2
IL LIVELLO APPLICATIVO

Introduzione

In questo capitolo verranno analizzati alcuni protocolli facenti parte di entrambi i modelli standard di rete informatica: il modello *OSI* ed il modello *TCP/IP*.
Benché lo standard internazionale per la comunicazione sia rappresentato dal modello OSI, questo è utilizzato più come un modello astratto di riferimento. Nella pratica comune lo standard è il modello TCP/IP, caratterizzato da un numero inferiore di livelli di rete e da una conseguente elevata praticità.
I livelli OSI riguardati da questa modifica sono i livelli che andremo ad analizzare in questo capitolo, definendoli ed approfondendone i principali servizi. Il *livello applicativo* (o applicazione) è il settimo e più esterno livello nel modello OSI, ed il quarto ed ultimo livello nel TCP/IP. Il parallelismo di tale strato nei due standard sarà motivo di studio nelle sezioni successive. Prima di andare ad approfondire i servizi offerti dai vari strati dei due standard, si tratterà di come si sono sviluppate le *applicazioni web*, a partire dalla nascita di internet fino ad arrivare ad oggi per avere un quadro generale di quello andremo successivamente a studiare nello specifico.

2.1 Funzionalità

Il livello applicazione è un livello astratto che ha la funzione di specificare all'interno di una rete le interfacce ed i protocolli di comunicazione utilizzati e più in generale di fornire servizi alle applicazione degli utenti della rete, al fine di garantire una corretta trasmissione.

Quando due programmi devono comunicare tra loro è necessario prima di tutto che esista una connessione logica tra le due entità logiche (mittente e ricevente). Tramite tale connessione si può andare a gestire i quattro livelli

dello stack TCP/IP tramite un insieme di istruzioni (o funzioni) denominate *API* (*Application Program Interface*) per mezzo delle quali è possibile aprire e chiudere connessioni ed inviare e ricevere dati.
Durante una trasmissione, ogni strato dello stack aggiunge un header al pacchetto dati che identifica il messaggio da trasmettere. Questa operazione è chiamata incapsulamento. Per mezzo di essa è possibile identificare il dato effettivo e l'header aggiunto dallo strato attuale come un unico pacchetto dati che verrà successivamente passato allo strato sottostante. Il pacchetto dati risultante dal passaggio nello stack verrà poi trasmesso e ricevuto. Il computer o l'applicazione ricevente dovrà poi effettuare l'operazione inversa, estraendo ad ogni livello l'header necessario a dirigere l'operazione di ricezione.

Il primo e più esterno strato dello stack, come anticipato, è il livello applicativo, il quale ha lo scopo di *standardizzare la comunicazione.* Nel TCP/IP infatti, il livello applicativo contiene i protocolli e le interfacce di comunicazione usati nelle trasmissioni processo a processo per completare le varie richieste dei programmi, svolte attraverso il protocollo internet (IP), all'interno di una rete. Tra i protocolli più comunemente usati troviamo HyperText Transfer Protocol (HTTP), File Transfer Protocol (FTP), Domain Name System (DNS) e Simple Mail Transfer Protocol (SMTP).
Un esempio di concreto utilizzo dei protocolli si ha ogni volta che richiediamo al nostro browser di accedere ad internet o di caricare o scaricare qualcosa da esso. Il livello applicativo per completare la richiesta, chiamerà nel primo caso il protocollo HTTP mentre nel secondo il protocollo FTP.
Per far arrivare all'utente i dati richiesti secondo lo standard TCP/IP, il livello applicazione deve passare i dati al livello trasporto attraverso degli indirizzi (o interfacce) logici chiamati *porte.* Sarà poi il *livello di trasporto* ad occuparsi della ricezione finale da parte dell'utente.
L'utilizzo delle porte rende molto più semplice al livello di trasporto, capire che tipo di dato gli sta venendo passato. Se ad esempio venisse inviato un dato sulla porta 25, significherebbe che il dato riguarda una richiesta e-mai. Se invece fosse stato mandato sulla porta 21 o 22, sarebbe stato un dato FTP.

