP, utilizzando il quale è possibile connettersi ai relativi siti web.

Ogni singolo URL è ovviamente univoco a livello globale: per facilitarne l'amministrazione gli URL che sono forniti alle organizzazioni, alle persone e alle aziende che li richiedono per i propri siti web sono strutturati in **domini**, termine con il quale si definisce il suffisso finale dell'URL stesso.

Esistono domini di tipo geografico («.it» per l'Italia, «.us» per gli USA, «.eu» per l'Unione Europea, ...), ma anche domini di tipo funzionale («.com» per le attività commerciali, «.org» per le organizzazioni *no-profit*, «.edu» per le istituzioni educative, ...). Ogni dominio ha un ente di gestione che è autorizzato a registrare gli URL – solo nel caso che non ne esista un altro identico ovviamente – per conto delle persone, delle aziende e delle organizzazioni o istituzioni che ne fanno richiesta, normalmente dietro pagamento di una quota annuale.

Un computer, o un dispositivo client, per connettersi a un computer server deve necessariamente conoscerne l'indirizzo IP, che è l'unico elemento che i *router* possono utilizzare per instradare correttamente i singoli pacchetti sulla rete WAN. A questo scopo in rete sono presenti alcuni computer server specializzati, denominati server **DNS** (***Domain Name System***), che sono in grado di restituire l'indirizzo IP associato a uno specifico URL gestendo richieste e risposte in base a quanto stabilito dal protocollo DNS.

ESEMPIO

Quando viene digitato un URL in un *browser* esso – prima di inviare la richiesta della pagina web specificata al computer server – ne richiede l'indirizzo IP a un server DNS; una volta ottenutolo in risposta, procede alla connessione con il computer server associato all'URL inizialmente specificato dall'utente. La FIGURA **15** illustra la sequenza di interazione tra computer client, server DNS e server web.

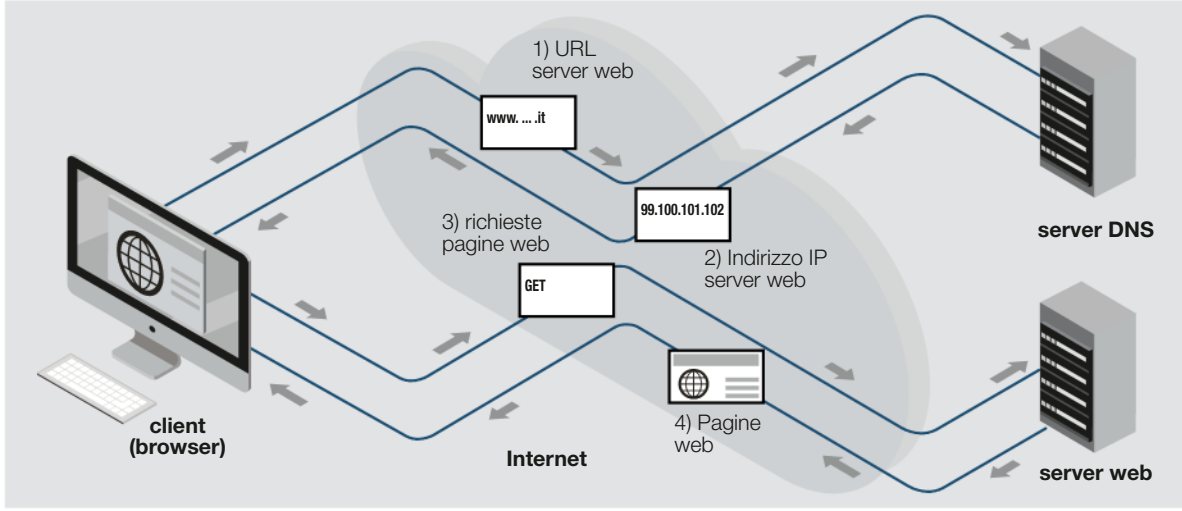


FIGURA **15**

L'indirizzo IP del server DNS cui richiedere di risolvere i nomi di dominio deve essere noto a priori ed è uno dei parametri di configurazione della rete che ogni computer o dispositivo richiede.

5 Il livello di trasporto e i protocolli UDP e TCP

Lo sviluppatore di applicazioni software che comunicano in rete ha la possibilità di scegliere tra i due diversi protocolli di trasporto seguenti.

- **UDP** (***User Datagram Protocol***): è un protocollo orientato allo scambio di messaggi denominati ***datagram*** (viene ricevuto dal destinatario con un'unica operazione di lettura il *datagram* trasmesso dal mittente in un'unica operazione di scrittura). Non è affidabile (non sono garantiti la ricezione dei *datagram* e il fatto che l'ordine in cui giungono al destinatario sia lo stesso con cui sono stati inviati dal mittente) e non presenta asimmetrie di ruolo tra i dispositivi che partecipano alla comunicazione.
- **TCP** (***Transmission Control Protocol***): è un protocollo affidabile orientato all'inoltro di un flusso di byte (garantisce la ricezione da parte del destinatario della sequenza ordinata di byte trasmessi, ma il destinatario riceve la sequenza di byte con operazioni di lettura non corrispondenti a quelle di scrittura impiegate per la trasmissione). Prevedendo la «connessione virtuale» tra i dispositivi che partecipano alla comunicazione, presenta una marcata asimmetria di ruolo tra il dispositivo server che accetta la connessione e il dispositivo client che richiede la connessione.

OSSERVAZIONE I due protocolli presentano caratteristiche molto differenziate. UDP è un protocollo molto efficiente, utilizzabile in quelle situazioni in cui l'applicazione software può permettersi la perdita di messaggi, o gestirne sotto la propria responsabilità l'eventuale ritrasmissione; inoltre UDP consente di gestire con relativa semplicità la comunicazione tra una molteplicità di dispositivi. TCP è un protocollo che offre all'applicazione software la possibilità di comunicare con un unico dispositivo, ma che si fa carico di garantire l'affidabilità della connessione liberando lo sviluppatore da qualsiasi responsabilità in merito al controllo della correttezza della comunicazione stessa.

Entrambi i protocolli condividono la necessità di individuare il processo mittente e il processo destinatario di un segmento di dati. Questa necessità è stata risolta dai progettisti dei due protocolli nello stesso modo: utilizzando un **numero di porta** a 16 bit (che assume quindi valori compresi tra 0 e 65 535) che identifica univocamente la sorgente o la destinazione dei segmenti di dati nel contesto di un dispositivo[7].

Dato che per iniziare la comunicazione UDP, o per richiedere la connessione a un server TCP, oltre che l'indirizzo IP del dispositivo destinatario è necessario conoscere anche il numero di porta del processo che risponderà alla richiesta, per alcuni protocolli di livello applicativo questo è stato

Protocolli sicuri TLS

Per rendere riservata e non manipolabile la comunicazione client/server è possibile adottare il protocollo TLS (*Transport Layer Security*) per autenticare e cifrare la comunicazione effettuata con il protocollo di trasporto TCP. Il protocollo TLS si interpone tra il livello di trasporto e il livello applicativo e sulla sua base sono state realizzate versioni «sicure» dei più comuni protocolli del livello applicazione come HTTPS (HTTP *Secure*) e FTPS (FTP *Secure*).

7. Il dispositivo è infatti identificato dall'indirizzo IP gestito dal protocollo del livello di rete.

standardizzato; la TABELLA **7** riporta a titolo di esempio alcuni *well-known port number*.

TABELLA **7**

Protocollo applicativo	Numero di porta
FTP (*File Transfer Protocol*)	20, 21
SMTP (*Simple Mail Transfer Protocol*)	25
DNS (*Domain Name Server*)	53
HTTP (*Hyper-text Transfer Protocol*)	80
POP (*Post Office Protocol*)	110
NTP (*Network Time Protocol*)	123
HTTPS (*Hyper-text Transfer Protocol Secure*)	443
Doom (*videogame*)	666
FTPS (*File Transfer Protocol Secure*)	989, 990

ESEMPIO Il *browser* a cui viene richiesto di connettersi al sito web con URL www.google.it invia una richiesta all'indirizzo IP del server DNS utilizzando il numero di porta 53 e, una volta ricevuto l'indirizzo IP utilizzato dalla versione italiana del motore di ricerca Google effettua la richiesta di connessione alla porta 80, cui risponde il processo server che utilizza il protocollo HTTP per redirezionare la richiesta alla porta 443 del protocollo HTTPS e infine trasferire la *home page* richiesta.

I numeri di porta di valore compreso tra 0 e 1023 sono riservati per i protocolli standard: uno sviluppatore non dovrebbe mai utilizzare questi numeri di porta per le applicazioni web che realizza[8].

L'intestazione di un pacchetto UDP presenta una struttura molto semplice (FIGURA **16**) e la TABELLA **8** ne descrive i campi.

