**Livello Applicazione e Paradigmi**

**1. Livello Applicazione**

* **Definizione**: Fornisce servizi all'utente finale. È il livello più alto del modello TCP/IP e si basa su connessioni logiche tra programmi.
* **Funzionamento**: Utilizza i servizi del livello di trasporto per scambiare messaggi tra dispositivi.
* **Protocolli**:
  + **Standard**: HTTP (web), FTP (trasferimento file), SMTP (posta elettronica).
  + **Non Standard**: Protocolli personalizzati per esigenze specifiche, creati da sviluppatori.

**2. Paradigmi del Livello Applicazione**

**a. Client/Server**

* **Funzionamento**:
  + Il client invia richieste, il server le elabora e risponde.
  + Esempi: Navigazione web, sistemi di file distribuiti.
* **Vantaggi**: Dati centralizzati e gestione unificata.
* **Svantaggi**: Carico sul server; interruzioni in caso di problemi.

**b. Peer-to-Peer (P2P)**

* **Funzionamento**:
  + Ogni dispositivo funge sia da client che da server.
  + Esempi: BitTorrent, applicazioni di file sharing.
* **Vantaggi**: Distribuzione del carico, indipendenza da server centrali.
* **Svantaggi**: Sicurezza ridotta, gestione complessa.

**c. Paradigma Misto**

* **Definizione**: Combina client/server e P2P.
* **Esempi**: Skype (server centrale per individuare nodi P2P).

**3. API (Application Programming Interface)**

* **Definizione**: Interfacce per far interagire programmi con il sistema operativo o altri software.
* **Esempi**: API di Google Maps, interfacce per database.

**Comunicazione e Protocolli**

**1. Socket e Comunicazione**

* **Definizione**: Interfaccia software per comunicare tra processi.
  + **Indirizzo IP**: Identifica il dispositivo.
  + **Porta**: Specifica il servizio.
* **Socket Address**:
  + Locale: Identifica il processo sul dispositivo.
  + Remoto: Indica il destinatario remoto.
* **Utilità**: Permette lo scambio bidirezionale di dati.

**2. Protocollo HTTP e HTTPS**

* **HTTP**:
  + Trasferisce pagine web (porta 80).
  + Stateless: Non mantiene lo stato tra richieste.
* **HTTPS**:
  + Versione sicura (porta 443), usa crittografia TLS/SSL.
* **Metodi HTTP**:
  + GET (richiesta dati), POST (invio dati), PUT (aggiornamento), DELETE (eliminazione), HEAD (solo intestazioni).
* **Connessione Persistente**:
  + Mantiene la connessione attiva per più richieste (default in HTTP/1.1).

**3. TCP vs UDP**

* **TCP**:
  + Affidabile, connessione orientata (es. email, HTTP).
  + Include un handshake iniziale.
* **UDP**:
  + Leggero, non affidabile (es. streaming, VoIP).
* **Scelte**:
  + TCP per affidabilità.
  + UDP per velocità e tolleranza agli errori.

**Servizi Applicativi**

**1. HTTP e Risposte del Server**

* **Codici di Stato**:
  + 2xx: Successo (200 OK).
  + 4xx: Errori client (404 Not Found).
  + 5xx: Errori server

(500 Internal Server Error).

* **Intestazioni di Risposta**:
  + Content-Length: Dimensione del contenuto.
  + Content-Type: Tipo di dati (es. HTML, JSON).
  + Set-Cookie: Invio di cookie al client.

**2. Cookie**

* **Definizione**: File memorizzati nel browser per identificare o tracciare l'utente.
* **Tipologie**:
  + Cookie di sessione: Eliminati alla chiusura del browser.
  + Cookie persistenti: Rimangono per un tempo definito.
  + Cookie di terze parti: Creati da domini esterni.
* **Utilizzi**:
  + Gestione delle sessioni, personalizzazione, tracciamento.

**3. Cache e Proxy**

* **Web Caching**:
  + Conserva copie delle risorse per migliorare le prestazioni.
  + Implementata nel browser o tramite server proxy.
* **Server Proxy**:
  + Intermediario tra client e server.
  + Riduce la latenza e aumenta la sicurezza.

**4. FTP (File Transfer Protocol)**

* **Funzionamento**:
  + Trasferisce file tra client e server.
  + Usa due connessioni:
    - Controllo: Invio comandi.
    - Dati: Trasferimento file.
* **Modalità**:
  + Attiva: Il server stabilisce la connessione dati.
  + Passiva: Il client gestisce entrambe le connessioni.
* **Comandi Comuni**:
  + LIST: Visualizza file.
  + RETR: Scarica file.
  + STOR: Carica file.

**5. Posta Elettronica**

* **Protocolli**:
  + SMTP: Invio di email.
  + POP3: Scarica email, elimina o conserva sul server (porta 110).
  + IMAP4: Sincronizza email su più dispositivi (porta 143).
* **Componenti**:
  + UA (User Agent): Applicazioni client (es. Outlook).
  + MTA (Message Transfer Agent): Gestisce la consegna tra server.
  + MAA (Message Access Agent): Consente l'accesso alle email salvate.

**DNS e Risoluzione dei Nomi**

**1. DNS (Domain Name System)**

* **Definizione**: Traduce nomi leggibili (es. www.google.com) in indirizzi IP.
* **Componenti**:
  + Sistema distribuito di server DNS.
  + Record DNS per ogni dominio:
    - A: Indirizzo IPv4.
    - AAAA: Indirizzo IPv6.
    - MX: Server di posta.
    - CNAME: Alias di un dominio.
* **Tipi di Messaggi**:
  + Query: Richiesta del resolver.
  + Risposta: Record DNS associati.

**2. Modalità di Risoluzione**

* **Ricorsiva**:
  + Il resolver delega tutto il processo a un server DNS.
  + Il server DNS interroga gli altri server e restituisce la risposta completa.
* **Iterativa**:
  + Il resolver interroga server successivi, ottenendo risposte parziali.

**3. Caching**

* **Definizione**: Memorizzazione delle risposte DNS per future richieste.
* **TTL (Time to Live)**:
  + Ogni risposta ha una durata predefinita dopo cui scade.

**TELNET e SSH**

* **TELNET**:
  + Protocollo per accedere a sistemi remoti tramite linea di comando.
  + Non sicuro: Trasmette dati in chiaro, vulnerabile ad attacchi.
* **SSH (Secure Shell)**:
  + Alternativa sicura a TELNET.
  + Fornisce crittografia, autenticazione e gestione sicura delle connessioni.

Questi concetti forniscono una visione completa dei protocolli di livello applicazione, della comunicazione su Internet e delle tecnologie fondamentali per reti e servizi online. Se desideri ulteriori dettagli, sono a disposizione!