Il livello applicazione nel TCP/IP non specifica nessun tipo di regola o formato di dati accettabili, che le applicazioni devono invece tenere di conto durante le trasmissioni. Per questo motivo nella specifica iniziale dello standard è fortemente raccomandato di seguire il *robustness principle* (principio di robustezza).

Principio di robustezza. *(o legge di Postel): "be conservative in what you do, be liberal in what you accept from others".*

In altre parole, quando viene inviato un messaggio ad un altro computer

o ad un'altra applicazione, questo deve essere perfettamente conforme alla specifica richiesta. In caso invece di messaggi in entrata non conformi, questi devono essere accettati fintanto che il loro significato è chiaro.

Nel modello OSI il livello applicativo è costituito, come nel TCP/IP, da un insieme di protocolli ed interfacce di comunicazione e come nel TCP/IP troviamo tra i protocolli più comunemente usati HTTP, FTP, DNS e SMTP. Una sostanziale differenza però la si trova nella struttura stessa del modello. Il livello applicativo del TCP/IP comprende sia il livello applicativo che i livelli presentazione e sessione del modello OSI. Questa maggiore modularità degli strati dello standard OSI è dovuta ad una più elevata distribuzione delle funzioni tra essi. Il livello applicazione del modello OSI infatti, lavora direttamente col software applicativo e si occupa solo di verificare la effettiva disponibilità di un partner per la comunicazione e delle risorse necessarie alla trasmissione. Una volta appurata la presenza di un partner e la disponibilità di risorse, il pacchetto dati composto dal dato da scambiare e dall'header aggiunto dal livello applicazione viene passato al sottostante livello presentazione.

2.2 Livelli Presentazione e Sessione

Il livello presentazione è il sesto livello del modello ISO/OSI ed è il primo livello ad occuparsi del significato delle informazioni trasmesse. Ha la funzione di assicurare la compatibilità dei dati durante una comunicazione tra entità che utilizzano codifiche diverse. Questo avviene tramite una traduzione del dato in una forma accettabile (quindi comprensibile) dal livello applicazione e dai livelli sottostanti del ricevente. Nello specifico: una volta iniziata la trasmissione, cambia il formato dei dati da quello del dispositivo mittente (*sintassi locale*, a quello del dispositivo ricevente (*sintassi di trasferimento*
Questo strato offre inoltre importanti servizi quali la formattazione, la codifica, la criptazione e la compressione di dati. Se ad esempio si volesse convertire un file di testo scritto in EBCDIC (Extended Binary Coded Decimal Interchange Code) in un file scritto in ASCII American Standard Code for Information Interchange, tale conversione avverrebbe a questo livello. Un altro esempio si ha quando si invia un file di testo dal contenuto sensibile (come una password), ed è dunque necessario che il messaggio venga criptato. Tale operazione avviene proprio al livello presentazione.
Una volta finita l'elaborazione del pacchetto dati, questo viene passato allo strato inferiore, il livello sessione.

Il livello sessione è il quinto livello del modello OSI e fornisce i meccanismi per stabilire, gestire ed infine concludere la connessione tra una applicazione

locale ed una remota, svolgendo inoltre un'operazione di verifica per garantire la corretta consegna del pacchetto dati. È solo a questo punto dello stack infatti, dopo che il livello applicativo e presentazione hanno adattato il pacchetto dati alla comunicazione, che avviene l'effettiva connessione. A questo livello vengono gestite anche le funzioni di autenticazione e autorizzazione e viene fornito il servizio di *session restoration* (checkpointing and recovery). Quest'ultimo servizio proverà, in caso di perdita di connessione, a ripristinarla riavviando se necessario la comunicazione.
Un esempio concreta del passaggio dal livello sessione si ha durante una video chiamata, durante la quale è essenziale che il video e l'audio siano sincronizzati per evitare *problemi di lip synch* (problemi di sincronizzazione del movimento delle labbra con l'audio). Se il quinto strato è ben progettato, la persona sullo schermo corrisponderà sempre all'audio in uscita, ed in caso di perdita di connessione, il meccanismo di session restoration, proverà a ristabilirla.
Date le sue caratteristiche, il livello sessione viene tipicamente implementato negli ambienti applicativi che fanno uso delle RPCs (chiamate a procedura remote), ma soprattutto nei Web browsers. In questi ultimi utilizza comunemente protocolli come lo ZIP (Zone Information Protocol) ed il SCP(Session Control Protocol).