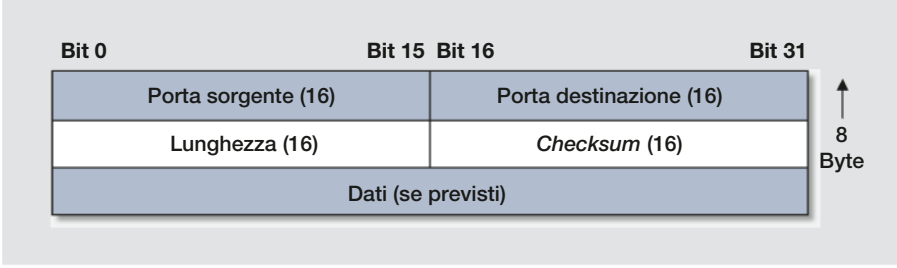


FIGURA **16**

TABELLA **8**

Porta sorgente	Sono rispettivamente il numero di porta usato dal processo mittente e il numero di porta del processo destinatario del segmento di dati.
Porta destinazione	
Lunghezza	È la dimensione in byte del pacchetto.
Checksum	Codice calcolato a partire dal contenuto del segmento di dati con un algoritmo predefinito; viene ricalcolato dal dispositivo destinatario per verificare che i dati ricevuti sono corretti.

L'intestazione di un pacchetto TCP è invece più complessa (FIGURA **17**) e la TABELLA **9** a pagina seguente ne descrive solo alcuni campi fondamentali.

8. In realtà molti numeri di porta compresi tra 1024 e 49 151 sono stati registrati da produttori di applicazioni software per i loro prodotti: se uno di questi prodotti è installato e attivo in un dispositivo, il relativo numero di porta non può essere utilizzato.

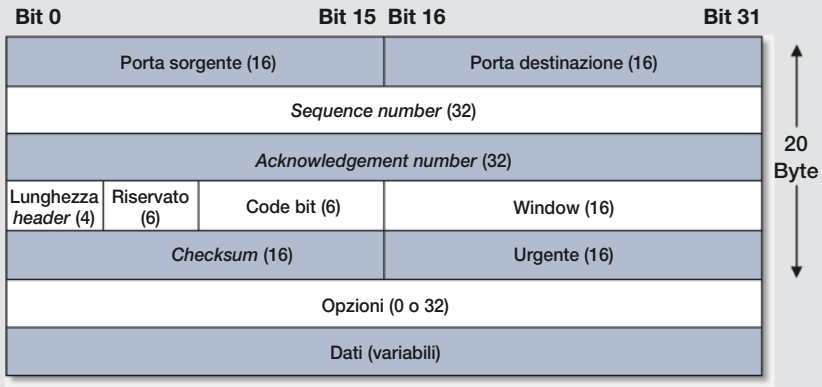


FIGURA **17**

TABELLA **9**

Porta sorgente	Sono rispettivamente il numero di porta usato dal processo mittente e il numero di porta del processo destinatario del segmento di dati.
Porta destinazione	
Sequence number	È il numero di byte inviati dall'inizio della connessione.
Acknowledgement number	È il numero di byte ricevuti dall'inizio della connessione.
Checksum	È il codice calcolato a partire dal contenuto del segmento di dati con un algoritmo predefinito; viene ricalcolato dal dispositivo destinatario per verificare che i dati ricevuti sono corretti.

OSSERVAZIONE I campi *Sequence number* e *Acknowledgement number* sono utilizzati per verificare, ed eventualmente correggere mediante ritrasmissione di parti mancanti, l'integrità del flusso di dati.

6 Il modello client/server e il protocollo applicativo HTTP: dal web al *cloud*

Molti protocolli di rete del livello applicativo che impiegano il protocollo di trasporto TCP realizzano un modello di cooperazione asimmetrica tra i due computer, o dispositivi, che partecipano alla comunicazione e che prendono rispettivamente il nome di client e server:

- il computer **server** ospita un processo – solitamente definito **servizio** – che, su una porta avente un numero noto, resta costantemente in attesa di ricevere delle richieste alle quali risponde inviando dati al computer o al dispositivo che ha effettuato la richiesta stessa;
- il computer o dispositivo **client** invia una richiesta a un computer server di cui conosce l'indirizzo IP e il numero di porta del servizio desiderato e dal quale si aspetta di ricevere una risposta.

ESEMPIO I siti web sono ospitati su computer server che, sulla porta 80 o 443, ricevono dai *browser* in esecuzione su computer o dispositivi client le richieste per singole pagine web che costituiscono il sito e alle quali rispondono inviando il contenuto delle pagine richieste. Come illustrato in FIGURA **18**, il protocollo applicativo HTTP stabilisce le regole di comunicazione per la richiesta e l'invio delle pagine web.

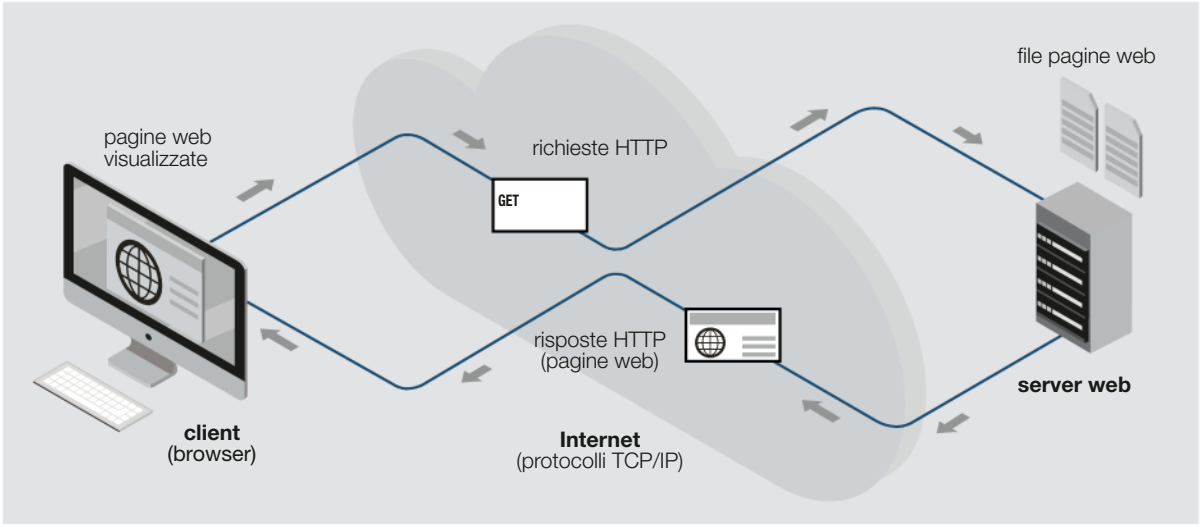


FIGURA **18**

La rete Internet è cresciuta per tutti gli anni Ottanta del secolo scorso interconnettendo università e centri di ricerca in tutto il mondo, ma fino ai primi anni Novanta è stata qualcosa che oggi nessuno di noi riconoscerebbe: non esistevano infatti i siti web! Gli utenti della rete, in prevalenza ricercatori e scienziati, si spedivano e-mail utilizzando programmi basati sui protocolli applicativi SMTP e POP e si scambiavano file con programmi basati sul protocollo FTP, ma il grande ipertesto multimediale rappresentato dal *World Wide Web*, che è oggi sinonimo di Internet, semplicemente non era stato ancora inventato.

L'idea di realizzare un nuovo protocollo applicativo che rendesse disponibili in rete pagine di testo formattato collegate le une alle altre e contenenti elementi multimediali (immagini, video, ...) la si deve al fisico e informatico inglese Tim Berners-Lee che, presso il CERN di Ginevra – il Centro Europeo per la Ricerca Nucleare, tutt'oggi il più moderno e grande laboratorio di fisica atomica al mondo –, la concretizzò nel corso del 1991. Scopo di Tim Berners-Lee era quello di rendere fruibili tramite Internet ai ricercatori e agli scienziati di tutto il mondo gli articoli scientifici in un formato ipertestuale comodamente leggibile, non a caso denominò il nuovo protocollo di rete **HTTP**, ***Hyper-Text Transfer Protocol***; comunque la sua visione del futuro sviluppo della nuova tecnologia era molto chiara, tanto da denominarla fin da subito **WWW**, ***World Wide Web***, letteralmente «ragnatela a estensione mondiale». Oggi Tim Berners-Lee presiede il W3C (*World Wide Web Consortium*), un'organizzazione internazionale avente lo scopo di promuovere lo sviluppo del web tramite la formalizzazione dei linguaggi e dei protocolli di livello applicativo.

Il linguaggio HTML

Le pagine web che un computer server invia in risposta alle richieste di un computer o dispositivo client sono codificate in un linguaggio ideato da Tim Berners-Lee, il linguaggio HTML, *Hyper-Text Mark-up Language*.
Il *browser* del computer o dispositivo client ha il compito di interpretare il linguaggio HTML con cui sono definite le pagine web e di visualizzarne il contenuto (testo e immagini, ma anche elementi multimediali quali suoni, musica, video, ...) con l'aspetto che il codice HTML stesso definisce.

OSSERVAZIONE Fin dall'inizio il protocollo applicativo HTTP fu concepito per gestire «risorse», in generale, mediante comandi di richiesta,

aggiornamento, creazione, eliminazione, ... Per un *browser* che effettua richieste a un server web le risorse sono le pagine che visualizza, ma oggi molte applicazioni e *APP* utilizzano il protocollo HTTP per richiedere, aggiornare, creare ed eliminare risorse software di altro tipo come, per esempio, i record della tabella di un database.

Il protocollo HTTP – di cui la versione attualmente più usata è la 1.1, standardizzata nel 1999 – è un protocollo prevalentemente testuale (le richieste effettuate e le relative risposte fornite sono normalmente stringhe di caratteri alfanumerici): il client invia al server richieste a cui il server risponde con una stringa di stato seguita dalla risposta vera e propria. Le richieste che un client può inoltrare a un server HTTP sono basate su alcune parole chiave che identificano altrettanti «**metodi**» corrispondenti, sempre riferiti a una risorsa identificata dall'URL compreso nella richiesta stessa ed eventualmente seguite da una o più specificazioni denominate ***header***. Nella TABELLA **10** sono riepilogati i metodi del protocollo HTTP. I metodi POST, PUT e TRACE comprendono il contenuto da inviare al server che costituisce il ***body*** della richiesta.

TABELLA **10**

Metodo	Operazione
GET	Richiede al server la risorsa identificata dall'URL specificato.
HEAD	Come GET, ma la risposta fornita dal server non comprende la risorsa (solo alcune informazioni relative alla risorsa note come *header*).
POST	Integra la risorsa identificata dall'URL specificato con le informazioni inviate.
PUT	Crea o aggiorna la risorsa identificata dall'URL specificato a partire dalle informazioni inviate.
DELETE	Elimina la risorsa identificata dall'URL specificato.
TRACE	Richiede al server una risposta costituita dalle informazioni inviate («eco»).
CONNECT	Richiede al server un accesso diretto al protocollo TCP.
OPTIONS	Richiede al server l'elenco dei metodi accettati per la risorsa identificata dall'URL specificato.

HTTP/2

Nel 2015 è stato pubblicato da IETF lo standard HTTP/2 (RFC 7540). Questa nuova versione del protocollo HTTP pur essendo binaria anziché testuale è compatibile con la precedente 1.1 in quanto non ne modifica i metodi, i codici di stato e gli *header*, ma permette di incrementare l'efficienza di caricamento di un sito web da parte di un *browser* in vari modi:

- effettuando il *multiplexing* della trasmissione/ricezione delle risorse su un'unica connessione TCP anziché su connessioni multiple;
- comprimendo gli *header*;
- permettendo al server di anticipare le richieste del client trasmettendo risorse in modalità *push*.

ESEMPIO

Accedendo con un client Telnet[9] a un web server in esecuzione sullo stesso computer e che ha nella propria directory di lavoro il file «HelloWorld.html» contenente il codice HTML di una semplice pagina web, si possono inviare richieste e visualizzare le relative risposte. Inviando la seguente richiesta[10]

```
GET /HelloWorld.html HTPP/1.1
Host: 127.0.0.1
```

si ottiene una risposta simile a questa:

```
HTTP/1.1 200 OK
Date: Tue, 15 Nov 2016 15:56:36 GMT
Server: Apache/2.4.18 (Win32) OpenSSL/1.0.2e PHP/7.0.2
Last-Modified: Mon, 14 Nov 2016 15:32:38 GMT
Content-Length: 117
Content-Type: text/html

<!DOCTYPE html>
<html>
```

9. Il programma *Telnet* consente a un utente di connettersi a un server TCP specificandone indirizzo IP (o URL) e numero di porta per digitare comandi e visualizzare le risposte inviate dal server.

10. La specificazione del riferimento del server che ospita la risorsa richiesta (in questo caso è stato fornito l'indirizzo IP) mediante lo *header* «Host» è obbligatoria e consente l'adozione di «server virtuali».

```
<head>
<title>Hello world</title>
</head>
<body>
<p>Hello world!</p>
</body>
</html>
```

Inviando invece la seguente richiesta

```
OPTIONS /HelloWorld.html HTPP/1.1
Host: 127.0.0.1
```

si ottiene una risposta simile a questa:

```
HTTP/1.1 200 OK
Date: Sat, 19 Nov 2016 14:19:55 GMT
Server: Apache/2.4.18 (Win32) OpenSSL/1.0.2e PHP/7.0.2
Allow: GET,HEAD,POST,OPTIONS,TRACE
Content-Length: 0
Content-Type: text/html
```

Nelle risposte fornite dal server nell'esempio precedente è chiaramente riconoscibile la linea di stato

```
HTTP/1.1 200 OK
```

seguita da più linee denominate *header*, ciascuna delle quali è composta da un identificatore seguito dal simbolo «:» e dal relativo valore (sono facilmente riconoscibili la data/ora della risposta, la data/ora di modifica della pagina, la descrizione del web server, la dimensione in byte del contenuto restituito e il relativo tipo). Gli *header* del protocollo HTTP sono eventualmente seguiti dalla risorsa richiesta, nel caso della prima richiesta dell'esempio dal contenuto del file «HelloWorld.html».

La risposta a una richiesta del protocollo HTTP è costituita da un codice di stato, da uno o più *header* e – per alcuni metodi – dal *body*, cioè dal contenuto della risorsa richiesta. Nelle TABELLE **11** e **12** sono rispettivamente elencate le categorie dei codici di stato e descritti alcuni codici di stato fondamentali.

TABELLA **11**

Formato del codice di stato	Categoria
1XX (100, 101, ...)	Informativa (codici di negoziazione tra client e server).
2XX (200, 201, ...)	Successo (codici che confermano al client che la richiesta avanzata al server è stata eseguita correttamente).
3XX (300, 301, ...)	Ridirezione (codici che comunicano al client che la richiesta avanzata al server NON è stata eseguita a causa di un'errata specificazione della risorsa).
4XX (400, 401, ...)	Errore del client (codici che comunicano al client che la richiesta avanzata al server NON è stata eseguita perché errata).
5XX (500, 501, ...)	Errore del server (codici che comunicano al client che la richiesta avanzata al server NON è stata eseguita a causa di un errore del server).

TABELLA **12**

Codice di stato	Descrizione
100 Continue	La richiesta avanzata parzialmente può essere completata.
200 OK	Richiesta eseguita con successo.
201 Created	Risorsa creata con successo.
202 Accepted	Richiesta accettata, ma non ancora eseguita.
204 No Content	Richiesta eseguita con successo, ma senza restituire un contenuto.
300 Multiple Choices	Esistono versioni multiple per la risorsa richiesta.
301 Moved Permanently	La risorsa specificata esiste, ma è diversamente identificata.
303 See Other	Il server non dispone della risorsa richiesta che può essere acceduta con un diverso URL specificato nello *header Location* della risposta.
400 Bad Request	Richiesta errata.
401 Unauthorized	Richiesta non autorizzata in mancanza di informazioni di autenticazione.
403 Forbidden	Richiesta proibita dai requisiti di sicurezza del server.
404 Not Found	Risorsa specificata non esistente.
405 Method Not Allowed	Metodo non accettato per la risorsa specificata.
406 Not Acceptable	Il server non può rispondere alla richiesta rispettando le condizioni definite dagli *header.*
408 Request Time-out	La richiesta non è stata ricevuta dal server in tempo utile.
409 Conflict	Richiesta che genera un conflitto di aggiornamento della risorsa specificata.
414 URL Too Long	L'URL della richiesta è troppo lungo.
415 Unsupported Media Type	La richiesta specifica un *media type* che il server non supporta per la risorsa specificata.
426 Upgrade required	Il server richiede il passaggio a una diversa versione del protocollo, normalmente la versione HTTPS che utilizza il protocollo sicuro TLS.
429 Too Many Request	Il server ha ricevuto troppe richieste in un determinato periodo di tempo.
500 Internal Server Error	Errore generico del server.
501 Not Implemented	Metodo non implementato dal server.
503 Service Unavailable	Server temporaneamente non disponibile.