2.3 Evoluzioni delle applicazioni Internet e Web

Dalla nascita di *ARPANET* negli anni '60 si sviluppò nel mondo un nuovo modo di comunicare: la rete internet. Sebbene inizialmente i fini fossero puramente militari, con la fine della Guerra Fredda l'esercito si disinteressò dello strumento, lasciandolo nel pieno controllo delle università dove divenne utile per scambiare conoscenze scientifiche e più in generale per comunicare. Negli stessi anni *Tim Berners-Lee* creò un'architettura che semplificò drasticamente l'utilizzo della rete ormai rinominata Internet: il *World Wide Web*. Nel 1991 lo stesso Tim Berners-Lee, creò il primo sito web, avviando una rivoluzione che oggi chiamiamo **Web 1.0**.

Il Web 1.0 era l'internet dei contenuti, caratterizzato da siti web semplici e statici dai quali era possibile solo accedere a risorse senza poterle modificare o aggiungere. Le pagine web erano scritte da una ristretta cerchia di persone ed erano gremite di collegamenti ipertestuali (link) ad altre pagine, in modo da dare all'utente più libertà di movimento possibile all'interno della rete. È col Web 1.0 che nacque il concetto di *applicazione internet* (web application), cioè un programma di tipo client-server che viene eseguito dal client tramite un *web browser*. Nel Web 1.0 il client, rappresentato dal browser, per mezzo di uno o più web server accede a pagine web statiche (figura web client -¿network-¿web server-¿web pages). I browser utilizzati per la

navigazione erano molto semplici poiché l'unico linguaggio che dovevano interpretare era l'HTML. Nel Web 1.0 non era inoltre prevista alcun tipo di separazione tra i dati e la loro rappresentazione, era tutto semplicemente ammassato nelle pagine web, rendendo queste ultime molto fragili. Se ad esempio doveva essere cambiata anche solo una parola, si doveva andare a modificare l'intera pagina.
Le comunicazioni erano gestite, come adesso, dal protocollo HTTP il quale ha la caratteristica di essere un protocollo "senza stato" (state-less protocol). Questo significa che ogni richiesta da client a server è trattata in modo indipendente dalle altre e di conseguenza ogni informazione scambiata tra browser e server alla fine della trasmissione viene persa. Se ad esempio avessimo messo qualcosa nel carrello di un qualche sito di acquisti ed avessimo poi navigato su altri siti o chiuso e riaperto il collegamento, una volta tornati al nostro sito di acquisti avremmo trovato il carrello vuoto. Questo problema venne risolto sempre in quel periodo con l'invenzione dei *cookies*. Per mezzo di essi, ogni volta che l'utente effettuava una richiesta il browser, se necessario, inviava al server i cookies per notificarlo di connessioni precedenti.

Col progredire degli anni e l'aumentare del numero degli utenti, si rendeva sempre più necessaria la possibilità di interagire con i contenuti. Il mutamento iniziò grazie all'ausilio dei nuovi linguaggi di programmazione come *PHP*, per mezzo dei quali iniziarono a crearsi i primi *blog*, cioè siti web sui quali era possibile inserire dei commenti. Questo primo cambiamento comportò l'avvento delle prime *community* e venne identificato come **Web 1.5**. Di lì a pochi anni la rete si espanse in modo esponenziale con l'introduzione dei *Wiki* e dei *Social Network*, ponendo l'interattività con l'utente in primo piano e facendo nascere così il **Web 2.0**.