ESEMPIO

Nelle stesse condizioni dell'esempio precedente la richiesta

```
GET /CiaoMondo.html HTTP/1.1
Host: 127.0.0.1
```

ottiene una risposta come la seguente:

```
HTTP/1.1 404 Not Found
Date: Wed, 16 Nov 2016 15:22:42 GMT
Server: Apache/2.4.18 (Win32) OpenSSL/1.0.2e PHP/7.0.2
Content-Length: 593
Content-Type: text/html; charset=utf-8
Content-Language: en

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
```

```
<title>Object not found!</title>
</head>
<body>
<h1>Object not found!</h1>
<p>
The requested URL was not found on this server.
If you entered the URL manually please check your spelling and try
again.
</p>
If you think is a server error, please contact the <a href="mailto:postmaster@localhost">
webmaster </a>
<h2>Error 404</h2>
<address>
<a href="/">127.0.0.1</a><br/>
<span>Apache/2.4.18 (Win32) OpenSSL/1.0.2e PHP/7.0.2</span>
</address>
</body>
</html>
```

il cui *body* contiene la tipica pagina visualizzata dal *browser* quando viene richiesto un URL non esistente. Invece la richiesta

```
XYZ /CiaoMondo.html HTTP/1.1
Host: 127.0.0.1
```

ottiene una risposta simile alla seguente:

```
HTTP/1.1 501 Not Implemented
Date: Sat, 19 Nov 2016 15:07:55 GMT
Server: Apache/2.4.18 (Win32) OpenSSL/1.0.2e PHP/7.0.2
Connection: close
Content-Type: text/html; charset=utf-8
Content-Language: en

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
<title>Cannot process request!</title>
</head>
<body>
<h1>Cannot process request!</h1>
<p>
The server does not support the action requested by the browser.
</p>
If you think is a server error, please contact the <a href="mailto:postmaster@localhost">
webmaster </a>
<h2>Error 501</h2>
<address>
<a href="/">127.0.0.1</a><br/>
<span>Apache/2.4.18 (Win32) OpenSSL/1.0.2e PHP/7.0.2</span>
</address>
</body>
</html>
```

il cui *body* contiene una pagina di errore che il *browser* visualizza.

Gli *header* più comunemente utilizzati nelle richieste/risposte del protocollo HTTP sono riportati nella TABELLA **13**.

TABELLA **13**

Header	Richiesta	Risposta	Descrizione
Host	•		Nome di dominio o indirizzo IP del server che ospita la risorsa.
Server		•	Nome del server HTTP che fornisce la risposta.
Date	•	•	Data/ora di invio della richiesta o della risposta.
Content-Length	•	•	Dimensione in byte del *body* della risposta o della richiesta.
Content-Type	•	•	Tipo MIME (*Internet Media Type*) del *body* (esempio: `text/html`).
Last-Modified		•	Data/ora di ultima modifica del *body* della risposta.
Accept	•		Elenco di tipi MIME accettati dal client (esempio: `text/xml`).
Accept-Encoding	•		Tipi di codifica/compressione accettati dal client (esempi: `identity`, `gzip`, ...).
Accept-Charset	•		Tipi di codifica dei caratteri accettati dal client (esempi: `utf-8`, `iso-8859-1`, ...).
User-Agent	•		Informazioni relative al client che esegue la richiesta.
Authorization	•		Informazioni di autenticazione del client.
Content-Encoding		•	Tipo di codifica/compressione del *body* della risposta (esempio: `gzip`).

Multipurpose Internet Mail Extension (MIME)

Lo standard MIME nasce per definire le estensioni di contenuto dei messaggi di posta elettronica e i formati degli allegati. Con il tempo la modalità di definizione dei formati dello standard MIME è divenuta uno standard per la definizione del formato dei dati scambiati sulla rete Internet indipendentemente dal protocollo utilizzato. Questa evoluzione dello standard prende il nome di *Internet Media Type* e prevede una classificazione dei formati in categorie:

- *application*: formati per specifiche applicazioni software (per esempio PDF);
- *audio*: formati per audio (per esempio MP3);
- *image*: formati per immagini (per esempio JPEG);
- *message*: protocolli per lo scambio di messaggi;
- *model*: formati per modelli tridimensionali (per esempio VRML);
- *multipart*: formati per oggetti internamente costituiti da più parti;
- *text*: formati testuali (per esempio CSV e HTML);
- *video*: formati per video (per esempio MPEG).

Semplificando si può affermare che oggi il *World Wide Web* è materialmente costituito da tre componenti fondamentali (FIGURA **19**):

- i computer server che ospitano i siti e i servizi web esposti prevalentemente mediante il protocollo applicativo HTTP utilizzando il protocollo di trasporto TCP;
- l'infrastruttura di comunicazione definita dal nome Internet, a sua volta costituita dai collegamenti delle WAN e dai *router* che instradano i pacchetti del protocollo IP;
- i computer e i dispositivi client che attraverso Internet si connettono ai computer server per riceverne la pagine web dei siti e accedere ai servizi utilizzando i protocolli di trasporto e applicativo TCP e HTTP.

I primi dieci anni successivi all'introduzione del protocollo HTTP, che ha permesso la nascita e lo sviluppo del *World Wide Web*, sono oggi noti come «web 1.0», per contrasto con la fase del cosiddetto «web 2.0»[11] che ha simbolicamente inizio con il nuovo millennio. I siti del web 2.0 sono fortemente dinamici e basati sull'interazione degli utenti con le pagine web che li costituiscono: blog, forum, chat, social network come Facebook, Twitter e Google+ e piattaforme di condivisione di contenuti multimediali come

11. Questa espressione è stata coniata dall'editore Tim O'Reilly in una conferenza relativa alle tecnologie web innovative nel 2004.

Flickr e YouTube hanno rivoluzionato la natura prevalentemente statica che aveva in precedenza il *World Wide Web*.

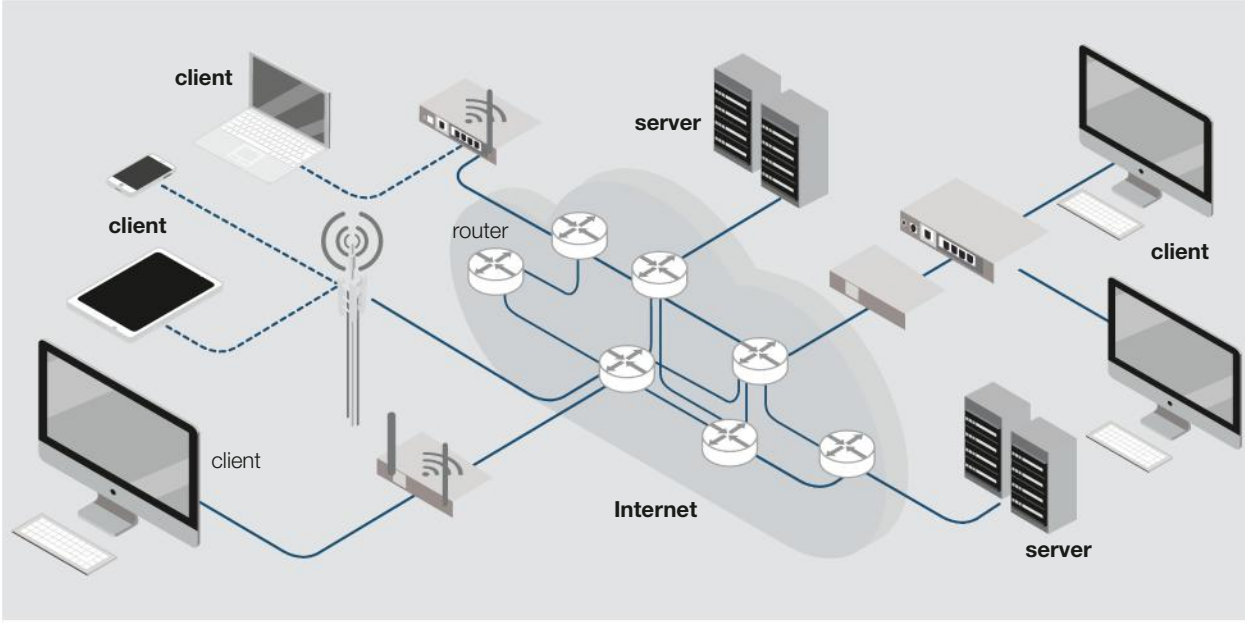


FIGURA **19** Il WWW è costituito dai computer server, dalla rete Internet (collegamenti WAN e *router*) e dai computer e dispositivi client.

Negli ultimi anni il termine che più viene utilizzato per definire le tecnologie web innovative è ***cloud***, che deriva ovviamente dall'uso dei tecnici informatici di rappresentare graficamente la complessità della rete Internet mediante una nuvola stilizzata. La trasformazione del web in *cloud* prevede che i dati degli utenti – sia privati sia aziendali – e le applicazioni software che essi utilizzano risiedano su computer localizzati nella rete, «nella nuvola», anziché sui propri computer o dispositivi personali, o sui computer dell'azienda o dell'organizzazione.

Oggi il protocollo HTTP sempre più spesso viene utilizzato non per trasferire pagine di siti web da visualizzare in un *browser* per un utente umano, ma come protocollo di accesso a servizi remoti di cui fanno uso applicazioni per computer e *APP* per dispositivi mobili.

I servizi software resi disponibili in rete ad applicazioni e *APP* prendono il nome di ***web-service***, in contrasto con i *web-site* a cui accedono gli utenti umani; l'insieme di servizi reso disponibile dai *web-service* – in analogia con il web reso disponibile dall'insieme di tutte le pagine che costituiscono i siti – è denominato ***programmable web***.

I CONCETTI CHIAVE

RETI LOCALI (LAN). Una rete locale (LAN, *Local Area Network*) moderna è realizzata connettendo con una topologia «a stella» i computer che costituiscono la rete a un dispositivo centrale denominato *switch*; nel caso di reti locali *wireless* (WLAN) il dispositivo al centro della stella di comunicazione radio è invece definito *access-point*.

RETI GEOGRAFICHE (WAN). La rete geografica (WAN, *Wide Area Network*) Internet interconnette tra loro milioni di reti LAN: il dispositivo che consente di collegare una rete LAN alla rete Internet è il *modem*. I *router* hanno invece il compito di smistare il traffico dei dati nella rete.

RETI DI COMPUTER. Le reti di computer sia di tipo LAN sia WAN sono oggi il contesto di riferimento ineludibile per lo sviluppo di applicazioni software e di *APP* per dispositivi mobili.

PACKET-SWITCHING. La tecnologia *packet-switching* è alla base del funzionamento di tutte le reti di telecomunicazione moderne. Essa prevede che un messaggio trasmesso da un dispositivo mittente a un dispositivo destinatario sia suddiviso in «pacchetti» di piccole dimensioni: ogni singolo pacchetto include l'«indirizzo» sia del dispositivo mittente sia del dispositivo destinatario.
La tecnologia *packet-switching* è stata fin dall'inizio concepita con l'idea di essere «robusta» rispetto ai guasti e ai malfunzionamenti degli apparati e dei collegamenti che costituiscono una rete di computer: anche se non frequentemente, un pacchetto che costituisce il frammento di un messaggio può essere perso senza compromettere l'integrità del messaggio ricevuto, ma solo un ritardo nella ricezione.

ROUTING. La comunicazione tra *router* per lo scambio di informazioni relative al loro stato di funzionamento, oltre a realizzare una valida gestione dei malfunzionamenti, permette anche di risolvere automaticamente il problema ricorrente della congestione del traffico di dati in una zona di una rete: al verificarsi di ritardi e di perdite di dati i *router* devieranno un numero sempre maggiore di pacchetti verso parti della rete meno congestionate.

RETI LAN *WIRED* E *WIRELESS*. Le reti LAN moderne sono tutte realizzate in base allo standard *Ethernet* e possono essere di due tipi: *wired* o cablata (utilizzano cavi in rame o fibre ottiche; ogni dispositivo è connesso a uno *switch* e la velocità di trasmissione, come la massima lunghezza di un collegamento, varia in funzione delle caratteristiche dei cavi e degli apparati di rete) e *wireless* o «senza fili» (utilizzano le onde radio nelle bande di frequenza comprese tra 2,4 e 2,5 GHz o tra 5,7 e 5,9 GHz; ogni dispositivo comunica con un *access-point* e la velocità di trasmissione e la distanza raggiungibile dipendono dalle caratteristiche dei dispositivi e degli apparati di rete).

STANDARD *ETHERNET*. Gli indirizzi «fisici» riportati in un *frame Ethernet* sono sequenze di 48 bit (sono sempre presenti l'indirizzo del dispositivo mittente che genera e trasmette il *frame* e quello del dispositivo destinatario del *frame*; nel caso che il *frame* sia destinato a tutti i dispositivi della rete esso è costituito da una sequenza di 48 bit impostati al valore «1»). I produttori di dispositivi che si possono connettere a una rete LAN o WLAN predefiniscono in fase di produzione indirizzi univoci per ciascuno di essi. I dispositivi connessi in una rete LAN o WLAN ricevono in molti casi anche *frame* destinati ad altri dispositivi, ma in base al proprio indirizzo fisico selezionano solo i *frame* a loro destinati.
Lo standard *Ethernet* prevede soluzioni hardware estremamente diversificate ma espone un modello unico di trasmissione da dispositivo a dispositivo di un *frame* nell'ambito di una rete locale.

PROTOCOLLO DI RETE. Un protocollo di rete è un insieme di regole tecniche formalmente e rigorosamente definite e documentate che consente la progettazione e la realizzazione di apparati di rete (*router*, *switch*, *access-point*, *modem*, schede di rete *wired* e *wireless* per i computer e per altri dispositivi, ...) e di applicazioni software che – come il *browser* – consentono di comunicare con altri computer o dispositivi connessi mediante una rete LAN e/o WAN.

STANDARD ISO/OSI. Nel 1994 l'Organizzazione Internazionale per la Standardizzazione (ISO, *International Standard Organization*) ha pubblicato lo standard OSI (*Open System Interconnection*) che definisce un modello di generica architettura di rete organizzata in 7 livelli gerarchici: Fisico, *Data-link*, Rete, Trasporto, Sessione, Presentazione e Applicazione.

PROTOCOLLI TCP/IP. I protocolli dei livelli intermedi di trasporto e di rete denominati rispettivamente TCP e IP sono gli stessi per tutte le reti LAN e WAN moderne: sono stati sviluppati nel corso degli anni Settanta del secolo scorso da Vinton Cerf e Robert Kahn presso i laboratori dell'università di Stanford per la realizzazione di Internet e ne sono ancora oggi alla base del funzionamento. Dal punto di vista tecnologico la rete Internet è infatti fondata sullo *stack* («pila», per sottolineare la struttura gerarchica articolata in livelli) di protocolli TCP/IP.

INCAPSULAZIONE. Il funzionamento congiunto di più protocolli a vari livelli gerarchici è basato sulla tecnica di incapsulazione dei singoli pacchetti costituiti ai livelli

più alti del modello nei pacchetti dei livelli più bassi come dati.

PROTOCOLLO IP. Il protocollo IP ha il compito di trasferire un pacchetto dal computer mittente al computer destinatario: a questo scopo ogni computer presente sulla stessa rete deve disporre di un indirizzo IP univoco.

INDIRIZZI IP. L'indirizzo IP versione 4 (IPv4) di un dispositivo è un numero a 32 bit che viene considerato suddiviso in due parti: l'indirizzo che identifica la rete cui appartiene il dispositivo (da un minimo di 1 bit a un massimo di 30 bit iniziando dal bit più significativo) e l'indirizzo del dispositivo all'interno della rete (da un minimo di 2 bit a un massimo di 31 bit iniziando dal bit meno significativo). Tutti i dispositivi presenti sulla stessa rete condividono la parte di rete del proprio indirizzo IP.
Il numero di bit della parte di rete di un indirizzo IP è definito dalla sua *netmask*, una sequenza di 32 bit in cui i bit meno significativi sono posti a «0» e i più significativi sono posti a «1» in numero corrispondente ai bit dedicati alla parte di rete dell'indirizzo.
Data la difficoltà di rappresentare gli indirizzi IP in formato binario, essi sono normalmente scritti come 4 numeri separati dal simbolo «.» corrispondenti al valore dei 4 byte che costituiscono l'indirizzo (i 4 numeri non possono quindi assumere un valore superiore a 255).

URL E DNS. Ogni singolo URL (*Uniform Resource Locator*) è ovviamente univoco a livello globale: per facilitarne l'amministrazione, gli URL che sono forniti alle organizzazioni, alle persone e alle aziende che li richiedono per i propri siti web sono strutturati in domini, termine con il quale si definisce il suffisso finale dell'URL stesso.
Un computer, o un dispositivo client, per connettersi a un computer server deve necessariamente conoscerne l'indirizzo IP, che è l'unico elemento che i *router* possono utilizzare per instradare correttamente i singoli pacchetti sulla rete WAN. A questo scopo in rete sono presenti alcuni computer server specializzati, denominati server DNS (*Domain Name System*), che sono in grado di restituire l'indirizzo IP associato a uno specifico URL gestendo richieste e risposte in base a quanto stabilito dallo specifico protocollo DNS.

PROTOCOLLI UDP E TCP. Esistono due principali e diversi protocolli di trasporto: UDP (*User Datagram Protocol*) è un protocollo orientato allo scambio di messaggi, denominati *datagram*, non affidabile e che non presenta asimmetrie di ruolo tra i dispositivi che partecipano alla comunicazione; TCP (*Transmission Control Protocol*) è un protocollo affidabile orientato all'inoltro di un flusso di byte e che – prevedendo la «connessione virtuale» tra i dispositivi che partecipano alla comunicazione – presenta una marcata asimmetria di ruolo tra il dispositivo server che accetta la connessione e il dispositivo client che richiede la connessione.