La rete da statica divenne quindi dinamica, apportando modifiche architetturali principalmente lato server, rendendolo decisamente più complesso. Non si avevano più semplici pagine web statiche che reperivano informazioni da server, ma pagine web dotate di un elevato numero di servizi e script costantemente in comunicazione con qualche *database*.
Due dei servizi più innovativi furono gli *RSS* ed i *Podcast*. L'RSS (Really Simple Syndacation) è uno dei più popolari formati di distribuzione dei contenuti web basato su XML (eXtensible Markup Language) per mezzo del quale divenne possibile, tramite il meccanismo di feed-RSS, monitorare gli aggiornamenti di un sito senza visitarlo. La funzionalità dei Podcast è molto simile, ma anziché monitorare il cambiamento di testo ed immagini, tramite tale meccanismo si rese possibile monitorare l'inserimento di file audio e video.
Conseguentemente al mutamento del lato server, i browser divennero più complessi eliminando le incompatibilità che vi erano in passato con le web

app ed offrendo una vasta gamma di servizi oggigiorno fondamentali. Un esempio si ha dall'arrivo di *JavaScript*, per mezzo del quale si rese possibile eseguire codici direttamente sul browser.
Col Web 2.0 si registrò un fortissimo aumento degli utenti e dei contenuti pubblicati [immagine crescita siti web e utenti web 1.0- 2.0- 3.0]. Basti infatti pensare che ogni post che venisse pubblicato da qualcuno su un qualsiasi social media diventava (e diventa ancora) effettivamente un contenuto reperibile da chiunque su internet.

Passando dal Web 1.0 al 2.0 e continuando esso a crescere in modo esponenziale si è venuto a creare un problema: se prima su internet scarseggiavano i contenuti e si rischiava di non riuscire a trovare una corrispondenza alle nostre ricerche, questi divennero poi troppi e districarsi tra essi per presentare una corrispondenza significativa divenne sempre più difficile.
Il **Web 3.0**, il *Web of Things*, è quello su cui oggigiorno navighiamo ed è basato sul paradigma dell'*intelligenza semantica*, ancora oggi in piena evoluzione. In questa fase, il World Wide Web è rappresentato da un ambiente in cui ogni file pubblicato (pagine HTML, immagini, video, testi ecc...) entra concettualmente a far parte di un enorme database , il *Web Database* nel quale ogni dato è interpretabile tramite metadati associati ad esso che ne specificano il contesto semantico in un formato adatto all'interrogazione, all'interpretazione e, più in generale, all'elaborazione automatica intelligente. L'intelligenza è infatti la caratteristica fondamentale di questa fase. Tramite algoritmi sempre più sofisticati che sfruttano l'intelligenza artificiale avremo in futuro programmi più potenti ed efficienti capaci di interagire e collaborare tra loro e con gli utenti, sfruttando la potenza della semantica. Si pensi ad esempio se un'applicazione calendario ed una di posta elettronica fossero in grado di comunicare tra loro. Ogni volta che venisse fissato un appuntamento, il calendario potrebbe aggiornarsi da solo, tramite le informazioni comunicategli dall'applicazione di posta elettronica.
Per far sì che ciò sia possibile, il Web 3.0 si basa sul concetto che tutte le fonti siano codificate secondo gli stessi criteri e che quindi tutti i documenti condividano la lingua con cui sono scritti.

La rete è in continuo mutamento e sul come si svilupperà il Web 3.0 si possono fare solo congetture.
Tim Berners-Lee vede il Web 3.0 come una rete raggiungibile da tutti, senza barriere, che tramite la semantica possa generare applicazioni web più potenti di qualsiasi altra applicazione mai creata. Tale potenza, oltre che dal punto di vista della semantica è intesa anche dal punto di vista grafico, dove la *grafica vettoriale scalabile* (SVG), secondo Lee, prenderà il sopravvento. Per mezzo di essa è possibile esprimere figure interattive che possono essere ridefinite in qualsiasi momento ed in qualsiasi punto, senza perdere un grammo di qualità. Ci sono tuttavia persone che teorizzano si tenderà a

portare la rete in una forma tridimensionale (tipo second life) fatta non più di pagine ma di spazi in cui poterci muovere per trovare ciò che cerchiamo. Altre teorie vedono il futuro del web come un'entità onnipresente nella vita quotidiana, capace di alterare letteralmente le percezioni umane. Da quest'ultimo punto di vista, i Google Glasses potrebbero essere visti come un primo piccolo tentativo di sviluppo, che non ha però trovato spazio nella società odierna.