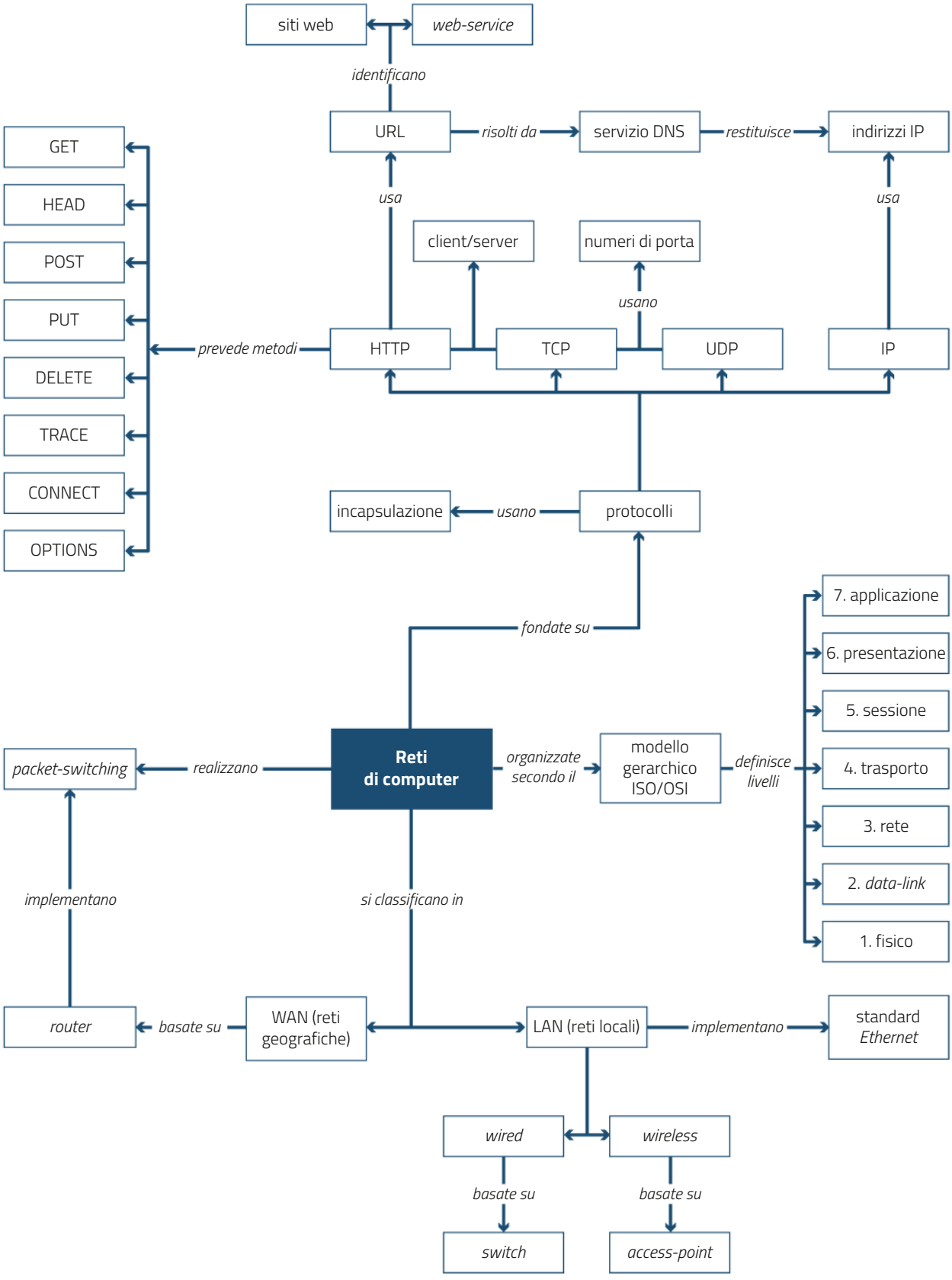
NUMERI DI PORTA. I protocolli UDP e TCP condividono la necessità di individuare il processo mittente e il processo destinatario di un segmento di dati. Questa necessità è stata risolta dai progettisti dei due protocolli nello stesso modo: utilizzando un numero di porta a 16 bit (che assume quindi valori compresi tra 0 e 65 535), che identifica univocamente la sorgente o la destinazione dei segmenti di dati nel contesto di un dispositivo che è identificato dall'indirizzo IP gestito dal protocollo del livello di rete.

MODELLO CLIENT/SERVER. Molti protocolli di rete del livello applicativo che impiegano il protocollo di trasporto TCP realizzano un modello di cooperazione asimmetrica tra i due computer, o dispositivi, che partecipano alla comunicazione e che prendono rispettivamente il nome di client e server. Il computer server ospita un processo – solitamente definito servizio – che, su una porta avente un numero noto, resta costantemente in attesa di ricevere delle richieste alle quali risponde inviando dati al computer o al dispositivo che ha effettuato la richiesta stessa; il computer o dispositivo client invia una richiesta a un computer server di cui conosce l'indirizzo IP e il numero di porta del servizio desiderato e dal quale si aspetta di ricevere una risposta.

PROTOCOLLO HTTP. Il protocollo HTTP – di cui la versione attualmente più utilizzata è la 1.1, standardizzata nel 1999 – è un protocollo prevalentemente testuale (le richieste effettuate e le relative risposte fornite sono normalmente stringhe di caratteri alfanumerici): il client invia al server richieste a cui il server risponde con una stringa di stato seguita dalla risposta vera e propria. Le richieste che un client può inoltrare a un server HTTP sono basate su alcune parole chiave che identificano altrettanti metodi (GET, HEAD, POST, PUT, DELETE, TRACE, CONNECT e OPTIONS), sempre riferiti a una risorsa identificata dall'URL compreso nella richiesta stessa ed eventualmente seguite da una o più specificazioni denominate *header*. I metodi POST, PUT e TRACE comprendono dati da inviare al server.
La risposta a una richiesta del protocollo HTTP è costituita da un codice di stato, da uno o più *header* e – per alcuni metodi – dal *body*, cioè dal contenuto della risorsa richiesta.

WEB-SERVICE. I servizi software resi disponibili in rete ad applicazioni e *APP* prendono il nome di *web-service*, in contrasto con i *web-site* a cui accedono gli utenti umani; l'insieme di servizi reso disponibile dai *web-service* – in analogia con il web reso disponibile dall'insieme di tutte le pagine che costituiscono i siti – è denominato *programmable web*.

RIPASSA CON LA MAPPA



SPIEGA IN UNA PAGINA

1 Descrivere il funzionamento di massima di una rete di computer di tipo *packet-switching*.

2 Descrivere in relazione ai livelli del modello ISO/OSI le funzionalità dei principali protocolli di rete.

QUESITI A RISPOSTA APERTA

3 Rappresentare graficamente il *frame Ethernet* descrivendone sinteticamente i campi principali.

4 Rappresentare graficamente lo *header* del protocollo IP descrivendone sinteticamente i campi principali.

5 Rappresentare graficamente lo *header* del protocollo UDP descrivendone sinteticamente i campi principali.

6 Rappresentare graficamente lo *header* del protocollo TCP descrivendone sinteticamente i campi principali.

7 Per ciascuna caratteristica, indicare la tipologia di rete corretta scegliendo tra LAN (L) e WAN (W).

A. Rete confinata in un edifico .....
B. Rete a scala geografica .....
C. Rete realizzata mediante *switch* e/o *access-point* .....
D. Rete realizzata mediante *router* .....

8 Indicare per i protocolli indicati i relativi livelli gerarchici del modello OSI dell'ISO.

A. FTP .....
B. HTTP .....
C. UDP .....
D. IP .....
E. TCP .....
F. IEEE-802.3 .....
G. PPP .....

9 Indicare se le seguenti caratteristiche appartengono al protocollo UDP (U) o TCP (T).

A. Scambio non affidabile di messaggi .....
B. Trasmissione affidabile di un flusso di byte .....
C. Simmetria del ruolo di più dispositivi intercomunicanti .....
D. Asimmetria del ruolo di due dispositivi connessi .....

10 Per ciascun protocollo applicativo indicare il numero di porta che impiega.

A. FTP .....
B. NTP .....
C. HTTP .....

11 Per ciascuna delle attività tipiche del protocollo HTTP elencate, indicare se il computer è server (S) o client (T).

A. Attesa di richieste .....
B. Invio di richieste da eseguire .....
C. Esecuzione delle richieste ricevute e invio delle relative risposte .....
D. Ricezione di risposte alle richieste inviate .....

12 Indicare per ciascun elemento dei messaggi del protocollo HTTP trasmessi se si tratta di una richiesta (RC) o una risposta (RS).

A. *Header* .....
B. *Body* .....
C. Metodo .....
D. Linea di stato .....

13 Indicare per ciascuna funzionalità del protocollo HTTP il rispettivo metodo.

A. Eliminazione di una risorsa identificata da un URL .....
B. Integrazione di una risorsa identificata da un URL .....
C. Creazione/aggiornamento di una risorsa identificata da un URL .....
D. Richiesta di una risorsa identificata da un URL .....

14 Indicare per ciascuna categoria di codici di stato del protocollo HTTP la relativa descrizione.

A. 200 .....
B. 300 .....
C. 400 .....
D. 500 .....

15 **Indicare per ciascun livello del modello gerarchico OSI dell'ISO la rispettiva funzionalità.**

A. Trasmissione/ricezione di segnali fisici .....
B. Rappresentazione dei dati in un formato comune .....
C. Trasmissione/ricezione di un *frame* di bit .....
D. Trasferimento di pacchetti dall'origine alla destinazione .....
E. Trasmissione/ricezione affidabile di segmenti di dati .....
F. Gestione sessione di comunicazione .....
G. Comunicazione di dati tra due processi su due computer .....

TEST A RISPOSTA MULTIPLA

16 **Da quanti bit è composto l'indirizzo MAC di un'interfaccia *Ethernet*?**

A 32
B 48
C 64
D 128

17 **Quanti sono i bit di un indirizzo IPv4 che identificano la rete?**

A 8
B 16
C 24
D Dipende dalla *netmask*.

18 **Che cosa indica un *header* HTTP di tipo *Content-Type*?**

A In una richiesta il tipo MIME del *body* accettabile per la risposta.
B In una richiesta il metodo HTTP.
C Il tipo MIME del *body* di una richiesta o di una risposta.
D Il tipo di codifica del contenuto testuale del *body* di una richiesta o di una risposta.

19 **La tecnologia *packet-switching*:**

A consente il funzionamento della rete anche in presenza di guasti.
B è nata per permettere il collegamento alla rete di dispositivi mobili.
C garantisce l'indivisibilità dei messaggi inoltrati sulla rete.
D Nessuna delle risposte precedenti è corretta.

20 **Quali sono le funzioni di un *router*?**

A Instradare ogni singolo pacchetto sul cammino migliore per raggiungere la propria destinazione.
B Comunicare ad altri *router* informazioni relative a eventuali variazioni relative alla rete.
C Controllare il contenuto dei pacchetti per verificare che siano stati trasmessi nell'ordine corretto.
D Richiedere al mittente la ritrasmissione dei pacchetti di un messaggio andati perduti.

21 **Un protocollo di rete:**

A è un insieme di regole tecniche rigorosamente definite e formalmente documentate per la progettazione e la realizzazione di hardware e software per la comunicazione di rete.
B è un dispositivo hardware realizzato secondo uno standard rigorosamente definito e formalmente documentato.
C è un'applicazione software realizzata secondo uno standard rigorosamente definito e formalmente documentato.
D Nessuna delle risposte precedenti è corretta.

22 **Fissata la *netmask* 255.255.0.0 quale dei seguenti indirizzi IP non appartiene alla rete 172.16.0.0?**

A 172.16.1.2
B 172.15.0.1
C 172.16.255.254
D Tutti gli indirizzi IP indicati appartengono alla rete.

23 **Il protocollo ARP:**

A consente di associare a un indirizzo IP il corrispondente indirizzo del livello *data-link*.

B è utilizzato da computer privi di configurazione di rete per ottenere il proprio indirizzo IP.

C permette di ottimizzare le prestazioni di una rete locale di computer di tipo *Ethernet*.

D trasforma indirizzi di 48 bit in indirizzi di 32 bit.

24 **Un *web-service*:**

A è un sito web accessibile con un protocollo alternativo a HTTP.

B è un servizio software reso disponibile in rete ad applicazioni e *APP*.

C è un servizio software finalizzato alla gestione dei siti web.

D Nessuna delle risposte precedenti è corretta.

PROBLEMI

25 **Ordinare temporalmente le fasi dell'algoritmo CSMA-CD utilizzato dalle schede di rete *Ethernet*.**

... Trasmissione del frame.
... Eventuale attesa di un tempo casuale.
... Verifica della disponibilità del collegamento.
... Rilevazione di un eventuale collisione.
... Eventuale ritrasmissione del *frame*.

26 **Ordinare temporalmente le fasi di connessione di un *browser* a un server web.**

... Riceve dal server DNS l'indirizzo IP del server web che ospita il sito.
... Invia al server DNS l'URL del sito web.
... Riceve e visualizza la pagina web ricevuta dal sito.
... Richiede al server web la pagina web identificata dall'URL.

27 **Per ciascuno dei seguenti indirizzi di rete IP completare la tabella con le informazioni richieste.**

Indirizzo IP	Numero di bit indirizzo rete	Numero di bit indirizzo dispositivo	Numero massimo di dispositivi nella rete	*Netmask*	Indirizzo di *broadcast*
10.0.0.0	8				
172.16.0.0	16				
192.168.1.0	24				
192.168.0.0	30				
172.31.0.0	20				

buzzword
termine in voga

to drone on
continuare a parlare (in modo monotono)

good manners
buone maniere

lower-tier
di livello inferiore

«nuts and bolts»
«dadi e bulloni»

unwillingness
riluttanza

upper-tier
di livello superiore

Computer Networks and the Internet

Today's Internet is arguably the largest engineered system ever created by ankind, with hundred of millions of connected computers, communication links, and switches; hundreds of millions of users who connect intermittently via cell phones and PDAs; and devices such as sensors, webcams, game consoles, picture frames, and even washing machines being connected to the Internet. Given that the Internet is so large and has so many diverse components and uses, is there any hope of understanding how it (and more generally computer networks) work? Are there guiding principles and structure that can provide a foundation for understanding such an amazingly large and complex sistem? And if so, is it possible that it actually could be both interesting *and* fun to learn about computer networks? Fortunately, the answers to all of these questions is a resounding YES! Indeed, it's our aim in this bookto provide you with a modern introduction to the dynamic field of computer networking, giving you the principles and practical insights you'll need to understand not only today's networks, but tomorrow's as well.

This first chapter presents a broad overview of computer networking and the Internet. Our goal here is to paint a broad picture and set the context for the rest of this book, to see the forest through the trees. We'll cover a lot of ground in this introductory chapter and discuss a lot of the pieces of a computer network, without losing sight of the big picture.

We'll structure our overview of computer networks in this chapter as follows. After introducing some basic terminology and concepts, we'll first examine the basic hardware and software components that make up a network. We'll begin at the network's edge and look at the end systems and network applications running in the network. We'll then explore the core of a computer network, examining the links and the switches that transport data, as well as the access networks and physical media that connect end systems to the network core. We'll learn that the Internet is a network of networks, and we'll learn how these networks connect with each other.

After having completed this overview of the edge and core of a computer network, we'll take the broader and more abstract view in the second half of this chapter. We'll examine delay, loss, and throughput in a computer network and provide simple quantitative models for end-to-end throughput and delay: models that take into account transmission, propagation, and queuing delays. We'll then introduce some of the key architectural principles in computer networking, namely, protocol layering and service models. We'll also learn that computer networks are vulnerable to many different types of attacks; we'll survey some of these attacks and consider how computer networks can be made more secure. Finally,we'll close this chapter with a brief history of computer networking.

What Is the Internet?

In this book, we'll use the public Internet, a specific computer network, as our principal vehicle for discussing computer networks and their protocols. But what is the Internet? There are a couple of ways to answer this question. First, we can describe the nuts and bolts of the Internet, that is, the basic hardware and software components that make up the Internet. Second, we can describe the Internet in terms of a networking infrastructure that provides services to distributed applications. Let's begin with the nuts-and-bolts description, using Figure 1.1 to illustrate our discussion.

A Nuts-and-Bolts Description

The Internet is a computer network that interconnects hundreds of millions of computing devices throughout the world. Not too long ago, these computing devices were primarily traditional desktop PCs. inux workstations, and so-called servers that store and transmit information such as Web pages and e-mail messages. Increasingly,

however, nontraditional Internet end systems such as TVs, laptops, gaming consoles, cell phones. Web ccams, automobiles, environmental sensing devices, picture frames, and home electrical and security systems are being connected to the Internet. Indeed, the term *computer network* is beginning to sound a bit dated, given the many nontraditional devices that are being hooked up to the Internet. In Internet jargon, all of these devices are called **hosts** or **end systems**. As of Juny 2008, there were nearly 600 million end systems attached to the Internet, not counting the cell phones, laptops, and other devices that are only intermittently connected to the Internet.

End systems are connected together by a network of **comunication links** and **packet switches**. We'll see that there are many tpes of communication links, which are made up of different types of physical media, including coaxial cable, copper wire, fiber optics, and radio spectrum. Different links can transmit data at different rates, with the **transmission rate** of a link measured in bits/second. When one end system has data to send to another end system, the sending end system segments the data and adds header bytes to each segment. The resulting packages of information, known as **packets** in the jargon of computer networks, are then sent through the network to the destination end system, where they are reassembled into the original data.

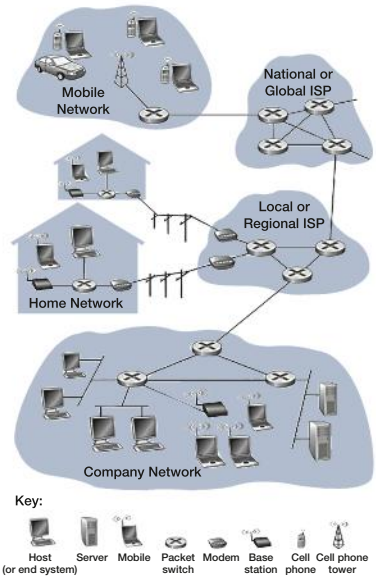


Figure 1.1. Some pieces of the Internet.

A packet switch takes a packet arriving on one of its incoming communication links and forwards that packet on one of its outgoing communication links. Packet switches come in many shapes and flavors, but the two most prominent types in today's Internet are **routers** and **link-layer switches**. Both types of switches forward packets toward their utimate destinations. Link-layer switches are typically used in access networks, while routers are typically used in the network core. The sequence of communication links and packet switches traversed by a packet from the sending end system to the receiving end system is known as a **route** or **path** through the network. The exact amount of traffic being carried in the Internet is difficult to estimate. PriMetrica estimates that to terabits per second of international capacity was used by public Internet providers in 2008, and that capacity doubles approximately every two years.

Packet-switched networks (which transport packets) are in many ways similar to transportation networks of highways, roads, and intersections (which transport vehicles). Consider, for example, a factory that needs to move a large amount of cargo to some destination warehouse located thousands of kilometers away. At the factory, the cargo is segmented and loaded into a fleet of trucks. Each of the trucks then independently travels through the network of highways, roads, and intersections to the destination warehouse. At the destination warehouse, the cargo is unloaded and grouped with the rest of the cargo arriving from the same shipment. Thus, in many ways, packets are analogous to trucks, communication links are analogous to highways and roads, packet switches are analogous to intersections, and end systems are analogous to buildings. Just as a truck takes a path through the transportation network, a packet takes a path through a computer network.

End systems access the Internet through **Internet Service Providers** (**ISPs**), including residential ISPs such as local cable or telephone companies; corporate ISPs; university ISPs; and ISPs that provide WiFi access in airports, hotels, coffee shops, and other public places. Each ISP is in itself a network of packet switches and communication links. ISPs provide a variety of types of network access to the end systems, including 56 kbps dial-up modem access, residential broadband access such as cable modem or DSL, high-speed local area network access, and wireless access. ISPs also provide Internet access to content providers, connecting Web sites directly to the Internet. The Internet is all about connecting end systems to each other, so the ISPs that provide access to end systems must also be interconnected. These lower-tier ISPs are interconnected through national and international upper-tier ISPs such as AT&T and Sprint. An upper-tier ISP consists of high-speed routers interconnected with high-speed fiber-optic links. Each ISP network, whether upper-tier or lower-tier, is managed independently, runs the IP protocol (see below), and conforms to certain naming and address conventions. [...]

End systems, packet switches, and other pieces of the Internet run **protocols** that control the sending and receiving of information within the Internet. The **Transmission Control Protocol** (**TCP**) and the **Internet Protocol** (**IP**) are two of the most important protocols in the Internet. The IP protocol specifies the format of the packets that are sent and received among routers and end systems. The Internet's principal protocols are collectively known as **TCP/IP**. We'll begin looking into protocols in this introductory chapter. But that's just a start – much of this book is concerned with computer network protocols!

Given the importance of protocols to the Internet, it's important that everyone aree on what each and every protocol does. This is where standards come into play. **Internet standards** are developed by the Internet Engineering Task Force (IETF). The IETF standards documents are called **requests for comments** (**RFCs**). RFCs started out as general requests for comments (hence the name) to resolve network and protocol design problems that faced the precursor to the Internet. RFCs tend to be quite technical and detailed. They define protocols such as TCP, IP, HTTP (for the Web), and SMTP (for e-mail). There are currently more than 5,000 RFCs. Other bodies also specify standards for network components, most notably for network links. The IEEE 802 LAN/MAN Standards Committee, for example, specifies the Ethernet and wireless WiFi standards.

[...]

What Is a Protocol?

Now that we've got a bit of a feel for what the Internet is, let's consider another important buzzword in computer networking: *protocol*. What is a protocol? What does a protocol do?

A Human Analogy

It is probably easiest to understand the notion of a computer network protocol by first considering some human analogies, since we humans execute procols all of the time. Consider what you do when you want to ask someone for the time of day. A Typical exchange is shown in Figure 1.2. Human protocol (or good manners, at least) dictates that one first offer a greeting (the first "Hi" in Figure 1.2) to initiate communication with someone else. The typical response to a "Hi" is a returned "Hi" message. Implicitly, one then takes a cordial "Hi" response as an indication that one can proceed and ask for the time of day. A different response to the initial "Hi" (such as "Don't bother me!" or "I don't speak English" or some unprintable reply) might indicate an unwillingness or inability to communicate. In this case, the human protocol would be not to ask for the time of day. Sometimes one gets no response at all to a question, in which case one typically gives up asking that person for the time. Note that in our human protocol, *there are specific messages we send, and specific actions we take in response to the received reply messages or other events* (such as no reply within some given amount of time); Clearly, transmitted and received messages, and actions taken when these messages are sent or received or other events occur, play a central role in a human protocol. If people run different protocols (for examle, if one person has manners but the other does not, or if one understands the concept of time and the other does not) the protocols do not interoperate and no useful work can be accomplished. The same is true in networking – it takes two (or more) communicating entities running the same protocol in order to accomplish a task.

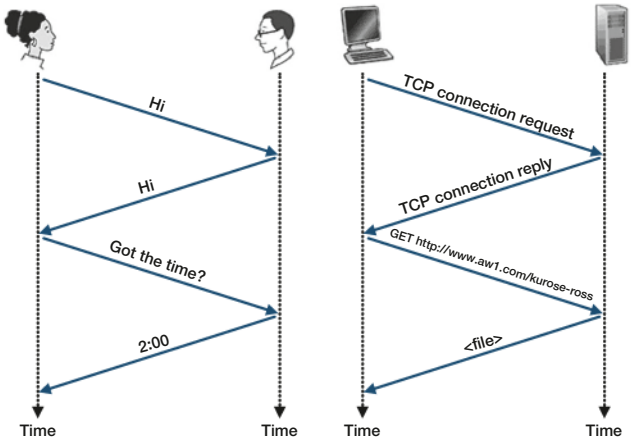


Figure 1.2. A human protocol and a computer network protocol.

Let's consider a second human analogy. Suppose you're in a college class (a computer networking class, for example!). The tescher is droning on about protocols

and you're confused. The teacher stops to ask, "Are there any questions?" (a message that is transmitted to, and received by, all students who are not sleeping). You raise your hand (transmitting an implicit message to the teacher). Your teacher acknowledges you with a smile, saying "Yes ..." (a transmitted message encouraging you to ask your question – teachers *love* to be asked question), and you then ask your question (that is, transmit our message to your teacher). Our teacher hears your question (receives your question essage) and answers (transmits a reply to you). Once again, we see that the transmission and receipt of messages, and a set of conventional actions taken when these messages are sent and received, are at the heart of this question-and-answer protocol.

Network Protocols

A network protocol is similar to a human protocol, except that the entities exchanging messages and taking actions are hardware or software components of some device (for example, computer, PDA, cellphone, router, or other network-capable device). All activity in the Internet that involves two or more communicating remote entities is governed by a protocol. For example, hardware-implemented protocols in the network interface cards of two physically connected computers control the flow of bits on the «wire» between the two network interface cards; congestion-control protocols in end systems control the rate at which packets are transmitted between sender and receiver; protocols in routers determine a packet's path from source to destination. Protocols are running everywhere in the Internet, and consequently much of this book is about computer network protocols.

As an example of a computer network protocol with which you are probably familiar, consider what happens when you make a request to a Web server, that is, when you type the URL of a Web page into your Web browser. The scenario is illustrated in the right half of Figure 1.2. First, your computer will send a connection request message to the Web server and wait for a reply. The Web server wrill eventually receive your connection request message and return a connection reply message. Knowing that it is now OK to request the Web document, your computer then sends the name of the Web page it wants to fetch from that Web server in a GET message. Finally, the Web server returns the Web page (file) to your computer.

Given the human and networking examples above, the exchange of messages and the actions taken when these messages are sent and teceived are the key defining elements of a protocol:

*A **protocol** defines the format and the order of messages exchanged between two or more communicating entities, as well as the actions taken on the transmission and/or receipt of a message or other event.*

The Internet, and computer networks in general, make extensive use of protocols. Different protocols are used to accomplish different communication tasks. As you read through this book, you will learn that someprotocols are simple and straightforward, while others are comples and intellectualy deep. Mastering the field of computer networking is equivalent to understanding the what, why, and how of networking protocols.

[J. Kurose, K. Ross, "Computer Networking. A Top-Down Approach", 5th edition, 2010]

QUESTIONS

a **Why Internet is a *packet-switching* network?**
b **What is a route or path through Internet?**
c **Which are the two most important Internet protocols?**
d **Give a definition of «protocol».**